

A HYBRID APPROACH DEVELOPMENT TO SOLVING THE STORAGE LOCATION ASSIGNMENT PROBLEM IN A PICKER-TO-PARTS SYSTEM

Marcele Elisa Fontana
marcelelisa@gmail.com
Federal University of Pernambuco,
Caruaru, PE, Brazil.

Vilmar Santos Nepomuceno
vnepomuceno@gmail.com
Pernambuco Federal Institute of
Education, Science, and Technology,
Caruaru, PE, Brazil.

Thalles Vitelli Garcez
tvgarcez@cidsid.org.br
Federal University of Pernambuco,
Caruaru, PE, Brazil.

ABSTRACT

Goal: This study developed a structured decision model capable of solving the storage location assignment problem (SLAP) in a picker-to-parts system, using multiples key performance indicators (KPIs).

Design / Methodology / Approach: A hybrid approach was developed. For that, a Multi-Objective Genetic Algorithm (MOGA) was used considering three fitness functions, but more functions may be considered. Through MOGA it was possible to verify a high number of solutions and reduce it into a Pareto frontier. After that, a Multiple-Criteria Decision-Making (MCDM) approach was used to choose the best solution.

Results: This model was able to find viable solutions considering multiples objectives, warehouse restrictions and decision makers' preferences, and the required processing time for the simulated cases was insignificant.

Limitations of the investigation: One limitation of this work was the consideration of known and predictable data.

Practical implications: The proposed model was developed with the purpose of assisting companies that face this type of problem, providing a solution for SLAP requiring the minimum information and operational actions.

Originality / Value: SLAP is a NP (Non-Deterministic Polynomial time) complex problem and, after the MOGA, the number of solution can be still high for the final decision making by the engineering manager (decision maker - DM). Thus, the MOGA-MCDM hybrid approach developed was able incorporate the DM' preferences into a compensatory view, vetoing alternatives that were worse in any of the KPIs, to recommend a final solution.

Keywords: Warehouse Operations; Order Picking System; Storage Policy; Non-Dominated Sorting Genetic Algorithm II (NSGA-II); Additive-Veto Model.

INTRODUCTION

Consumers are demanding more agility in their orders, especially in e-commerce, and both quantity and size of orders have changed in this new trend. In this scenario, the warehouses are crucial to provide supply chain efficiency and customer satisfaction. Thus, increasing productivity and reducing operating costs become crucial, especially in warehouse with labour or capital intensive or both (Accorsi et al., 2014; Ballestín et al., 2013).

A suitable order picking (OP) system is needed in order to minimize the throughput time of picking an order while it maximizes the use of space, equipment and labour, as well as the accessibility to all items (Chan and Chan, 2011). One of the ways used to optimize the OP operations is the study of storage location assignment problem (SLAP), which is applied to “find an effective way to locate products in a warehouse in order to improve the operational efficiency of order picking” (Pan et al., 2015).

When the SLAP seeks all possible solutions, it can be characterized by combinatorial optimization problems. Thus, SLAP is a NP-hard problem, where Evolutionary Algorithms have become very popular to solve it. However, in the literature, authors considered only one objective function. But, several factors can affect the performance of SLAP, such as: order picking method, size and layout of the storage system, material handling system, product characteristics, demand trends, turnover rates and space requirements (Chan and Chan 2011). For this, many authors argue that, for a decision maker (DM) consider all of these aspects simultaneously to define the best storage location is extremely difficult (Fontana and Cavalcante, 2013; 2014; Fontana and Nepomuceno, 2017). Thus, a Multi-Objective Genetic Algorithms (MOGA) can be used (Rabbani et al., 2018). “The Genetic Algorithms are considered to be a powerful technique of stochastic optimization and, probably the most important evolutionary computer techniques” (Serra Costa, 2011).

However, in multi-objective optimization, the DM's preferences are not taken into consideration. This means that a specific final solution is not indicated, since a DM's preferences are not incorporated into the model for combining objectives. In multiple-criteria decision-making (MCDM) approaches, the objectives are combined based on the DM's preferences. These preferences consist of the DM's subjective evaluation of the criteria (i.e., objectives). This subjectivity is an inherent part of the problem, and it cannot be avoided (Almeida et al., 2015).

Nevertheless, there is a need to use MCDM to evaluate non-dominated alternatives from the Multi-Objective

Optimization. In this sense but in another context, Srivastav and Agrawal (2017) used Multi-objective particle swarm optimization (MOPSO) which is used to generate Pareto curves, and then, they used the technique of order preference by similarity to ideal solution (TOPSIS) to rank these non-dominated solutions, and determine the best compromise solution among them.

However, the compensatory nature of the additive aggregation model may recommend a solution with a very low performance in one of the criteria (e.g., highest total picking time to service all customer in warehouse), which is compensated by high performance in one or more of the other criteria (e.g., lowest total space required to store all SKUs in warehouse). For any of these cases, in order to choose the best SLAP, DM may consider having a total compensation between the undesirable criteria. Especially, when a solution reaches a very low degree in one or more of the criteria.

Therefore, at least a minimum performance to be achieved in each criterion should be predetermined. Hence, the additive-veto model, proposed by De Almeida (2013), may solve this problem by vetoing alternatives in such situations. The veto concept has the role of eliminating alternatives that do not match up to the DM's preferences, even if their overall performance in the additive model is good.

Therefore, this work developed a hybrid model to solve the SLAP for a picker-to-parts system and generating key performance indicators (KPIs) in warehouse. According to Bahrami et al. (2019), future research should focus on other performance measures, beside economics measures. Thus, three objectives, or KPIs, were considered, that were: (a) storage space used, (b) travel distance, and (c) travel time for retrieval operations. Moreover, proximity restrictions between SKUs were considered. A Multi-Objective Evolutionary Algorithm, the non-dominated sorting genetic algorithm II (NSGA-II) was proposed in order to generate solutions. Finally, all admissible solutions in the Pareto Front were observed and one of them was chosen as the final recommendation through a MCDM method, using the additive-veto model, given the compensatory rationality of DM.

The major contribution of this model is to give a SLAP solution considering the subjective opinion from the engineering manager (DM) and not only the use of technical aspects, as it is commonly considered in the literature. In addition, studies on the adoption of MOGA and MCDM hybrid methodologies are still recent, especially in SLAP. Finally, Horne et al. (2015) argued that “research suggests that one possible key to development may be the growth

of micro-entrepreneurial firms”. Moreover, “application of information systems in warehouse management is a growing tendency” (Bahrami et al., 2019). However, many of these firms suffer with the same pressures as large enterprises for better service levels, without the same economic support and skilled labour, which prevents the acquisition of complete Warehouse Management Systems, for example. Furthermore, these systems require special and expensive training, because of their complex interfaces. Hence, employees can misuse it, so operations made based on daily experiences, and not on critical factors. Consequently, major consequences can be generated, such as: product movements unnecessarily, loss storage location of SKUs and delays in shipment. In this case, for micro-entrepreneurial firms, any improvement in terms of cost and management may bring competitive advantage for these companies, and so the proposed hybrid model is able to make this improvement, requiring the minimum information and operational actions.

This paper was structured as follows. Section 2 presents the decision model development, considering the MOGA and MCDM approaches. After that, a realistic case demonstrates the applicability of the decision model proposed. Finally, some discussions and concluding remarks are made.

LITERATURE REVIEW

“Storage, per se, is not only an important process in a warehouse, also it has the greatest influence on the most expensive one, i.e., order picking” (Bahrami et al., 2019). The Order Picking (OP) system can be grouped basically into two: (1) Parts-to-picker systems and (2) Picker-to-parts systems. The first is related to the automatic storage and retrieval systems (AS/RS) (Reis et al., 2017); while in the second the warehouses employ humans in order picking operations. In this last, pickers use manual storage and retrieval system – they can use order-pick truck or not – and, usually, the picker picks an order in small quantities (Calzavara et al., 2019; Chen et al., 2016; Fontana and Nepomuceno, 2017; Pan et al., 2015).

In this sense, the right storage location assignment (SLA) is important mainly to minimize the OP cost and required storage space (Ang et al., 2012; Öztürkoğlu, 2018). In this sense, Hausman et al. (1976) submitted three basic storage policies: random storage, dedicated storage, and class-based storage.

