Shelf Space Allocation with Rectangular Arrangement: A Comparative Study of Three MILP Formulations

Daniel Martins[1(✉)], Maria João Alves[1,2], and Cristela Bairrada[1]

[1] Faculty of Economics, Univ Coimbra, CeBER, Av Dias da Silva 165, 3004-512 Coimbra, Portugal
danielmmail@gmail.com, {mjalves,cristela.bairrada}@fe.uc.pt
[2] INESC Coimbra, Coimbra, Portugal

Abstract. In this work, we compare three mixed-integer linear programming formulations for the shelf space allocation problem with rectangular arrangement: an original model proposed in the literature and two new formulations. The new proposed models differ in how they enforce rectangular arrangements with respect to the original model: one is a compact-variable formulation, reducing the number of variables at the expense of additional constraints, while the other keeps a similar size to the original model but uses a more direct representation of the rectangularity conditions. Using selected real instances, we evaluate solution times and best objective values with a general-purpose solver (Gurobi) under the original setting, where only row effectiveness is considered, and settings that also include column attractiveness under several patterns. Overall, when column attractiveness is included, the new models consistently outperformed the original one, with the third model achieving the best performance in most cases. The experiments confirm that the problem is very difficult to solve at optimality when the size of instances increases, and explicitly modeling both row and column attractiveness can significantly affect the solution landscape and performance of the models.

Keywords: Shelf space allocation · Retailing · Mixed-integer linear programming

1 Introduction

Shelf space allocation has been a central topic in retail research for decades, given its direct impact on sales and profitability. Earlier studies have shown that demand responds to the amount of space provided (space elasticity) and to where items are placed (location elasticity) [1, 2]. More recent studies further link layout to the effectiveness of marketing instruments [3] and provide comprehensive reviews of shelf planning models and rules [4]. The challenge lies in optimally distributing limited shelf space among numerous products to maximize returns, considering space elasticity, location effects and merchandising rules such as rectangular arrangement [5].

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 179–191, 2027.
https://doi.org/10.1007/978-3-032-30491-9_12

In this work we focus on the Shelf Space Allocation Problem with Rectangular Arrangement (SSAP-RA) introduced by Gencosman and Begen [5], who proposed a mixed-integer linear programming (MILP) formulation to address the problem.

The SSAP-RA model is intended to determine the number of facings (defined as the number of product units visible at the front of the shelf) for each product, in both the vertical and horizontal dimensions, and their allocation in a rectangular shelf layout. The objective of the retailer is to maximize profit considering space-elasticity, i.e., the effect of the number of facings on the demand, and location-elasticity, i.e., the effect of products' position on the demand of products. In this model, only row position effectiveness is considered. As the use of a general MILP solver for the SSAP-RA model has not allowed to solve realistic instances in acceptable times, two exact solution approaches were developed in [5]. The first method is based on logic-based Benders decomposition, which decomposes the optimization problem into a master problem and a sub-problem. The master problem contains only some decision variables and the values found for these variables are parameters to the subproblem to obtain possible solutions. Feasibility cuts and optimality cuts are iteratively imposed until an optimal solution is found. Although this method can handle larger instances compared to solving the MILP formulation, it still fails to solve some instances. Therefore, inspired by the previous method, a two-stage iterative approach was built by the authors, which separates the MILP model into two simple models considering row and column consecutiveness conditions separately. This approach has proven to be effective and could deal with larger instances. However, these methods are specific to the model addressed and rely on the assumption that the column position of the product has no effect on the optimization problem, that is, attractiveness varies only across rows. As a result, these approaches do not apply if placing a product in one column has a different effect on demand than placing it in another. The vertical dimension, however, also influences product, with the eye and hand level positions identified in the literature as the best locations [2, 6]. With respect to the horizontal dimension some authors support the best placement on the edges and others advocate for placement in the central columns of a cabinet [3, 6].

Our goal is to consider both row and column attractiveness, comparing the SSAP-RA formulation with two new more compact formulations - Shelf Space Allocation Problem with Rectangular Fulfilment Arrangement (SSAP-RFA) and Shelf Space Allocation Problem with Interval-Based Rectangular Arrangement (SSAP-IBRA). Both models accommodate horizontal (column) and vertical (row) attractiveness. SSAP-RFA model reduces the number of variables but significantly increases the number of constraints. SSAP-IBRA model has a comparable number of variables and constraints to the SSAP-RA model, but uses a more direct representation of the rectangularity conditions, which is easier to understand. Using selected real instances from [5], we evaluate solution times and best objective values with a general-purpose solver (Gurobi) under the original setting, where only row effectiveness is considered, and settings that also include column attractiveness under several patterns (center-most and edge-most columns, and an alternative multiplicative row $\times$ column attractiveness model). Computational experiments have shown that the new models consistently outperform the original model when both row and column attractiveness are both considered. The experiments also confirmed that

solving the problem at optimality is very difficult and different row/column attractiveness scenarios influence the performance of the models.

