Journal of Retailing and Consumer Services 58 (2021) 102274

Contents lists available at ScienceDirect

Journal of Retailing and Consumer Services

journal homepage: http://www.elsevier.com/locate/jretconser

Integrated planning for product selection, shelf-space allocation, and replenishment decision with elasticity and positioning effects

Gwang Kim [a], Ilkyeong Moon [b,*]

[a] Department of Industrial Engineering, Seoul National University, 1, Gwanak-ro, Gwanak-gu, Seoul, Republic of Korea
[b] Department of Industrial Engineering and Institute for Industrial Systems Innovation, Seoul National University, 1, Gwanak-ro, Gwanak-gu, Seoul, Republic of Korea

ARTICLE INFO

Keywords:
Shelf-space allocation
Product selection
Replenishment decisions
Non-linear programming

ABSTRACT

As the retail industry is growing larger and more diversified, retailers' decisions about product selection, shelf-space-allocation, and replenishment become more important and challenging. This paper is to present a model for shelf-space allocation with product selection and replenishment decisions to maximize the retailer's profit. The model is based on a two-dimensional display space in which all shelves and products have widths and heights and includes factors that influence demand for each product, such as space and cross-space elasticities and positioning effects. The integrated model presented is mixed-integer non-linear programming (MINLP) because the demand function is non-convex. This research proposes two heuristic algorithms (tabu search and genetic) to solve the MINLP problem. The results show the effectiveness and efficiency of these algorithms by comparing the outputs to the MINLP optimal solution for small data sets and comparing the algorithm performances for large data sets. The solution methodologies expect to support a simultaneous decision-making process for retailers to maximize their revenue.

1. Introduction

Shelf-space allocation, among various retail decision-making processes, is one of the essential issues for retailers. It can affect customers' demand and cost of products, which results in retailers' profit (Corstjens and Doyle, 1981; Gajjar and Adil, 2011; Pizzi and Scarpi, 2016). Most importantly, managers of small retail shops, which do not have sufficient shelf space, should weigh decisions about the types and numbers of products to sell and when and where to display them on the shelves. These decisions influence the sales of products (Chen et al., 2011; Eisend, 2014; Hübner, 2017; Vincent et al., 2020). Therefore, retailers need effective and efficient systems that help them in decision making about shelf-space allocation.

Recent research on the shelf-space allocation has focused on two factors retailers should consider. First, to display products (assuming that each product has a rectangular shape) on shelves, retailers should solve a two-dimensional allocation problem because products have length and height. Second, customer demand for products can be influenced by the proximity of similar products and by the location where the products are displayed (Chen et al., 2006). Fig. 1 shows a shelf-space allocation example that has ten products on six shelves.

Before (or while) retailers make decisions on shelf-space allocations, they should take product selection into account. The product selection means to determine a set of products within a category that a retailer intends to display on the shelves to maximize retailers' profit. The solution can help retailers reduce costs associated with displaying slow-selling instead of fast-selling products (Borle et al., 2005; Kahn and Lehmann, 1991; Shioda et al., 2011). It can also reduce the likelihood of out-of-stock (OOS) events that cause substantial loss of profits (Dass & Kumar, 2012; García-Arca et al., 2020). As the number of brands within a category increases and customer brand perceptions diversifies, retailers should more consider both product selection and shelf-space allocation problems because not all the brands can be displayed in stores (Eisend, 2014; Spieth et al., 2019).

After deciding the product selection and shelf-space allocation, retailers should consider inventory management of the selected products. It means that retailers decide the timing and amount of orders for each product to minimize the costs related to the inventory and orders of the products (Benkherouf et al., 2017; Düsterhöft et al., 2020; Hariga et al., 2007). Some papers presented a shelf-space allocation model with the replenishment decisions (Önal et al., 2016; Rabbani et al., 2016; Zhao et al., 2016). For example, when a fixed cost of replenishment is high,

* Corresponding author.
E-mail addresses: gwangkim91@snu.ac.kr (G. Kim), ikmoon@snu.ac.kr (I. Moon).

https://doi.org/10.1016/j.jretconser.2020.102274
Received 9 January 2020; Received in revised form 28 July 2020; Accepted 28 July 2020

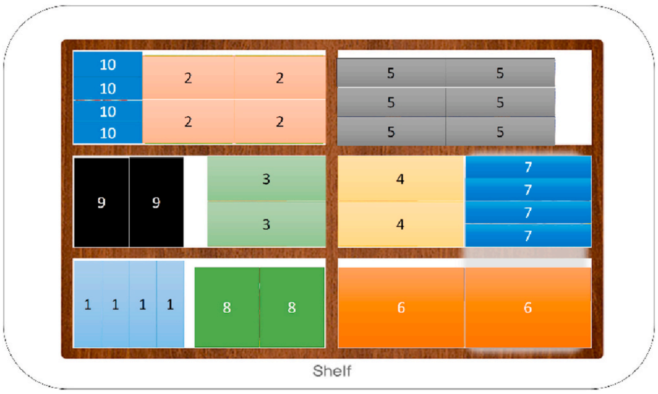

Fig. 1. Example of shelf-space allocation.

retailers want to order a large number of products to reduce the replenishment frequency (Feng, 2018).

In general, retailers conduct the decision-making process in the following order: product selection, shelf-space allocation, and replenishment decision. If retailers manage these three operations individually, they may obtain sub-optimal results. To avoid sub-optimal results, recent research has provided integrated models (Hariga et al., 2007; Hübner and Schaal, 2017a; Rabbani et al., 2018; Zhao et al., 2016). This study also aims at providing an integrated model for the shelf-space allocation considering product selection and replenishment decisions.

This study not only develops an integrated model but also reflects realistic factors of the retail environment. To reflect the realities of the retail environment, this model considers two-dimensional shelves/products and includes factors that influence customer demand for products. We formulate the integrated model considering realistic situation in Section 3. To solve the model effectively and efficiently, we provide solution methodologies based on the greedy and meta-heuristic algorithms. It presents comparisons of the results to show the accuracy and efficiency of the algorithms in finding solutions. Subsequently, we present experimental results of the algorithms to show that the algorithms are competitive at solving the large data sets for which LINGO cannot be used to find optimal solutions within a specific time.

The remainder of the paper is organized as follows. Section 2 presents a review of the previous studies on the problems addressed in this paper. Section 3 describes assumptions and formulation that features the shelf-space allocation according to product selection and replenishment decisions. Sections 4 shows that the two proposed heuristics solve the shelf-space allocation model in an efficient and robust way. Section 5 provides experimental results for small and large problems. In Section 6, conclusions about the research and future studies suggested.

2. Literature review

Many researchers have focused on shelf-space allocation in the context of optimization problems because sales of products depend on efficient shelf-space management (Bookbinder and Zarour, 2001; Bultez and Naert, 1988; Corstjens and Doyle, 1981; Curhan, 1973; Zufryden, 1986). Reisi et al. (2019) and Kuiti et al. (2019) presented strategic shelf-space allocation decisions of retailers in a supply chain context. In addition to developing the space-allocation model, researchers have been interested in algorithm development for the shelf-space allocation problem (Borin et al., 1994; Lim et al., 2004; Yang, 2001).

Some studies have provided integrated models that include product selection and replenishment decisions. Borin et al. (1994), Hansen and Heinsbroek (1979), and Irion et al. (2012) attempted to combine the product assortment (selection) problem with the shelf-space allocation problem. Urban (1998) developed a joint optimization model that integrates inventory management, product assortment, and shelf-space allocation; in this formulation, demand for a product is affected by the inventory information on it. Hwang et al. (2005) developed an integrated model that includes replenishment decisions and considered vertical location (positioning) and space and cross-space elasticity effects. Hariga et al. (2007) extended the model proposed by Hwang et al. (2005) by including the product assortment problem and considering showroom and backroom (storage) inventories separately. However, the results for only a small data set (less than four products and four shelves) were presented in this study.

Recent studies on shelf-space allocation not only provided integrated models but also considered factors that influence customer demand, as shown in Table 1. Hwang et al. (2009) and Murray et al. (2010) dealt with two-dimensional shelf-space design and proposed two types of shelf space configurations. Hansen et al. (2010) presented a shelf-space allocation model by considering vertical and horizontal locations. Russell and Urban (2010) introduced product family integrity to the shelf-space allocation problem as a means of product selection, which means that all brands belonging to a product family are displayed together.