Random storage (RS) is widely used in many warehouses because it is simple to use, and often requires less space than other storage methods (Petersen and Aase 2004). In this case, a storage keep unit (SKU) may be assigned

to any empty location. While, dedicated storage (DS) prescribes a specific location where each material must be stored (Rouwenhorst et al., 2000); no other item can be stored there, even if it is empty. According to Fontana and Cavalcante (2013), in general, there is an increase in the space required cost when uses a DS, while the RS increase the OP effort. It is because RS has all SKUs in a single class, while DS has each SKU assigned to a separate class (Chan and Chan 2011).

Class-based storage (CBS) combines feature of both RS and DS. It divides the SKUs into classes and assigns to each class a set of areas in which the items are located in whatever way, i.e., randomized (Guerriero et al., 2013). Thus, it can be a good and more efficient alternative in terms of storage space used and the order picking operation, minimizing storage costs (Fontana and Cavalcante 2013). Therefore, according to Muppani and Adil (2008a), the implementation of the CBS involves determining the number of classes, which SKUs are assigned into these classes and what will be the place for each class in the warehouse.

Some authors have proposed to optimize this problem by: robust optimization (Ang et al., 2012; Ang and Lim, 2019) and mixed integer linear programmin (Muppani and Adil, 2008a; Yener et al., 2019), for example.

However, there is a NP-hard problem (n^N alternatives) when a SLAP is solved in a CBS policy and by combinatorial optimization way. In this way, some heuristic methods were used, such as: Simulated Annealing Algorithm (Muppani and Adil, 2008b), Tabu search algorithm (Chen et al., 2010), Minimum Delay Algorithm (MDA) (Wutthisirisart et al., 2015), Differential Evolution (DE) and Global Local and Near-Neighbor Particle Swarm Optimization (GLNPSO) (Wisittipanich and Kasemset, 2015), Genetic Programming (Xie et al., 2014), and Genetic Algorithm (GA) (Ene and Öztürk, 2012; Pan et al., 2015). However, all of these works used only one objective function.

The inclusion of multiple-objectives can lead to Pareto frontier with multiple solutions, which will not help the decision maker. In this regard, some works used Multiple-Criteria Decision-Making (MCDM) approach in SLAP: non-compensatory rationality, as PROMETHEE and ELECTRE methods (Fontana and Cavalcante, 2013; 2014; Fontana and Nepomuceno, 2017), and compensatory rationality, as SMARTER method (Silva et al., 2015). These MCDM authors aimed to establish one dedicated storage or class-based storage solution or choice one solution from a small set of them. However, none managed to generate n^N alternatives from MDCM. Therefore, the MOGA and MCDM methods hybridization may be relevant to generate n^N alternatives and analyse multicriteria simultaneously.

METHODOLOGICAL PROCEDURES AND PROPOSED MODEL

The hybrid decision model developed in this paper is able to solve the SLAP and it improves the key performance indicators (KPIs) in warehouse. Figure 1 shows the step of this proposed model.

INPUTS: Assumptions and warehouse features

The purpose of this step is to enter the system with all the necessary information that will be classified, such as, product information, warehouse information (layout and operations) and DM's preferences. Firstly, the model allowed considers both warehouse in single layer (2D) and multilayer warehouse (3D). In this case, the DM will previously select the type of warehouse. Thus, there is n SKUs p_j , or SKUs, such that $j = \{1, 2, \dots, n\}$ and, based on (Fontana and Nepomuceno, 2017), it can be assumed that:

- The picker-to-parts systems and unit-load warehouse are considered;
- Each pallet contains only one SKU and it may contain multiple units thereof;
- The material handling will do always by full pallet, i.e., it will not be allowed the fragmentation of the load on the pallet and loads are not relocated;
- The order picking operation occurs by single-command operation, i.e., the pallet is removed from the storage place and placed to the shipping area (p/d point);
- All times required in the storage/retrieval process are considered independent of storage allocation, except travel times;
- The pallet unitizing must comply with the resistance of product packaging, shelf capacity and the restrictions of material handling equipment (in multilayer warehouse, pallet truck or turret-trucks);

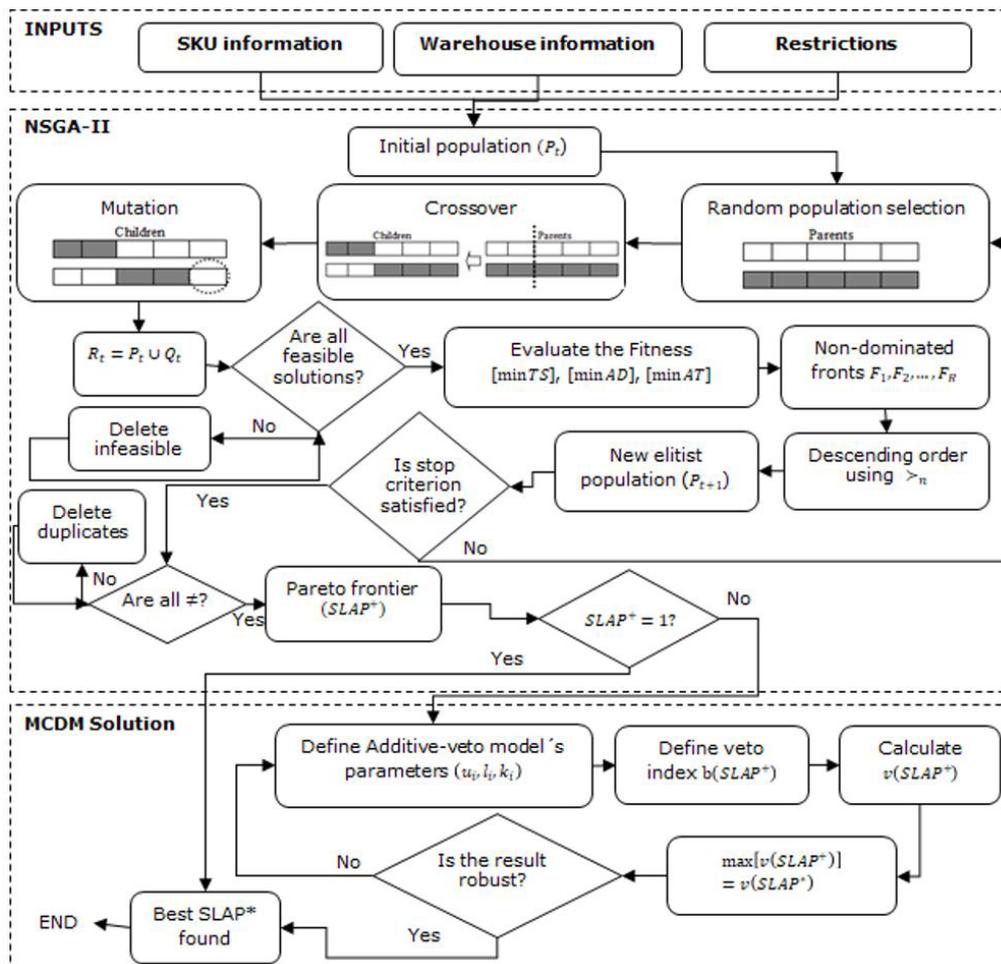


Figure 1. Steps of the model developed
Source: The authors themselves

- No pallet may be placed over each other, regardless of the quantity of items.

Therefore, the spaces used to storage any SKU will be standardized and adjusted to the pallet size. Information on the number of SKUs per pallet will only serve to inventory control by the company.

Storage locations are at both sides of shelves, except on the shelves near the warehouse walls. Thus, since there are m available spaces, each space s_i can be represented by a coordinate $s_i = \{x_i, y_i, z_i\}$, in which $i = \{1, 2, \dots, m\}$ and:

- x represents one side of the shelves, such that $x = \{1, 2, \dots, n_x\}$
- y represents the row of the shelves, such that $y = \{1, 2, \dots, n_y\}$ and;
- z represents the shelf level, such that $z = \{1, 2, \dots, n_z\}$.

This coordinate serves as an identifier of each space and it addresses the SKU. The number n_y is not fixed, because it is assumed that there is enough space for all SKUs. When single layer (2D) are considered, the value of z will be disregarded.

Figure 2 shows an illustration of the warehouse layout considered here. Note that the warehouse can be rectangular

or square, this is not relevant. However, the shelves must be in the perpendicular direction in relation to the door (pick-up/drop-off (p/d) point). Figure 2 is not represented on a real scale, and it does not represent the physical structure of the shelves, only the blocks that represent the available spaces.

