



\*Correspondence:  
Romit S. Beed, St. Xavier's  
College (Autonomous),  
Kolkata, India, rbeed@  
yahoo.com

## A Study of the Effect of the Parameters for Optimizing Profit Using Simulated Annealing to Solve Shelf Space Allocation Problem

Romit S. Beed, Ankita Sarkar, Raya Sinha, Deboshruti Dasgupta

*St. Xavier's College (Autonomous), Kolkata, India, rbeed@yahoo.com, ankita1397sarkar@gmail.com, rayamomsinha@gmail.com, deboshrutidg.96@gmail.com*

### Abstract

Shelf space allocation has always remained a crucial issue for any retail store, as space is a limited resource. This work proposes a model that uses a hyper-heuristic approach to allocate products on shelves to maximize the retailer's profit. This work has concentrated on providing a solution specifically for a consumer packaged goods store. There exist multiple conflicting objectives and constraints which influence the profit. The consequence is a non-linear programming model having a complex objective function, which is solved by using multiple neighborhood approaches using simulated annealing as simulated annealing is a useful tool for solving complex combinatorial optimization problems. Detailed analysis of the proposed technique of using annealing and reheating has revealed the effectiveness in profit maximization in the shelf space allocation problem. Various simulated annealing parameters have been studied in this article, which provides optimum values for maximizing profit.

**Keyword:** Shelf Space Allocation, Optimization Problem, Simulated Annealing, Hyper-Heuristic.

### 1. Introduction

Shelf space is always a finite resource for a retailer of a multi-purpose utility store. The Shelf space allocation problem is defined as the distribution of the appropriate amount of shelf space among various products, along with their locations, in a store, such that the total profit and the purchaser fulfillment are maximized. A shelf is defined as a horizontal, stable, and quadrangular structure used to withstand, hold, and display objects. Again, the rack is a stack of shelves. The objective of Shelf Space Allocation Problem is to maximize category sales and profits, without any regard to any particular bound (Retailer's goal) or to improve the sales of any particular brand, thus wanting to allocate as much space as possible to one particular product (Manufacturer's goal). Such a nature whose decision variables have discrete and finite domains is termed a combinatorial problem. As a result, a combinatorial problem has a finite number of solutions, although typically exponential in the number of variables.

In most cases of such problems, an instance may have more than one feasible solution. Although it is enough to find any solution that satisfies the constraints for search and decision problems, optimization problems discover a better solution than the others according to some measure. The customary way to evaluate the goodness of a solution with respect to the others is to define a fitness function to assess the utility of the solution that may influence the decision process.

In many problem formulations, constraints are split into hard constraints and soft constraints. Hard constraints define the restrictions that cannot be violated for any reason. Their violation prevents a solution from being feasible at all, thereby defining the search space's boundaries. Soft constraints, on the other hand, maybe violated, as their violation does not hinder feasibility. However, it is done at the price of a higher solution cost. Thus, care should be taken to minimize their violations by the optimization process. Soft constraints are the most common way to define an objective function. The in-store factors positively influence the choice of consumers. Rather than just displaying the merchandising, a creative product arrangement on the shelves becomes quintessential to increasing the perceptibility, consumer responsiveness, and demand for the products, in turn resulting in better performance.

A multi-objective optimization problem (MOOP) combines several objectives that decide the ultimate outcome. Generally, these objectives influence one another in a complex and conflicting manner (Ombuki et al., 2006). The main aim is to find a set of values for these objectives to optimize the overall problem. Goldberg states that multi-objective optimization is the method of optimizing multiple conflicting objectives, subject to a set of constraints. In these problems, it is observed that there is no specific solution that minimizes, at the same time, each objective completely, but to a limit beyond which the other objective(s) will be compromised as a consequence (Goldberg, 1989). After procuring a specific solution, one of the chief aims of MOOPs is to compare it with other solutions and measure the improvement in this solution with respect to the current set of solutions (Haupt and Haupt, 2004). A multi-objective problem with several conflicting objectives may be framed into a one-objective scalar function. This well-known technique, known as the weighted-sum method or Single Objective Evolutionary Algorithm (SOEA), is an a priori technique based on the "linear aggregation of functions" principle (Arulmozhiyal and Jubril, 2012). The weighted-sum method cuts down to a positively weighted convex sum of the objectives, as follows:

$$\min \sum_{i=1}^n w_i f_i(x), \text{ where } \sum_{i=1}^n w_i = 1; w_i > 0; w_i \forall i$$

Various scalarization procedures have been anticipated in the past. Zadeh promoted the weighted sum technique as a classical approach for explaining such problems (Zadeh, 1963). In this technique, the scalarization of a set of contradictory objective functions is done by pre-multiplying each objective function with already defined weights. However, despite the technique being straightforward and computationally proficient, it may fail to explore all solutions given that the true Pareto front is non-convex (Konak et al., 2006).

Simulated annealing (SA) is a useful tool for solving complex combinatorial optimization problems. Simulated annealing is a type of local search heuristic which is enthused from the physical process of annealing of solids. An advantage of this method is that it never gets trapped in local maxima, which are useful in such complex problems. It also provides a better solution quality and enhances computational efficiency. However, evolutionary algorithms, such as the SA, were mostly used to solve a single objective function due to its 'search-from-a-point' nature. This can be extended to solve the multi-objective optimization problem where the general aim is to find a set of solutions called the pareto set, all of which are equally important in the search space and obtains global optimal solutions. SA can be applied for such multi-objective problems by using a weighted sum method. Simulated Annealing has the upper hand due to its probabilistic nature. Shelf Space Optimization is an important topic being worked upon to capitalize on the goods' availability in their product line at the least cost to operations. This project focuses on the retailer's perspective and aims to allocate shelf space in a way to maximize profit (sales).