Hübner and Kuhn (2011) analyzed the necessity of substitution effects and basic supply-level concepts for shelf-space allocation. They did this according to the results from a model that can solve cases in the "200 or smaller" data set category. Bai et al. (2013) developed a model for two-dimensional shelf-space allocation with positioning effects. They obtained results by using a simulated annealing hyper-heuristic, and their sensitivity analysis showed the robustness of their algorithm. Geismar et al. (2015) solved the two-dimensional shelf-space allocation problem by using a combination of integer programming and an algorithm that was based on a maximum-weight independent set problem.

Chen et al. (2016) and Zhao et al. (2016) presented shelf-space allocation models with replenishment decisions. Their models showed that results based on decisions about considering joint replenishment policies were the most realistic. Some papers proposed the first stochastic shelf-space optimization model from which retailers could effectively use the results obtained under uncertain demands (Hübner & Schaal, 2017a, 2017b; Schaal and Hübner, 2017). However, the model only dealt with the shelf-space allocation problem.

Flamand et al. (2018) proposed retail assortment planning along with store-wide shelf-space allocation by considering four types of product category interdependence. However, the problem they constructed does not include positioning effects and replenishment decisions. Rabbani et al. (2018) and Düsterhöft et al. (2020) designed three-dimensional shelf-space models with positioning effects. The studies did not consider either replenishment decisions or product selection, respectively.

There are some distinctive features of our model that we show in Table 1. First, this study presents an integrated model for the two-dimensional shelf-space allocation with product selection and replenishment decisions. This model is represented as an optimization problem because retailers want optimal decision-making under constraints associated with products and shelves. Second, the integrated model we propose considers elasticity (space and cross-space) and positioning (vertical and horizontal) effects to reflect the realities of the retail environment. The effects are different depending on the state of products displayed, and therefore influence customers' demands. Corstjens and Doyle (1981) and Hariga et al. (2007) presented a demand function that reflects elasticity and positioning effects mathematically. Most of the research in Table 1 presents their models by referring to the demand function. Likewise, this study uses the demand function as a term of the objective function. Because the demand function is non-linear, the mathematical model becomes a mixed-integer non-linear programming (MINLP) model. The details of the effects and formulation are presented in Section 3. Third, retailers can manage three operations simultaneously by using our integrated model. However, the model may be somewhat complex to solve the problem. This study presents two solution methodologies (a hybrid tabu search and a hybrid genetic algorithm) to find effective solutions within a reasonable time (see Section

Table 1
Recent studies related to shelf-space allocation.

Author (year)	Elasticity Effect		Positioning Effect		2-Dimensional Shelf-Space	Product Selection	Replenishment Decisions	Solution Methodology
	Space	Cross-Space	Vertical	Horizontal				
Hwang et al. (2009)	✓	✓	✓		✓			GA (genetic algorithm)
Murray et al. (2010)	✓	✓	✓		✓			Algorithms based on branch-and-bound
Hansen et al. (2010)	✓	✓	✓	✓				Heuristic based on Yang (2001), GA
Russell and Urban (2010)	✓		✓	✓		✓		Discrete shelf-partitioning model
Hübner and Kuhn (2011)	✓	✓				✓	✓	MINLP (transformed to a specialized knapsack problem)
Bai et al. (2013)	✓	✓	✓	✓	✓			Approach based on gradient, simulated annealing
Geismar et al. (2015)	✓		✓		✓			Heuristic based on network formulation
Chen et al. (2016)	✓						✓	Optimization under mild assumptions (concave fractional programming)
Zhao et al. (2016)	✓	✓	✓	✓			✓	Simulated annealing based hyper heuristic algorithm
Schaal and Hübner (2017)	✓	✓	✓					Stochastic model
Flamand et al. (2018)	✓	✓				✓		Optimization-based heuristic
Rabbani et al. (2018)	✓	✓	✓	✓	✓	✓		GAMS, GA
Düsterhöft et al. (2020)	✓		✓	✓	✓		✓	Three-step algorithm
This study	✓	✓	✓	✓	✓	✓	✓	Hybrid GA, Hybrid tabu search

4).

3. Mathematical model

In this section, we present an MINLP model for a shelf-space allocation problem in which the product selection and replenishment decisions are considered. The objective of the model is to maximize the total profit per unit of time by simultaneously determining a subset of products to be displayed, the amounts of products displayed on each shelf, the order quantity, and the replenishment cycle time for each product. The replenishment cycle time means the time interval between the time an order is placed and the time the next order is placed. The replenishment cycle time is equivalent to the length of the replenishment cycle. Retailers can decide the unit of time, which can be a minute, an hour, or a day, or another chosen unit of time, depending on the problem situation. The definitions of parameters and decision variables used in the model are presented as follows:

Parameters	
N	Set of product types
W	Width of a shelf
H	Height of a shelf
K	Set of shelf columns
R	Set of shelf rows
w_i	Width of product i
h_i	Height of product i
$g_i\left(=\frac{H}{h_i}\right)$	Maximum quantity of product i when products are stacked only vertically on a shelf
s_i^{max}	Maximum quantity of product i that can be displayed on shelf
s_i^{min}	Minimum quantity of product i displayed on shelf if it is selected
p_i	Unit selling price of product i
c_i	Unit purchasing cost of product i
h_{I_i}	Unit inventory cost of product i
h_{b_i}	Unit backroom storage cost of product i
$h_{s_{ikr}}$	Unit display cost of product i displayed on shelf $k \times r$
A_i	Ordering cost of product i
T_i^{max}	Maximum length of replenishment cycle for product i
α_i	Space scale parameter of product i
β_{ikr}	Main space elasticity of product i on shelf $k \times r$
δ_{ij}	Cross-elasticity between product i and j
Decision variables	
y_i	1: If product i is selected in the product selection; 0: otherwise
y_{ikr}	1: If product i is displayed on shelf k(column) $\times$ r(row); 0: otherwise
s_{ikr}	Quantity of product i displayed on shelf k (column) $\times$ r (row)
T_i	Length of replenishment cycle of product i

This model uses the demand function for product i displayed on shelf k (column) $\times$ r (row) as follows ($\forall i \in N,\ \forall k \in K,\ \forall r \in R,\ \forall D_{ikr} \geq 0$):

$$D_{ikr} = \alpha_i (s_{ikr})^{\beta_{ikr}} \prod_{\forall j \neq i} \sum_{m=1}^{|K|} \sum_{n=1}^{|R|} \left(1 - y_j + s_{jmn} y_j\right)^{\delta_{ij}} \tag{1}$$

Corstjens and Doyle (1981) developed a non-linear demand function that takes elasticity effects into account. This study considers two types of elasticity effects (main space and cross space). The main-space elasticity effect is based on the demand of product i that is positively affected by an increase in the shelf space that it occupies. β_i reflects the main-space elasticity. The demand of product i is also influenced by the amount of space that other products occupy, referred to as the cross-space elasticity. δ_{ij} reflects the cross-space elasticity between products i and j. If two products are substitutable, then the cross-space elasticity effect is negative ($\delta_{ij} < 0$); if two products are complements of each other, then the effect is positive ($\delta_{ij} > 0$).

Hariga et al. (2007) integrated positioning effects into the demand function developed by Corstjens and Doyle (1981). The positioning effect refers to the effect of changes in product location on product sales. It is related to consumer's eye-level position. Dreze et al. (1994) experimentally found the importance of the positioning effect. To consider the positioning effect, this study uses β_{ikr} instead of β_i. β_{ikr} combines the positioning effect (shelf $k \times r$) with the main-space elasticity (β_i). A shelf $k \times r$ means a k^{th} column and r^{th} row shelf, as shown in Fig. 2. Previous studies have considered the positioning effect with both vertical and horizontal locations (Bai et al., 2013; Hansen et al., 2010; Rabbani et al., 2018; Zhao et al., 2016).