Non-dominated sorting genetic algorithm II (NSGA-II)

A solution in NSGA II is represented by a chromosome. Each chromosome $Cr_i = \{[g_1], [g_2], \dots, [g_n]\}$ is composed by n genes $[g_j]$, since n is the total number of SKU. Each gene represents a class and local for a specific SKU, and it is composed by alleles. In this case, the alleles received just one interior value j , such that $1 \leq j \leq n$.

Let us consider an example. Given a set F with 04 SKUs, such that $F = \{a, b, c, d\}$, $n_x = 6$, $n_y = 6$ and $n_z = 3$. If $Cr_i = \{[2], [3], [4], [1]\}$, this means a dedicated storage policy in which the rank (position) from p/d point to back is following: d , a , b and c . Additionally, let's consider the maximum warehouse space allocated by each SKU as: $d = 16$, $a = 32$, $b = 25$, and $c = 35$. Thus, the allocation rule follows the sequence given by the addressed spaces in Figure 2, i.e., from s_1 to $s_{n_x \times n_y \times n_z}$. In this case, the SLAP solution can be seen in Figure 3.

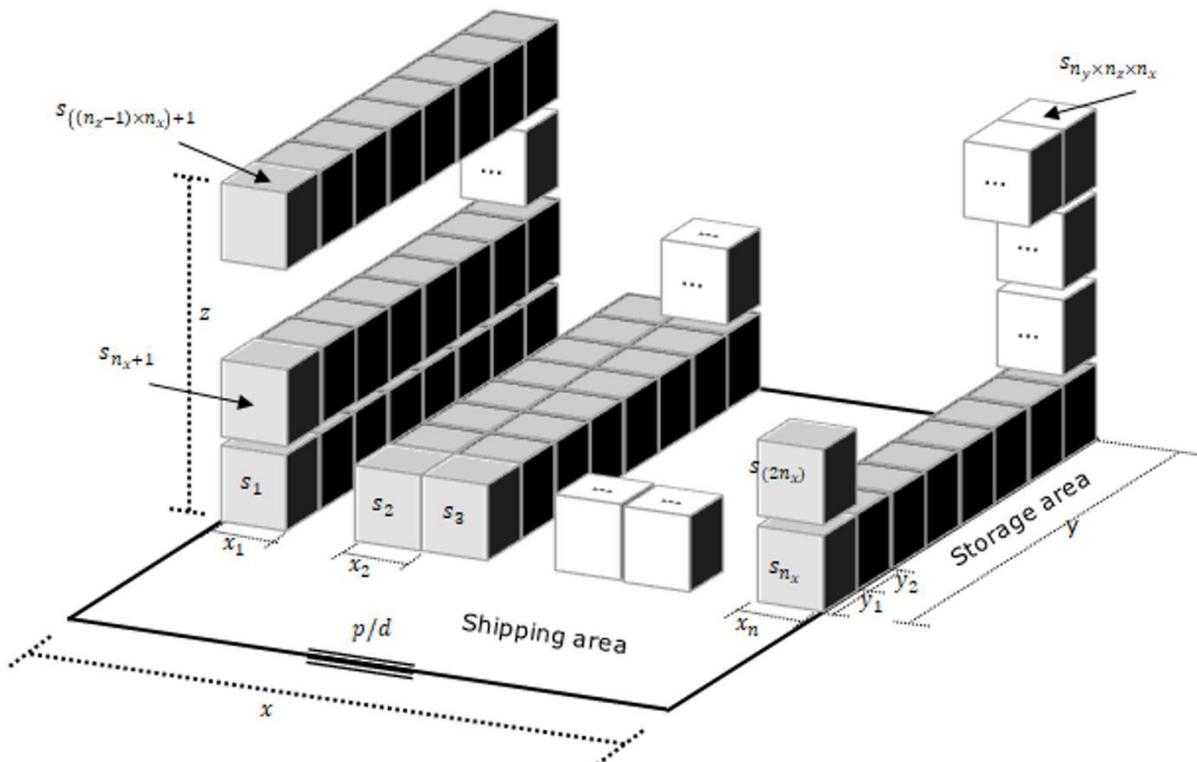


Figure 2. Warehouse layout: shelf levels (3D)
 Source: The authors themselves

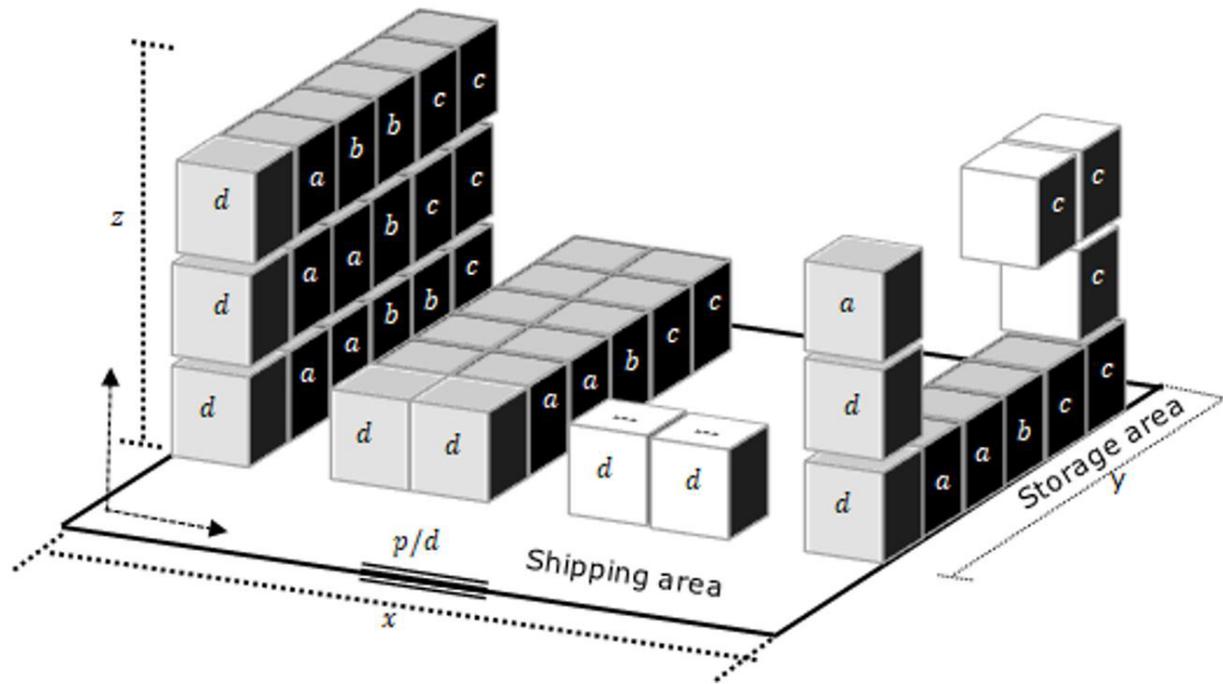


Figure 3. Location of SKUs in warehouse
 Source: The authors themselves

Moreover, when two genes receive the same number, this features an aggregation in class. For example, if a new chromosome is $Cr_i = \{[2],[3],[3],[1]\}$, this means a class formed by b and c . In this case, a random storage policy is represented by $Cr_r = \{[1],[1],[1],[1]\}$.

The NSGA-II uses a fixed population size of N . Thus, the framework of NSGA II used in this work is an adaptation of (Deb et al., 2002; Shi et al., 2017):

- Step 1. Set $t = 0$, and generate an initial random parent population P_t ,
- Step 2. Generate an offspring population Q_t of size N by selection, crossover and mutation operations.
- Step 3. Combine parents and offspring population, $R_t = P_t \cup Q_t$,
- Step 4. Eliminate not feasible solutions and calculate fitness,
- Step 5. Perform non-dominated fronts F_1, F_2, \dots, F_R
- Step 6. Generate the next population P_{t+1} :
 - Step 6.1. Until the parent population is filled, i.e., $|P_{t+1}| + |F_i| \leq N$ calculate crowding distances in F_i , and $P_{t+1} = P_{t+1} \cup F_i$,
 - Step 6.2. Check the next front for inclusion, $i = i + 1$, and sort in descending order using \succ_n ,

- Step 6.3. Choose the first $(N - |P_{t+1}|)$ solutions in F_i to added in P_{t+1} , i.e., $P_{t+1} = P_{t+1} \cup F_i [1 : (N - |P_{t+1}|)]$,
- Step 7. Increment de generation counter: $t = t + 1$
- Step 8. Go to step 2 until the stopping conditions are satisfied.