## *2. Literature Survey*

Research in shelf space allocation started back in the 1960s when a pragmatic study was made for three products from eight chain stores. A significant relationship was found between shelf facings and sales (Kotzan and Evanson, 1969). Similar experiments were carried out to include products from two brands of two classes, salt and coffee cream. A relationship was established between the facings of shelf and sales (Cox, 1970). Space elasticity was defined as "the ratio of relative change in unit sales to relative change in unit sales in shelf space" (Curhan,1972). A study for the affinity effects between products where space was manipulated to improve complimentary shopping by inserting such products together was performed (Dreze et al.,1994). The outcomes depicted that co-related merchandising resulted in a positive boost of sales (over 5%) on tested products. Moreover, it was concluded that the location effect of a shelf has a more significant impact on the number of facings.

A model was presented for Shelf Space Allocation in 1981 using geometric programming and heuristics constraints for optimization (Corstjens and Doyle, 1981). Their idea was based on the demands of the product along with their allowable capacities. A simplified version of the previous model was presented (Yang and Chen, 1999), which was the first to consider the number of facings of a product using a greedy knapsack approach to optimize. An extension of this model, considering product affinity, was produced. They used multiple problem neighborhood moves to optimize the non-linear function. A different model from the Yang and Chen model was proposed and worked on(Hwang et al.,2005). The difference was because the new model was formulated by amalgamating location effects. A location effect factor was implemented in the objective function, with constraints similar to previous models. Russell and Urban were the first two authors who considered the products as part of the family that can be grounded on diversity in characteristics like brand, flavor, price, etc. (Russell and Urban, 2010). Products of the same family should be kept together.

A model was developed based on a retailer's decisions for product prices, display facing areas, display orientations, and shelf-space locations in a product category using branch and bound (Murray et al., 2010).

Simulated annealing (SA) was proposed by S. Kirkpatrick (Kirkpatrick et al., 1983). Various engineering planning and manufacturing problems can be modeled as a cost function that needs to be minimized or maximized over a set of distinct variables. Simulated Annealing is used to deal with problems that cannot be solved in polynomial time: NP-hard problems. Simulated annealing is applied to different types of problems, including traveling salesman problems (Wang et al., 2009), Job Shop scheduling (Laarhoven et al., 1992), and Shelf space allocation problems (Erol et al., 2015). The results produced by SA are susceptible to the annealing schedule and how the neighborhood is searched; that is, values of initial temperature, final temperature, cooling coefficient, and the number of moves in each loop's neighborhood affect the results produced. The main essence of the SA algorithm lies in the cooling schedule. Better results are obtained if the cooling schedule is big. However, if the cooling schedule is too big, it wastes time in the loop. If the cooling schedule is too short, it may fail to find the optimal solution. The main disadvantage of the algorithm is the considerable running time. Nevertheless, it is recouped by the simplicity and ease of application to different problems. This paper suggests an innovative technique to optimize profit through better store layout in the light of modern-day customer demands.

### ***3. Problem Formulation***

'Shelf space' with respect to any retail store is an essential and limited resource. In this paper, allocating products on the shelves has been optimized by keeping in mind the various conflicting objectives and the constraints to which this problem is subjected to. This paper proposes a 3-dimensional shelf space allocation model where the third-dimension stems from the height of the rack. The first two dimensions are the length of the shelf and the height of a shelf. The height of a rack can be considered to be the summation of each shelf's height within the rack. The following information has been collected from a retail store for evaluation: the selling price of each item present in the retail store, length and height of the shelves, number of shelves in a rack, length and height of each item, minimum and maximum units of facings that can be provided to each product, total number of shelves being considered, total number of products that are considered, affinity matrix of the products that will enable the identification of frequent item-sets from retailer's transaction history, location effect of each shelf, scale parameter for demand function of an item, space elasticity factor of an item.

Profit is calculated based on the selling price of an item and the value of an item's demand function over time. The demand model of a product depends on elasticity, and in this model, it is expressed as a product of direct space elasticity, cross elasticity, and location effect of a shelf. To fulfill the overall objective, it is required to consider certain underlying objectives that may be conflicting in nature. The number of items sold is directly related to the number of facings displayed of the product. Increasing

the number of facings of a product has a considerable effect on the sales of an item. However, studies have suggested that the number of product facings has an upper and lower boundary, a constraint considered by the model. Items that are frequently purchased together are placed in close proximity to boost the sale of such items together (Han et al., 2012). However, reducing the distance between items that are frequently purchased together also reduces individual products' display, thus the conflict. Since this problem is subject to multiple conflicting constraints, so, in order to maximize the retailer's profit, multi-objective optimization becomes a necessity. A pilot survey was conducted among 60 consumers to understand the preferences one places while purchasing an item from the retail store. The weights obtained from them were used to scalarize the multiple objective problems into a single objective optimization problem.

Table 1 : Survey of factors that can affect the sale of an item based on customer priorities

Factor Affecting Choice Of Customer		Weights assigned	Votes In Favour	Ratio Of Weights
Selling Price of an item ( $p_i$ )		$w_1$	22	0.367
Demand of an item ( $F_i$ )	Number Of Units Displayed ( $S_i$ )	$w_2$	8	0.133
	Affinity factor ( $\rho$ )	$w_3$	13	0.217
	Shelf Location of the Item ( $Y_j$ )	$w_4$	17	0.283

The problem is formulated for a given number of racks  $m$  in a shelf, with each rack  $j$  having a demand impact factor  $Y_j$ , where  $Y_j \geq 1$ . Racks at the eye level or nearer impact customer sales than the shelves located in the upper or lower regions. Length facing has been defined as the number of facings allocated to each item along the rack's length. Stack coefficient is the number of facings that can be assigned height-wise. The product of length facing and stack coefficient gives the total number of facings of an item.