The following assumptions are used in the integrated model:

- $|N|$ types of products are available; some of them are selected for display.
- Each product has a rectangular shape; it is displayed in a two-dimensional display space.
- No product can be rotated because most products feature a face of fixed width and height designed. It reflects real-world display practices in retail spaces.
- There are $|K| \times |R|$ shelves that have the same sizes.
- Each product type can be displayed on only one of the $|K| \times |R|$ shelves.
- Different types of products should not be stacked up and down of a product. In other words, if a product (type i) is located on a shelf first, only the same type can be stacked in the residual space above the product.
- Each product is continuously sent from the storage area in a non-public space (e.g., backroom) to the shelf without incurring a delivery cost.
- Replenishment to the storage area is instantaneous (no lead time).
- The price and purchasing cost of each product are deterministic.
- Inventory level for product i decreases linearly at the rate of D_{ikr} during the replenishment time interval $[0, T_i]$.
- Shortages are not allowed, and no discount factor is considered.

This study focuses on the value of D_{ikr}, which varies depending on where (on which shelf) products are displayed. The value of D_{ikr} becomes constant once the shelf-space allocation is decided. Then, the replenishment cycle time to the storage area, T_i, and order quantity, $q_i = \left(\sum_{k=1}^{|K|}\sum_{r=1}^{|R|} D_{ikr}\right) \times T_i$, also are decided by maximizing the net profit per replenishment cycle time of the product. The time of the order may change with lead time and shortage cost; however, the replenishment cycle time to the storage area does not change because the value of D_{ikr} is constant. Therefore, research on shelf-space allocation in relation to replenishment decisions assumed zero lead time (Düsterhöft et al., 2020; Önal et al., 2016; Rabbani et al., 2016; Zhao et al., 2016). As in previous research, we assume that replenishment to the storage area is instantaneous (no lead time).

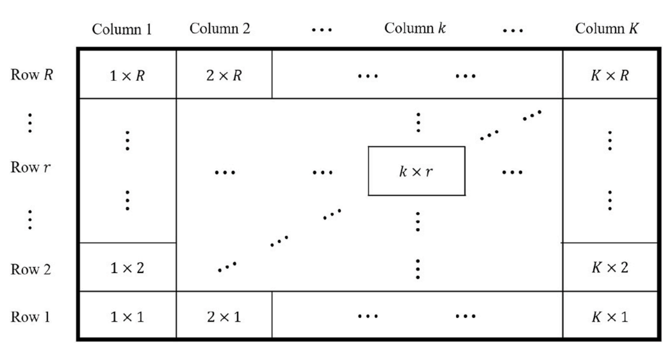

Fig. 2. Graphical representation of $K \times R$ shelves.

3.1. Mixed-integer non-linear programming formulation

The net profit per unit of time of product i ($\forall i \in N$) is defined as follows (Zhao et al., 2016):

$$\pi_i = \sum_{k=1}^{|K|}\sum_{r=1}^{|R|}(p_i - c_i)D_{ikr} - h_{I_i}\left(\sum_{k=1}^{|K|}\sum_{r=1}^{|R|} s_{ikr} + q_i/2\right) - h_{b_i}q_i - \sum_{k=1}^{|K|}\sum_{r=1}^{|R|} h_{s_{ikr}} s_{ikr} - \frac{A_i}{(1 - y_i + T_i y_i)} y_i \tag{2}$$

q_i is the economic order quantity of product i in the model and $q_i = \left(\sum_{k=1}^{|K|}\sum_{r=1}^{|R|} D_{ikr}\right) \times T_i$, ($\forall q_i \geq 0$). According to the assumption, the value of D_{ikr} means the demand for product i on shelf $k \times r$ per unit of time. To help understand Equation (2), we present the net profit per replenishment cycle time of product i ($\forall i \in N$), which consists of five elements:

- Gross profit is equal to $\sum_{k=1}^{|K|}\sum_{r=1}^{|R|}(p_i - c_i)D_{ikr}T_i$
- Inventory cost is equal to $h_{I_i}\left(\sum_{k=1}^{|K|}\sum_{r=1}^{|R|} s_{ikr} + q_i/2\right)T_i$ (including backroom and display inventory)
- Backroom storage cost is equal to $h_{b_i}q_iT_i$
- Display cost is equal to $\left(\sum_{k=1}^{|K|}\sum_{r=1}^{|R|} h_{s_{ikr}} s_{ikr}\right)T_i$
- Ordering cost is equal to $A_i y_i$

The above elements are based on the five elements of the profit function presented by Zhao et al. (2016). The first element is the gross profit where $(p_i - c_i)$ is the unit profit of product i and q_i is the amount of the product sold (order quantity). The second element is the inventory cost where $\sum_{k=1}^{|K|}\sum_{r=1}^{|R|} s_{ikr} + q_i/2$ means the average inventory, including backroom and display inventory, for product i during its replenishment cycle time. The third element is the backroom storage cost where the cost is proportional to order quantity q_i. The fourth element is the display cost. The fifth element is the ordering cost which is charged if product i is ordered.

In our model, some products might not be displayed because a store does not have sufficient shelf space. That is, if product i is not selected, the value of T_i is set to zero; and the denominator of the fifth element in π_i is equivalent to zero if $A_i y_i / T_i$ is used in the ordering cost per unit of time. To prevent from setting zero to the denominator, we used $A_i y_i / (1 - y_i + T_i y_i)$ instead of $A_i y_i / T_i$. The objective function is to maximize the total net profit of all displayed products per unit of time. To deal with the net profit per unit of time, the net profit per unit of time of product i, π_i, is defined as the net profit per replenishment cycle time of the product divided by the replenishment cycle time of the product.

The MINLP formulation of the integrated model is

$$\text{Max} \quad \sum_{i=1}^{|N|} \pi_i \tag{3}$$

$$y_{ikr} \leq y_i, \quad \forall i \in N, k \in K, r \in R \tag{4}$$

$$y_{ikr} \cdot s_i^{min} \leq s_{ikr} \leq y_{ikr} \cdot s_i^{max}, \quad \forall i \in N, k \in K, r \in R \tag{5}$$

$$\sum_{k=1}^{|K|} \sum_{r=1}^{M} y_{ikr} = y_i, \quad \forall i \in N \tag{6}$$

$$T_i \leq y_i \cdot T_i^{max}, \quad \forall i \in N \tag{7}$$

$$\sum_{i=1}^{|N|} w_i \cdot \left\lceil \frac{s_{ikr}}{g_i} \right\rceil \leq W, \quad \forall k \in K, r \in R \tag{8}$$

$$y_i \in \{0, 1\}, \quad \forall i \in N \tag{9}$$

$$y_{ikr} \in \{0, 1\}, \quad \forall i \in N, k \in K, r \in R \tag{10}$$

$$s_{ikr} \in Z^+, \quad \forall i \in N, k \in K, r \in R \tag{11}$$

$$T_i \geq 0, \quad \forall i \in N \tag{12}$$

Constraint (4) means that if a product is not selected, then the products cannot be displayed on any shelves. Constraint (5) ensures lower and upper bounds on the displayed quantity for selected products. Constraint (6) guarantees that each selected product must be displayed on only one shelf among $|K| \times |R|$ shelves. Constraint (7) limits the replenishment cycle time for each product selected. Constraint (8) follows geometric rules such that no selected products overlap according to the two-dimensional display space. In addition, Constraint (8) ensures the same type of products should be first stacked vertically. Then, the remaining products can be stacked in the next residual space within the shelf only if there is no room to be stacked above the product. For example, for product i, we assume that $g_i = 3$ and $s_{ikr} = 7$. This means that two vertical bundles of three products and one vertical bundle of one product ($2 \times 3 + 1 \times 1 = 7$) are displayed on shelf $k \times r$. Constraint (8) has a ceiling function that includes a decision variable, but it can be converted into linear constraints by adding auxiliary variables. For each product, we can calculate the value of $g_i\left(= \frac{H}{h_i}\right)$. If $g_i = M_i$ for product i, M_i auxiliary integer variables $(AU_{ikr}^1, AU_{ikr}^2, \ldots, AU_{ikr}^{M_i})$ are used to linearize Constraint (8). The M_i auxiliary variables mean the number of bundles consisting of one, two, …, and M_i products stacked, respectively, on shelf $k \times r$. The revised MINLP formulation with the two types of linear constraints is