In each interaction t , the pair of parents is formed randomly from the selected individuals. For each pair of parents there are two possibilities: (1) a crossover occurs (generation of children) or (2) it doesn't. Given a uniform probability distribution, each pair of parents has a $\alpha\%$ chance to have children, that is, the crossover rate is α . The crossover is used with 1-partition, i.e., a cut point is selected randomly, and the portions of the parents before, and after this point are combined generating two children (Yang et al., 2007).

Additionally, a mutation rate is adopted to prevent premature convergence of the solution. "Mutation operator is to alter randomly one or more gene codes in a chromosome from its initial state with a certain probability" (Pan et al., 2015). Thus, the process begins with the choice of a random point (gene) of an individual; a very small likelihood rate ρ is then applied to exchange this point for another. Therefore, given a random variable β , with $0 \leq \beta \leq 1$, in each chromosome this variable is rotated to each of its

genes (as a simple roulette). If $\beta > \rho$ no mutation, otherwise the gene is mutated.

Restrictions

In sequence, for a solution to be considered, it must respect the restrictions of the objective function established by the DM. These restrictions are related to the undesired proximity between some SKUs in the warehouse, as in Equation 1 and Equation 2.

$$F = \{p_r, p_s\} \rightarrow Cr_i = \left\{ \left[g_{p_r} \right], \left[g_{p_s} \right] \right\} \forall g_{p_r} \neq g_{p_s} \quad (1)$$

$$F = \{p_r, p_s\} \rightarrow Cr_i = \left\{ \left[g_{p_r} \right], \left[g_{p_s} \right] \right\} \forall g_{p_r} = g_{p_s} \quad (2)$$

In which: (1) defines that the SKU p_r must not be in the same class as a SKU p_s ; and (2) defines that the SKU p_r must be in the same class as a SKU p_s ; After that, the fitness functions of each solution are analyzed.

Fitness functions

In this work, there are three fitness functions or key performance indicators (KPIs), but others may be considered. In SLAP studied, it is desired to decrease the total distance travelled during the order picking operations, and decrease the total space required to minimize the operating cost of a warehousing system. In order to improve consumer service level, decrease the total time travelled in order picking activity is also desired. In cases where single layer (2D) is considered, this time can be ignored, because it will be proportional to the travel distance. However, in multilayer warehouse (3D) the effort required for picking the SKUs can be different at each level of the shelf.

Therefore, the fitness functions are given by Equation 3 to 5.

$$\text{minimize} \left[TS = \sum \max N_{pc}^t \right] \quad (3)$$

$$\text{minimize} \left[AD = \sum AD_c \times D_{pc}^t \right] \quad (4)$$

$$\text{minimize} \left[AT = \sum AT_c \times D_{pc}^t \right] \quad (5)$$

In which,

TS : is the total space required to store all SKUs in warehouse;

AD : is the total distance travelled to serve all customer in warehouse;

AT : is the total picking time to serve all customer in warehouse;

AD_c : is the total distance travelled to pick all SKUs in class c ;

AT_c : is the total picking time to pick all SKUs in class c ;

D_{pc}^t : is total number of pick (number of retrieval operations per unit time period or popularity), in unit loads, for SKU p in class c during the planning period t .

N_{pc}^t : is the storage level in number of spaces required planned for all SKU p in class c during the planning period t (in units).

The total warehouse space reserved N_p^t to storage each SKU p can be estimated by the total number of pallets stored at a period of time t . While, for the distance AD_c can be used the Equation 6. This considered as reference point the centre of the aisle, and the movement of the SKUs is done by complete pallet, which represents exactly one storage space. Moreover, all movement to the left or right inside the aisle, in order to reach an item, will be the same for all spaces (s_i). Thus, in this problem, this displacement can be neglected.

$$AD_c = \frac{\sum_{i=0}^{N_{pc}^t} D_{s_i}}{N_{pc}^t} \quad (6)$$

and,

$$D_{s_i}(x, y, z) = d_{xi} + d_{yi} + d_{zi} \quad (7)$$

$$\begin{cases} d_{xi} = \frac{a_{i+1}}{2} \forall x_i = \text{odd} \\ d_{xi} = a_i \forall x_i = \text{even} \end{cases} \quad (8)$$

$$d_{yi}(y, w_p) = w_p \left(y_i - \frac{l}{2} \right) \quad (9)$$

$$d_{zi}(z, h_p) = (z_i - l) h_p \quad (10)$$

$$a_i(w_a, n_a, n_x, l_p) = \left[\left(\frac{w_a \times n_a}{2} + \frac{l_p \times n_x}{2} \right) \right] - \left[(2i - l) \times \left(l_p + \frac{w_a}{2} \right) \right] \quad (11)$$

In which,

D_{s_i} : is the total travel distance from s_i to p/d point [m];

d_{xi} : is the distance travelled in axis x [m];

d_{yi} : is the distance travelled in axis y [m];

d_{zi} : is the distance travelled in axis z [m];

x_i : is the corresponding the shelf of the space s_i [unit];

y_i : is the corresponding row for the space s_i [unit];

z_i : is the corresponding level of shelf for the space s_i [unit];

a_i : is the distance for the corresponding aisle of the space s_i [m];

n_a : is the total number of aisles [unit];

n_x : is the total number of sides of the shelves [unit];

w_a : is the width of the aisle [m];

l_p : is the space length for a pallet on shelf [m];

w_p is the space width for a pallet on shelf [m];

h_p : is space height for a pallet in shelf [m].

According to Chan and Chan (2011), a storage location assignment with less distance trip is not necessarily one with less picking time. Thus, the total picking time to service all customer orders in class c (AT_c), during period t , will be given by Equation 12.

$$AT_c = \frac{\sum_{i=0}^{N_{pc}^t} \left[\left(\frac{d_{xi} + d_{yi}}{r_{xy}} \right) + \left(\frac{d_{zi}}{r_z} \right) \right]}{N_{pc}^t} \quad (12)$$

In which,

r_{xy} : is average picking speed in horizontal direction by pallet truck (axis x and/or y) [m/s];

r_z : is the average picking speed in vertical direction for loading and unloading by pallet truck (axis z) [m/s].

Therefore, the NSGA-II process is repeated until it reaches the stop criterion T . The stop criterion adopted was to limit the number of generations that the algorithm will execute.

MCDM solution

After the NSGA-II phase, only non-dominated solutions are considered in MCDM method. In other words, if there is more than one non-dominated alternative, the additive-veto model is performed. Otherwise, the best SLAP* is found. Thus, in the additive-veto model, the level of veto, which is at the limit placed on accepting the performance of alternatives for each criterion, is represented by two thresholds (Upper and Lower veto thresholds), representing an indeterminate region of some vagueness in the DM's specification of level of veto.

In this case, the set of criteria are the fitness functions, or key performance indicators (KPIs), Equations 3-5. For each

criterion i , a veto function b_i is assumed. The veto function $b_i(a)$ for alternative a is given by Equation 13. Therefore, the veto function $b_i(a)$ is integrated into the additive model to provide the global value $V(a)$ for alternative a , as given by Equation 14. In this particular study, each alternative a represents a SLAP solution.

$$b_i(a) = \begin{cases} 0, & \text{if } v_i(a) \leq l_i \\ 1, & \text{if } v_i(a) \geq u_i \\ \frac{v_i(a) - l_i}{u_i - l_i}, & \text{if } l_i < v_i(a) < u_i \end{cases} \quad (13)$$

$$V(a) = \left[\sum_{i=1}^m (k_i b_i(a)) \right] \left[\sum_{i=1}^m (k_i v_i(a)) \right] \quad (14)$$

In which,

u_i : is the upper veto threshold;

l_i : is the lower veto threshold;

$v_i(a)$: is the value function for criterion i ;

k_i : is the scaling constant for criterion i , where $\sum_{i=1}^m k_i = 1$;

m is the number of criteria.

