International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 13, Number 11 (2020), pp. 3456-3463
© International Research Publication House. https://dx.doi.org/10.37624/IJERT/13.11.2020.3456-3463

Smart Solution for Enhancing Storage Location Assignments in WMS Using Genetic Algorithm

Hanan Saleet

Industrial Engineering Department, Applied Science Private University, Jordan.

Abstract

Nowadays because of COVID-19 pandemic, the warehouse management system (WMS) solutions in the lean supply chain became more challenging. The WMS concerned with the logistics problem of stock keeping units (SKUs) being assigned to a proper storage location, i.e the Storage Location Assignment Problem (SLAP), is considered in this paper. Solving SLAP helps with the decision making affecting the efficiency of order-picking operations. This paper proposes a generic solution for SLAP; it is a smart logistics solution using data analytics and genetic algorithm where it introduces generic association-based assignment algorithm which helps placing SKUs nearby each other on the racking systems within the warehouse based on their frequency of being ordered together in the same picking orders. The percentage difference between the optimal solution and the solution achieved from the proposed approach is around 5.6%. This implies that the proposed smart logistic solution approach is successful in solving the SLAP and then enhances the WMS efficiency.

Keywords: Smart Logistics Solutions, WMS, Genetic Algorithm, Data Analytics.

I. INTRODUCTION

Optimally warehousing products is crucial to the speed and cost of delivery in the lean supply chain management [1] [2]. Warehouse management system (WMS) is an important component of the supply chain management. Usually warehouses keep hunderds or thousands of stock items called stock keep units (SKUs). Recently, data analytics and machine learning are utilized to design and analyze smart systems for WMS. It would enable the use of key resources, time, and energy to distribute and store all types of SKUs: fast, median and slow-moving SKUs [3]-[5].

Nowadays and because of COVID-19 pandemic, the logistic solutions in the lean supply chain became more challenging [6]. There are larger order fulfillment operational challenges when receiving, storing, order picking, and shipping products; showing an increased uncertainty in storage time, higher turnover velocities, and more types of product to fulfill. Due to labor intensity in such systems, the order picking process alone costs 50–75% of the total operating costs [7]. This shows urgent need for smart and intelligent solutions to enhance the effectiveness of storage policies, and guarantee more efficient order picking within a shorter timeframe [8] [9].

The importance of products being assigned to a proper storage location brought studies into different methods; which includes collecting data for decision making on optimal location assignment [1]. This research considers Storage Location Assignment Problem (SLAP), which helps with the decision making affecting the efficiency of the logistic system including receiving, storage, and order-picking operations. Usually, SKUs may be correlated by being ordered together; i.e within the same picking order (PO).

Warehouse management system is an important part of the supply chain, where it deals with the internal logistics in any facility [10]. Warehouse management is the act of planning and controlling all operations within a warehouse; and making sure it is all executed in an optimal way [11]. The main operations included in WMS are summarized in Figure 1.

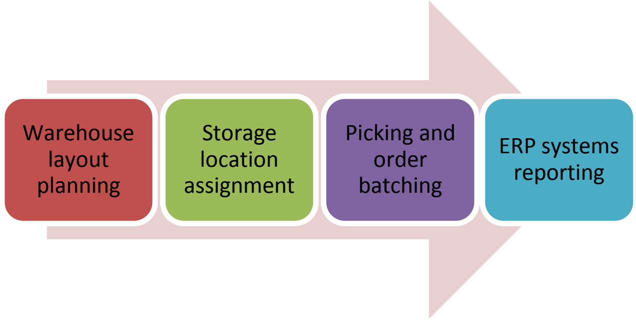


Figure 1: Warehouse management system

Data analytics and machine learning is proposed in the literature to be used in WMS. In the supply chain in general, and in the WMS, there is large data that is produced daily. This data is heterogeneous and it grows very fast. Then this data can be considered big data which needs data-analytics and machine learning tools to discover knowledge that is hidden in this data. [12]. There are previous studies that introduced the use of smart logistics solutions using data analytics, artificial intelligence and machine learning techniques to solve the SLAP under different conditions.

