European Research Studies Journal
Volume XXIV, 3, 2021

pp. 316-334

Shelf Space Allocation for Specific Products on Shelves Selected in Advance

Submitted 28/05/21, 1st revision 17/06/21, 2nd revision 11/07/21, accepted 25/0821

Kateryna Czerniachowska[1], Marcin Hernes[2]

Abstarct:

Purpose: *The aim of the research is to develop the shelf space allocation model for specific products on shelves selected in advance, based on the practical retail requirements. Retailers and manufacturers may impose particular conditions for the appearance of products on the shelf based on its package type, brand, price, form and size. Shelf space allocation must be in line with the store positioning strategy of the retailer.*
Design/Methodology/Approach: *This paper proposed dynamic programming to solve the profit maximization problem on small problem sizes considering extra allocation parameters such as capping and nesting. The distribution of shelf space to products has a direct effect on the competitiveness of the retail store.*
Findings: *The paper presented major profit differences between shelf space allocation without capping and nesting parameters and including them. The computational experiments were performed to test the additional gains received with the usage of capping and nesting parameters.*
Practical Implications: *This research provided qualitative insights for the retailers by comparing the profit gained with and without the capping and nesting allocation possibilities proposed in the model.*
Originality/Value: *In this paper the basic shelf space allocation problem was simplified with selection of the shelf to place the products in advance, next the basic shelf space allocation model was extended with capping and nesting on-shelf allocation methods.*

Keywords: *Retailing, decision making/process, merchandising, shelf levels, product package*

JEL codes: *C44, C61, L81.*

Paper Type: *Research Paper.*

Acknowledgements: *The project is financed by the Ministry of Science and Higher Education in Poland under the program "Regional Initiative of Excellence" 2019-2022, project number 015/RID/2018/19, total funding amount 10,721,040.00 PLN.*

---

[1]*Department of Process Management, Wroclaw University of Economics and Business. E-mail: kateryna.czerniachowska@ue.wroc.pl*
[2]*Corresponding Author, Department of Process Management, Wroclaw University of Economics and Business. E-mail: marcin.hernes@ue.wroc.pl*

1. Introduction

In a competitive environment retailer should enable the customer to make choice not only on product but also on package size and purchase quantity. The prices of products vary in such cases very often. Since the retail segment is growing rapidly, some vendors often create dedicated special smaller packages of well-known national brands for low-income customers.

The main concern of the shelf space allocation problem (SSAP) is to select the products and assign the amount of shelf space to each selected product (Coskun, 2012). Typically, the task of the retailer within the retail chain is to offer sufficient choices of products in limited quantities through outlets located close to relevant groups of customers (Varley, 2001). In supermarkets very often one could find the same product presented in different package sizes with different per-unit prices (Burke, 2006).

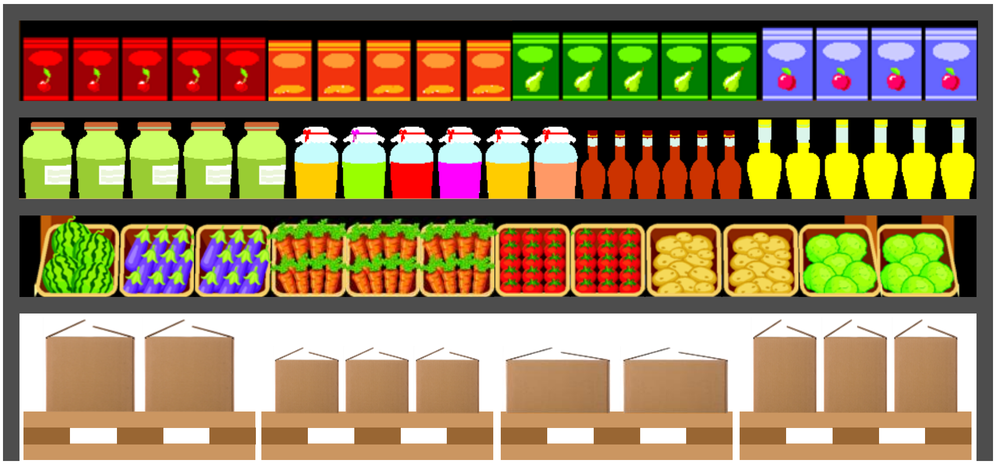
Retailers and manufacturers may apply specific requirements to display products on the shelves such as shelf level or neighboring on shelf products. The arrangement of products on appropriate shelf levels needs to be considered in the light of the package type, brand, price, form and size. The variety of package types and allocation of them in specific shelf levels should satisfy the long term strategic objectives obtaining productivity gains.

Merchandising is a store aesthetic strategy which means that the products are displayed on shelves in retail stores. The goal is to have an eye-level interaction with customers and encourage them to make impulse purchases, minimize the workforce ratio and increase the obtained profit (Randhawa and Saluja, 2017). Retail merchandisers recognize the value of on-shelf merchandising. Appropriate shelves, display levels are the key factors in effective shelf marketing.

The local customer may pretend that some shelves are more convenient for the products they require. For example women are not as tall as men, for this reason the products dedicated to be used by women should not be placed on the highest shelves. Sweets and toys should be placed on lower levels to enable children to see them. Therefore the retailers make an effort to meet clients' requirements more closely.