The remainder of the paper is organized as follows: Sect. 2 describes the original model (SSAP-RA). Section 3 presents our new models – SSAP-RFA and SSAP-IBRA models. Section 4 reports and discusses the computational results. Section 5 concludes the paper and outlines future work.

2 The SSAP-RA Model

The SSAP-RA model by Gencosman and Begen [5] aims to decide how products should be displayed on shelves to maximize the total expected profit. The products should be placed in rectangular blocks (consecutive rows and columns), which reflects practical merchandising practices. The key variables represent the number of facings (units visible on the shelf) of each product and where each product is placed on the shelf. Since the motivation for this study has been an application to a local bookstore, the *facings* represent the number of copies of a book title that are displayed on the shelf so that their front cover is visible to customers.

The problem addressed in [5] considers K cabinets. Each cabinet k includes R_k rows and C_k columns, so each cabinet has $R_k \times C_k$ slots. A slot is the space where the thinnest book can be allocated. Thus, a facing of a product can be allocated to more than one slot, depending on the thickness of each product. Empty slots are not allowed. All the facings of a product must be placed in the same cabinet in a rectangular layout. Without loss of generality, the present work considers only one cabinet ($K = 1$), meaning that the k index can be omitted from certain parameters and variables. The original model only considers the effect of vertical position on demand (i.e., the effectiveness of the rows A_i) but not that of the columns. In the formulation below, we have adapted the model to also incorporate the impact of horizontal position (replacing A_i with A_{ij} and adjusting the objective function accordingly). Table 1 shows the notation of this model.

For the sake of completeness, we now present the full MILP formulation of SSAP-RA. Further explanations can be found in [5].

$$\max \text{Profit} = \sum_{d \in D} \sum_{i \in I} \sum_{j \in J} V_d A_{ij} t_{dij} / T_d \tag{1}$$

s.t.

$$\sum\nolimits_{\mathrm{d} \in \mathrm{D}} t_{dij} = 1, \qquad \forall i \epsilon I, j \epsilon J \tag{2}$$

$$\sum\nolimits_{j \in J} t_{dij} = n_{di} T_d, \qquad \forall d \epsilon D, i \epsilon I \tag{3}$$

$$\sum\nolimits_{i \in I} n_{di} = m_d, \qquad \forall d \epsilon D \tag{4}$$

$$\sum\nolimits_{i \in I} t_{dij} = p_{dj}, \qquad \forall d \epsilon D, j \epsilon J \tag{5}$$

$$L_d \leq m_d \leq U_d, \qquad \forall d \epsilon D \tag{6}$$

Table 1. Notation for the SSAP-RA model

Sets and indices:	*Parameters:*
D- Set of products, $D = \{1,2,\ldots,D_{max}\}$	D_{max}- Maximum number of products
I- Set of rows, $I = \{1,2,\ldots,R\}$	R- Number of rows in the cabinet
J- Set of columns, $J = \{1,2,\ldots,C\}, j \in J$	C- Number of columns in the cabinet
d- Index of products, $d \in D$	A_{ij}- Effectiveness of slot (i,j)
i- Index of rows, $i \in I$	V_d- Profit of product d
j- Index of columns, $j \in J$	L_d, U_d - Lower and upper bound for the number of facings of product d
	T_d- Thickness of product d
Decision variables:	
$y_{di} \in \{0,1\}$: 1, if product d is assigned to row i; 0, otherwise	
$z_{dj} \in \{0,1\}$: 1, if product d is assigned to column j; 0, otherwise	
$t_{dij} \in \{0,1\}$: 1, if product d is assigned to row i and column j; 0, otherwise	
m_d: Total number of facings of product d	
n_{di}: Number of facings of product d in row i	
p_{dj}: Number of slots occupied by product d in column j (which also represents the number of facings of product d in column j)	
w_d: : Width of the rectangle (number of slots = columns) assigned to product d	
h_d: : Height of the rectangle (number of rows) assigned to product d	