A case-study in ABC Company in Hong Kong implemented fuzzy logic to assists decision makers during the problem-solving process by retrieving data and testing alternatives to improve the SKU allocation time [13]. Li et al, proposed an optimization model that integrates ABC classification technique with SLAP. They used data mining techniques associated with greedy genetic algorithm to mitigate the complexity of the problem [14].

Kofler, et al studied the SLAP in a logistics center of an Austrian company in the automotive e-sector; where they employed simulated annealing to decrease picking effort [15]. Silva, et al. presented a General Variable Neighborhood Search metaheuristic, which is observed to be efficient for both small and large instances in WMS [7]. Syafrudin proposed a model that utilizes IoT-based sensors, big data processing, and a hybrid prediction using Random Forest classification. The model was evaluated and tested at an automotive manufacturing assembly line in Korea [16]. Alyahya et al. propose an RFID-enabled storage and retrieval mechanism that works without any human intervention. They developed a selection algorithm that aims to maximize efficiency in material-handling operations and minimize operational costs[17]

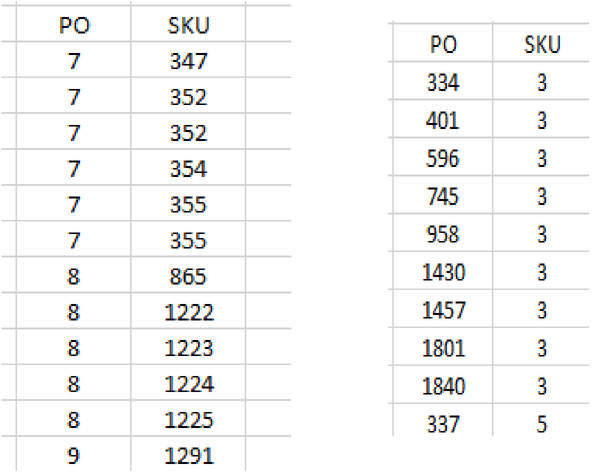
In this paper, SLAP problem is solved by introducing a smart logistics solution using a generic association-based assignment algorithm which helps placing SKUs nearby each other based on their frequency of being ordered together in the same picking orders. The proposed solution uses Genetic algorithm to solve the formulated SLAP problem. This will aid workers when picking multiple items per order, reducing distance moved hence time taken to retrieve the products.

II. SMART SYSTEM DESIGN

II.I System Description

In order picking environments a picker usually retrieves multiple SKUs per order. Items that are frequently ordered together are said to be correlated or *associated*. Storing *associated* SKUs close to each other may reduce the total travel time of the order pickers. One way to calculate the association index between two SKUs is finding the frequency at which those two SKUs appear in the same PO. The POs can be retrieved from the ERP system. Therefore, for each SKU the total number of picking orders in which that SKUs occurs is retrieved. Using this information an association matrix that stores how often two SKUs are ordered together is generated. This matrix is referred to as *inm* and is presented in equation (1).

Storage Location Assignment Problem (SLAP) solution requires the rearrangement of rows and columns of the matrix inm_{i2i3} to create blocks (i.e. racking systems within the WMS). several scores have been proposed to solve the SLAP e.g pick up delay, distance moved, etc. The proposed big data analytics approach presented here propose a two generic grouping scores that requires just the information presented in *inm* introduced in equation (1). The results of this research are used to get a baseline of how successful are the proposed grouping scores in solving the SLAP. Thus the innovation of this paper is the formulation of the generic objective function that is used in modeling the SLAP. In addition to the use of Genetic algorithm to solve this model.



PO	SKU
7	347
7	352
7	352
7	354
7	355
7	355
8	865
8	1222
8	1223
8	1224
8	1225
9	1291

PO	SKU
334	3
401	3
596	3
745	3
958	3
1430	3
1457	3
1801	3
1840	3
337	5

Figure 2: Each Picking order lists a number of SKUS that are order together.