While there are many literature related to the shelf space allocation, in the most models the goal is to determine the shelf for the product and the number of facings of them on each shelf. But sometimes it may be necessary to present the products on the shelves selected in advance. Figure 1 presents the planogram on which each shelf is dedicated for the specific type of package. Obviously as it could be observed, the products from different shelves can't be moved to other shelves because of their weight, size, package type or application. This significantly simplifies the problem and enables to propose exact solutions to solve small instances. But the problem complicates the possibility of product capping (Figure 2) or nesting (Figure 3) while allocation on shelf. To the best of our knowledge there is no any literature which considers cappings or nestings parameters. The paper contributes to filling this gap in the literature. The aim of research is to develop the shelf space allocation model for specific products on shelves selected in advance, based on the practical retail requirements. In this paper, we propose a dynamical programming approach, in which the capping and nesting allocation methods are considered. Next, the experiments present the profit comparison between allocation with or without such parameters.

Figure 1. *Planogram with shelves for specific products.*



Source: *Own elaboration.*

The rest of the paper is organized as follows. Section 2 provides a literature background with regard to quality and shelf levels as well as package types. Section 3 presents the SSAP and the dynamic programming to solve it on small instances. Further, section 4 gives computational experiments. The paper is concluded in section 5.

2. Literature Background

2.1 Quality Levels

Usually retailers make an effort to show the product on fixtures that are suitable for the products, and the items themselves must be shown in such a manner to appeal to the consumers (Varley, 2001). The number of assortment quality levels offered is one of the characteristics of the product assortments. Several stores, like many outlets, provide only one quality level, while some other offers two, three or even four quality levels. For example, a store assortment can contain a generic brand, a store brand, and a national brand (Simonson, 1999).

Extensions of studies on the compromise effect show the amount of quality levels that customers expect influence the probability that a comparatively more costly, higher margin choice will be chosen. Simonson and Tversky (1992) have found that, in many situations, adding a third, higher-quality level to the range of choices considered causes customers to choose a higher-quality, higher-priced alternative, with the cheapest choice losing the most. Contradictory, adding a third, lower quality alternative to a set does not move the customers' preference to lower quality levels.

The goal of a high-low strategy is to offer as much merchandise as necessary at "full price" with a high profit margin for as long time as possible, and then reduce the profit for some time frame in order to clear out older products and to make the opportunity for new ones. High-low pricing strategy fits better for prestige pricing, but it's out of date. Branded goods should be traded off at a discount in order to keep the product lines new and creative (Varley, 2001).

Based on the above mentioned literature the importance of allocating products on appropriate quality shelf levels could be understood.

2.2 Shelf Levels

Allocation of products on vertical shelf levels has a more significant influence on sales than horizontal positioning (Hansen, Raut, and Swami, 2010; Elbers, 2016; Wongkitrungrueng, Valenzuela, and Sen, 2018). As shown by Ebster and Garaus (2011), the highest customer focus is paid to eye-level and touch-level zones. Research further suggests that the stretch level relatively raises more attention than the stoop level. The tendency of growing emphasis from the bottom shelf is noticeable in the majority of categories.

There is no doubt that it is impossible to place all products on an eye-level, but the retailer should try to distribute products on different shelf levels according to the merchandising strategies which are the most relevant in the defined retail chain (Elbers, 2016). Furthermore, the product may be displayed on more than one shelf levels. Vertical shelves levels are divided into 4 zones (Ebster and Garaus, 2011):

- Stoop-level – below 0.9 m, the lowest shelf, to which customers must crouch while viewing the products. It is used for lower margin products.
- Touch-level – 0.9 m — 1.2 m, lower shelf which customers can easily reach. Generally it is used for sweets, cakes, snacks, toys and other products dedicated for children.

- Eye-level – 0.9m — 1.2m, the most important supermarket shelves with a highest customers' attention. It is used for high-profit margin products.
- Stretch-level – above 1.8m, two levels higher than the touch level shelf, therefore it receives little attention from customers. Supermarkets avoid placing heavy products there (Ebster and Garaus, 2011).

The literature above gives the examples of retail decisions in which products are assigned to the different shelf levels in advance.

2.3 Package Types

Product package management initially attracts a potential customer to the shelf. Different packaged products are managed in fundamentally different ways. Merchandise depends on the aesthetic attractiveness to draw interest. It is fundamentally different for packaged goods (e.g., biscuits) or staple products (e.g., yogurt), so several of them could be more visually appealing to customers. Since the product itself serves as the key stimulus, there is huge research interest to consider exactly what catches visual focus and how to control of the merchandized atmosphere to increase sales (Behe *et al.*, 2013).

Product, packaging and advertisement design choices with a specific positioning (category) can deliver significant benefits. Retailer performs product categorization based on brand name, price, thickness, package type, style and package size (Desrochers and Nelson, 2006). Manufacturers can use awareness of the growing success of the brand in a specific segment to make a more effective presentation of the promoted product, simplify the tasks of store category manager in finding the suitable location on the shelf. Of course, advertisements, packages, exposition and product layout may be used for categorization on the shelves (Desrochers and Nelson, 2006).

Package variations enable the retailers to more efficiently set up their shelf displays. For example, jams could be sold in small jars and tins, cereals and biscuits in paper packages, fruit and vegetables in cans or specific containers. Therefore different kinds of orders are available even in the most disorderly of stores as well as categorized grocery store (Evans, 2011). As it was mentioned, the literature explains a lot of cases in which the shelf for the product is selected in advance with regard to its package type.

3. Problem Definition

Parameters and indices used in a model:
P - total number of products.
j - product index, $j = 1,...,P$.

Shelf parameters:

s^l - shelf length.

s^d - shelf depth.

s^h - shelf height.

Product parameters:

p_j^w - width of the product j.

p_j^d - depth of the product j.

p_j^h - height of the product j.

p_j^s - supply limit of the product j.

p_j^u - unit profit of the product j.

p_j^n - nesting coefficient of the product j, $p_j^n < 1$, or $p_j^n = 0$ if product can't be nested.

$p_j^{o_1} = \begin{cases} 1, & \text{if front orientation is available for product } j \\ 0, & \text{otherwise} \end{cases}$ - front orientation binary parameter.