If the index $\sum_{i=1}^m b_i(a) k_i = 1$, then the performance of the alternative is acceptable for all the criteria, i.e., $b_i(a) = 1$, for all i . Case $0 < \sum_{i=1}^m b_i(a) k_i < 1$, then the alternative is being partially vetoed for any criterion. Case $\sum_{i=1}^m b_i(a) k_i = 0$, then the performance of the alternative is unacceptable for all the criteria. It concludes that the value of the alternative $v(a)$ is being reduced by a proportion related to the weight of the criterion by which the alternative is vetoed. Therefore, indicates that the value of the alternative $v(a)$ is reduced proportionally to the summation of the weights of the criteria for which this alternative is vetoed.

This procedure is repeated for other alternatives found by NSGA-II ($SLAP^+$). Therefore, the best alternative ($SLAP^*$) is the $\max [v(SLAP^+)]$. Moreover, a sensitivity analysis step should be conducted. The objective of this step is to verify whether the recommendation of the alternative is robust or sensitive to the input data, model resources, or DM's preference parameters. The values of the veto thresholds and scaling constant for criterion are altered to verify if it changes the alternative chosen. Therefore, this step may show that the additive-veto model's parameters should be reevaluated or not by the DM.

In order to facilitate the process for define the parameters u_i and l_i , here is suggested that the DM use a percentage



of performance (1%, 5%, 10%, etc.) of the worst values in each criterion. For example, let's consider AT^- to be the worst total picking time possible to serve all customers in warehouse. Thus, the DM may define to be 10% of this value as unacceptable, defining the $l_i = 0.10AT^-$. I.e., this is the minimum value of performance in criterion given by Equation 5, which the DM is sure to reject the alternative, independently of its performance in other criteria. Similarly, the DM define the upper veto threshold ($u_i = 0.15AT^-$), defining the minimum value of performance for any alternative to criterion i , in which it is widely acceptable for DM in this criterion.

RESULTS

A realistic case was considered based on a warehouse similar as in Figure 2. Thus, a small case with only 03 SKUs will be presented to validate and demonstrate the applicability of

the developed model. The inputs used can be seen in Table 1. All previously reported assumptions have been respected.

In this case, the number of possible solutions is $n^n = 27$. However, some solutions are considered identical, called here as duplicates. All possible solutions, generated in a deterministic manner, can be seen in Table 2, and all duplicates are allocated in the same place.

In GA, the following parameters were considered: crossover rate $\alpha = 0.1$; mutation rate $\rho = 0.2$; population size $N=15$; number of generations $T=30$. Thus, in scenario 1 the proposed model resulted in the following solutions: N^o 2: ($Cr_2 = \{[1], [3], [2]\}$); N^o 4: ($Cr_4 = \{[2], [3], [1]\}$); N^o 7: ($Cr_7 = \{[1], [1], [1]\}$, $Cr_8 = \{[2], [2], [2]\}$, $Cr_9 = \{[3], [3], [3]\}$); and N^o 9: ($Cr_{14} = \{[1], [3], [1]\}$, $Cr_{15} = \{[2], [3], [2]\}$). Only N^o 7: ($Cr_{13} = \{[1], [2], [1]\}$) was not found.

In scenario 2 and 3, due to the number of viable solutions, the population size decrease from 15 to 5, other parameters

Table 1. Inputs of SLAP problem

SKU information							
n	SKU Code (p_j)	Warehouse space reserved				Popularity	
		N_p^1	N_p^2	N_p^3	N_p^4	N_p^t	
1	p_1	27	38	25	20	50	
2	p_2	30	18	18	44	10	
3	p_3	09	20	00	15	25	
Warehouse information						Restrictions	
n_x	6.0 m	w_p	1.0 m	l_p	1.0 m	Scenario 1	---
n_a	3.0 m	w_a	1.0 m	r_{xy}	0.5 m/s	Scenario 2	$g_{p_1} \neq g_{p_2}$
n_z	3.0 m	h_p	1.5 m	r_z	0.2 m/s	Scenario 3	$g_{p_1} = g_{p_2}$

Source: The authors themselves

Table 2. All possible results by deterministic model

N ^o	Solutions $Cr_i = \{[p_1], [p_2], [p_3]\}$	TS	AD	AT	Pareto Front		
					Scenario 1	Scenario 2	Scenario 3
1	$Cr_1 = \{[1], [2], [3]\}$	102	511.49	1389.55	No	No	Eliminated
2	$Cr_2 = \{[1], [3], [2]\}$	102	453.20	1263.76	Yes	Yes	Eliminated
3	$Cr_3 = \{[2], [1], [3]\}$	102	608.37	1593.11	No	No	Eliminated
4	$Cr_4 = \{[2], [3], [1]\}$	102	455.57	1268.50	Yes	Yes	Eliminated
5	$Cr_5 = \{[3], [1], [2]\}$	102	611.53	1599.42	No	No	Eliminated
6	$Cr_6 = \{[3], [2], [1]\}$	102	565.51	1498.18	No	No	Eliminated
7	$Cr_7 = \{[1], [1], [1]\}$; $Cr_8 = \{[2], [2], [2]\}$; $Cr_9 = \{[3], [3], [3]\}$	79	477.72	1308.89	Yes	Eliminated	Yes
8	$Cr_{10} = \{[1], [1], [2]\}$; $Cr_{11} = \{[1], [1], [3]\}$; $Cr_{12} = \{[2], [2], [3]\}$	84	496.56	1350.31	No	Eliminated	No
9	$Cr_{13} = \{[1], [2], [1]\}$; $Cr_{14} = \{[1], [3], [1]\}$; $Cr_{15} = \{[2], [3], [2]\}$	102	454.32	1265.82	Yes	Yes	Eliminated
10	$Cr_{16} = \{[1], [2], [2]\}$; $Cr_{17} = \{[1], [3], [3]\}$; $Cr_{18} = \{[2], [3], [3]\}$	97	477.97	1315.92	No	Yes	Eliminated
11	$Cr_{19} = \{[2], [1], [1]\}$; $Cr_{20} = \{[3], [1], [1]\}$; $Cr_{21} = \{[3], [2], [2]\}$	97	568.65	1500.54	No	Yes	Eliminated
12	$Cr_{22} = \{[2], [1], [2]\}$; $Cr_{23} = \{[3], [1], [3]\}$; $Cr_{24} = \{[3], [2], [3]\}$	102	609.93	1596.22	No	No	Eliminated
13	$Cr_{25} = \{[2], [2], [1]\}$; $Cr_{26} = \{[3], [3], [1]\}$; $Cr_{27} = \{[3], [3], [2]\}$	84	478.12	1310.62	No	Eliminated	No

Source: The authors themselves



remaining the same. In scenario 2, the proposed model resulted in N° 2: ($Cr_2 = \{[1], [3], [2]\}$); N° 4: ($Cr_4 = \{[2], [3], [1]\}$); N° 9: ($Cr_{13} = \{[1], [2], [1]\}$, $Cr_{14} = \{[1], [3], [1]\}$); N° 10: ($Cr_{16} = \{[1], [2], [2]\}$, $Cr_{17} = \{[1], [3], [3]\}$, $Cr_{18} = \{[2], [3], [3]\}$); and N° 11: ($Cr_{19} = \{[2], [1], [1]\}$, $Cr_{20} = \{[3], [1], [1]\}$, $Cr_{21} = \{[3], [2], [2]\}$). Only N° 9: ($Cr_{15} = \{[2], [3], [2]\}$) was not found. Finally, in scenario 3, the proposed model found exactly three solutions that are: N° 7: ($Cr_7 = \{[1], [1], [1]\}$, $Cr_8 = \{[2], [2], [2]\}$, $Cr_9 = \{[3], [3], [3]\}$).

Those solutions do not found are duplicated and they are eliminated from the next phase of this proposed model. Thus, it does not represent a problem here. Therefore, by eliminating duplicates, scenario 3 has only one acceptable solution, that is, the random storage policy. In this way, the additive-veto model was not performed. For other cases, the veto thresholds were established from the maximum values for each criterion defined by the DM, as shown in Table 3. In this work, three different levels of maximum values were used to verify the consequences, as a sensitivity analysis.