The adopted process involves the following main decision steps for the design and configuration of the warehouse management system.

1. Retrieve data from POs: The analysis of the POs will result in listing the POs and what SKUs are requested in those POs; see Figure (2).
2. Association indices evaluation: the association index belongs to a pair of SKUs. It represents the degree to which they need to be assigned to locations that are close to each other.
3. Clustering analysis: in this step the SLAP is formulated and the GA algorithm is used to solve the model. It results in finally grouping the SKUs.
4. Location assignment: decide on the number of distinct locations, or racking systems, where the SKUs should be located within the warehouse; then assign the SKUs accordingly.
5. Plant layout configuration: this step deals with the determination of the location of each group (racking system) within the warehouse area.

The five decision steps mentioned above belong to a hierarchical process that involves several steps. The result of each step depends on the quality of the input data. For example, how effective the clustering task depends on the quality and significance of the association index adopted, whose values must be evaluated correctly, and which provides the input data for the algorithm. This algorithm also influences the output of location assignment problem.

II.II Data Analytics

This section introduces the data analytics mechanism used to deal with the research problem. As mentioned earlier, this study refers to a location assignment process based on the introduction of association indices and the application of heuristic technique. The adopted data analytic mechanism involves the following four main stages; shown in Figure (3).

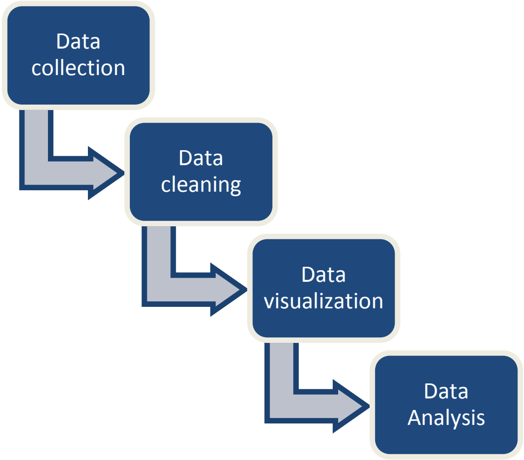


Figure 3: General steps for data analytics

II.II.I Data collection

The picking order (PO) contains the SKUs that must be retrieved from the racking systems at a point of time. As depicted in Figure (2), the data set contains a unique ID for each PO (column 1); each PO includes a set of SKUs (column 2). In this stage the following should be applied:

- if a PO includes just one SKU, exclude this record from the data; where it is considered an outlier. Such records can be summarized and sent to the manager, so he/she can decide where to locate that SKU.
- SKUs that have very high or very low frequency are also excluded. SKUs with very high frequency (high demand) are called fast moving SKUs. These are excluded because they have POs in common with so many of the SKUs; which cause significant imbalance in the final location assignment of SKUs. Since they are fast moving, they need to be placed close to the iterance of the warehouse, or placed near the end of racking systems. SKUs that has very low frequency (low demand) are called slow moving. Usually they need large storage space. They are stored far from the entrance of the warehouse and then they may need more time for retrieval. Nevertheless, since they are slow moving SKUs, their added picking time will not affect the efficiency of the warehouse management system.
- rearrange the dataset so that the ID for SKUs are ranked in an ascending order.

II.II.II Data cleaning

- if a PO includes an SKU that is (are) repeated more than once, only just one record will be kept. In this paper, the association between SKUs is considered regardless of their quantity. Further study could be conducted to include the number of batches (of the same SKU) which are requested in each PO.
- map the ordered list of SKU IDs to the numbers sequences from 1 to n; n represents the total number of SKU IDs that are stored in the warehouse.
- The initial matrix (inm_{i2i3}) will be modified to eliminate the association index between SKU and itself. This is logical since there will be no decision built on that information.

II.II.III Data visualization

The data listed in Figure (2) is transform into a square matrix that contains SKUs in the rows, and in the columns it contains the same order of those SKUs.