$$\sum\nolimits_{d \in D} n_{di} T_d = C, \qquad \forall i \epsilon I \tag{7}$$

$$\sum\nolimits_{d \in D} p_{dj} = R, \qquad \forall j \epsilon J \tag{8}$$

$$w_d - n_{di} T_d \leq C(1 - y_{di}), \qquad \forall d \epsilon D, i \epsilon I \tag{9}$$

$$n_{di} T_d \leq C y_{di}, \qquad \forall d \epsilon D, i \epsilon I \tag{10}$$

$$h_d - p_{dj} \leq R\left(1 - z_{dj}\right), \qquad \forall d \epsilon D, j \epsilon J \tag{11}$$

$$p_{dj} \leq R z_{dj}, \qquad \forall d \epsilon D, j \epsilon J \tag{12}$$

$$n_{di} T_d \leq w_d, \qquad \forall d \epsilon D, i \epsilon I \tag{13}$$

$$\sum\nolimits_{i \in I} y_{di} \leq h_d, \qquad \forall d \epsilon D \tag{14}$$

$$y_{d,i+1} + w_d \geq y_{di} + n_{d,i+u} T_d, \qquad \forall d \epsilon D, i \epsilon I, u \epsilon I | 1 \leq u \leq R-1; \& i+u \leq R \quad (15)$$

$$z_{d,j+1} + h_d \geq z_{dj} + p_{d,j+u}, \qquad \forall d \epsilon D, j \epsilon J, u \epsilon J | 1 \leq u \leq C-1; \& j+u \leq C \quad (16)$$

$$y_{di}, z_{dj}, t_{dij} \epsilon \{0,1\}, \qquad \forall d \epsilon D, i \epsilon I, j \epsilon J \quad (17)$$

$$m_d, n_{di}, p_{dj}, w_d, h_d \epsilon \mathbb{Z}_0^+, \qquad \forall d \epsilon D, i \epsilon I, j \epsilon J \quad (18)$$

The objective function (1) maximizes the expected profit of displaying products. Since $\sum_{j \in J} t_{dij} = n_{di} T_d$, then the variable n_{di} in the objective function in [5] is replaced by $\sum_{j \in J} t_{dij} / T_d$, to also include the position of the facing in column. Constraints (2) assign exactly one facing unit to each slot. Constraints (3) compute the number of facings of each product d in each row i, n_{di}, and (4) calculate the total number of facings of each product d, m_d; (5) determine the number of facings of each product d in each column j, p_{dj}; (6) restrict the number of assigned facings of each product d with respect to its lower and upper bounds, L_d and U_d, respectively; (7) and (8) limit the display quantities of products with the dimensions of the cabinet that contains C columns and R rows; (9) and (10) ensure that, if product d exists in a row, the number of slots allocated to product d should be equal to the width of product d, w_d; (11) and (12) ensure that the number of facings of a product in different columns is equal to each other, h_d; (13) and (14) determine the width w_d and the height h_d of each product d, respectively; (15) guarantee that the product facings in a row are allocated consecutively and (16) do the same for columns, ensuring a rectangular arrangement of products. The expressions (17) and (18) define the solution space of the decision variables.

3 The New Proposed Models

We propose two more compact MILP formulations – SSAP-RFA model and SSAP-IBRA model – which differ in the way the rectangular arrangement is ensured. SSAP-RFA keeps only the assignment variables to slots and the row-wise facing counts, while enforcing rectangularity using rectangle fulfilment constraints. SSAP-IBRA introduces additional variables that define the rows and columns used by each product, as well as the start/end row/column for each rectangle. The aim is to compare the performance of these models with that of the original one (SSAP-RA). The sets, indices and parameters for these models are the same as presented in Table 1. The decision variables of SSAP-RA are only the following:

$t_{dij} \in \{0,1\}$: 1, if product d occupies slot (i, j); 0, otherwise.

$n_{di} \geq 0$ integer: Number of facings of product d in row i.

SSAP-RFA and SSAP-IBRA both have the same objective function (19) and the set of constraints (20)–(26). These correspond to the objective function (1) and constraints (2), (3), (6)–(8) and (17)–(18) of SSAP-RA. Each model also includes specific constraints, which will be presented and explained below.

$$\max \text{Profit} = \sum_{d \epsilon D} \sum_{i \epsilon I} \sum_{j \epsilon J} V_d A_{ij} t_{dij} / T_d \quad (19)$$

s.t.