The final rank by use of the additive-veto model changes only in scenario 2, as can be seen in Table 3. The alternative N° 10 ($Cr_{16} = \{[1], [2], [2]\}$, $Cr_{17} = \{[1], [3], [3]\}$, $Cr_{18} = \{[2], [3], [3]\}$) is 4th rank when applying the additive model

without veto. However using the additive-veto model, it is the 1st placed. There is a promotion of three positions. Furthermore, the scaling constants of criteria were varied by $\pm 30\%$. The final rank was not changed.

In order to test the model parameters, a new simulation with 10 SKUs was performed. A similar warehouse with $n_x = 10$ was considered, where the dimensions $w_p = 0.8$ and $l_p = 1.2$ were established. The space required (N_p^t) and popularity (D_p^t) were randomly generated. Moreover, 05 shelf levels were considered, $n_z = \{1, 2, 3, 4 \text{ and } 5\}$, except to $n_z = 1$, 02 space height for a pallet in shelf were simulated: $h_p = 1.5$ and $h_p = 3.0$, totalling 09 different scenarios.

For each scenario, the mutation rate was set at $\beta = 10\%$, because it believed that it was already a high enough rate to prevent early convergence. The number of generations and the population size were set at 200 and 30, respectively. The crossover rate was alternated at three levels: (a) $\alpha = 40\%$, (b) $\alpha = 60\%$ and (c) $\alpha = 80\%$. Table 4 presents the results obtained considering the following parameters for the Additive-veto model: $\max(TS) = 1575.0$; $\max(AD) = 128741.8$; $\max(AT) = 257483.5$; $u_i(TS) = 50\%$; $u_i(AD) = 50\%$; $u_i(AT) = 50\%$; $l_i(TS) = 25\%$; $l_i(AD) = 25\%$; $l_i(AT) = 25\%$; $k_i(TS) = 0.333$; $k_i(AD) = 0.333$; and, $k_i(AT) = 0.333$.

Table 3. Additive-veto model result

Maximum value				
		TS	AD	AT
Situation 1		117	703	1839
Situation 2		112	673	1759
Situation 3		107	642	1679
Parameters				
u_i		50%	50%	50%
l_i		25%	25%	25%
k_i		0.333	0.333	0.333
Ranking				
	N° Alternative			
	Additive model Without veto	Additive-veto model		
		Situation 1	Situation 2	Situation 3
Scenario 1				
1 st	7	7	7	7
2 nd	2	2*	2*	2*
3 rd	9	9*	9*	9*
4 th	4	4*	4*	4*
Scenario 2				
1 st	2	10 (up 3 positions)*	10 (up 3 positions)*	10 (up 3 positions)*
2 nd	9	2 (down 1 position)*	2 (down 1 position)*	2 (down 1 position)*
3 rd	4	9 (down 1 position)*	9 (down 1 position)*	9 (down 1 position)*
4 th	10	4 (down 1 position)*	4 (down 1 position)*	4 (down 1 position)*
5 th	11	11*	11*	11*

*Partially vetoed by criterion TS. Source: The authors themselves

Table 4. Proposed model results for 10 SKUs

Scenarios	Ranking partially vetoed with changed ranking	SENSITIVITY ANALYSIS (-pp%)			SENSITIVITY ANALYSIS (+pp%)			
		Criterion TS	Criterion AD	Criterion AT	Criterion TS	Criterion AD	Criterion AT	
Code	$n_z; h_p$	Rank with veto	Rank c/ veto	Rank c/ veto	Rank c/ veto	Rank c/ veto	Rank c/ veto	
S1(a)	1; 1.5	Change from 3 rd to 7 th	-99%: 1 st &2 nd	-92%: 1 st &2 nd	NAR<-100%	97%: 1 st &2 nd	<100%NAR: 1 st	NAR<-100%
S1(b)	1; 1.5	Change from 2 nd to 7 th	-57%: 1 st &2 nd	<-100%NAR: 1 st	<-100%NAR: 1 st	<100%NAR: 1 st &2 nd	95%: 1 st &2 nd	<100%NAR: 1 st
S1(c)	1; 1.5	Change from 1 st to 7 th	-60%: 1 st &3 rd	-8%: 1 st &2 nd	-29%: 1 st &2 nd	7%: 1 st &2 nd	98%: 1 st &3 rd	<100%NAR: 1 st
S2(a)	2; 1.5	NAR	-90%: 1 st &3 rd	<-100%NAR: 1 st	<-100%NAR: 1 st	64%: 1 st &6 th	NAR<-100%	NAR<-100%
S2(b)	2; 1.5	NAR	-35%: 1 st &5 th	<-100%NAR: 1 st	<-100%NAR: 1 st	<100%NAR: 1 st	73%: 1 st &5 th	67%: 1 st &5 th
S2(c)	2; 1.5	NAR	-47%: 1 st &3 rd	<-100%NAR: 1 st	<-100%NAR: 1 st	<100%NAR: 1 st	97%: 1 st &3 rd	90%: 1 st &3 rd
S3(a)	2; 3.0	NAR	-9%: 1 st &3 rd	<-100%NAR: 1 st	<-100%NAR: 1 st	<100%NAR: 1 st	20%: 1 st &3 rd	17%: 1 st &3 rd
S3(b)	2; 3.0	NAR	-19%: 1 st &2 nd	<-100%NAR: 1 st	<-100%NAR: 1 st	73%: 1 st &6 th	40%: 1 st &2 nd	36%: 1 st &2 nd
S3(c)	2; 3.0	NAR	-90%: 1 st &4 th	-70%: 1 st &2 nd	-60%: 1 st &2 nd	32%: 1 st &2 nd	<100%NAR: 1 st	<100%NAR: 1 st
S4(a)	3; 1.5	NAR	-93%: 1 st &2 nd	<-100%NAR: 1 st	-99%: 1 st &5 th	55%: 1 st &5 th	<100%NAR: 1 st	<100%NAR: 1 st
S4(b)	3; 1.5	NAR	-13%: 1 st &2 nd	<-100%NAR: 1 st	<-100%NAR: 1 st	<100%NAR: 1 st	28%: 1 st &2 nd	25%: 1 st &2 nd
S4(c)	3; 1.5	NAR	-48%: 1 st &2 nd	<-100%NAR: 1 st	<-100%NAR: 1 st	<100%NAR: 1 st	99%: 1 st &2 nd	93%: 1 st &2 nd
S5(a)	3; 3.0	NAR	-84%: 1 st &8 th	<-100%NAR: 1 st	<-100%NAR: 1 st	<100%NAR: 1 st	<100%NAR: 1 st	<100%NAR: 1 st
S5(b)	3; 3.0	NAR	-66%: 1 st &4 th	NAR<-100%	NAR<-100%	66%: 1 st &5 th	<100%NAR: 1 st	<100%NAR: 1 st
S5(c)	3; 3.0	NAR	-73%: 1 st &6 th	<-100%NAR: 1 st	<-100%NAR: 1 st	<100%NAR: 1 st	<100%NAR: 1 st	<100%NAR: 1 st
S6(a)	4; 1.5	NAR	-41%: 1 st &3 rd	NAR<-100%	NAR<-100%	84%: 1 st &2 nd	85%: 1 st &3 rd	79%: 1 st &3 rd
S6(b)	4; 1.5	NAR	-80%: 1 st &4 th	<-100%NAR: 1 st	-95%: 1 st &3 rd	52%: 1 st &2 nd	<100%NAR: 1 st	<100%NAR: 1 st
S6(c)	4; 1.5	NAR	-75%: 1 st &2 nd	<-100%NAR: 1 st	<-100%NAR: 1 st	<100%NAR: 1 st	<100%NAR: 1 st	<100%NAR: 1 st
S7(a)	4; 3.0	NAR	-3%: 1 st &3 rd	-3%: 1 st &2 nd	-3%: 1 st &2 nd	2%: 1 st &2 nd	7%: 1 st &3 rd	6%: 1 st &3 rd
S7(b)	4; 3.0	NAR	-99%: 1 st &12 th	-94%: 1 st &9 th	-78%: 1 st &9 th	43%: 1 st &9 th	<100%NAR: 1 st	<100%NAR: 1 st
S7(c)	4; 3.0	NAR	-79%: 1 st &10 th	<-100%NAR: 1 st	<-100%NAR: 1 st	<100%NAR: 1 st	<100%NAR: 1 st	<100%NAR: 1 st
S8(a)	5; 1.5	NAR	-81%: 1 st &3 rd	NAR<-100%	NAR<-100%	NAR<-100%	<100%NAR: 1 st	<100%NAR: 1 st
S8(b)	5; 1.5	NAR	-80%: 1 st &6 th	-12%: 1 st &2 nd	-10%: 1 st &2 nd	6%: 1 st &2 nd	<100%NAR: 1 st	<100%NAR: 1 st
S8(c)	5; 1.5	NAR	-72%: 1 st &5 th	-76%: 1 st &6 th	-65%: 1 st &6 th	<100%NAR: 1 st	<100%NAR: 1 st	<100%NAR: 1 st
S9(a)	5; 3.0	NAR	-62%: 1 st &3 rd	<-100%NAR: 1 st	-65%: 1 st &6 th	96%: 1 st &8 th	<100%NAR: 1 st	<100%NAR: 1 st
S9(b)	5; 3.0	NAR	-65%: 1 st &3 rd	<-100%NAR: 1 st	<-100%NAR: 1 st	90%: 1 st &13 th	<100%NAR: 1 st	<100%NAR: 1 st
S9(c)	5; 3.0	NAR	-63%: 1 st &3 rd	<-100%NAR: 1 st	<-100%NAR: 1 st	74%: 1 st &4 th	<100%NAR: 1 st	<100%NAR: 1 st