$p_j^{o_2} = \begin{cases} 1, & \text{if side orientation is available for product } j \\ 0, & \text{otherwise} \end{cases}$ - side orientation binary parameter.

$f_j^{\min}$ - minimum number of facings of the product j.

$f_j^{\max}$ - maximum number of facings of the product j.

$c_j^{\min}$ - minimum number of caps per facings group of the product j.

$c_j^{\max}$ - maximum number of caps per facings group of the product j.

$n_j^{\min}$ - minimum number of nests of one facing of the product j.

$n_j^{\max}$ - maximum number of nests of one facing of the product j.

Decision variables:

f_j - the number of facings of the product.

c_j - the number of caps of the product.

n_j - the number of nests of the product.

$$y_j^{o_1} = \begin{cases} 1, & \text{if product } j \text{ is put on the shelf on front orientation} \\ 0, & \text{otherwise} \end{cases}.$$

$$y_j^{o_2} = \begin{cases} 1, & \text{if product } j \text{ is put on the shelf on side orientation} \\ 0, & \text{otherwise} \end{cases}.$$

The SSAP is defined as follows. A set of P products must be placed on the shelf of a planogram. In this research only one shelf is investigated at each moment because in this SSAP the shelves are allocated on the different levels and all products are prepared to be placed on the specific level, e. g.:

- Stretch-level for products of lighter weight on shorter shelves above;
- Eye-level for high profit, branded or expensive products;
- Touch-level for higher profits products which allow less customer attention than the eye-level;
- Low-level for children's or cheep products.
- Stoop-level for products of low profit, heavy or big, sometimes this shelf is substituted by a pallet.

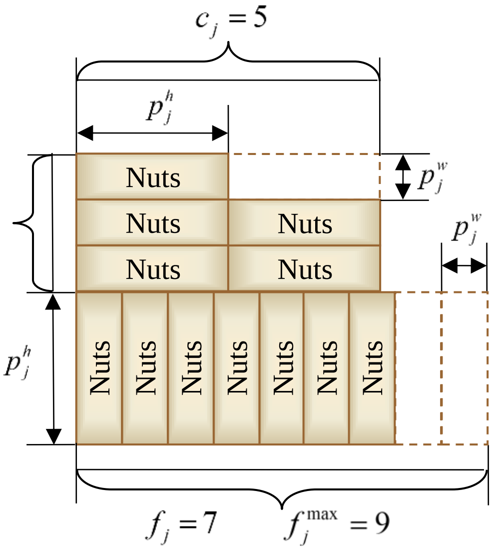
The product supply limit p_j^s represents the maximum number of items of the given product. Products can be placed on the shelves only with facings, facings with cappings or facings with nestings. Figure 2 shows the capping rule when the products could be placed above the facings (as light cookies or tea packages). Figure 3 shows the nesting rule when products could be placed inside each other (as baskets or plates). In this case p_j^n represents the percentage of the height which gives one nested product.

The minimum number of facings $f_j^{\min}$ signifies the product on shelf visibility. The maximum number of facings $f_j^{\max}$ is defined by the retailer based on the product movement. The minimum number of caps per facings group $c_j^{\min}$ shows if the product can be capped. The maximum number of caps per facings group $c_j^{\max}$ means how many cappings can be placed above facings without destroying the facings below due to their weight. $\lceil p_j^h / p_j^w \rceil$ is the minimal number of facings in one group in order to support the cappings above. The minimum number of nests of one facing $n_j^{\min}$ shows if the product can be nested. The maximum number of nests of one facing $n_j^{\max}$ means how many nestings could be placed inside one facing without destroying them.

Therefore $(f_j + c_j + n_j)$ means the total number of product on the shelf. The orientation parameters show if the product could be placed on the shelf on its front orientation $p_j^{o_1}$ or side orientation $p_j^{o_2}$ (Figure 4). The possible product orientations are defined based on the package type, printed text on the product cover or its dimensions. If the product is on front orientated, its width p_j^w is taken as line parameter. If the product is on side orientated, its depth p_j^d is taken as line parameter.

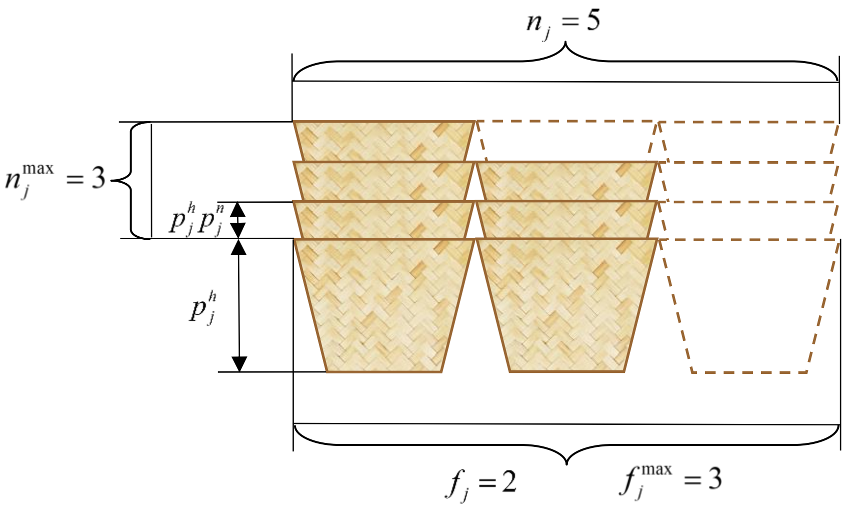
To solve the problem, one has to determine the number of facings f_j, caps c_j and nests n_j of a product j allocated to the shelf on its front $y_j^{o_1}$ or side $y_j^{o_2}$ orientation with regard to shelf and product constraints.