Exactly one product per slot

$$\sum_{d \in D} t_{dij} = 1, \qquad \forall i \epsilon I, j \epsilon J \tag{20}$$

Facings per row

$$\sum_{j \in J} t_{dij} = n_{di} T_d, \qquad \forall d \epsilon D, i \epsilon I \tag{21}$$

Facing bounds

$$L_d \leq \sum_{i \in I} n_{di} \leq U_d, \qquad \forall d \epsilon D \tag{22}$$

Capacity in number of columns

$$\sum_{d \in D} n_{di} T_d = C, \qquad \forall i \epsilon I \tag{23}$$

Capacity in number of rows

$$\sum_{d \in D} \sum_{i \in I} t_{dij} = R, \qquad \forall j \epsilon J \tag{24}$$

Type of variables

$$t_{dij} \epsilon \{0,1\}, \qquad \forall d \epsilon D, i \epsilon I, j \epsilon J \tag{25}$$

$$n_{di} \epsilon \mathbb{Z}_0^+, \qquad \forall d \epsilon D, i \epsilon I \tag{26}$$

To ensure the fulfillment of the rectangle for each product, the following principle is followed in SSAP-RFA: if slots (i_1, j_1) and (i_2, j_2) are occupied by the same product d, as shown in Fig. 1, then all the slots between these two corners should also be occupied by product d. The idea is to interpret any pair of occupied slots as defining the corners of a potential rectangle and then impose the same product within this region.

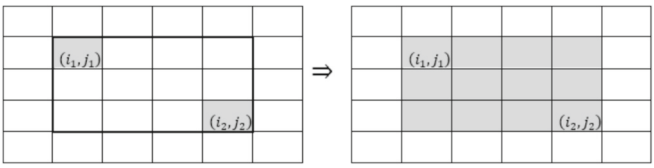

Fig. 1. Example of filling in the area from the top left to the bottom right.

Constraints (27) ensure the fulfillment illustrated in Fig. 1, in which $(i_2 - i_1 + 1)(j_2 - j_1 + 1)$ represents the area of the rectangle; the condition is redundant if either $t_{di_1j_1} = 0$ or $t_{di_2j_2} = 0$ (or both) and it must be satisfied if $t_{di_1j_1} = t_{di_2j_2} = 1$. M is a sufficiently large constant. Although the SSAP-RFA formulation relies on Big-M constraints, which can generally affect computational performance, in this case M can

be set explicitly to $R \times C$. This is valid because the area of the rectangle is never greater than $R \times C$. Therefore, the usual difficulty of determining appropriate Big-M values is avoided.

$$\sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} t_{dij} \geq (i_2 - i_1 + 1)(j_2 - j_1 + 1) - M\left(2 - t_{di_1j_1} - t_{di_2j_2}\right),$$

$$\forall d \epsilon D, i_1, i_2 \epsilon I, j_1, j_2 \epsilon J, i_2 \geq i_1, j_2 \geq j_1 \tag{27}$$

Constraints (27) search to the right and downward from each slot. However, searching in only one direction is not sufficient to cover all possible corner configurations, which is why a symmetric set of constraints is also necessary. To fill in the rectangles in all situations, we also need to search in another direction, downwards and to the left (as illustrated in Fig. 2) or upwards and to the right (corresponding to search from (i_2, j_2) to (i_1, j_1) in Fig. 2). Either one ensures that, together with (27), all cases are considered. Constraints (28) implement the former (in Fig. 2).

$$\sum_{i=i_1}^{i_2} \sum_{j=j_2}^{j_1} t_{dij} \geq (i_2 - i_1 + 1)(j_1 - j_2 + 1) - M\left(2 - t_{di_1j_1} - t_{di_2j_2}\right),$$

$$\forall d \epsilon D, i_1, i_2 \epsilon I, j_1, j_2 \epsilon J, i_2 \geq i_1, j_2 \leq j_1 \tag{28}$$

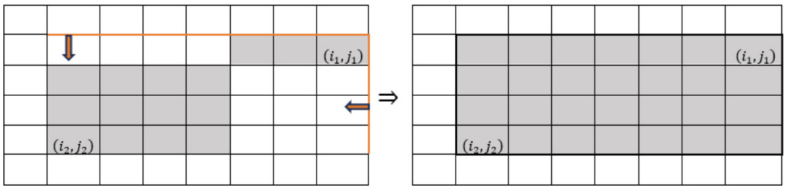

Fig. 2. Example of filling in the area from the top right to the bottom left.