NAR <± 100%: means that there was no change in any ranking position with change of up to ± 100% of the scaling constants; <± 100% NAR: 1st: means that there was no change in 1st rank position with change of up to ± 100% of the scaling constants; and, ± pp%: 1st & 2nd = means there was change in 1st rank position for the 2nd rank position with change of ± pp% of the scaling constants. Source: The authors themselves

In this case, only S1(c) scenario presented changing of the first-position when applied additive-veto model. Subsequently, a sensitivity analysis was performed by changing the scaling constants of the criteria, as is also shown in Table 4. In a few cases, a reversal in the alternatives first-position of rank was observed with the variation of the scaling constant to ±10% (highlighted in Table 4).

DISCUSSIONS

In this work, two realistic situations were simulated: 03 SKUs and 10 SKUs. In both the required computational time by the proposed model was negligible. However, comparing it with other heuristic methods would not

be prudent due to the subjectivity inherent in MCDM. Therefore, for the case with 03 SKUs was considered a deterministic methodology to generated solutions, i.e., all the alternatives were generated by means of combinatorial analysis and, then, they were compared to the proposed hybrid decision model. On the other hand, for 10 SKUs the deterministic method is not computationally adequate and, thus, the results from the proposed model were compared to random storage and dedicated storage. The dedicated storage assignment was made based on cube-per-order index (COI) (Fontana and Cavalcante 2014; Fontana and Nepomuceno 2017).

For the first analysed case, the results proved the robustness of the proposed hybrid. In some situations,



there were partially vetoed alternatives by the Additive-veto model. It means, the value of these alternatives were reduced by a proportion related to the weight of the criterion for which these alternatives were partially vetoed in some criterion, but not in all criteria, because $0 < \sum_{i=1}^m b_i(a) k_i < 1$. In this case, there was no significant impact between criteria AD and AT, because the criterion AT had greater impact the higher the number of levels (n_z) and the lower the average picking speed (r_z). However, this was sufficient to demonstrate that the model developed here is able to incorporate more than two objectives simultaneously.

For the second analysed case, in all simulations made the dedicated storage alternative was dominated. Even considering this alternative, it would be vetoed in all situations. The random storage alternative, though not dominated by any scenario, it is not the winner in any of them either. The best position of this alternative was in the scenarios: S6(a) and S8(a) as second, and S6(b) and S6(c) as third. Class-based storage (CBS) combines feature of both random storage (RS) and dedicated storage (DS), and then it can be a more efficient alternative, especially when multi-criteria are considered (Muppani and Adil 2008a; 2008b; Fontana and Cavalcante 2013). Thus, our results prove that the developed model was able to return better alternatives than those traditionally used (RS and DS).

Moreover, although the sensitivity analysis showed some order reversals in the solutions ranking, in all these cases, the winning alternative would be another alternative generated by the developed model, i.e., the random or dedicate storage alternatives would not be chosen in any of the cases. This proves the robustness of the proposed model, which was able to find better solutions in all cases.

CONCLUSION

This work presented a new hybrid decision model to solve SLAP (MOGA+MCDM). Hybrid methodologies are still recent and have not been observed for SLAP. The simulation done proved that when considering only a heuristic or deterministic method to get the Pareto front, the manager should decide on this set of solutions without methodological support. The additive-veto model is able to help the decision maker in this analysis. Moreover, this method allows considering the possibility of vetoing alternatives that showed unacceptable performance in one of the criteria, since the additive model may compensate this low performance in one of the criteria as a result of high performance in other criteria.

In strategic decision-making environments, each problem situation can be unique. Thus, decision-making based on

multiple key performance indicators allows for more realistic results and a greater volume of information relevant to the process. This research aimed to address this research gap. Here only three indicators were considered, however, others are possible, depending on the preferences of the DM and reality of the warehouse. In addition, from the Pareto Front, the DM can consider other qualitative criteria, in the additive-veto model, that would be impossible on a large scale case, i.e., considering all the admissible solutions. This work did not aim to present the best evolutionary or multicriteria method for the problem, but to show that hybrid methodology is able to return interesting results for complex problems such as SLAP.

One limitation of this work was the consideration of known and predictable data, though it is known that in some situations warehouse operations deal with imprecise information. Thus, the study of the system's behaviour, considering the uncertainties of the demand, for example, and risks with the lack of spaces, and delays in the lead time of resupply, that compromise the availability of SKUs, are left for future studies.

ACKNOWLEDGMENTS

This work had partial support from Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

REFERENCES

- Accorsi, R., Manzini, R. and Maranesi, F. (2014), "A decision-support system for the design and management of warehousing systems", *Computers in Industry*, Vol. 65, pp. 175-86. <http://dx.doi.org/10.1016/j.compind.2013.08.007>.
- Almeida, A.T., Cavalcante, C.A.V., Alencar, M.H. et al. (2015), Multicriteria and multiobjective models for risk, reliability and maintenance decision analysis. International Series in Operations Research & Management Science, Vol. 281, Springer International Publishing, Switzerland. <https://doi.org/10.1007/978-3-319-17969-8>
- Ang, M. and Lim, Y.F. (2019), "How to optimize storage classes in a unit-load warehouse", *European Journal of Operational Research*, Vol. 278, No. 1, pp. 186-201. <http://dx.doi.org/10.1016/j.ejor.2019.03.046>.
- Ang, M., Lim, Y.F. and Sim, M. (2012), "Robust storage assignment in unit-load warehouses", *Management Science*, Vol. 58, No. 11, pp. 2114-30. <http://dx.doi.org/10.1287/mnsc.1120.1543>.