- Count the number of times each SKU (SKU_{i2}) appeared in POs at the same time as another SKU (SKU_{i3}).
- Fill this count in an n x n matrix which is the association matrix (inm_{i2i3});
- This matrix is visualized in Figure (5). The number of SKUs in this example is 35. The cold colors represent low association index. The initial matrix usually has a mixed color since the SKUs are not yet grouped (or formed into blocks).

II.II.IV Data analysis

A detailed mathematical formulation of the generic objective function is presented next. The objective function is a weighted average of two generic grouping scores. The storage location assignment problem has been shown to be NP-Hard [7]. Thus the objective function is formulated to solve this combinatorial optimization problem. This formulation is presented in section III.I. A heuristic based on Genetic Algorithm (GA) is developed to find the optimal or near optimal solution. The proposed GA is presented in section 3.2

III. MODEL FORMULATION

III.I Objective Function

As mentioned earlier, the generic objective function is a weighted average of two grouping scores M_1 and M_2. The two scores are formulated based on the following analysis.

Step1:

The entries to the matrix inm_{i2i3} are a_{i2i3} which represented the frequency SKU_{i2} and SKU_{i3} are requested together

where $i2=1,\ldots,n$ and $i3=1,\ldots,n$; n: number of SKUs

$inm=[a_{i2i3}]$

$$inm = \begin{bmatrix} a_{11}\ a_{12}\ \ldots\ a_{1n} \\ a_{21}\ a_{22}\ \ldots\ a_{1n} \\ \vdots \quad \vdots \quad \ddots \quad \vdots \\ a_{n1}\ a_{n2}\ a_{n3}\ \ldots\ a_{n4} \end{bmatrix} \quad (1)$$

Step2:

The solution sol_{mk} will take the form of a matrix with number of rows equals the population size (specified by GA) whereas the number of columns equals the number of SKUs.

$sol = [b_{mk}]$

$$sol_{mk} = \begin{bmatrix} b_{11}\, b_{12} \;\dots\; b_{1n} \\ b_{21}\, b_{22} \;\dots\; b_{1n} \\ \vdots \quad \vdots \quad \ddots \quad \vdots \\ b_{p1}\, b_{p2}\, b_{p3} \dots\, b_{pn} \end{bmatrix} \quad (2)$$

$m=1,\dots, p$; p is the population size of the Genetic Algorithm, as will be explained later.

$k=1,\dots, n$; which represents the location of the SKU within *Sol* matrix. b_{mk} can take the values from 1 to the number of racking systems (number of blocks): c.

Thus, b_{mk} represents the location of the k^{th} SKU within the solution (chromosome in GA) m in the population within the current iteration (generation in GA).

Step3:

The objective function formulation: it consists of two generic grouping scores: M_1 and M_2

Finding Grouping score 1 (M_1):

$$M_1 = \frac{0.5 * sum_1}{e} + \frac{0.5 * sum_1}{sumdLargestSumd} \qquad (3)$$

M_1 looks for the largest association indexes within *inm*. Its optimization assures that the association indexes inside the blocks are the largest ones. This assure that the SKUs that are ordered together are located in the same racking system.

sum_1: represents the sum of the association indexes found inside the blocks

e: represents the sum of all associating indexes in the matrix *inm*

$$\mathrm{e} = \sum_{i2=1}^{n} \sum_{i3=1}^{n} a_{i2i3} \quad (4)$$

To find the sum_1, sol_{mk} is used to find the SKUs that need to be assigned together.

Taking the first solution in the population:

$Sol_{1k}=\ [b_{11}\, b_{12}\, b_{13} \dots b_{1k}]$

mu matrix can be found by searching Sol_1 for the SKUs that are assigned to the same racking system c.

$$mu_{i1k} = \begin{cases} 1 & if\ b_{mk} = i1 \\ 0 & otherwise \end{cases} \qquad (5)$$

$i1 = 1, 2, \dots, c$

$$\mathrm{sum1} = \sum_{i1=1}^{c} \sum_{i2=}^{n} \sum_{i3=1}^{n} a_{i2i3} * mu_{i1i2} * mu_{i1i3} \qquad (6)$$

To find *sumdLargestSumd:*

-First, find the number of association indexes located in each block, then sum them all to find the total number of indexes inside all the blocks; *sumd*.