The **SSAP-RFA** model is composed of (19)–(28).

The SSAP-IBRA model includes all the variables of SSAP-RFA, i.e., t_{dij} and n_{di}, and also the following binary variables:

$r_{di} \in \{0,1\}$: 1, if product d uses row i; 0, otherwise.

$c_{dj} \in \{0,1\}$: 1, if product d uses column j; 0, otherwise.

$r_{di}^1, r_{di}^2 \in \{0,1\}$: 1, if row i is the start/end interval row for product d.

$c_{dj}^1, c_{dj}^2 \in \{0,1\}$: 1, if column j is the start/end interval column for product d.

SSAP-IBRA explicitly specifies which rows and columns are occupied by each product, allowing the rectangle to emerge from choosing contiguous rows and columns.

A principle like the one in SSAP-RFA is considered to force each product d to occupy a single solid rectangle in $R \times C$ cabinet, but it is implemented in a different way: if (i_1, j_1) and (i_2, j_2) are assigned to d, then all (i, j) with $i \in [\min(i_1, i_2), \max(i_1, i_2)]$ and $j \in \left[\min(j_1, j_2), \max(j_1, j_2)\right]$ must also be assigned to d. The rectangular arrangement constraints ensure:

- Linkage between t_{dij} (use of slot (i, j) by product d) and r_{di}, c_{dj}, such that $t_{dij} = r_{di}.c_{dj}$; these products between binary variables are linearized in (29);
- Contiguity in rows linking variables of using row i, r_{di}, to the start/end rows for d, r^1_{di}, r^2_{di}, which is modelled in constraints (30);
- Contiguity in columns, linking variables of using column j, c_{dj}, to the start/end columns for d, c^1_{dj}, c^2_{dj}, which is modelled in constraints (31).

Linearization of $t_{dij} = r_{di}.\, c_{dj}$

$$\begin{aligned} &t_{dij} \le r_{di}, \qquad \forall d \epsilon D, i \epsilon I, j \epsilon J \\ &t_{dij} \le c_{dj}, \qquad \forall d \epsilon D, i \epsilon I, j \epsilon J \\ &t_{dij} \ge r_{di} + c_{dj} - 1, \qquad \forall d \epsilon D, i \epsilon I, j \epsilon J \end{aligned} \tag{29}$$

Contiguity in rows.

$$\begin{aligned} &\textstyle\sum_{i \in I} r^1_{di} \le 1, \sum_{i \in I} r^2_{di} \le 1, \sum_{i \in I} r^1_{di} = \sum_{i \in I} r^2_{di}, \forall d \in D \\ &\textstyle r_{di} = \sum_{k \in I : k \le i} r^1_{dk} - \sum_{k \in I : k < i} r^2_{dk}, \forall d \in D,\ i \in I \end{aligned} \tag{30}$$

Contiguity in columns.

$$\begin{aligned} &\textstyle\sum_{j \in J} c^1_{dj} \le 1, \sum_{j \in J} c^2_{dj} \le 1, \sum_{j \in J} c^1_{dj} = \sum_{j \in J} c^2_{dj}, \qquad \forall d \in D \\ &\textstyle c_{dj} = \sum_{h \in J : h \le j} c^1_{dh} - \sum_{h \in J : h < j} c^2_{dh}, \qquad \forall d \in D,\ j \in j \end{aligned} \tag{31}$$

$$\begin{aligned} &r_{di}, r^1_{di}, r^2_{dt} \epsilon \{0,1\}, \qquad \forall d \epsilon D, i \epsilon I \\ &c_{dj}, c^1_{dj}, c^2_{dj} \epsilon \{0,1\}, d \epsilon D, j \epsilon J \end{aligned} \tag{32}$$

Constraints (30) ensure contiguity in rows by imposing that (i) there is at most one start and one end per product ($\sum_{i \in I} r^1_{di} \le 1, \sum_{i \in I} r^2_{di} \le 1$), ($ii$) the start and end are balanced ($\sum_{i \in I} r^1_{di} = \sum_{i \in I} r^2_{di}$) and ($iii$) the row usage r_{di} is defined as the following: before the start row, $r_{di} = 0$ (because $\sum_{k \in I : k \le i} r^1_{dk} = \sum_{k \in I : k < i} r^2_{dk} = 0$); from the start row up to the end row (inclusive), $r_{di} = 1$ (because $\sum_{k \in I : k \le i} r^1_{dk} = 1$ and $\sum_{k \in I : k < i} r^2_{dk} = 0$); and after the end row, $r_{di} = 0$ (because $\sum_{k \in I : k \le i} r^1_{dk} = \sum_{k \in I : k < i} r^2_{dk} = 1$). Thus, the rows used by product d form one contiguous interval. Constraints (31) ensure similar contiguity in columns.