$$\text{Max} \sum_{i=1}^{|N|} \pi_i \tag{13}$$

$$y_{ikr} \leq y_i, \quad \forall i \in N, k \in K, r \in R \tag{14}$$

$$y_{ikr} \cdot s_i^{min} \leq s_{ikr} \leq y_{ikr} \cdot s_i^{max}, \quad \forall i \in N, k \in K, r \in R \tag{15}$$

$$\sum_{k=1}^{|K|} \sum_{r=1}^{M} y_{ikr} = y_i, \quad \forall i \in N \tag{16}$$

$$T_i \leq y_i \cdot T_i^{max}, \quad \forall i \in N \tag{17}$$

$$s_{ikr} = 1 \cdot AU_{ikr}^1 + 2 \cdot AU_{ikr}^2 + \ldots + M_i \cdot AU_{ikr}^{M_i}, \ \forall i \in N, k \in K, r \in R \tag{18a}$$

$$\sum_{i=1}^{|N|} w_i \cdot \left(AU_{ikr}^1 + AU_{ikr}^2 + \ldots + AU_{ikr}^{M_i}\right) \leq W, \quad \forall k \in K, r \in R \tag{18b}$$

$$y_i \in \{0, 1\}, \quad \forall i \in N \tag{19}$$

$$y_{ikr} \in \{0, 1\}, \quad \forall i \in N, k \in K, r \in R \tag{20}$$

$$s_{ikr} \in Z^+, \quad \forall i \in N, k \in K, r \in R \tag{21}$$

$$T_i \geq 0, \quad \forall i \in N \tag{22}$$

$$AU_{ikr}^{a_i} \in Z^+, \quad \forall i \in N, k \in K, r \in R, \ \forall a_i \in \{1, 2, \ldots, M_i\} \tag{23}$$

Using the $(M_1 + M_2 + \ldots + M_N)$ auxiliary variables for all products, two types of linear constraints (Constraints (18a) and (18b)) can be introduced to replace the nonlinear constraint. We use the two types of the linear constraints to solve the problem, instead of Constraint (8). Constraints (9) and (10) define y_i and y_{ikr} as binary variables, respectively. Constraint (11) ensures that s_{ikr} is a positive integer. Constraint (12) indicates that the replenishment cycle time for each product is a positive number. Constraint (23) added in the revised MINLP means that $AU_{ikr}^{a_i}$ is a positive integer.

The MINLP model can be reformulated by using a piecewise linearization technique (Gajjar and Adil, 2010; Irion et al., 2012; Tsao et al., 2014). However, Gajjar and Adil (2010) excluded cross-space elasticity from the MINLP model. Irion et al. (2012) and Tsao et al. (2014) assumed that cross-space elasticity is negative ($\delta_{ij} < 0$). If a positive effect is considered, the gap between upper and lower bounds becomes larger. Thus, we develop two heuristic algorithms to solve the MINLP problem in Section 4.

4. Solution algorithms

The optimization model presented in Section 3 has a non-linear and non-convex objective function because of the demand function. Therefore, in a large data set, whose optimal solution can be intractable to compute, the problem might not be solved directly using optimization solvers within a reasonable time (Öztürkoğlu, 2018). In this study, two hybrid heuristic approaches are provided to solve the model efficiently and effectively. The approaches include a greedy algorithm to generate a good initial solution and meta-heuristic methods to find solutions. They can find solutions close to the optimal solution within a reasonable time in large data sets. Section 4 describes two algorithms used to solve the shelf-space allocation model with product selection and replenishment decisions considering elasticity and positioning effects.

4.1. Hybrid tabu search (Algorithm 1)

Tabu search, which is one of the meta-heuristic methods, generates a new solution obtained by relatively simple modifications to the original solution at every iteration. It can also keep the current solution from being trapped in a local optimum. In this paper, a hybrid tabu search is a solution methodology that combines a tabu search with a greedy algorithm. The greedy algorithm is used to generate a good initial solution.

The greedy algorithm focuses on the profitability of each product, which changes depending on the selection of the shelf. The greedy algorithm uses R_{ikr} values to decide the order to allocate the products on shelves. R_{ikr} is defined as the profit per unit length for g_i quantity of product i displayed on shelf $k \times r$ under the same replenishment cycle time ($T_i = 1, \ \forall i \in N$) regardless of the allocation of other products:

$$R_{ikr} = \frac{1}{w_i}\left\{\left((p_i - c_i) - \frac{h_{I_i}}{2} - h_{s_{ikr}}\right)\alpha_i (g_i)^{\beta_{ikr}} - \left(h_{s_{ikr}} + h_{I_i}\right) g_i - A_i\right\} \tag{24}$$

According to the descending order of R_{ikr}, display products i to shelf $k \times r$ as many as possible with satisfying the constraints. The allocation can be an initial solution.

Then, the tabu search changes the descending order of R_{ikr} by switching two consecutive values with each other, so $N\cdot K\cdot R-1$ orders are generated. New solutions are obtained by each new order. A new solution is updated if the profit is better than that of the initial solution. New $N{\cdot}K{\cdot}R-1$ orders are generated according to the best order generated from previous $N{\cdot}K{\cdot}R-1$ orders. While conducting the tabu search, a set of the tabu order is also set to reduce the computation time by preventing the same order from being applied to the heuristic again. The algorithm is repeated until the tabu iteration is terminated. The proposed hybrid tabu search algorithm is referred to as *Algorithm 1*.

Algorithm 1: Hybrid tabu search

```
Input: S (initial sequence), W(width of a shelf)
N (number of product types), K (number of shelf columns), R (number of shelf rows)
Initialize: PROFIT ⟵ 0; tabuSEQ ⟵ ∅; num1 ⟵ random(0,1);
(*random(0,1): an arbitrary constant between 0 and 1)
SEQ_sol ⟵ S; SEQ ⟵ S;
PROFIT ⟵ Algorithm 3(SEQ);
while time limit is not reached do:
  k ⟵ 1;
  while k < N × K × R do:
    SEQ_k ⟵ sequence by switching the k^th and (k + 1)^th cells of SEQ;
    if SEQ_k ∈ tabuSEQ then:
      PROFIT_k ← 0; k ← k + 1;
      continue
    else
      PROFIT_k ⟵ Algorithm 3(SEQ_k);
      k ← k + 1;
    end
  end
  k ⟵ 1;
  PROFIT_temp ⟵ 0; temp ⟵ 0;
  while k < N × K × R do:
    if PROFIT_temp < PROFIT_k then:
      temp ← k;
    end
    k ← k + 1;
  end
  if PROFIT < PROFIT_temp then:
    PROFIT ← PROFIT_temp;
    SEQ_sol ← SEQ_temp;
  end
tabuSEQ ⟵ tabuSEQ ∪ SEQ_temp;
SEQ ⟵ SEQ_temp;
end
Output: SEQ_sol, PROFIT
```

We present the details of the greedy algorithm that generates an initial sequence of the hybrid tabu search and the procedure of the hybrid tabu search (Algorithm 1) as follows:

Phase 1. Calculate R_{ikr} of each product displayed on each shelf. R_{ikr} is defined as the profit per unit length for g_i quantity of product i displayed on shelf $k\times r$ under the same replenishment cycle time ($T_i=1,\ \forall i\in N$) regardless of the allocation of other products:

$$R_{ikr}=\frac{1}{w_i}\left\{\left((p_i-c_i)-\frac{h_{I_i}}{2}-h_{s_{ikr}}\right)\alpha_i(g_i)^{\beta_{ikr}}-\left(h_{s_{ikr}}+h_{I_i}\right)g_i-A_i\right\}$$

Phase 2. Generate an initial sequence (S) that has $N\times K\times R$ cells according to the descending order of R_{ikr}. Each cell includes $(a,\ b,\ c)$: $a\in\{1,2,3,\dots,N\},\ b\in\{1,2,3,\dots,K\},\ c\in\{1,2,3,\dots,R\}$

$$R_{i_1k_1r_1}\geq R_{i_2k_2r_2}\geq R_{i_3k_3,r_3}\geq\dots\geq R_{i_{NKR}k_{NKR}r_{NKR}}$$

$$(i_1,k_1,r_1)\rightarrow(i_2,k_2,r_2)\rightarrow(i_3,k_3,r_3)\rightarrow\dots\rightarrow(i_{NKR},k_{NKR},r_{NKR})$$