Figure 2. *Example of caps allocation.*



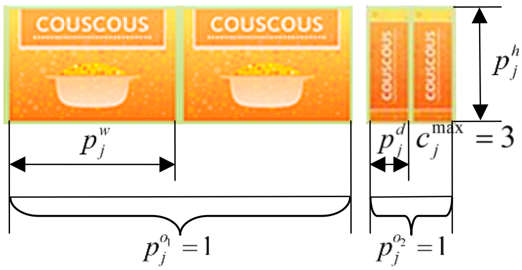
Source: *Own elaboration.*

Figure 3. *Example of nests allocation.*



Source: *Own elaboration.*

Figure 4. *Orientation possibilities.*



Source: *Own elaboration.*

3.1 Problem Formulation

The problem can then be formulated as follows:

$$\max \sum_{j=1}^{P} p_j^u (f_j + c_j + n_j) \qquad (1)$$

subject to:

$$\sum_{j=1}^{P} f_j (y_j^{o_1} p_j^w + y_j^{o_2} p_j^d) \le s^l \qquad \text{(shelf length)} \qquad (2)$$

$$\forall(j)\left[\left(p_j^h + \left\lceil \frac{c_j}{\left\lfloor \frac{f_j (y_j^{o_1} p_j^w + y_j^{o_2} p_j^d)}{p_j^h} \right\rfloor} \right\rceil \cdot (y_j^{o_1} p_j^w + y_j^{o_2} p_j^d) + \left\lceil \frac{n_j}{f_j} \right\rceil \cdot p_j^h p_j^n \right) \le s^h \right],$$

$$\text{(shelf height)}^* \qquad (3)$$

*Constraint (3) represents the base product height, caps height and nests height correspondingly to the Figure 2 and Figure 3.

$$\forall(j)[y_j^{o_1} p_j^d + y_j^{o_2} p_j^w \le s^d] \qquad \text{(shelf depth)} \qquad (4)$$

$$\forall(j)[f_j + c_j + n_j \le p_j^s] \qquad \text{(supply limit)} \qquad (5)$$

$$\forall(j)[f_j^{min} \le f_j \le f_j^{max}] \qquad \text{(minimum and maximum number of facings)} \qquad (6)$$

$$\forall(j)\left[c_j^{min} \le c_j \le c_j^{max} \cdot \left\lfloor \frac{f_j (y_j^{o_1} p_j^w + y_j^{o_2} p_j^d)}{p_j^h} \right\rfloor \right]$$

$$\text{(minimum and maximum number of caps)}^* \qquad (7)$$

*Constraint (7) assures that if capping exists, the number of product facings is enough so that the capping can be placed above facings.

$$\forall(j)[n_j^{min} \le n_j \le n_j^{max} f_j] \qquad \text{(minimum and maximum number of nests)} \qquad (8)$$

Decision variables:

$$f_j = \{f_j^{min} \ldots f_j^{max}\} \qquad \text{(number of facings)} \qquad (9)$$

$$c_j = \{c_j^{\min} ... c_j^{\max} \cdot \left\lfloor \frac{f_j (y_j^{o_1} p_j^w + y_j^{o_2} p_j^d)}{p_j^h} \right\rfloor \} \quad \text{(number of caps)} \quad (10)$$

$$n_j = \{n_j^{\min} ... n_j^{\max} \cdot f_j^{\max}\} \quad \text{(number of nests)} \quad (11)$$

$$y_j^{o_1} \in \{0,1\} \quad \text{(front orientation)} \quad (12)$$

$$y_j^{o_2} \in \{0,1\} \quad \text{(side orientation)} \quad (13)$$

3.2 Dynamic Programming on a Dedicated Shelf Level

All product sets are dedicated to the specific shelf levels and can't be placed on other shelves. Allocating products on only one possible shelf is similar to choosing items in a knapsack problem. In the case of a knapsack problem, a set of items is chosen, each of which has a particular weight and value, such that the overall value is maximized without exceeding the capacity of the knapsack. Due to the practical value many versions of this knapsack problem have been studied by the researches, but even the simplest discrete one is a binary NP-hard. Most of the known knapsack problems are pseudo-polynomially solvable, i.e. the complexity is limited by the number of variables in the instance and the magnitude of the largest coefficient (Pisinger, 1995).

In the current study, each product is defined by the width of the facing (weight), and the unit profit (value). The shelf length corresponds to the knapsack weight capacity. The knapsack problem is an example of a combinatorial optimization problem, in which the goal is to maximize the values of the items chosen to the knapsack without exceeding its capacity. Due to this fact we propose a knapsack algorithm for each shelf level which we extended by additional capping and nesting parameters. Dynamic programming is one of the exact techniques for solving such problems. In the dynamic programming the solutions for each of the sub-problems are calculated once and next it is saved in a table in order to use it on later steps.

We propose a method for solving the classical knapsack problem with caps and nests parameters and a shelf length constraint which stores the calculated values in the table. We propose to create an auxiliary table in which the number of rows corresponds the number of products and a number of columns corresponds the shelf length. Before calling the function the auxiliary table is checked to see if the required value was calculated earlier, next the profit value is calculated and saved for future use.

The proposed algorithms executes as follows. Create a set of facings items, capped items, nested items. The items in this algorithm may be created by single facing or grouped by some facings with cappings (or nestings) above facings. The profit of such group is higher than the profit of the same facings number comparing to single facings. So there will be a set of $\tilde{N}$ items $(k = 1,..., \tilde{N})$, each with its width w_k and profit p_k.