-second, Find the sum of first *sumd* maximum elements (a_{i2i3}) in the matrix *inm* ; this sum is *sumdLargestSumd* . If *sumdLargestSumd* is close to *sum1* , it means that the blocks are well structured.

To find *sumd* , find k_{i1} which is a vector representing how many SKU assigned to each racking system

$$k_{i1} = \sum_{k=1}^{n} mu_{i1k} \quad (7)$$

$$sumd = \sum_{i1=1}^{c} k_{i1} * k_{i1} \quad (8)$$

Then *sumdLargestSumd* can be found by the following iterative heuristic:

sumdLargestSumd = 0

Max_value = max(*inm*(:))

sumdLargestSumd = *sumdLargestSumd* + *Max_ value*

max(*inm*(:)) = 0

Repeat

Finding Grouping score 2 (M_2):

M_2 measures the blocks' density compared to the neighboring indexes. In Figure(4), the blocks are colored in blue. The neighboring indexes are located outside each block. For example, the first block's neighboring indexes are colored in grey, the second block's neighboring indexes are colored in red, and so on.

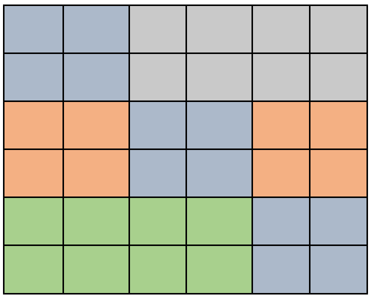


Figure 4: Blocks are colored in blue

-First find S*olM* which is the solution matrix. Based on the first row elements in sol_{mk} matrix which represents the first alternative solution (candidate solution; or chromosome in GA), the *inm* can be transformed into a solution matrix (*SolM*) which is a block diagram matrix. Each block in *SolM* represents one racking system in WMS. All SKUs within that block can be assigned to one racking system.

$$M_{i1k} = \begin{cases} b_{mk} & if\ b_{mk} = i1 \\ 0 & otherwise \end{cases} \ ; i1 = 1, \dots, c; k = 1, \dots, n \ (9)$$

To find *SolM,* find the SKUs assigned to each block *skus*

$$skus_{i2} = M_{i1k}\ if\ M_{i1k} > 0 \quad (10)$$

$$SolM_{i2i3} = inm_{i2i3}(skus_{i2}, skus_{i3})\ ; i2, i3 = 1, \dots, n \quad (11)$$

Find the sum of association indexes inside each block (i1)

$$QM_{1,1} = \sum_{z=1}^{k_1} \sum_{y=1}^{k_1} SolM_{zy} \quad (12)$$

$$QM_{i1,1} = \sum_{z=1+\sum_{l=1}^{k_{i1-1}} k_l}^{k_{i1}} \ \sum_{y=1+k_{i1-1}}^{k_{i1}} SolM_{zy} \quad (13)$$

$$QM_{i1,2} = \sum_{z=1}^{k_{i1}} \sum_{y=1}^{n} SolM_{zy} \quad (14)$$

$$M_2 = \frac{\sum_{i1=1}^{k_{i1}} QM_{i1,1}}{\sum_{i1=1}^{k_{i1}} QM_{i1,2}} \quad (15)$$

Step 4:

Calculating the objective function to find the fitness value F for each row in Sol : it is a weighted average between M1 and M2:

$$F = 0.5 \text{ x } M_1 + 0.5 \text{ x } M_2 \quad (16)$$

Step 5:

Find a measure for the grouping efficiency. Matrix Quality Measure MQ can be used here:

MQ measures the ratio between how much similarity can be found inside the blocks to how much similarity can be found in the whole matrix *inm:*

$$MQ = 1 - \frac{e - sum1}{e} \quad (17)$$

III.II Genetic algorithm

The formulation and implementation of GA to SLAP formation required using a MATLAB code, which includes several functions. The MATLAB code employed the objective function and its initialization, mutation and crossover techniques. The MATHLAB code performs the GA procedures utilizing a dataset that contains POs. It generated an initial population of solutions Sol_{mk} (chromosomes) randomly. These solutions are then reproduced in order to find the optimal or near optimal solution [18].