The **SSAP-IBRA** model is composed of (19)–(26), (29)–(32).

4 Computational Results

First, we compare the performance of the three models in instances R1 to R9 of the *RealisticData* in [5] derived from bookstore data. As these instances only consider one cabinet ($K = 1$), the SSAP-RA model presented herein is exactly the same as the one in [5] when instantiated with these problems. The original SSAP-RA only considers the effect of vertical position on demand, i.e., the effectiveness of rows (A_i). Thus, for the

first comparison with the original instances R1-R9, we have set $A_{ij} = A_i$ for all column j. The parameters A_{ij} were then modified to consider column attractiveness under several patterns. A total of 44 instances were tested. The number of rows is 8 for all instances, while the number of books and columns varies.

Table 2 reports the number of decision variables and constraints for the instances in sets R1–R9 under each model.

Table 2. Number of variables and constraints of each set of problems

Prob	SSAP-RA		SSAP-RFA		SSAP-IBRA	
	Variables	Constraints	Variables	Constraints	Variables	Constraints
R1, R2, R3	990	1240	720	26100	1200	2320
R4, R5, R6	1980	2400	1440	52120	2400	4560
R7, R8, R9	4170	5096	3120	168896	4920	9836

We used Gurobi 12.0.3 with the AMPL modelling language. All runs were carried out on a standard laptop with an Intel Core i5-1335U@1.30 GHz, 16 GB RAM. A maximum time limit of 1800s per run was imposed for all formulations (as in the original paper). The tables below show the objective values and the running times (in seconds). The best times are highlighted in bold. If the time limit of 1800s is reached with an MIP gap larger than the default Gurobi value, this is also shown in the result tables.

A - Rows-Only Effectiveness

Table 3 reports the results obtained by the three models for the original R1-R9 instances. When columns are neutral (no horizontal attractiveness), the original model remains competitive and is sometimes faster than the alternative formulations. SSAP-RFA shows the weakest performance on this set of instances, with the least consistent running times. SSAP-IBRA achieves the best running times in 7 of the 9 instances.

Regarding column attractiveness, there is a lack of agreement in the literature about the best position along the horizontal axis within a cabinet. Some authors support the best placement on the edges, while others advocate for placement in the center. In contrast, in the vertical dimension, eye and hand level is often referred to as the best location [2, 6].

In the first scenario considered, the middle column is the most effective and effectiveness decreases towards the columns furthest to the left and right of the cabinet [6, 3]. To facilitate identification, the original dataset's instances have been labelled with "_B".

B - Center-Most Columns Preferred

Table 4 shows the results for the three models in the "_B" instances. These instances appear to be more challenging than the previous ones, typically requiring longer running times. For this and the subsequent experiments, we tested seven instances, from R1 to R7, since R7_B ran out of time on SSAP-RA and SSAP-RFA without ensuring that the optimal solution had been found.

In this test set, SSAP-IBRA outperformed the other models in 6 of the 7 instances.

Table 3. Comparison results considering rows-only effectiveness

Prob	# Books	#Cols	SSAP-RA		SSAP-RFA		SSAP-IBRA	
			Obj	Time	Obj	Time	Obj	Time
R1	10	8	2260	**0.20**	2260	0.66	2260	0.31
R2	10	8	1996	0.89	1996	0.69	1996	**0.47**
R3	10	8	1652	14.34	1652	139.39	1652	**13.84**
R4	20	8	2241	**7.91**	2241	125.00	2241	18.34
R5	20	8	2316	4.64	2316	18.78	2316	**1.61**
R6	20	8	2432	1.11	2432	2.22	2432	**0.39**
R7	30	12	3646	5.69	3646	8.11	3646	**4.70**
R8	30	12	2965	13.64	2965	13.58	2965	**5.17**
R9	30	12	2791	8.02	2791	10.97	2791	**6.69**