Thus, formulating the main objective function, as follows:

$$\text{Max } \sum_{i=1}^n w_1 p_i * (1 - w_1) F_i \tag{1}$$

Here,  $p_i$  is the selling price of item  $i$ , and there are  $n$  items.  $F_i$  is the demand function of item  $i$ .

Maximizing the demand for an item is done by maximizing the amount of visibility of that product. The total number of facings that can be allotted to an item  $i$  can be given as:

$$S_i = \sum_{j=1}^m x_{ij} * \pi_{ij} \tag{2}$$

where  $x_{ij}$  is the length of the shelf  $j$  allocated to item  $i$  or length facings.

$\pi_{ij}$  is the height stack coefficient or the number of facings assigned to an item along with shelf height.

$$\pi_{ij} = \left\lfloor \frac{H_i}{h_i} \right\rfloor \tag{3}$$

Where  $H_i$  is the height of the shelf and  $h_i$  is the height of item  $i$ .

The demand for an item is also affected by the location of the shelf it is assorted to. Thus, the average location effect can be calculated as:

$$Y_i = \frac{\sum_{j=1}^m x_{ij} * \pi_{ij} * Y_j}{S_i} \tag{4}$$

The affinity factor for a pair of products is defined as a complex relationship between the number of facings of product  $k$  and the cross-elasticity factor of item  $i$  with item  $k$ .  $\beta_{ik}$  is the cross elasticity or affinity of an item  $k$  on item  $i$ . Cross-elasticity is defined as the influence of the number of facings allotted to one item on another item's sale. The affinity matrix is constructed with values in the range of 1 and -1. The closer the value of affinity to 1, the more affine are the products. If the affinity value is zero, then the affinity among the categories is indifferent. The affinity factor is represented using the equation:

$$\rho = \prod_{k \neq i}^n S_k^{\beta_{ik}} \tag{5}$$

Now the demand function of item  $i$  is calculated as follows:

$$F_i = a_i * w_2 S_i^{\beta_i} * w_3 \rho * w_4 Y_i \tag{6}$$

Here,  $a_i$  is the scale factor for the demand function of item  $i$ ,  $\beta_i$  is the space elasticity for item  $i$ , and Space elasticity for an item  $i$  is the measurement of the impact on a product sale performance by increasing or decreasing its allocation of space within a shelf.

The problem described in this paper that is the objective function is subject to all the following constraints:

$$\sum_{i=1}^n l_i x_{ij} \leq L_j, \forall j \tag{7}$$

$$y_{ij} \leq \pi_{ij}, \forall i, \tag{8}$$

$$s_i^{min} \leq s_i \leq s_i^{max}, \forall i \tag{9}$$

$$y_{ij} \in \{0, 1\}, \forall i, j \tag{10}$$

$$x_{ij} \in \{0\} \cup Z^+, \forall i \tag{11}$$

$$y_{ij} \leq x_{ij}, \forall i, j \tag{12}$$

$$\sum_{j=1}^m y_{ij} = 1, \forall i \tag{13}$$

$$y_{ij} * \frac{L_j}{l_i} \geq x_{ij}, \forall j \tag{14}$$

In constraint (7), signifies the length of item  $i$ ,  $x_{ij}$  is the length facing allotted to an item  $i$ ,  $n$  is the total number of items that are considered,  $j$  refers to the current shelf,  $L_j$  is the shelf length. In constraint (8),  $y_{ij}$  denotes an item's height facings, which should be less than the calculated height coefficient for a given shelf. Constraints (7) and (8) ensure that the items assigned to each shelf do not exceed the shelf's capacity, both in terms of length and height. Constraint (9) determines the minimum and the maximum number of facings that can be designated for each item.  $s_i$  denotes the total facing

allotted to item  $i$ , which should be within a given minimum ( $s_i^{min}$ ) and maximum ( $s_i^{max}$ ) boundary value. Constraints (10), (11), (12), (14) define the relationships of the variables  $x_{ij}$  and  $y_{ij}$ . Constraint (13) is a cluster constraint that ensures that the same type of items have to be displayed together on the shelf, where  $m$  denotes the total number of shelves that the problem considers. Usually, retailers wish to preserve the related items together in order to exhibit a large attractive block.

The allocation process is optimized using the hyper-heuristic learning mechanism and simulated annealing to get the optimized placement of products, and finally, the shelf space layout maximizes the retailer's profit. This paper considers one medium instance ( $m=5, n=29$ ) of data items. The details regarding the first ten items are displayed in table 2. During implementation, all the values given in the table have been normalized. The multiple neighborhood approach suggested in this paper uses a collection of neighborhoods in hybridization with the simulated annealing algorithm and a hyper-heuristic learning mechanism. Based on an initial solution and a set of neighborhoods, this algorithm changes the neighborhood inclination during the search, and the procedure of simulated annealing is used to govern whether a given neighborhood move is acknowledged or vetoed. The neighborhood approach basically uses a set of heuristics. Each neighborhood is associated with weight  $w_i$  that represents its preference in comparison with other neighborhoods. At each iteration, a neighborhood is ranked by the probability.

$$p_i = w_i / \sum_{i=1}^n w_i \tag{15}$$

Initially, the weights are set to  $w = w_{min}$ , all the neighborhoods are assigned equal weights ( $1/n$ ). However, the weights are updated at the end of each Learning Period. Learning Period refers to the period that the algorithm takes to understand the problem space to obtain a better selection of neighborhood moves, which can then be accepted or rejected by the simulated annealing acceptance criteria. During this time, the algorithm iterates to analyze the different neighborhood moves and reassigns their weights depending upon each move's performance. Each neighborhood structure is associated with a set of counters depending on the number of generated solutions and passed the simulated annealing acceptance criteria, the new solutions generated using a neighborhood, and the total solutions generated by heuristic moves. The performance of each neighborhood can be determined by observing associated counters. At each iteration, a neighborhood is stochastically ranked with a probability, as mentioned in equation (15).