1. For each product j define the set of N'_j items $(k' = 1,..., N'_j)$ by 1 facing with the

following parameters:

$$p_{k'} = p_j^u \qquad \text{(profit)}$$

$$w_{k'} = \begin{Bmatrix} p_j^d, & \text{if } p_j^{o_2} = 1 \wedge p_j^w \le s^d \\ p_j^w, & \text{if } p_j^{o_1} = 1 \wedge p_j^d \le s^d \\ \min(p_j^w, p_j^d) & \text{if } (p_j^{o_2} = 1 \wedge p_j^w \le s^d) \wedge (p_j^{o_1} = 1 \wedge p_j^d \le s^d) \end{Bmatrix} \qquad \text{(width)}$$

$$h_{k'} = p_j^h \qquad \text{(height)}$$

2. For each product j where caps exist ($c_j^{\max} > 0$) define the set of N_j'' grouped items

   $(k'' = 1,...,N_j'')$ by $\left\lceil \frac{p_j^h}{w_{k'}} \right\rceil$ facings with $\min(\left\lfloor \frac{s^h - p_j^h}{w_{k'}} \right\rfloor, c_j^{\max})$ caps above them with the

   following parameters:

$$p_{k''} = p_j^u (\left\lceil \frac{p_j^h}{w_{k'}} \right\rceil + \min(\left\lfloor \frac{s^h - p_j^h}{w_{k'}} \right\rfloor, c_j^{\max})) \qquad \text{(profit)}$$

$$w_{k''} = w_{k'} \left\lceil \frac{p_j^h}{w_{k'}} \right\rceil \qquad \text{(width)}$$

$$h_{k''} = p_j^h + \min(\left\lfloor \frac{s^h - p_j^h}{w_{k'}} \right\rfloor, c_j^{\max}) w_{k'} \qquad \text{(height)}$$

3. For each product j where nests exist ($n_j^{\max} > 0$) define the set of N_j''' grouped items

   $(k''' = 1,...,N_j''')$ by 1 facing with $\min(\left\lfloor \frac{s^h - p_j^h}{p_j^h p_j^n} \right\rfloor, n_j^{\max})$ nests above them with

$$p_{k'''} = p_j^u (1 + \min(\left\lfloor \frac{s^h - p_j^h}{p_j^h p_j^n} \right\rfloor, n_j^{\max})) \qquad \text{(profit)}$$

$$w_{k'''} = w_{k'} \qquad \text{(width)}$$

$$h_{k'''} = p_j^h + \min(\left\lfloor \frac{s^h - p_j^h}{p_j^h p_j^n} \right\rfloor, n_j^{\max}) p_j^h p_j^n \qquad \text{(height)}$$

The numbers of generated items are following:

$\forall (j)[N_j' \le f_j^{\max}]$ (number of single items of facings)

$\forall(j)[N''_j \leq \left\lfloor \frac{f_j^{\max}}{\left\lceil \frac{p_j^h}{w_{k'}} \right\rceil} \right\rfloor]$ (number of grouped facings with caps)

$\forall(j)[N'''_j \leq f_j^{\max}]$ (number of single facings with nests)

$\forall(j)[N'_j + N''_j + N'''_j \leq 2p_j^s]$ (supply limit constraint)

$\forall(j)[N_j = N'_j + N''_j + N'''_j]$ (number of generated based on the product j)

$\tilde{N} = \sum_{j=1}^{P}(N'_j + N''_j + N'''_j)$ (total number of generated items)

After defining the sets of products the decision about what item to put into the knapsack is conducted. The items are defined as grouped and single ones but only one item could be placed into the knapsack. For this reason the items must be marked in order to identify which ones could be placed into the knapsack and which one not.

4. Mark the set of capped items by $g_{jk''}$ $(g_{jk''} = 1,\dots,N''_j)$ and corresponding to them single facings items with the same mark.
5. Mark the set of nested items by $g_{jk'''}$ $(g_{jk'''} = 1,\dots,N'''_j)$ and corresponding to them single facings items with the same mark.
6. The rest separate neither capped nor nested items are marked by $g_{jk'}$

$$g_{jk'} = \begin{cases} 1,\dots,N'_j, & \text{if } c_j^{\max} = 0 \wedge n_j^{\max} = 0 \\ N''+1,\dots,N'_j, & \text{if } c_j^{\max} > 0 \\ N'''+1,\dots,N'_j, & \text{if } n_j^{\max} > 0 \end{cases}.$$

The SSAP can be rewritten as follows. The goal is to decide what items from the newly created sets can be placed on the shelf so that the total items width does not exceed the shelf length and the total profit is as large as possible.

Maximize

$$\max \sum_{k=1}^{\tilde{N}} p_k \tilde{x}_k \tag{14}$$

Subject to the constraints:

$$\sum_{k=1}^{\tilde{N}} p_k \tilde{x}_k \leq s^l \quad \text{(shelf length)} \tag{15}$$

$$\forall(j)[\sum_{k=1}^{N_j} w_k \tilde{x}_k \geq f_j^{\min} w_{k'}] \quad \text{(minimum number of facings)} \tag{16}$$

$\forall(j)[\max_{k''=1,\dots,N''_j}(\tilde{x}_{k''}(h_{k''}-p^h_j)) \ge c^{\min}_j w_{k'}]$ (minimum number of caps) (17)

$\forall(j)[\max_{k'''=1,\dots,N'''_j}\tilde{x}_{k'''}(h_{k'''}-p^h_j) \ge n^{\min}_j p^h_j p^n_j]$ (minimum number of nests) (18)

$$\forall(k',k'': g_{jk'} = g_{jk''})[\max(\sum_{k'=1}^{N'_j}\tilde{x}_{k'}g_{jk'}) + \sum_{k''=1}^{N''_j}\tilde{x}_{k''}g_{jk''} \le g_{jk'}]$$