III.II.I Initialization

Initialization involves generation of possible solutions to the problem. In our implementation, the initial population is generated randomly. Each individual or chromosome is made up of a sequence of genes from a certain alphabet. An integer alphabet {1, …, c} is employed, where c represents an upper bound on the number of blocks. For example if $n = 5$ and $c = 2$ then:

Sol_{lk}= [b_{11} b_{12} b_{13} … b_{1n}] = [1 1 2 2 2]

Each gene position represents an SKU (k; $k=1,\dots,n$) and each gene value (b_{mk}) equals the block number assigned to SKU. In the above example, $b_{12} = 1$, indicates that SKU with ID 2 is assigned to block 1. Table 1 represents the initial population of solutions (chromosomes). Each chromosome has $n = 5$ genes, and the population size $p = 10$.

Table 1: Initialization of the objective function.

Chromosome no.	Initial population (randomly selected)
Chromosome1	2 2 1 1 2
Chromosome2	2 2 2 2 1
. . .	. . .
Chromosome10	1 2 1 2 1

III.II.II Evaluation function

Evaluation functions (objective function) is used to evaluate the solutions (chromosomes) within the population. According to the fitness value scored, chromosomes are selected as parents to produce offsprings in the next generation or to disappear in the next generation.

The generic objective function formulated in this research aims to divide the association indexes matrix into blocks; which means that the SKUs assigned to that block or racking system are more frequently ordered together. In a solution matrix, a block diagonal form is often desirable because the blocks may be easily identified to facilitate the subsequent location assignment as shown in Figure (6).

III.II.III Reproduction (Parent selection technique)

Reproduction is a process in which individuals are copied according to their objective function values, i.e., fitness values. Individuals with a higher value have a higher probability of contributing one or more offspring in the next generation. The roulette wheel method of reproduction is to be used in this research.

III.II.IV Crossover and mutation

The purpose of crossover is to create children whose genetic material resembles their parent's genes in some fashion. Crossover combines building blocks of good solutions from different chromosomes based on the crossover rate.

Mutation is applied to an offspring individually after crossover according to the mutation rate. It provides a small amount of random search and helps ensure that no point in the search space has a zero probability of being examined.

In this paper, the GA uses simple float crossover; which selects a cut point. The two parents are then split at this point, and concatenating the segments from both parents creates offsprings. For mutation, the GA uses a combination of uniform mutation operator which randomly selects one of the genes from a parent and sets it equal to a random number uniformly distributed between the variable's lower and upper bounds. And, the boundary mutation operator which randomly selects one of the variables from a parent and sets it equal to its lower or upper bound.

III.II.V Replacement

After mutation, the new population is evaluated using the objective function. Using the elitism strategy of replacement, the best solution in the current population is added to the new population replacing the worst solution.

III.II.VI Termination

The GA iterates from generation to generation until a termination criterion is met. In this research, the GA uses a hybrid technique between two parameters: setting the maximum number of generations, and a threshold for the standard deviation among the fitness values. This would cancel any redundant computations, hence minimize the CPU time.

IV. RESULTS AND DISCUSSION

This paper mainly focuses on finding optimal solutions for instances of the SLAP combinatorial optimization problem. The lack of commonly shared benchmark problems makes it challenging to compare between research results. Despite this, this research effort is still dominated by the search for best possible solutions by relying on randomly generated datasets.

Seven datasets are generated as shown in Table (2). The initial population contains random solutions. When one of the solutions is used to construct the matrix *inm*, it will lack any blocks as shown in Figure (5). After running GA, the final solution will be optimal or near optimal which contains blocks as shown in Figure (6).