Table 4. Comparison results considering center-most columns preferred

Prob	SSAP-RA		SSAP-RFA		SSAP-IBRA	
	Obj	Time	Obj	Time	Obj	Time
R1_B	4456	157.45	4456	85.05	4456	**46.77**
R2_B	3903.6	195.41	3903.6	76.95	3903.6	**47.72**
R3_B	3235.2	195.41	3235.2	349.75	3235	**53.55**
R4_B	4391.9	496.09	4391.9	212.16	4391.9	**37.06**
R5_B	4538.1	230.59	4538.1	**41.61**	4538.1	52.45
R6_B	4752	670.91	4752	164.55	4752	**64.45**
R7_B	8783.5 *mipgap* = 0.14%	1800	**8787** *mipgap* = 0.03%	1800	**8787**	**1027.47**

B2 - Center-Most Columns Preferred with $T_d = 2$.

With the aim of undertaking further analysis and determining whether product thickness can affect model performance, at this stage we consider the same A_{ij} values as in study B, but assume that the thickness of all products is 2 ($T_d = 2, \forall d \epsilon D$). This change has been made to the previous dataset. The instances are identified as "_B2". Table 5 shows the comparison results for the three models.

In this test set, SSAP-RFA and SSAP-IBRA produced the best results, with SSAP-IBRA achieving better overall results than SSAP-RFA. SSAP-RA was not the best in any instance.

Table 5. Comparison results considering center-most columns preferred with $T_d = 2$

Prob	SSAP-RA		SSAP-RFA		SSAP-IBRA	
	Obj	Time	Obj	Time	Obj	Time
R1_B2	4719.5	16.56	4719.5	16.97	4719.5	**6.03**
R2_B2	4441	324.64	4441	143.91	4441	**46.66**
R3_B2	3919.5	21.53	3919.5	76.27	3919.5	**16.89**
R4_B2	4756.5	63.27	4756.5	11.27	4756.5	**5.70**
R5_B2	4769.5	160.78	4769.5	**10.38**	4769.5	16.02
R6_B2	4572.5	181.44	4572.5	**16.83**	4572.5	32.27
R7_B2	8652 *mipgap* = 0.05%	1800	8652	827.5	8652	**247.39**

C2 - Edges preferred with $T_d = 2$.

In another scenario, the columns are considered to be most effective at the far left and right edges of the cabinet, and less effective in the middle column [6]. We tested this by changing A_{ij} in the B2 instances, where the thickness of all products is 2 ($T_d = 2, \forall d \epsilon D$). Table 6 shows the comparison results for the three models. The performance of the models in this test set is similar to that observed in experiment B2, with SSAP-IBRA showing an even stronger advantage over the other models. SSAP-IBRA ensured the optimal solution in all instances and did not hit the time limit in any case. Moreover, it achieved shorter running times in all but one instance.

Table 6. Comparison results considering edges preferred with $T_d = 2$

Prob	SSAP-RA		SSAP-RFA		SSAP-IBRA	
	Obj	Time	Obj	Time	Obj	Time
R1_C2	4714.5	26.36	4714.5	99.11	4714.5	**8.64**
R2_C2	4440	81.99	4440	78.50	4440	**21.83**
R3_C2	3915.5	104.89	3915.5	85.29	3915.5	**16.36**
R4_C2	4756.5	111.20	4756.5	28.42	4756.5	**14.94**
R5_C2	4767.5	205.72	4767.5	59.10	4767.5	**31.34**
R6_C2	4572.5	210.67	4572.5	**34.02**	4572.5	37.44
R7_C2	8643 *mipgap* = 0.13%	1800	8643 *mipgap* = 0.08%	1800	8643	**580.20**

D - Multiplicative Row × Column Attractiveness

In this scenario, the slot attractiveness is considered to be the product of row and column factors, $A_{ij} = A_i \cdot A_j$, where A_i are the original parameters and A_j are the parameters for the center-most columns scenario. The results are shown in Table 7. These results

corroborate previous findings, showing the superiority of SSAP-IBRA in all cases but one.

Table 7. Comparison results considering $A_i \times A_j$ (center-most columns preferred)

Prob	SSAP-RA		SSAP-RFA		SSAP-IBRA	
	Obj	Time	Obj	Time	Obj	Time
R1_D	14785	63.83	14785	48.34	14785	**25.20**
R2_D	13069	83.41	13069	68.67	13069	**14.75**
R3_D	10915	90.91	10915	79.06	10915	**32.36**
R4_D	14956.5	860.66	14956.5	**119.02**	14956.5	291.22
R5_D	15275	245.47	15275	43.73	15275	**33.14**
R6_D	16200.5	623.70	16200.5	103.27	16200.5	**43.92**
R7_D	35947.67	1515.73	35947.67	290.09	35947.67	**87.47**

D2 - Multiplicative row × column attractiveness with $T_d = 2$.