Table 2: Item specifications of 10 sample items.

Item	Height	Length	Price	Min Facing	Max Facing	scale factor	space elasticity	Cross Elasticity													
								1	2	3	4	5	6	7	8	9	10				

1	143	47	301.12	7	12	658.52	0.98	-9999	0.023	0.003	0.013	0.011	-0.03	0.029	0.018	0.021	-0.027
2	143	47	200.92	2	8	18.18	1.02	0	-9999	0.001	-0.014	0.029	0.011	-0.007	-0.004	0.01	0.01
3	143	47	362.57	8	16	685.4	0.35	-0.002	-0.016	-9999	-0.009	0.001	-0.02	-0.01	0.011	-0.007	-0.021
4	143	47	500.04	3	10	45.8	0.88	0.026	-0.025	0.028	-9999	-0.001	0.008	-0.028	-0.023	-0.018	-0.025
5	143	47	448.68	2	8	93.57	0.73	-0.012	0.002	-0.012	0.029	-9999	0.022	0.017	-0.017	-0.024	-0.02
6	150	50	346.71	3	8	42.59	0.88	-0.005	0.027	-0.006	0.016	0.026	-9999	-0.023	-0.001	-0.011	-0.027
7	143	47	311.20	3	10	125.21	1.06	-0.002	-0.009	-0.007	-0.008	-0.03	0.007	-9999	0.02	0.021	0.016
8	180	65	578.60	4	6	134.29	1.11	-0.025	-0.029	0.023	-0.017	0.022	-0.002	-0.012	-9999	-0.023	-0.023
9	180	65	196.39	7	10	70.81	1.12	-0.008	-0.002	-0.008	0.003	-0.012	-0.026	-0.2	-0.015	-9999	-0.004
10	153	56	441.88	3	8	671.83	0.15	0.015	-0.02	-0.022	0.021	0.029	-0.011	0.029	0.009	0.021	-9999

Temperature plays an important role in the annealing and reheating process used in this algorithm. The starting and the stopping temperatures are initially estimated as  $t_s$  and  $t_e$ , respectively. During the annealing process, the temperature is gradually reduced. This happens when the acceptance ratio is improving with respect to



temperature, i.e., better solutions are generated. When the algorithm fails to generate better solutions, the reheating phase is triggered to explore other neighboring solutions in an expanded problem space. The acceptance ratio is defined as the ratio of the number of accepted solutions ( $C_a$ ) to the length of a single learning period (LP). The learning period plays a pivotal role in deciding whether the algorithm performs annealing or reheating functions. The acceptance ratio for a given Learning Period is compared with the stopping non-improving acceptance ratio ( $r_e$ ), which decides whether reheating is required. If the annealing phase continues, the temperature reduction takes place in accordance to the function.

$$t = t / (1 + \eta t) \quad (16)$$

$$\text{where, } \eta = (t_s - t_e) \cdot n_{rep} / (K \cdot t_s \cdot t_e) \quad (17)$$

$n_{rep}$  represents the number of iterations at each temperature and  $K$  is the number of total iterations.

Within a learning period, if the acceptance ratio drops below a minimum threshold ( $r_e$ ), then the algorithm switches to the reheating phase, and it continues until it finds a new, better solution. During the reheating phase, the temperature is incremented using the temperature deduction rate  $\eta$ . The equation used for updating the temperature is current temperature

$$t = t / (1 - \eta \cdot t) \quad (18)$$

Also, at the end of each learning period, for the reheating phase, the weights of the neighborhoods are updated based on the number of total solutions and new solutions that are generated,

$$w_i = c_{new_i} / c_{total_i} + w_i \quad (19)$$

However, if the algorithm encounters acceptable solutions without the need for reheating, the annealing phase continues. The weights are updated after each Learning Period for the annealing phase based on the total solutions and the number of accepted solutions, i.e.

$$w_i = c_{accept_i} / c_{total_i} + w_i \quad (20)$$

A candidate solution  $S'$  is generated randomly from the current solution  $S$  in neighborhood  $N_i$ . The two solutions are compared based on the objective function. The new solution generated is accepted or rejected based on the Simulated Annealing property. However, suppose the new solution is not better than the one already existing, in that case, the counter for the newly generated solution is incremented, and it is accepted with a probability of  $e^{-\frac{\delta}{t}}$  (Aarts et al., 2005) where  $\delta$  is the difference between the current solution and the newly generated solution and  $t$  is the current temperature of the loop.

The authors have proposed the following neighborhood moves to reach an optimal solution: (i) Exchange\_facing – This move includes all the conceivable solutions produced by exchanging one shelf length facing an item with another item are sharing the same shelf. (ii) Alter\_shelf – This move includes all the conceivable solutions that may be produced by moving all the facings of a selected item from one shelf to another. (iii) Switch – This move includes the concept of both the neighborhood mentioned above. The move includes all the potential solutions produced by swapping the total

number of facings of one item  $i$ , which is placed on a shelf with another item  $k$  placed on a different shelf. (iv) Remove\_facing - This move includes all the conceivable solutions produced by deleting one shelf length facing an item placed on a shelf. (v) Increment\_facing - This move includes all the conceivable solutions that may be produced by adding one shelf length of an item placed on a shelf. (vi) Duplet\_swap - This move includes all the conceivable solutions that may be produced by moving two different items ( $i, k$ ) placed on two different shelves ( $m_0, m_1$ ) with a high positive value of cross elasticity factor to another shelf ( $m_2$ ) wherein they are to be placed together in turn, replacing the total facings of two other items ( $x, y$ ) already placed on that shelf ( $m_2$ ). The items ( $x, y$ ) which are being replaced should have poor affinity between them.