4.2. Hybrid genetic algorithm (Algorithm 2)

In contrast to the tabu search, a genetic algorithm (GA) is a method that produces a population as a solution at each iteration. This approach uses the allocation rule based on R_{ikr} to decide the first chromosome of the first generation. Other chromosomes in the first generation are determined by using $N{\cdot}K{\cdot}R$ random numbers between 0 and 1. If a population consists of n chromosomes, n chromosomes are generated in this step. A chromosome consists of $N{\cdot}K{\cdot}R$ genes. The p^{th} gene is represented as p_1^{th} products, p_2^{th} column, and p_3^{th} row. p_1 is equal to $\frac{p-1}{K\cdot R}+1$, p_2 is equal to $\frac{p-\{K\cdot R(p_1-1)\}}{R}+1$, and p_3 is equal to $p-K{\cdot}R(p_1-1)-R(p_2-1)$. The first chromosome in the initial population is decided by the descending order of R_{ikr}. The higher R_{ikr} is, the higher the value the gene corresponding to $(i,\ k,\ r)$ assigns (between 0 and 1). In other chromosomes, each gene is assigned to a value that is a randomly generated number between 0 and 1. The value that the p^{th} gene has is called GV_p. Fig. 3 shows an example of an initial population where N, K, and R are 3, 2, and 2, respectively.

A chromosome is generally represented as a solution. However, in this hybrid GA, a chromosome represents an allocation rule, not a solution. The gene that has the greatest value in the chromosome is selected, then allocation proceeds from the information of the gene. The allocation rule is determined according to the descending order of each gene value. If two values in one chromosome are the same, the gene that has the bigger $(p_i-c_i)/(w_i{\cdot}h_i)$ value is first selected for the allocation rule. If the $(p_i-c_i)/(w_i{\cdot}h_i)$ values are the same, the gene that is the farthest to the left is first selected. If a chromosome that has $N\times K\times R$ genes is selected, we can generate a gene sequence (S) according to the descending order of the gene value (GV_p), $p\in\{1,2,3,\ldots,N\times K\times R\}$:

$$GV_1\geq GV_2\geq GV_3\geq\ldots\geq GV_{NKR}$$

$$(i_1,k_1,r_1)\rightarrow(i_2,k_2,r_2)\rightarrow(i_3,k_3,r_3)\rightarrow\ldots\rightarrow(i_{NKR},k_{NKR},r_{NKR})$$

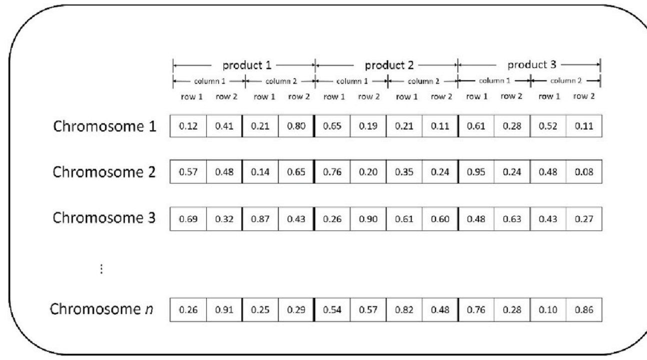

Fig. 3. Example of an initial population.

Algorithm 2 present the procedure of the hybrid genetic algorithm. Each chromosome in the initial population has a solution and profit according to the allocation process. The best solution among all solutions obtained for every allocation iteration becomes the solution of the population.

Algorithm 2: Hybrid genetic algorithm

Input: $P\ (population)=\{SEQ_1,\cdots,SEQ_{NKR}\}\ (SEQ_i{:}\ sequence\ made\ from\ i^{th}\ chromosome)$
$W(width\ of\ a\ shelf), N\ (number\ of\ product\ types),$
$K\ (number\ of\ shelf\ columns), R(number\ of\ shelf\ rows)$
Initialize: $PROFIT\leftarrow 0; num1\leftarrow random(0,1);$
(*$random(0,1)$: an arbitrary constant between 0 and 1)
$P_{cur}\leftarrow P\ (current\ population);$
while *time limit is not reached* **do:**
  $k\leftarrow 1;$
  while $k\ \leq N\times K\ \times R$ **do:**
    $SEQ_k\leftarrow sequence\ made\ from\ k^{th}\ chromosome\ in\ P_{cur};$
    $PROFIT_k\ \leftarrow$ **Algorithm 3**$(SEQ_k);$
    $k\ \leftarrow k+1;$
  end
  $k\leftarrow 1;$
  $PROFIT_{temp}\leftarrow 0; temp\leftarrow 0;$
  while $k\ \leq N\times K\ \times R$ **do:**
    if $PROFIT_{temp}<PROFIT_k$ **then:**
      $temp\ \leftarrow k;$
    end
    $k\ \leftarrow k+1;$
  end
  if $PROFIT<PROFIT_{temp}$ **then:**
    $PROFIT\ \leftarrow PROFIT_{temp};$
  end
  $P_{cur}\leftarrow SEQ_{best}\ \ (sequence\ from\ chromosome\ that\ had\ the\ best\ solution\ in\ P_{cur}$
  $\cup\ n-1\ offspring\ sequences\ from\ chromosomes$
  $generated\ by\ selection, crossover, and\ mutation)$
end
Output: $SEQ_{best},\ PROFIT$

Algorithm 3: Allocating products to shelves using SEQ

Input: $SEQ\ (sequence)$
Initialize: $W_{kr} \longleftarrow W\ \forall k, r; s_{ikr} \longleftarrow 0\ \forall i, k, r; r_i \longleftarrow random(0,1)\ \forall i$
(*$random(0,1)$: an arbitrary constant between 0 and 1)

\# Assigning advertisements to time slots
$p \leftarrow 1;$
while $p \le N \times K \times R$ **do:**
  $i(k,r) \leftarrow i_p(k_p, r_p)$ *in the* p^{th} *cell of* SEQ;
  if $r_i < num1$ or $R_{ikr} < 0$ or Product i is already displayed **then:**
    $p \leftarrow p + 1;$
  continue
  end
  if $w_i \le W_{kr}$ **then:**
    $s_{ikr} \leftarrow \min\left\{s_i^{max}, \left\lfloor \frac{W_{kr}}{w_i} \right\rfloor \cdot g_i\right\};$
    $W_{kr} \leftarrow W_{kr} - w_i \cdot \left\lceil \frac{s_{ikr}}{g_i} \right\rceil;$
  end
  $p \leftarrow p + 1;$
end

\# Updating the objective value
$D_{ikr} \leftarrow \alpha_i (s_{ikr})^{\beta_{ikr}} \prod_{\forall j \neq i} \sum_{m=1}^{|K|} \sum_{n=1}^{|R|} (1 - y_j + s_{jmn} y_j)^{\delta_{ij}}\ \forall i, k, r;$
$T_i \leftarrow \sqrt{A / \left(\frac{h_{I_i}}{2} + h_{b_i}\right) D_{ikr}}$ if $s_{ikr} > 0\ \forall i, k, r;$
forall i **do:**
  $\pi_i \leftarrow \sum_{k=1}^{K} \sum_{r=1}^{M} (p_i - c_i) D_{ikr} - h_{I_i}\left(\sum_{k=1}^{|K|} \sum_{r=1}^{|R|} s_{ikr} + {}^{q_i}/_{2}\right) - h_{b_i} q_i - \sum_{k=1}^{|K|} \sum_{r=1}^{|R|} h_{s_{ikr}} s_{ikr} - \frac{A_i}{(1 - y_i + T_i y_i)} y_i$
end
if $\sum_{i=1}^{|N|} \pi_i > PROFIT$ **then:**
  $PROFIT \leftarrow \sum_{i=1}^{|N|} \pi_i;$
end
Output: $PROFIT$

Table 2
Parameters of the two algorithms.

Algorithm	Parameter	Value
Algorithm 1	# of iterations	$10 \times N$
	# of tabu search iterations	5
Algorithm 2	Population size	N (or 10 if $N \le 10$)
	# of generations	25
	# of iterations for each chromosome	N (or 10 if $N \le 10$)
	Tournament size	5
	Mutation rate	0.05

Table 3
Parameter sets.