(capped grouped or single items with the same mark) (19)

$$\forall(k',k'': g_{jk'} = g_{jk'''})[\max(\sum_{k'=1}^{N'_j}\tilde{x}_{k'}g_{jk'}) + \sum_{k'''=1}^{N'''_j}\tilde{x}_{k'''}g_{jk'''} \le g_{jk'}]$$

(nested or single items with the same mark) (20)

Decision variable:

$$\tilde{x}_k = \begin{cases} 1, & \text{if item } k \text{ is placed on the shelf} \\ 0, & \text{otherwise} \end{cases}$$

$\tilde{x}_k \in \{0,1\}$

In the proposed algorithm we model the 0-1 knapsack problem where there are defined families of items so that the items of each family must occupy the part of the shelf within the defined for each family shelf width and height bounds. This means that items from each family must exist on the shelf. Generally the dynamic problem approach is very time consuming for large instances but it is appropriate for small problem cases. The running time of the proposed dynamic programming algorithm is $O(\tilde{N}\cdot s^l)$

7. Decompose the problem into subproblems. Generally 2-dimensional array $M[0..\tilde{N},0..s^l]$ could be used, but for convenience of the separate processing of the items with the same j we propose to create a 3-dimensional array with flexible second dimension which represent one product j.

$\hat{N}_j = |\{j \mid j = k,\ j = 1,\dots,P, k = 1,\dots,\tilde{N}\}|$ - the number items for a product j.

$\hat{k}$ - item index for a product j, $\hat{k} = 1,\dots,\hat{N}_j$.

$\hat{L}_j$ - the occupied space by item j.

$\hat{l}$ - item index for a product j, $\hat{l} = 0,\dots,\hat{L}_j$.

8. Sort $\tilde{N}$ items in the order non-decreasing group g_{jk}, take neither capped nor nested items, next take capped of nested items, increasing of its order generated number based on the number j of the products they were formed from.

9. The 3-dimensional array is $M[1..P][0..\hat{N}_j][0..s^l]$. So for $1 \le j \le P$, $0 \le \hat{k} \le \hat{N}_j$ and $0 \le l \le s^l$ $M[j][\hat{k}][l]$ contains the maximum total profit as a solution of the subproblem. $M[1..P][\hat{N}_j][s^l]$ represents the solutions of the subproblems for each product j.

```
{
  for j = 1 to P
    Initialize M[j][0..N̂_j][0..s^l]
  end
  for j = 1 to P
    Find best L̂_j
  end
  Define what items to place on the shelf X[1..P][0..N̂_j][L̂_j]
}
```

The initial settings are:

$M[j][\hat{k}][0] = 0$ for $\hat{k} = 1,..., \hat{N}_j$ - no space for items.

$M[j][0][l] = 0$ for $0 \le l \le s^l$ - no items selected.

Add some temporary parameters.

$\tilde{y}^{sg} = \begin{Bmatrix} 1, \text{ if } g_{j\hat{k}} = g_{j\hat{k}-1} \\ 0, \text{ otherwise} \end{Bmatrix}$ - the same groups of this and previous item.

$\tilde{y}^{pcn} = \begin{Bmatrix} 1, \text{ if } h_{\hat{k}-1} > p_j^h \\ 0, \text{ otherwise} \end{Bmatrix}$ - previous item is capped or nested.

$\tilde{y}^{tcn} = \begin{Bmatrix} 1, \text{ if } h_{\hat{k}} > p_j^h \\ 0, \text{ otherwise} \end{Bmatrix}$ - this item is capped or nested.

t_1 - profit if the item in situation 1 is taken or not.

t_2 - profit if the item in situation 2 is taken or not.

t_3 - profit if the item in situation 3 is taken or not.

b_2 - item backstep for situation 2.

b_3 - item backstep for situation 3.

$\hat{k}''$ - index for items from another group among processed earlier for situation 2.

$\hat{k}'''$ - index for items from another group among processed earlier for situation 3.

The value of an optimal solution considering solutions of the subproblems could be found recursively. Below the possible situations while considering current and the previous item are described:

1. Processed items are from different groups or both of them are neither capped nor nested $(\tilde{y}^{sg}=0 \vee (\tilde{y}^{sg}=1 \wedge \tilde{y}^{pcn}=0 \wedge \tilde{y}^{tcn}=0))$.

$$t_1 = \begin{cases} \max(M[j][\hat{k}][l],\ p_{\hat{k}} + M[j][\hat{k}][l-w_{\hat{k}}]), & \text{if } w_{\hat{k}} \le l \\ M[j][\hat{k}][l], & \text{if } w_{\hat{k}} > l \end{cases}$$

2. Processed items are from the same group, previous item is neither capped nor nested, this item is capped or nested $(\tilde{y}^{sg}=1 \wedge \tilde{y}^{pcn}=0 \wedge \tilde{y}^{tcn}=1)$.

Find the last item from another group among investigated earlier items.

If such item exists, then define b_2 backstep to that item.

$$t_2 = \begin{cases} \max(M[j][\hat{k}][l],\ p_{\hat{k}} + M[j][\hat{k}-b_2][l-w_{\hat{k}}]), & \text{if } w_{\hat{k}} \le l \\ M[j][\hat{k}][l], & \text{if } w_{\hat{k}} > l \end{cases}$$

Else

$$t_2 = \begin{cases} \max(M[j][\hat{k}][l],\ p_{\hat{k}} + M[j][\hat{k}][l-w_{\hat{k}}]), & \text{if } w_{\hat{k}} \le l \\ M[j][\hat{k}][l], & \text{if } w_{\hat{k}} > l \end{cases}$$

Next try not to take all the processed at this point items $(\tilde{y}^{sg}=1 \wedge h_{\hat{k}} \ne h_{\hat{k}-1})$.

$$t_2 = \begin{cases} \max(M[j][\hat{k}][l],\ t_2 - M[j][\hat{k}''][l-w_{\hat{k}}]), & \text{if } w_{\hat{k}} \le l \\ M[j][\hat{k}][l], & \text{if } w_{\hat{k}} > l \end{cases}$$

3. Items are from the same group, previous item is capped or nested, this item is neither capped nor nested $(\tilde{y}^{sg}=1 \wedge \tilde{y}^{pcn}=1 \wedge \tilde{y}^{tcn}=0)$.