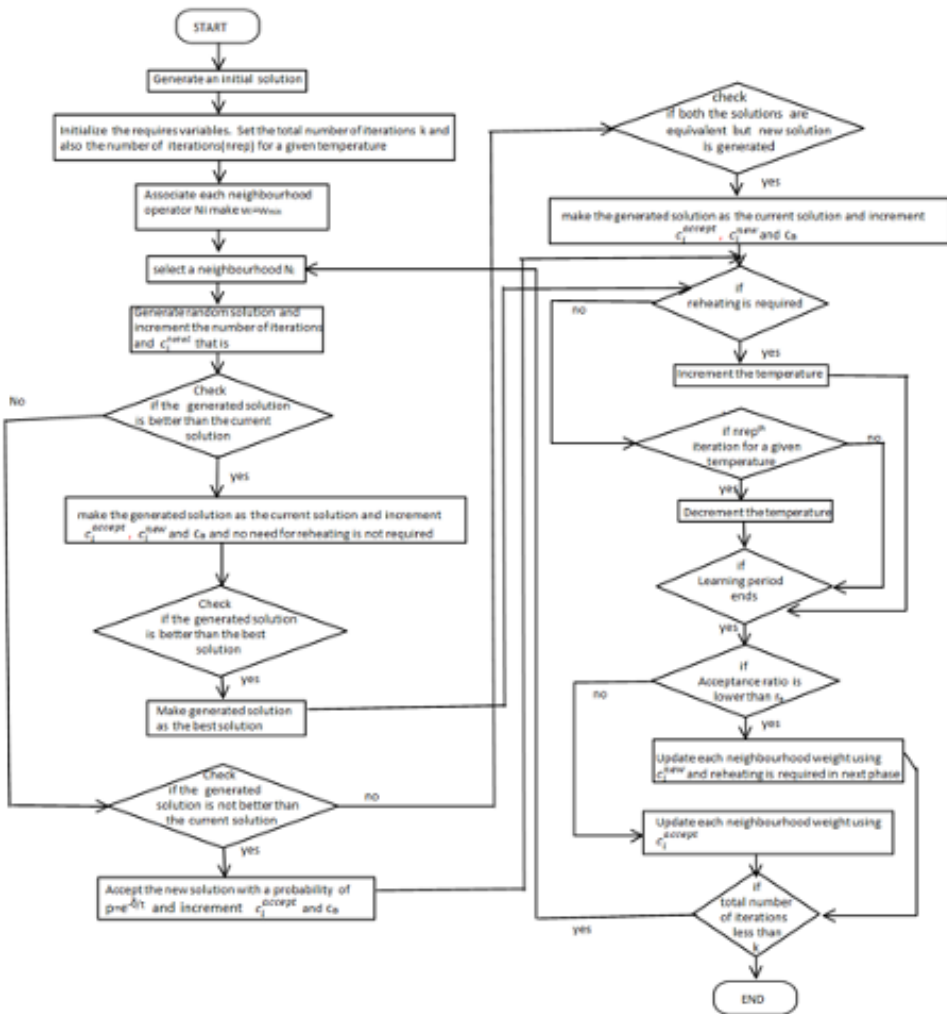


Fig. 1. Flowchart for the proposed multi-neighbourhood approach

#### 4. Results and Analysis

The proposed algorithm was executed for several iterations, and the results were summarized in Figure 2. It has been observed that the profit increases almost linearly

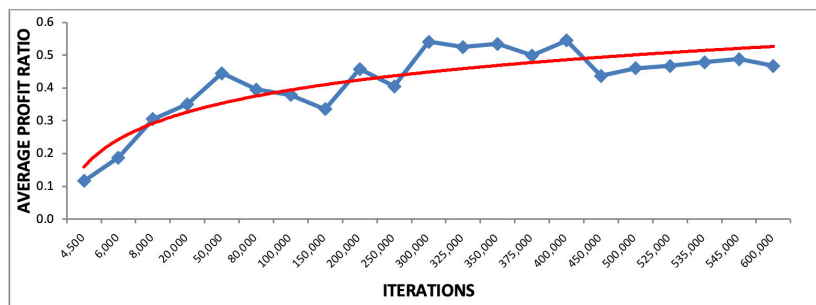


Fig. 2. Iteration stabilization graph

with an increase in the total number of iterations. However, it is also not feasible for the process to iterate indefinitely as it takes more time to execute. The total number of iterations highly influences the improvement in profit. After a certain number of iterations, the profit stabilizes.

From figure 2, it can be inferred that when the number of iterations is very small ( $K=4500$  to  $20,000$ ), the improvement in profit is minimal. However, the graph in this section linearly increases with increased iteration values. Between  $80,000$  to  $2,50,000$  the graph shows variability in profit improvement. The maximum profit is obtained between iterations  $3,00,000$  to  $4,00,000$ , after which the graph stabilizes at a slightly lower profit value.