Parameter	Value
$(\|\boldsymbol{N}\|, \|\boldsymbol{K}\|, \|\boldsymbol{R}\|)$	Small: (4,2,1); (4,3,1); (5,2,1); (5,3,1); (5,2,2); (6,2,1); (6,3,1); (6,2,2); Large: (10,2,1); (15,3,2); (20,3,3); (30,3,3); (50,3,3); (50,5,2); (75,5,2); (100,5,3);
$\boldsymbol{W}$	30
$\boldsymbol{H}$	20
$\boldsymbol{p_i}$	(integer) Random[20, 40]
$\boldsymbol{c_i}$	(integer) Random[10, 20]
$\boldsymbol{w_i}$	(integer) Random[6, 8]
$\boldsymbol{h_i}$	(integer) Random[4, 6]
$\boldsymbol{\alpha_i}$	(integer) Random[5, 10]
$\boldsymbol{h_{I_i}}$	U[2.0, 4.0]
$\boldsymbol{h_{b_i}}$	U[1.0, 2.0]
$\boldsymbol{A_i}$	50
$\boldsymbol{s_i^{max}}$	12
$\boldsymbol{s_i^{min}}$	1
$\boldsymbol{T_i^{max}}$	3
$\boldsymbol{\beta_{ikr}}$	U[0.2, 0.4]
$\boldsymbol{h_{s_{ikr}}}$	U[1.0, 3.0]
$\boldsymbol{\delta_{ij}}$	U[-0.05, 0.05]

Chromosome selection is one of the generic GA operations in which two parent chromosomes are selected from the current population. Two parent chromosomes are selected through the tournament selection method in the hybrid GA. Also, three types of crossover operations are used in the hybrid GA: arithmetic, uniform, and one point. A mutation is also used to maintain diversity in the population. The population of the next generation consists of the one chromosome that had the best solution in the previous generation and the n-1 offspring chromosomes generated by the operations of selection, crossover, and mutation. If n chromosomes of the next generation are determined, the allocation process, as explained in Algorithm 2, with the n chromosomes is conducted to improve the profit. Updating the population is repeated until the GA iteration is terminated.

4.3. Allocating products to shelves

When a sequence (SEQ_k) is selected from the hybrid tabu search or hybrid genetic algorithm, we allocate products to shelves using the sequence information through Algorithm 3. After D_{ikr} is updated, the corresponding T_i should be updated. If all decision variables but T_i related to π_i are given, then π_i can be a convex function on T_i (the sign of the second derivative of π_i, with respect to T_i, is negative). Thus, the optimal value of T_i can be equal to $\sqrt{A/\left(\frac{h_{I_i}}{2}+h_{b_i}\right)D_{ikr}}$. The details of the procedure are as follows:

5. Computational experiments

The performances of the proposed model for small and large data sets, respectively, are described. The MINLP formulation was solved with LINGO software version 17.0 for the optimal solution. The two solution methodologies proposed in this paper were run with JAVA language in Windows 7 on a PC with an Intel(R) Core (TM) i5-4690 CPU 3.5 GHz with 16.00 GB of RAM.

The MINLP with small data sets could be solved within a reasonable time because the problems have relatively small solution spaces. The performances of the two algorithms were also evaluated by comparing the optimal solutions and computational times. However, for large data sets, only the results obtained by the two algorithms were analyzed because the MINLP could not be solved within a reasonable time. The parameters of the two algorithms are shown in Table 2. *num*1 was set to 0.05, and *num*2 was set to 0.7 in Algorithms 1 and 2. For the crossover operation of Algorithm 2, an arithmetic crossover was used for random numbers less than 0.1, and one-point crossover was used for numbers greater than 0.9; for other numbers, a uniform crossover was used (critical probability = 0.5).

Table 3 shows the parameter sets used for problems with a small and large data set. In most of the parameters, the values or ranges of parameters were adopted from the existing studies on the shelf-space allocation problem (Hariga et al., 2007; Murray et al., 2010; Zhao et al., 2016). U[*a*, *b*] refers to a uniform distribution between *a* and *b*. (integer) Random[a, b] represents a random integer value between *a* and *b*.

5.1. Results for problems with small data sets

We conducted numerical experiments in small data sets that LINGO could solve within a reasonable time. In the experiments, the time limit was set to 3600 s. However, the problems that exceeded the (6, 2, 2) instance could not provide good solutions as well as optimal solutions within the time limit.

Twenty different samples were tested for each problem size. We also used the sample data for the analysis of the effectiveness and efficiency of the algorithms, which was conducted by comparing the optimal solutions and computation times from LINGO. Table 4 shows the results for problems with small data sets.

LINGO software could find optimal solutions for all instances. The average computation time of LINGO ranged between 58.19 and 803.08 s. In contrast, the average computation time of the two algorithms was less than 0.5 s. Algorithm 1 found the optimal solutions in 136 of 160 instances, and Algorithm 2 found the optimal solutions in 147 of 160

Table 4
Results for problems with small data sets.

(N, K, R)	Computation time in seconds (avg, max)			% performance gap (avg, max)	
	LINGO	Algorithm 1	Algorithm 2	Algorithm 1[a]	Algorithm 2[a]
(4, 2, 1)	(58.19, 160.03)	(0.013, 0.02)	(0.08, 0.09)	(0.00, 0.00)	(0.00, 0.00)
(4, 3, 1)	(72.05, 121.80)	(0.023, 0.04)	(0.24, 0.36)	(0.72, 3.50)	(0.00, 0.00)
(5, 2, 1)	(87.22, 144.10)	(0.021, 0.03)	(0.15, 0.37)	(0.63, 6.96)	(0.00, 0.00)
(5, 3, 1)	(236.42, 796.15)	(0.079, 0.14)	(0.31, 0.53)	(0.93, 5.97)	(0.03, 0.70)
(5, 2, 2)	(244.98, 1076.5)	(0.11, 0.14)	(0.50, 0.68)	(0.44, 4.38)	(0.14, 1.15)
(6, 2, 1)	(367.98, 577.8)	(0.045, 0.08)	(0.42, 0.57)	(0.43, 4.85)	(0.00, 0.00)
(6, 3, 1)	(496.03, 963.6)	(0.13, 0.16)	(0.50, 0.78)	(0.10, 1.10)	(0.08, 0.88)
(6, 2, 2)	(803.08, 1010.8)	(0.16, 0.38)	(0.33, 0.94)	(0.75, 7.22)	(0.34, 4.99)

[a] $\left(1-\frac{\textit{objective function value obtained by each algorithm}}{\textit{objective function value obtained by LINGO}}\right)\times 100\%$

Table 5
Results for problems with large data sets.

(N, K, R)	Computation time in seconds (avg, max)		% performance gap (avg, max)			
	Algorithm 1	Algorithm 2	Algorithm 1[a]	Algorithm 2[a]	Algorithm 1[b]	Algorithm 2[b]
(10, 2, 2)	(0.23, 0.41)	(0.46, 0.94)	(0.51, 5.58)	(0.77, 4.36)	(46.36, 62.22)	(46.45, 62.42)
(15, 3, 2)	(1.48, 2.04)	(2.42, .3.75)	(0.23, 2.18)	(3.10, 7.28)	(72,92, 75.50)	(72.05, 74.34)
(20, 3, 3)	(6.76, 8.00)	(10.34, 11.18)	(0.06, 1.94)	(4.02, 8.88)	(83.07, 86.00)	(82.26, 85.43)
(30, 3, 3)	(26.26, 29.38)	(44.35, 46.52)	(0.14, 2.99)	(3.94, 9.35)	–	–
(50, 3, 3)	(101.8, 110.3)	(261.8, 270.7)	(0.005, 0.29)	(4.78, 9.47)	–	–
(50, 5, 2)	(142.5, 154.7)	(322.6, 327.7)	(0.06, 1.67)	(4.80, 9.23)	–	–
(75, 5, 2)	(418.0, 445.5)	(1435.2, 1456.7)	(0.00, 0.00)	(5.07, 9.23)	–	–
(100, 5, 3)	(1035.1, 1093.5)	(4239.9, 4373.1)	(0.01, 0.85)	(0.28, 1.13)	–	–

[a] $\left(1-\frac{\textit{objective function value obtained by each algorithm}}{\textit{best objective function value between Algorithm 1 and 2}}\right)\times 100\%$

[b] $\left(1-\frac{\textit{objective function value obtained by LINGO at 12 hours}}{\textit{objective function value obtained by Algorithm 1(or 2)}}\right)\times 100\%$

instances. For most of the instances, Algorithm 1 and 2 found the optimal solutions. The worst optimality gaps were 7.22% in Algorithm 1 ((6, 2, 2) instance) and 4.99% in Algorithm 2 ((6, 2, 2) instance). Still, the average optimality gaps were less than 1% in Algorithms 1 and 2. It means that Algorithms 1 and 2 can find the optimal solutions for the majority of small data sets.