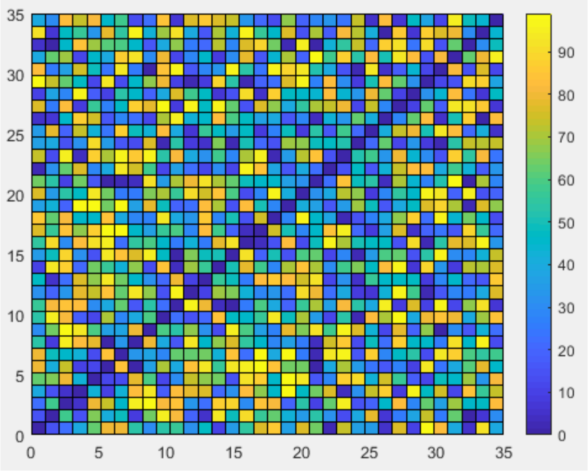


Figure 5: Hot/cold association between SKUs-random assignments.

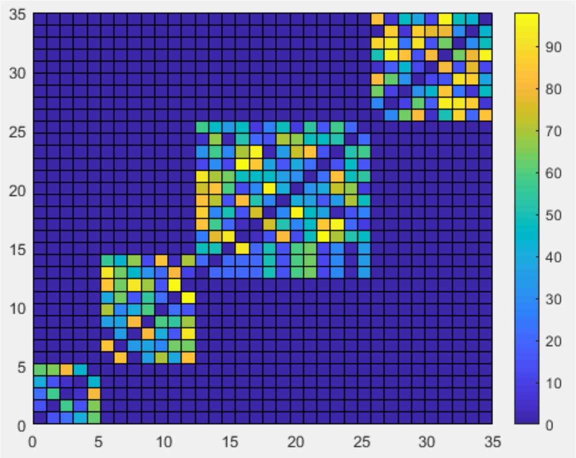


Figure 6: Hot/cold association between SKUs- bock diagram assignment

Because GA is stochastic search algorithm, number of generations required to solve a problem depends on the composition of the initial population. Figure (7) shows the great improvement on the solution of the GA, from the first generation till the generation, which gives the best solution. Note the great improvement in the best solution which changed from 105 to 280 in nearly 70 generations i.e., 166.7% improvement. In this figure the fitness value is scaled to make it clear.

Specifying a maximum number of blocks for the GA simply places an upper bound c. This is realistic since in real situations the warehouse will have a certain capacity limitation which constraint the number of racking systems (blocks) c_{max}. Let c_{max} be the maximum number of blocks permitted and $c*$ be the optimal number of blocks. If $c_{max} \geq c*$, then in the GA will find $c*$. However, if $c_{max} < c*$, the constraint is binding, and the genetic algorithm will produce a solution with, at most, c_{max} cells at a reduced objective function, because of the warehouse capacity constraint. In this research the data sets are generated randomly. Thus the optimal number of cells $c*$ is known. In the experimental work c_{max} was set to be $c_{max} \geq c*$.

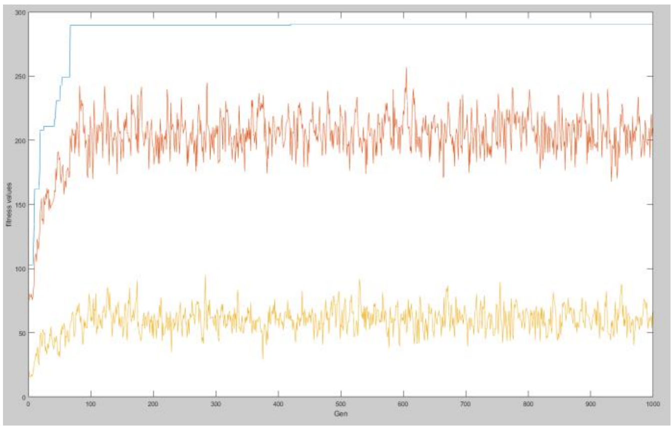


Figure 7: The fitness value as a function of number of generations; it is scaled for clarification.