The algorithm analyses the problem space, trying to find better solutions controlled by the number of iterations. In this paper, to estimate the most desirable learning period, a range of different learning periods between  $500$  and  $3,00,000$  have been considered. Within a learning period, the algorithm explores the different suggested neighborhoods using either annealing or reheating process. Based on the performance of each neighborhood, the weights associated with each heuristic move are updated at the end of each learning period, and again, the learning period works with those neighborhoods that are, in turn, selected based on their updated weights (using a probability function). Thus, the learning period for a given iteration must not be so small that there is no ample time for the process to learn about the problem. Also, it should not, on the other hand, be so large as to prohibit the opportunities for the algorithm to utilize the information gathered during the learning period.

Shows the graph for the learning period  $20000$ , which has been used in the implementation. For instance, observing the case of a minimum learning period,  $LP=5000$ , as shown in figure 4, the number of variations in the graph for the same number of total iterations is much less than that  $LP=20000$ . This proves that if the learning period is too small, the algorithm fails to gather sufficient knowledge about the heuristic moves whose performance is pivotal to determining the improvement in profit, which again serves the whole work's primary objective.

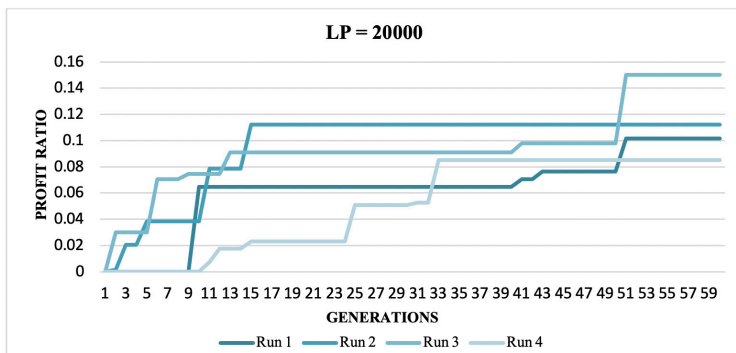


Fig. 3. Graph for learning period 20,000

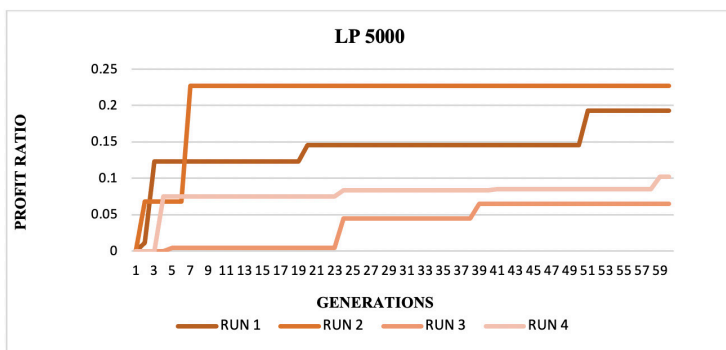


Fig. 4. Graph for learning period 5,000

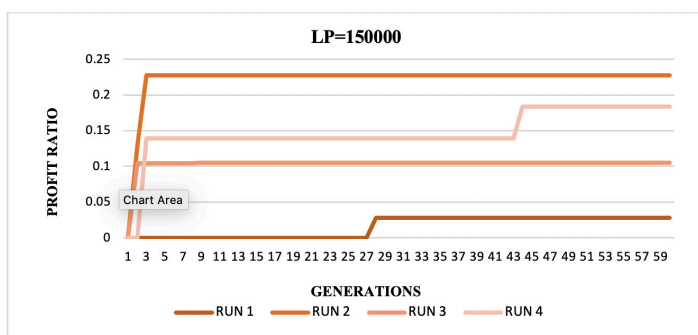


Fig. 5. Graph for learning period 1,50,000

Again, analyzing the algorithm by keeping the learning period as large as 1,50,000 shows that the graph in figure 5 is almost stable, with no considerable change. This proves that although the algorithm gets a substantial amount of time to learn the search space, the given iteration of 6,00,000, falls short of applying the knowledge gathered. In other words, the neighborhoods' weights cannot get adequately updated at the end of a learning period and then implemented in the next as most of the iterations are lost

while executing a single learning period. Analysis has been done by executing the algorithm several times, and every time, it has shown similar results. The gradient coloring schemes in the graphs in figure 3, 4, 5 shows the different executions wherein the darkest shade stands for the first execution.

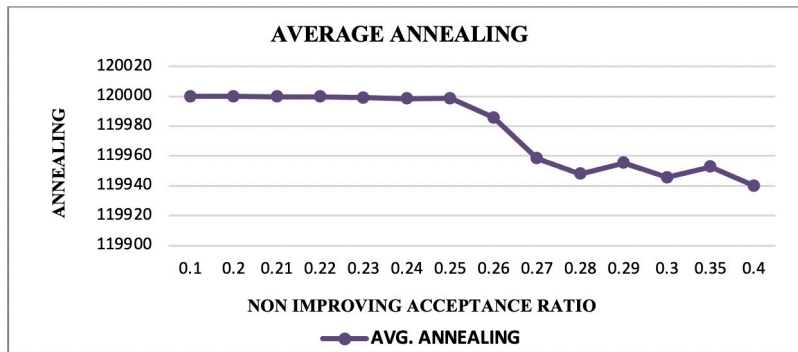


Fig. 6. Graph representing the average number of times annealing process takes place by varying re values