For the comparison between Algorithms 1 and 2, Algorithm 2 performed a little better than Algorithm 1 for the small data sets. The two algorithms found near-optimal solutions within a second of each other in the experiments with small data sets. The results showed that Algorithms 1 and 2 are appropriate approaches in terms of effectiveness and efficiency. In Section 5.2, we compare the performances of the two algorithms for large data sets.

5.2. Results for problems with large data sets

In this section, we present numerical experiments with data from many products and shelves. Contrary to those presented in Section 5.1, this section mainly deals with the results obtained from the two algorithms because the data sets were too large to solve the MINLP using LINGO within the time limit. Fifty different samples were generated for each problem size to see the results from the two algorithms. A total of 400 instances were used. Table 5 shows the results for problems with large data sets. The last two columns of Table 5 deal with a comparison between feasible solutions obtained by Algorithms and LINGO at 12 h (43,200 s).

The average computation time ranged between 0.25 and 1035.1 s in Algorithm 1 and between 0.46 and 4239.9 s in Algorithm 2. As shown in the fourth and fifth columns of Table 5, the average % performance gaps of Algorithm 1 were lower than those of Algorithm 2. It means that Algorithm 1 found better feasible solutions than Algorithm 2 on average. Specifically, Algorithm 1 was better than Algorithm 2 for all (75, 5, 2) instances. On average, Algorithm 1 performed better and found solutions faster than Algorithm 2 did for large data sets. The results also revealed that Algorithm 2, which was based on a GA, could explore a wider space of possible solutions than Algorithm 1, but it was less likely to find remarkable solutions as the instance size increased. Although Algorithm 1 proved a good approach to this problem, it took more than 1000 s to compute the (100, 5, 3) instance on average.

The last two columns of Table 5 show the gap between the best possible objective value obtained by LINGO at 12 h and Algorithm 1 (or Algorithm 2). The instances ranged from (10, 2, 2) to (20, 3, 3), and we used three instances for each category. In these instances, the objective values of the solutions given by LINGO were approximately 20–50% of those given by Algorithms 1 and 2. For 12 h, upper bounds given by LINGO were decreasing, but the decreasing rates were not significant. As the computation time increased, the optimality gap that LINGO presented tended to decrease. However, the reduction of the optimality gap was less than 5%, on average, from 1 h to 12 h, which was not significant. In other cases, LINGO presented either meaningless upper bounds or feasible solutions whose objective values were small. The average optimality gaps were more than 90%. This implies that alternate solution methodologies are needed to solve the problem effectively and efficiently in large data sets.

5.3. Managerial insights

We analyzed a model for the shelf-space allocation problem with product selection and replenishment decisions. There are some managerial insights into the findings of the model, the heuristic approach, and the analysis. First, the problem is rather complex to solve because many decision processes, which might be intercorrelated, are combined. As mentioned above, retailers need alternate algorithms to obtain good solutions within a reasonable time. The two approaches presented in this paper would be helpful in making decisions on displaying the selected products on the shelves.

Second, it is not always good to display as many brands or products as possible on the shelves. As the number of brands displayed increases, effects on main-space and cross-space elasticities grow and diversify. In particular, the negative cross-space elasticity has a negative effect on product sales. In addition, the positioning effect related to the consumer's eye-level position is important to display the products. Thus, it is recommended that retailers choose and display brands or products by considering these effects, which might be positive or negative.

Third, the expected demands from the two same groups of products displayed on the shelves can be different depending on how much or which shelf each product is assigned. Also, it can be affected depending on what brands near the product are displayed. Therefore, it is recommended that retailers analyze the solutions by using our heuristic approaches, then identify groups of brands and the number of products displayed that will maximize the retailers' revenue. The approaches will help retailers understand the logic of the problem efficiently and execute the allocation effectively.

6. Conclusions

As the retail industry is growing larger and more diversified, retailers' decisions become more crucial and challenging. Decisions about product selection, shelf-space-allocation, and replenishment influence profit, so retailers need a tool to help make determinations on these issues to earn the highest profit. The objective of this paper was to show the development of an integrated model that reflects both the two-dimensional shape of shelves and factors that influence customers' demand for selected products.

The integrated model was based on a mixed-integer non-linear programming that featured a non-convex objective function with linear constraints. This paper also described heuristic and meta-heuristic algorithms of the integrated model to solve the problem efficiently. The problems were solved by using LINGO for problems with small data sets. The performances of the two algorithms for various data sets were also evaluated. The results showed that Algorithms 1 and 2 not only provide high-quality solutions that are equal to or close to the objective function values of the optimal solutions but also efficiently found the solutions in small data sets. Algorithm 1 performed most efficiently and effectively for the larger data sets in the experiments. We expect that these algorithms are useful to many retailers considering product selection, shelf-space allocation, and replenishment decisions in their industry.

There are several research topics for future research. The shelf-space allocation problem features a variety of parameters related to products and shelves that should be accurately estimated according to the historical data of retail stores. If the problem uses parameters obtained by various estimation methods, retailers might find more realistic solutions. Also, retailers should consider the decisions under uncertain demand for each product. To reflect this uncertainty, stochastic programming or robust optimization might be introduced to solve the problem.

Moreover, in particular, because there is no information or incomplete information in the case of new products, it is difficult to estimate the parameters used in our model. Therefore, more realistic estimations of the demand rate might be needed. For example, artificial neural networks can provide a method to forecast the demand of the displayed products using various input information of the product. Also, we can define an action space indicating a change of displayed location for products through reinforcement learning. This might give retailers a strategy that maximizes the expected reward over time. That is, artificial intelligence and machine learning might be effective methodologies to infer the demand or retailers' revenues for the displayed products such as was researched by Lee et al. (2014) and Jenkins et al. (2020). Furthermore, as online retailing evolves, retailers that provide product both online and in stores should deliberate more on the product selection, shelf-space allocation, and replenishment decisions.

Acknowledgments

The authors are grateful for the valuable comments from the anonymous reviewers. This research was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning [Grant number NRF-2019R1A2C2084616].