- Bahrami, B., Piri, H. and Aghezzaf, E.H. (2019), "Class-based storage location assignment: An overview of the literature" in *Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics ICINCO, 2019*, Prague, Czech Republic.
- Ballestín, F., Pérez, A., Lino, P. et al. (2013), "Static and dynamic policies with RFID for the scheduling of retrieval and storage warehouse operations", *Computers & Industrial Engineering*, Vol. 66, pp. 696-709. <http://dx.doi.org/10.1016/j.cie.2013.09.020>.
- Calzavara, M., Glock, C.H., Grosse, E.H. et al. (2019), "An integrated storage assignment method for manual order picking warehouses considering cost, workload and posture", *International Journal of Production Research*, Vol. 57, No. 8, pp. 2392-408. <http://dx.doi.org/10.1080/00207543.2018.1518609>.
- Chan, F.T.S. and Chan, H.K. (2011), "Improving the productivity of order picking of a manual-pick and multi-level rack distribution warehouse through the implementation of class-based storage", *Expert Systems with Applications*, Vol. 38, pp. 2686-700. <http://dx.doi.org/10.1016/j.eswa.2010.08.058>.
- Chen, L., Langevin, A. and Riopel, D. (2010), "The storage location assignment and interleaving problem in an automated storage/retrieval system with shared storage", *International Journal of Production Research*, Vol. 48, No. 4, pp. 991-1011. <http://dx.doi.org/10.1080/00207540802506218>.
- Chen, Z., Li, X. and Gupta, J.N.D. (2016), "Sequencing the storages and retrievals for flow-rack automated storage and retrieval systems with duration-of-stay storage policy", *International Journal of Production Research*, Vol. 54, No. 4, pp. 984-98. <http://dx.doi.org/10.1080/00207543.2015.1035816>.
- Da Serra Costa, J. (2011), "A genetic algorithm to obtain consistency in analytic hierarchy process", *Brazilian Journal of Operations & Production Management*, Vol. 8, No. 1, pp. 55-64. <http://dx.doi.org/10.4322/bjopm.2011.003>.
- De Almeida, A.T. (2013), "Additive-Veto models for choice and ranking multicriteria decision problems", *Asia-Pacific Journal of Operational Research*, Vol. 30, No. 6, pp. 1350026. <http://dx.doi.org/10.1142/S0217595913500267>.
- Deb, K., Pratap, A., Agarwal, S. et al. (2002), "Fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, pp. 182-97. <http://dx.doi.org/10.1109/4235.996017>.
- Ene, S. and Öztürk, N. (2012), "Storage location assignment and order picking optimization in the automotive industry", *International Journal of Advanced Manufacturing Technology*, Vol. 60, No. 5, pp. 787-97. <http://dx.doi.org/10.1007/s00170-011-3593-y>.
- Fontana, M.E. and Cavalcante, C.A.V. (2013), "ELECTRE TRI method used to storage location assignment into categories", *Pesquisa Operacional*, Vol. 33, No. 2, pp. 283-303. <http://dx.doi.org/10.1590/S0101-74382013000200009>.
- Fontana, M.E. and Cavalcante, C.A.V. (2014), "Use of PROMETHEE method to determine the best alternative for warehouse storage location assignment", *International Journal of Advanced Manufacturing Technology*, Vol. 70, pp. 1615-24. <http://dx.doi.org/10.1007/s00170-013-5405-z>.
- Fontana, M.E. and Nepomuceno, V.S. (2017), "Multi-criteria approach for products classification and their storage location assignment", *International Journal of Advanced Manufacturing Technology*, Vol. 80, pp. 3205-16. <http://dx.doi.org/10.1007/s00170-016-9040-3>.
- Guerriero, F., Musmanno, R., Pisacane, O. et al. (2013), "A mathematical model for the multi-levels product allocation problem in a warehouse with compatibility constraints", *Applied Mathematical Modelling*, Vol. 37, pp. 4385-98. <http://dx.doi.org/10.1016/j.apm.2012.09.015>.
- Hausman, W.H., Schwarz, L.B. and Graves, S.C. (1976), "Optimal storage assignment in automatic warehousing systems", *Management Science*, Vol. 22, No. 6, pp. 629-38. <http://dx.doi.org/10.1287/mnsc.22.6.629>.
- Horne, D.R., Nickerson, D. and DeFanti, M. (2015), "Improving supply chain efficiency through electronic payments: the case of micro-entrepreneurs in Kenya and Tanzania", *Journal of Marketing Channels*, Vol. 22, No. 2, pp. 83-92. <http://dx.doi.org/10.1080/1046669X.2015.1018074>.
- Muppani, V.R. and Adil, G.K. (2008a), "A branch and bound algorithm for class based storage location assignment", *European Journal of Operational Research*, Vol. 189, pp. 492-507. <http://dx.doi.org/10.1016/j.ejor.2007.05.050>.
- Muppani, V.R. and Adil, G.K. (2008b), "Efficient formation of storage classes for warehouse storage location assignment: A simulated annealing approach", *International Journal of Management Sciences*, Vol. 36, pp. 609-18. <http://dx.doi.org/10.1016/j.omega.2007.01.006>.
- Öztürkoğlu, O. (2018), "A bi-objective mathematical model for product allocation in block stacking warehouses", *International Transactions in Operational Research*, Vol. 6, pp. 1-27. <http://dx.doi.org/10.1111/itor.12506>.
- Pan, J.C.-H., Shih, P.-H., Wu, M.-H. et al. (2015), "A storage assignment heuristic method based on genetic algorithm for a pick-and-pass warehousing system", *Computers & Industrial*



Engineering, Vol. 81, pp. 1-13. <http://dx.doi.org/10.1016/j.cie.2014.12.010>.

Petersen, C.G. and Aase, G. (2004), "A comparison of picking, storage, and routing policies in manual order picking", *International Journal of Production Economics*, Vol. 92, pp. 11-9. <http://dx.doi.org/10.1016/j.ijpe.2003.09.006>.

Rabbani, M., Alipour, F., Farrokhi-Asl, H. et al. (2018), "Using metaheuristic algorithms for solving a mixed model assembly line balancing problem considering express parallel line and learning effect", *Brazilian Journal of Operations & Production Management*, Vol. 15, No. 2, pp. 254-69. <http://dx.doi.org/10.14488/BJOPM.2018.v15.n2.a8>.

Reis, A.C., Souza, C.G., Costa, N.N. et al. (2017), "Warehouse design: a systematic literature review", *Brazilian Journal of Operations & Production Management*, Vol. 14, pp. 542-55. <http://dx.doi.org/10.14488/BJOPM.2017.v14.n4.a10>.

Rouwenhorst, B., Reuter, B., Stockrahm, V. et al. (2000), "Warehouse design and control: framework and literature review", *European Journal of Operational Research*, Vol. 122, pp. 515-33. [http://dx.doi.org/10.1016/S0377-2217\(99\)00020-X](http://dx.doi.org/10.1016/S0377-2217(99)00020-X).

Shi, J., Liu, Z., Tang, L. et al. (2017), "Multi-objective optimization for a closed-loop network design problem using an improved genetic algorithm", *Applied Mathematical Modelling*, Vol. 45, pp. 14-30. <http://dx.doi.org/10.1016/j.apm.2016.11.004>.

Silva, D.D., Vasconcelos, N.V.C. and Cavalcante, C.A.V. (2015), "Multicriteria decision model to support the assignment of storage location of products in a warehouse",

Mathematical Problems in Engineering, Vol. 481950. <http://dx.doi.org/10.1155/2015/481950>.

Srivastav, A. and Agrawal, S. (2017), "Multi-objective optimization of a mixture inventory system using a MOPSO-TOPSIS hybrid approach", *Transactions of the Institute of Measurement and Control*, Vol. 39, No. 4, pp. 555-66. <http://dx.doi.org/10.1177/0142331215611211>.

Xie, J., Mei, Y., Ernst, A.T. et al. (2014), "Scaling up solutions to storage location assignment problems by genetic programming", in Dick, G. (Ed.), *Simulated evolution and learning*, Lecture Notes in Computer Science, Springer, Cham, pp. 691-702. https://doi.org/10.1007/978-3-319-13563-2_58

Wisittipanich, W. and Kasemset, C. (2015), "Metaheuristics for warehouse storage location assignment problems", *Logistics and Supply Chain Systems*, Vol. 14, No. 4, pp. 361-77. <http://dx.doi.org/10.12982/cmujns.2015.0093>.

Wutthisirisart, P., Noble, J.S. and Chang, C.A. (2015), "A two-phased heuristic for relation-based item location", *Computers & Industrial Engineering*, Vol. 82, pp. 94-102. <http://dx.doi.org/10.1016/j.cie.2015.01.020>.

Yang, J., Wen, Y. and Li, P. (2007), "The genetic-algorithm-enhanced blind system identification for water distribution pipeline leak detection", *Measurement Science & Technology*, Vol. 18, pp. 2178-84. <http://dx.doi.org/10.1088/0957-0233/18/7/049>.

Yener, F., Kir, S. and Cömert, S.E. (2019), "Class based storage with mixed integer linear programming approach for hazardous materials storage", *Acta Physica Polonica A*, Vol. 135, No. 4, pp. 678-80. <http://dx.doi.org/10.12693/APhysPolA.135.678>.

Received: 08 May 2019

Approved: 17 Dez 2019

How to cite: Fontana, M.E., Nepomuceno, V.S. and Garcez, T.V. et al. (2020), "HYBRID APPROACH DEVELOPMENT TO SOLVING THE STORAGE LOCATION ASSIGNMENT PROBLEM IN A PICKER-TO-PARTS SYSTEM", *Brazilian Journal of Operations & Production Management*, Vol. 17, No. 1, e2020853. <https://doi.org/10.14488/BJOPM.2020.005>