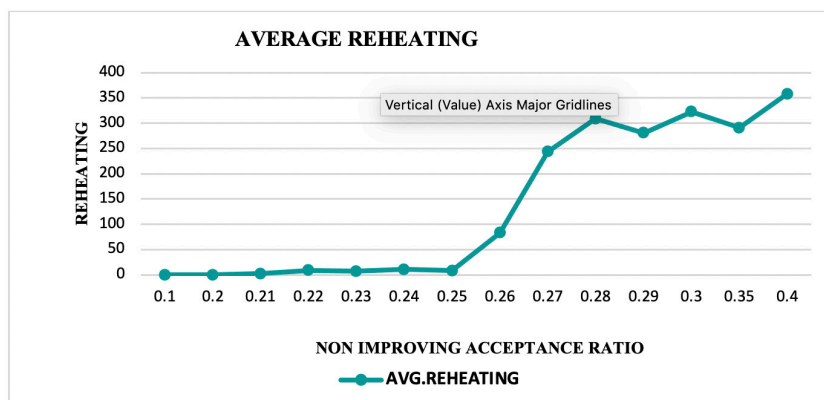


Fig. 7. Graph representing the average number of times reheating process takes place by varying re values

At each temperature, the algorithm iterates five times ( $nrep=5$ ). On increasing the temperature by reheating, it is intuitive that the program's execution time will increase. Moreover, as the value of re affects the reheating and annealing process within the algorithm, it indirectly affects the fitness value (profit). From the graphs in Figures 6 and 7, it is observed that with increasing non improving acceptance ratio values (re), the average number of times annealing takes place steadily decreases after  $re = 0.23$ , and simultaneously the reheating process starts increasing from the same point. For re value greater than 0.28, both the graphs show unpredictability. So, it may be concluded from the two graphs that the value of re within range 0.23-0.28 results in optimized solutions.

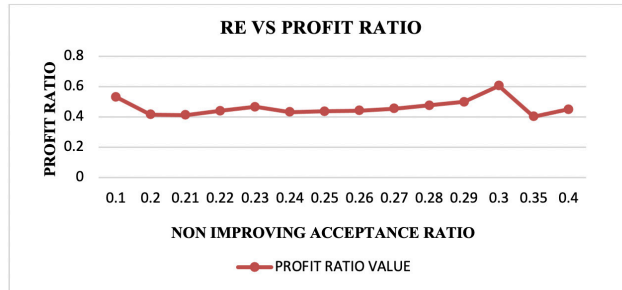


Fig. 8. Graph representing the change in average profit ratio with increasing  $r_e$  values.

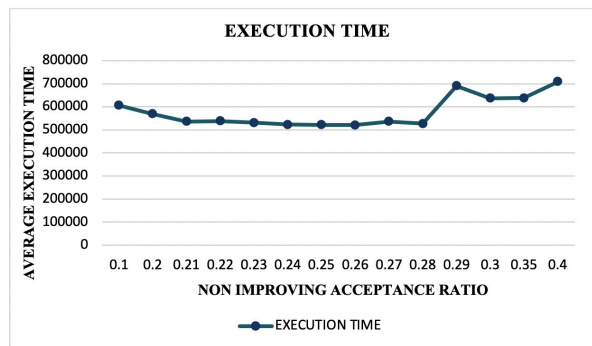


Fig. 9. Graph representing the average execution time taken with increasing  $r_e$  values.

It can be inferred from Figure 8 that for the chosen range of  $r_e$  values (0.23 – 0.28), the profit obtained is quite stable and obtained as a result of both annealing and reheating. An abrupt increase in profit can be observed on increasing the  $r_e$  value, which is marked by an abrupt increase in the reheating process. However, such abrupt increase is unrealistic and may be considered as outliers in comparison with other values. Similarly, for smaller values of  $r_e$ , the outcome is unpredictable and does not provide stable or reliable solutions for every execution. The graph in Figure 9 shows that the execution time does not increase much for the aforesaid chosen range of  $r_e$  values (0.23-0.28). For  $r_e$  values more than 0.28, the reheating process increases abruptly, leading to abrupt increase in the execution time, which is unwanted. Also, for smaller  $r_e$  values, it is observed that the execution time is relatively high even though no reheating takes place. This may be accounted for because the program tries to find a better solution within the same search space as it is unable to satisfy the reheating condition and increase the search domain.

The algorithm has been executed multiple times, changing the various parameters and summarized in Table 3. It can be observed that the algorithm obtains the most optimal solution for  $K=6,00,000$ ,  $LP=20,000$ , and  $r_e=0.25$ . The profit, in this case, has maximum improvement while the execution time is also rational. Both annealing and reheating processes take place, enabling the algorithm to yield better results by thoroughly exploring the problem's search space.

Table 3: Comparison of performance of algorithm depending on different values of iterations( $k$ ), learning period(LP), and non-improving acceptance ratio( $r_e$ )

K	LP	$r_e$	Initial Profit	Final Profit	Improvement	Annealing	Reheating	Execution Time
4,500	1,125	0.08	32,107.14	35,481.01	3,373.87	900	-	5,525
20,000	5,000	0.08	33,068.91	41,611.76	8,542.85	4,000	-	23,147
100,000	25,000	0.08	29,844.89	40,165.82	10,320.93	20,000	-	118,180
300,000	75,000	0.08	34,131.41	47,322.70	13,191.28	60,000	-	318,640
450,000	112,500	0.08	33,211.78	47,755.21	14,543.43	90,000	-	502,185
525,000	131,250	0.08	32,572.58	45,609.31	13,036.74	105,000	-	58,228
580,000	145,000	0.08	33,463.30	44,829.86	11,366.56	16,000	-	540,236
600,000	150,000	0.08	32,946.95	47,768.38	14,821.43	20,000	-	671,459
600,000	5,000	0.08	31,486.47	46,491.25	15,004.78	20,000	-	561,561
600,000	20,000	0.08	36,221.32	46,048.99	9,827.67	20,000	-	528,742
600,000	75,000	0.08	29,967.24	45,274.86	15,307.62	120,000	-	671,804
600,000	150,000	0.08	32,552.15	46,377.38	13,825.23	120,000	-	573,000
600,000	20,000	0.10	31,875.67	46,643.73	14,768.06	120,000	-	554,845
600,000	20,000	0.20	33,564.30	45,042.11	11,477.81	120,000	-	607,424
600,000	20,000	0.25	31,710.37	48,279.44	16,569.07	119,999	9	529,692
600,000	20,000	0.27	32,042.85	48,000.70	15,957.85	119,938	340	527,931
600,000	20,000	0.30	28,266.87	46,651.72	18,384.85	119,944	327	659,856
600,000	20,000	0.40	34,076.49	44,884.43	10,807.93	119,920	453	723,050