Repeat the situation 2. Find the last item from <u>another group</u> among processed earlier items.

If such item exists, then define b_2 backstep to that item.

$$t_2 = \begin{cases} \max(M[j][\hat{k}][l],\ p_{\hat{k}} + M[j][\hat{k}-b_2][l-w_{\hat{k}}]), & \text{if } w_{\hat{k}} \le l \\ M[j][\hat{k}][l], & \text{if } w_{\hat{k}} > l \end{cases}$$

Else

$$t_2 = \begin{cases} \max(M[j][\hat{k}][l],\ p_{\hat{k}} + M[j][\hat{k}][l-w_{\hat{k}}]), & \text{if } w_{\hat{k}} \le l \\ M[j][\hat{k}][l], & \text{if } w_{\hat{k}} > l \end{cases}$$

Next try not to take all the processed at this point items $(\tilde{y}^{sg}=1 \wedge h_{\hat{k}} \ne h_{\hat{k}-1})$.

$$t_2 = \begin{cases} \max(M[j][\hat{k}][l],\ t_2 - M[j][\hat{k}''][l-w_{\hat{k}}]), & \text{if } w_{\hat{k}} \le l \\ M[j][\hat{k}][l], & \text{if } w_{\hat{k}} > l \end{cases}$$

Find the last item from <u>this group</u> among investigated earlier items which is neither capped nor nested.

If such item exists, then define b_3 backstep to that item.

$$t_3 = \begin{cases} \max(M[j][\hat{k}][l],\ p_{\hat{k}} + M[j][\hat{k} - b_3][l - w_{\hat{k}}]), & \text{if } w_{\hat{k}} \le l \\ M[j][\hat{k}][l], & \text{if } w_{\hat{k}} > l \end{cases}$$

Else

$$t_3 = \begin{cases} \max(M[j][\hat{k}][l],\ p_{\hat{k}} + M[j][\hat{k}][l - w_{\hat{k}}]), & \text{if } w_{\hat{k}} \le l \\ M[j][\hat{k}][l], & \text{if } w_{\hat{k}} > l \end{cases}$$

Next try not to take all the investigated at this point items $(\tilde{y}^{sg} = 1 \wedge h_{\hat{k}} \ne h_{\hat{k}-1})$.

$$t_3 = \begin{cases} \max(M[j][\hat{k}][l],\ t_3 - M[j][\hat{k}'''][l - w_{\hat{k}}]), & \text{if } w_{\hat{k}} \le l \\ M[j][\hat{k}][l], & \text{if } w_{\hat{k}} > l \end{cases}$$

Choose the best situation with maximum profit.

$M[j][\hat{k}+1, l] = \max(t_1, t_2, t_3)$

In the above mentioned situations some notation mean:

$p_{\hat{k}} + M[j][\hat{k}][l - w_{\hat{k}}]$ - take this item in situation 1.

$p_{\hat{k}} + M[j][\hat{k} - b_2][l - w_{\hat{k}}]$ - take this item in situation 2.

$p_{\hat{k}} + M[j][\hat{k} - b_3][l - w_{\hat{k}}]$ - take this item in situation 3.

$M[j][\hat{k}][l]$ - don't take this item.

The notation above means that the best subset of items that has total length l is either the best subset of items with the same total profit, or the item $\hat{k}$ plus the best subset of items with length $l - w_{\hat{k}}$.

For each j find the appropriate space $\hat{L}_j$ occupied by item considering minimum linear dimensions (16)-(18) as well as shelf length (15) so that $\sum_{j=1}^{P} M[j][\hat{k}][\hat{L}_j] \to \max$. $\sum_{j=1}^{P} M[j][\hat{k}][\hat{L}_j]$ is the solution of the main problem.

10. To find items that must be placed on the shelf generating maximum possible total value, create $X[1..P][0..\hat{N}_j][0..\hat{L}_j]$ array. Items that could be placed on the shelf are defined by t_1, t_2 and t_3 parameters.

$$X[j][\hat{k}][\hat{l}] = \begin{cases} 1, & \text{if item } \hat{k} \text{ of product } j \text{ is placed on the shelf of length } \hat{l} \\ 0, & \text{otherwise} \end{cases}$$

4. Computational Experiments

The computational experiments compare the profit considering capping and nesting parameters (formulas (7)-(8)) and basic problem without capping and nesting. The product parameters were simulated randomly by a normal distribution according to Yang (Yang, 2001) and Bai and Kendall (Bai and Kendall, 2005). Five planogram products sets were generated. The tested shelf lengths were: 250 cm, 375 cm, 500 cm, 625 cm, 750, 875 cm, 1000 cm, 1125 cm, 1375 cm, 1500 cm. The numbers of products that must be allocated on the shelf on the planogram were: 10, 20, 30, 40, 50.

Optimal or feasible solution has been found in IBM ILOG CPLEX Optimization Studio Version: 12.7.1.0. The results were achieved the same within 1 min, 5 min and 15 time limit. Optimal solution was found in 11 cases both with and without caps/nests parameters.

Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε. and 0 compare the profit when the products were allocated with caps/nest method and without it. They show a significant difference of the profit when capping or nesting parameters were included. For the case of 40 products it even raise up to 79,16%. The lowest value of such difference is with the case of 30 products and equals 10,13%. The tables also show in which cases the problem size was small enough that optimal solution has been found. In these 11 cases the dynamic programming could be appropriate.

Table 1. *Comparison of the profit when the product were allocated with caps/nest method and without it;.* - optimal solution*

Products	Shelf width	Profit including caps/nests parameters is better	Products	Shelf width	Profit including caps/nests parameters is better
	250	10,22%*	30	500	10,13%*
	375	22,73%*		625	29,19%*
	500	24,67%*		750	40,51%
	625	23,12%*		875	47,90%
	750	19,94%*		1000	47,90%
10	875	17,31%		1125	43,58%
	1000	15,81%		1250	40,09%
	1125	14,20%		1375	36,91%
	1250	13,38%		1500	33,94%
	1375	12,43%	40	625	61,56%*
	1500	11,87%		750	74,00%
	500	27,85%*		875	71,55%
	625	32,79%*		1000	76,59%
	750	39,75%*		1125	79,13%
20	875	49,58%		1250	78,66%
	1000	52,05%		1375	79,16%
	1125	54,89%		1500	78,68%
	1250	56,57%	50	875	36,68%

	1375	56,72%		1000	45,51%
	1500	56,64%		1125	37,39%
				1250	33,32%
				1375	30,02%
				1500	27,50%

Source: *Own elaboration.*

Table 2. *Statistics on comparison of the profit when the product were allocated with caps/nest method and without it*

Products	All widths	Profit including caps/nests parameters is better
10	Min	10,22%
	Avg	16,88%
	Max	24,67%
20	Min	27,85%
	Avg	47,43%
	Max	56,72%
30	Min	10,13%
	Avg	36,68%
	Max	47,90%
40	Min	61,56%
	Avg	74,92%
	Max	79,16%
50	Min	27,50%
	Avg	35,07%
	Max	45,51%

Source: *Own elaboration.*

5. Conclusions and Future Research

In retail outlets, shelf space is considered to be a scarce resource and its management is central to ensuring convenience to customers and conducting profitable business by the retailers.

In this paper, we investigate SSAP in which there is no need to find the appropriate shelf for each product, i. e. the shelves for the defined types of the products are selected in advance. We explained the cases in which the product may be assigned to the shelf in advance based on its quality level, vertical shelf level or package type weight and size.

Via computational experiments we demonstrate the usefulness of considering capping and nesting parameters in the shelf space allocation models. The experiments also analyzed the problem sizes and suggests in which 11 cases from 55 cases the problem size was small enough so that dynamic programming could be used. The future research would consider implementation of the dynamic programming in more complicated SSAP where some products could be assigned to shelf in advance.

References:

Bai, R., Kendall, G. 2005. An investigation of automated planograms using a simulated annealing based hyper-heuristic. Operations Research/ Computer Science Interfaces Series, 32, 87-108.

Behe, B.K., Zhao, J., Sage, L., Huddleston, P.T., Minahan, S. 2013. Display signs and involvement: The visual path to purchase intention. International Review of Retail, Distribution and Consumer Research, 23(5), 511-522.

Burke, R.R. 2006. The third wave of marketing intelligence. In Retailing in the 21st Century: Current and Future Trends, 113-125.

Coskun, M.E. 2012. Shelf Space Allocation: A Critical Review and a Model with Price Changes and Adjustable Shelf Heights. M.Sc., Thesis. http://digitalcommons.mcmaster.ca/opendissertations/7100/.

Desrochers, D.M., Nelson, P. 2006. Adding consumer behavior insights to category management: Improving item placement decisions. Journal of Retailing, 82(4), 357-365.

Ebster, C., Garaus, M. 2011. Store Design and Visual Merchandising: Creating Store Space That Encourages Buying. Store Design and Visual Merchandising: Creating Store Space That Encourages Buying. Business Expert Press.

Elbers, T. 2016. The Effects of In-Store Layout- and Shelf Designs on Consumer Behaviour. Wageningen UR, 1-22.

Evans, J.R. 2011. Retailing in perspective: The past is a prologue to the future. International Review of Retail, Distribution and Consumer Research, 21(1), 1-31.

Hansen, J.M., Raut, S., Swami, S. 2010. Retail Shelf Allocation: A Comparative Analysis of Heuristic and Meta-Heuristic Approaches. Journal of Retailing, 86(1), 94-105.

Pisinger, D. 1995. Algorithms for Knapsack Problems. DIKU University of Copenhagen Denmark Diss. Dover. http://www.diku.dk/OLD/publikationer/tekniske.rapporter/1995/95-1.ps.gz.

Randhawa, K., Saluja, R. 2017. Does Visual Merchandising have an Effect on Consumer Impulse Buying Behavior? Journal of General Management Research, 4(2), 58-71.

Simonson, I. 1999. The effect of product assortment on buyer preferences. Journal of Retailing, 75(3), 347-370.

Simonson, I., Tversky, A. 1992. Choice in Context: Tradeoff Contrast and Extremeness Aversion. Journal of Marketing Research, 29(3), 281-295.

Varley, R. 2005. Retail Product Management: Buying and Merchandising, Second Edition. Retail Product Management: Buying and Merchandising, Second Edition. Routledge Taylor & Francis Group.

Wongkitrungrueng, A., Valenzuela, A., Sen, S. 2018. The Cake Looks Yummy on the Shelf up There: The Interactive Effect of Retail Shelf Position and Consumers' Personal Sense of Power on Indulgent Choice. Journal of Retailing, 94(3), 280-295.

Yang, M.H. 2001. Efficient Algorithm to Allocate Shelf Space. European Journal of Operational Research, 131(1), 107-118.