The last experiment used the previous test set, but considering $T_d = 2, \forall d \epsilon D$. Table 8 shows the comparison results. This experiment reinforces the conclusions of experiments B2 and C2, showing that, when all products have the same thickness, SSAP-RFA performs better than in other scenarios, sharing the best times with SSAP-IBRA.

Table 8. Comparison results considering $A_i \times A_j$ (center-most columns preferred) with $T_d = 2$

Prob	SSAP-RA		SSAP-RFA		SSAP-IBRA	
	Obj	Time	Obj	Time	Obj	Time
R1_D2	15188.5	32.13	15188.5	69.47	15188.5	**15.00**
R2_D2	14762	561.52	14762	113.80	14762	**36.80**
R3_D2	13035.5	76.02	13035.5	38.92	13035.5	**8.92**
R4_D2	15725	86.39	15725	**6.91**	15725	8.56
R5_D2	15820.5	94.67	15820.5	**9.47**	15820.5	15.06
R6_D2	15216	235.44	15216	35.77	15216	**27.11**
R7_D2	34363.5 $mipgap = 0.05\%$	1800	**34364**	568.99	**34364**	**208.05**

5 Conclusions

This study compared three MILP formulations for a shelf space allocation problem with rectangular arrangement. It extended the original model SSAP-RA in [5] to incorporate both row and column attractiveness and proposed two new alternative formulations -

SSAP-RFA and SSAP-IBRA models. Ignoring column effects can lead to biased allocations by considering equivalent positions on shelves that, in practice, result in different demand responses. However, including these effects may increase the difficulty of the problems, change the performance of the solvers and make specialized methods not applicable.

Computational experiments showed that the new models consistently outperformed the original model when both row and column attractiveness are considered. Overall, the SSAP-IBRA model achieved the best performance by providing the optimal solution in the shortest time in 35 of the 44 instances. The SSAP-RFA model was the second best, providing the fastest solution in 7 of the 44 instances. The original SSAP-RA model performed best in only 2 instances.

Computational experiments also confirm that solving this problem to optimality is very challenging. Due to the large number of variables and constraints of the MILP formulations, the use of general solvers is limited to small problem sizes. Future work will focus on two directions: (i) linking optimization to behavioral insights by estimating average attractiveness indices from planograms and consumer surveys [7], a pathway that we have already started to explore; (ii) developing metaheuristics or hybrid approaches to handle larger instances efficiently.

Acknowledgments. This work has been funded by national funds through FCT – Fundação para a Ciência e a Tecnologia, I.P., through project UID/05037/2025 with DOI: https://doi.org/10.54499/UID/05037/2025.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Corstjens, M., Doyle, P.: A model for optimizing retail space allocations. Manage. Sci. **27**, 822–833 (1981). https://doi.org/10.1287/mnsc.27.7.822
2. Drèze, X., Hoch, S.J., Purk, M.E.: Shelf management and space elasticity. J. Retail. **70**, 301–326 (1994). https://doi.org/10.1016/0022-4359(94)90002-7
3. van Nierop, E., Fok, D., Franses, P.H.: Interaction between shelf layout and marketing effectiveness and its impact on optimizing shelf arrangements. Mark. Sci. **27**, 1065–1082 (2008). https://doi.org/10.1287/mksc.1080.0365
4. Bianchi-Aguiar, T., Hübner, A., Carravilla, M.A., Oliveira, J.F.: Retail shelf space planning problems: a comprehensive review and classification framework. Eur. J. Oper. Res. **289**, 1–16 (2021). https://doi.org/10.1016/j.ejor.2020.06.018
5. Gencosman, B.C., Begen, M.A.: Exact optimization and decomposition approaches for shelf space allocation. Eur. J. Oper. Res. **299**, 432–447 (2022). https://doi.org/10.1016/j.ejor.2021.08.047
6. Bai, R.: An investigation of novel approaches for optimising retail shelf space allocation. PhD Thesis, University of Nottingham (2005). https://eprints.nottingham.ac.uk/10153
7. Rooderkerk, R.P., Lehmann, D.R.: Incorporating consumer product categorizations into shelf layout design. J. Mark. Res. **58**, 50–73 (2021). https://doi.org/10.1177/0022243720964127