## 5. Conclusion

A shelf space allocation problem usually involves a large number of parameters. Obtaining a reliable estimation of such parameters is generally challenging and time-consuming; therefore, it is challenging to obtain fruitful solutions in real life. This paper's proposed model is distinguishable by hyper-heuristics and the ability to solve more extensive, more realistic sized instances. This model considers the second dimension as height or stack coefficient instead of only considering the length facings of the items and the third dimension, which is the shelves' height in a rack. Thus, this model can be used to design layouts for the store to enable the retailer to maximize his profit. Moreover, it applies a hyper-heuristics method to calculate the solution using the process of simulated annealing. The algorithm used in this work is confined to shelf space allocation problems and can be used to optimize any other allocation problem. The paper tried to solve the problem of allocating shelf space by proposing a novel approach.

This paper solves the optimization problem by converting a multi-objective optimization into a single objective using the popular weighted sum technique. The proposed model increases the retailer's profit subject to various objectives and constraints. Analysis has been carried out thoroughly to determine the appropriate values of the different parameters that are quintessential to determining the retailer's profit with the aim of maximum utilization of shelf space. The weights of individual parameters result from a systematic survey done using the transaction history of various retailers. Moreover, the proposed model works on a hyper-heuristic algorithm that can provide better results than traditional heuristics, which are generally applied for optimization problems.

This work can be extended to include Pareto optimality to wholly implement multi-objective optimization considering conflicting objectives such as retailers' investments, human resources, etc. The model does not consider the effect of replenishment and backroom on retail shelf space planning. Also, improvements may be made in the probability selection of the neighborhood moves (hyper-heuristics). The selection of the neighborhoods can be based on an online learning mechanism as that selection process might provide a slightly better result for the instances. When implemented in reality, this model will help in optimizing profit but may help in planning marketing strategies and designing new store layouts.

### References

- Aarts, E., Korst, J., & Michiels, W. (2005). Simulated annealing. In *Search Methodologies* (pp. 187-210). Springer, Boston, MA.
- Bingüler, H. E., Bulkan, S., & Ağaoğlu, M. (2016). A heuristic approach for a shelf space allocation problem. *Journal of Management and Information Science*, 4(1), 38-44.
- Corstjens, M., & Doyle, P. (1981). A model for optimizing retail space allocations. *Management Science*, 27(7), 822-833.
- Cox, K. K. (1970). The effect of shelf space upon sales of branded products. *Journal of Marketing Research*, 7(1), 55-58.
- Curhan, R. C. (1972). The relationship between shelf space and unit sales in supermarkets. *Journal of Marketing Research*, 9(4), 406-412.
- Goldberg DE (1989) Genetic Algorithm in Search, Optimization, and Machine Learning Addison-Wesley publishing company, INC.
- Deb, K. (1998, January). Genetic algorithm in search and optimization: The technique and applications. In *Proceedings of International Workshop on Soft Computing and Intelligent Systems* (pp. 58-87). Machine Intelligence Unit, Indian Statistical Institute Calcutta, India.
- Dreze, X., Hoch, S. J., & Purk, M. E. (1994). Shelf management and space elasticity. *Journal of Retailing*, 70(4), 301-326.
- Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.



Haupt, R. L., & Ellen Haupt, S. (2004). Practical genetic algorithms. John Wiley & Sons.

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. *International Journal of Production Economics*, 97(2), 185-195.

Jubril, A. M. (2012). A nonlinear weights selection in weighted sum for convex multiobjective optimization. *Facta Universitatis*, 27(3), 357-372.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671-680.

Konak, A., Coit, D. W., & Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, 91(9), 992-1007.

Kotzan, J. A., & Evanson, R. V. (1969). Responsiveness of drug store sales to shelf space allocations. *Journal of Marketing Research*, 6(4), 465-469.

Laarhoven, P. J. M., Aarts, E. H. L., Lenstra, J. K. (1992). Job shop scheduling by simulated annealing. *Operations Research*, 40(1), 113-125.

Murray, C. C., Talukdar, D., & Gosavi, A. (2010). Joint optimization of product price, display orientation and shelf-space allocation in retail category management. *Journal of Retailing*, 86(2), 125-136.

Ombuki, B., Ross, B. J., & Hanshar, F. (2006). Multi-objective genetic algorithms for vehicle routing problem with time windows. *Applied Intelligence*, 24(1), 17-30.

Russell, R. A., & Urban, T. L. (2010). The location and allocation of products and product families on retail shelves. *Annals of Operations Research*, 179(1), 131-147.

Wang, Z., Geng, X., & Shao, Z. (2009). An effective simulated annealing algorithm for solving the traveling salesman problem. *Journal of Computational and Theoretical Nanoscience*, 6(7), 1680-1686.

Yang, M. H., & Chen, W. C. (1999). A study on shelf space allocation and management. *International Journal of Production Economics*, 60, 309-317.

Zadeh, L. (1963). Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*, 8(1), 59-60.

**Submitted: 26.08.2020**

**Accepted: 27.11.2020**