References

Bai, R., Van Woensel, T., Kendall, G., Burke, E.K., 2013. A new model and a hyper-heuristic approach for two-dimensional shelf space allocation. 4OR 11 (1), 31–55.
Benkherouf, L., Skouri, K., Konstantaras, I., 2017. Inventory decisions for a finite horizon problem with product substitution options and time varying demand. Appl. Math. Model. 51, 669–685.
Bookbinder, J.H., Zarour, F.H., 2001. Direct product profitability and retail shelf-space allocation models. J. Bus. Logist. 22 (2), 183–208.
Borin, N., Farris, P.W., Freeland, J.R., 1994. A model for determining retail product category assortment and shelf space allocation. Decis. Sci. J. 25 (3), 359–384.
Borle, S., Boatwright, P., Kadane, J.B., Nunes, J.C., Galit, S., 2005. The effect of product assortment changes on customer retention. Market. Sci. 24 (4), 616–622.
Bultez, A., Naert, P., 1988. SH. ARP: shelf allocation for retailers' profit. Market. Sci. 7 (3), 211–231.
Chen, Chen, J.-M., Tung, C.-W., 2006. A data mining approach for retail knowledge discovery with consideration of the effect of shelf-space adjacency on sales. Decis. Support Syst. 42 (3), 1503–1520.
Chen, J.-M., Cheng, H.-L., Chien, M.-C., 2011. On channel coordination through revenue-sharing contracts with price and shelf-space dependent demand. Appl. Math. Model. 35 (10), 4886–4901.
Chen, S.-C., Min, J., Teng, J.-T., Li, F., 2016. Inventory and shelf-space optimization for fresh produce with expiration date under freshness-and-stock-dependent demand rate. J. Oper. Res. Soc. 67 (6), 884–896.
Corstjens, M., Doyle, P., 1981. A model for optimizing retail space allocations. Manag. Sci. 27 (7), 822–833.
Curhan, R.C., 1973. Shelf space allocation and profit maximization in mass retailing. J. Market. 37 (3), 54–60.
Dass, M., Kumar, P., 2012. Assessing category vulnerability across retail product assortments. Int. J. Retail Distrib. Manag. 40 (1), 64–81.
Dreze, X., Hoch, S.J., Purk, M.E., 1994. Shelf management and space elasticity. J. Retailing 70 (4), 301–326.
Düsterhöft, T., Hübner, A., Schaal, K., 2020. A practical approach to the shelf-space allocation and replenishment problem with heterogeneously sized shelves. Eur. J. Oper. Res. 282 (1), 252–266.
Eisend, M., 2014. Shelf space elasticity: a meta-analysis. J. Retailing 90 (2), 168–181.
Feng, L., 2018. Dynamic pricing, quality investment, and replenishment model for perishable items. Int. Trans. Oper. Res. 1–18.
Flamand, T., Ghoniem, A., Haouari, M., Maddah, B., 2018. Integrated assortment planning and store-wide shelf space allocation: an optimization-based approach. Omega 81, 134–149.
Gajjar, H.K., Adil, G.K., 2010. A piecewise linearization for retail shelf space allocation problem and a local search heuristic. Ann. Oper. Res. 179 (1), 149–167.
Gajjar, H.K., Adil, G.K., 2011. Heuristics for retail shelf space allocation problem with linear profit function. Int. J. Retail Distrib. Manag. 39 (2), 144–155.
García-Arca, J., Prado-Prado, J.C., Garrido, A.T.G.-P., 2020. On-shelf availability and logistics rationalization. A participative methodology for supply chain improvement. J. Retailing Consum. Serv. 52, 101889.
Geismar, H.N., Dawande, M., Murthi, B., Sriskandarajah, C., 2015. Maximizing revenue through two-dimensional shelf-space allocation. Prod. Oper. Manag. 24 (7), 1148–1163.
Hansen, Heinsbroek, H., 1979. Product selection and space allocation in supermarkets. Eur. J. Oper. Res. 3 (6), 474–484.
Hansen, Raut, S., Swami, S., 2010. Retail shelf allocation: a comparative analysis of heuristic and meta-heuristic approaches. J. Retailing 86 (1), 94–105.
Hariga, M.A., Al-Ahmari, A., Mohamed, A.-R.A., 2007. A joint optimisation model for inventory replenishment, product assortment, shelf space and display area allocation decisions. Eur. J. Oper. Res. 181 (1), 239–251.
Hübner, 2017. A decision support system for retail assortment planning. Int. J. Retail Distrib. Manag. 45 (7/8), 808–825.
Hübner, Kuhn, H., 2011. Retail shelf space management model with space-elastic demand and consumer-driven substitution effects. Technical ReportKU Eichstätt-Ingolstadt. https://doi.org/10.2139/ssrn.1534665.
Hübner, Schaal, K., 2017a. A shelf-space optimization model when demand is stochastic and space-elastic. Omega 68, 139–154.
Hübner, A., Schaal, K., 2017b. An integrated assortment and shelf-space optimization model with demand substitution and space-elasticity effects. Eur. J. Oper. Res. 261 (1), 302–316.
Hwang, H., Choi, B., Lee, G., 2009. A genetic algorithm approach to an integrated problem of shelf space design and item allocation. Comput. Ind. Eng. 56 (3), 809–820.
Hwang, H., Choi, B., Lee, M.-J., 2005. A model for shelf space allocation and inventory control considering location and inventory level effects on demand. Int. J. Prod. Econ. 97 (2), 185–195.
Irion, J., Lu, J.-C., Al-Khayyal, F., Tsao, Y.-C., 2012. A piecewise linearization framework for retail shelf space management models. Eur. J. Oper. Res. 222 (1), 122–136.
Jenkins, P., Wei, H., Jenkins, J.S., Li, Z., 2020. A Probabilistic Simulator of Spatial Demand for Product Allocation arXiv preprint arXiv:2001.03210.
Kahn, B.E., Lehmann, D.R., 1991. Modeling choice among assortments. J. Retailing 67 (3), 274.
Kuiti, M.R., Ghosh, D., Gouda, S., Swami, S., Shankar, R., 2019. Integrated product design, shelf-space allocation and transportation decisions in green supply chains. Int. J. Prod. Res. 1–21.
Lee, H., Kim, S.G., Park, H.-w., Kang, P.J.T.F., Change, S., 2014. Pre-launch new product demand forecasting using the Bass model: a statistical and machine learning-based approach, 86, 49–64.
Lim, A., Rodrigues, B., Zhang, X., 2004. Metaheuristics with local search techniques for retail shelf-space optimization. Manag. Sci. 50 (1), 117–131.
Murray, C.C., Talukdar, D., Gosavi, A., 2010. Joint optimization of product price, display orientation and shelf-space allocation in retail category management. J. Retailing 86 (2), 125–136.
Önal, M., Yenipazarli, A., Kundakcioglu, O.E., 2016. A mathematical model for perishable products with price-and displayed-stock-dependent demand. Comput. Ind. Eng. 102, 246–258.
Öztürkoğlu, Ö., 2018. A bi-objective mathematical model for product allocation in block stacking warehouses. Int. Trans. Oper. Res. 1–27.
Pizzi, G., Scarpi, D., 2016. The effect of shelf layout on satisfaction and perceived assortment size: an empirical assessment. J. Retailing Consum. Serv. 28, 67–77.
Rabbani, M., Salmanzadeh-Meydani, N., Farshbaf-Geranmayeh, A., Fadakar-Gabalou, V., 2018. Profit maximizing through 3D shelf space allocation of 2D display orientation items with variable heights of the shelves. Opsearch 55 (2), 337–360.
Rabbani, M., Zia, N.P., Rafiei, H., 2016. Joint optimal dynamic pricing and replenishment policies for items with simultaneous quality and physical quantity deterioration. Appl. Math. Comput. 287, 149–160.
Reisi, M., Gabriel, S.A., Fahimnia, B., 2019. Supply chain competition on shelf space and pricing for soft drinks: a bilevel optimization approach. Int. J. Prod. Econ. 211, 237–250.
Russell, R.A., Urban, T.L., 2010. The location and allocation of products and product families on retail shelves. Ann. Oper. Res. 179 (1), 131–147.
Schaal, K., Hübner, A., 2017. When does cross-space elasticity matter in shelf-space planning? A decision analytics approach. Omega 1–18.
Shioda, R., Tunçel, L., Myklebust, T.G., 2011. Maximum utility product pricing models and algorithms based on reservation price. Comput. Optim. Appl. 48 (2), 157–198.
Spieth, P., Roeth, T., Meissner, S., 2019. Reinventing a business model in industrial networks: implications for customers' brand perceptions. Ind. Market. Manag. 83, 275–287.
Tsao, Y.-C., Lu, J.-C., An, N., Al-Khayyal, F., Lu, R.W., Han, G., 2014. Retailer shelf-space management with trade allowance: a Stackelberg game between retailer and manufacturers. Int. J. Prod. Econ. 148, 133–144.
Urban, T.L., 1998. An inventory-theoretic approach to product assortment and shelf-space allocation. J. Retailing 74 (1), 15–35.
Vincent, F.Y., Maglasang, R., Tsao, Y.-C., 2020. A reduced variable neighborhood search-based hyperheuristic for the shelf space allocation problem. Comput. Ind. Eng. 106420.
Yang, M.-H., 2001. An efficient algorithm to allocate shelf space. Eur. J. Oper. Res. 131 (1), 107–118.
Zhao, J., Zhou, Y.-W., Wahab, M., 2016. Joint optimization models for shelf display and inventory control considering the impact of spatial relationship on demand. Eur. J. Oper. Res. 255 (3), 797–808.
Zufryden, F.S., 1986. A dynamic programming approach for product selection and supermarket shelf-space allocation. J. Oper. Res. Soc. 37 (4), 413–422.