Table 2: Datasets

	n	Optimal solution				GA solution				%dif
		M1	*M2*	*F*	*MQ*	*M1*	*M2*	*F*	*MQ*	
1	35	0.9586	0.9571	0.95785	0.9784	0.888	0.8868	0.8874	0.9059	7.4%
2	50	0.8772	0.8784	0.8778	0.9359	0.8145	0.8162	0.81535	0.9301	7.1%
3	100	0.8523	0.8432	0.84775	0.9145	0.8088	0.8001	0.80445	0.9234	5.1%
4	200	0.8178	0.8077	0.81275	0.9066	0.7945	0.7601	0.7773	0.8897	4.4%
5	400	0.7988	0.799	0.7989	0.8778	0.7433	0.7397	0.7415	0.8582	7.2%
6	800	0.7357	0.7257	0.7307	0.8529	0.706	0.7053	0.70565	0.8184	3.4%
7	1600	0.695	0.6415	0.66825	0.8461	0.6686	0.6098	0.6392	0.8139	4.3%

Experimentation focused on comparing the GA solution to the optimal solution. Several experiments were conducted on the seven datasets shown in Table (2). The following parameters were recorded in the table for both the optimal solution and the best solution found using GA: the grouping score 1 and 2, the fitness value and the quality measure. It is important to notice that the average percent difference between the optimal value and the best solution is 5.6% which validates the proposed smart logistics solution for the different number of SKUs.

V. CONCLUSIONS

This research presents a smart logistics solution that relays on generic scores for grouping to solve the SLAP. It helps decision makers to assign SKUs to nearby locations inside racking system within the warehouse. Their location assignment is based on their frequency of being ordered together in the same picking orders. The validation of the proposed smart logistics solution was presented, with the aim to optimize the storage location assignment. Finding the optimal or near optimal solution is achieved using genetic algorithm. The percentage difference between the optimal solution and the solution achieved from GA is around 5.6%. This implies that the proposed smart solution is successful in solving the SLAP and then enhances the WMS efficiency.

Future research should aim to include different SLAP scores related to pick up delay, distance moved, storage capacity, etc. also physical constraint on pallet size and shape.

ACKNOWLEDGEMENT

The author is grateful to the Applied Science Private University, Amman, Jordan for the full financial support granted to this research.

REFERENCES

[1] M. Kovac & D. Djurdjevic, “Optimization of Order-Picking Systems through Tactical and Operational Decision Making”, International Journal of Simulation Modelling 19 (2020) 1, 89-99

[2] N. Andiyappillai, Dr.T.Prakash, “Implementing Warehouse Management Systems (WMS) in Logistics: A Case Study”, International Journal of Logistics Systems and Management, February 2019.

[3] T. Nguyena, L. ZHOU, V. Spieglerb, P. Ieromonachoua, Y. Lina, “Big data analytics in supply chain management: A state-of-the-art literature review”, Computers & Operations Research, July 2017. DOI: 10.1016/j.cor.2017.07.004

[4] S. S. Darvazeh, I. Vanani and F. M. Musolu, “Big Data Analytics and Its Applications in Supply Chain Management”, Chapter in DOI: http://dx.doi.org/10.5772/intechopen.89426

[5] S. Alswedani, M. Saleh, "Big Data Analytics: Importance, Challenges, Categories, Techniques, and Tools", International Journal of Advanced Trends in Computer Science and Engineering, 9(4), July – August 2020, 5384 – 5392.

[6] Eric Carter. Warehouse crisis management – How a WMS manages COVID-19 demand peaks. Industry Talk, https://itsupplychain.com/warehouse-crisis-management-how-a-wms-manages-covid-19-demand-peaks/, 25 March, 2020

[7] A., Silva, L. C. Coelhoa, M. Darvisha, J. Renauda, “Integrating storage location and order picking problems in warehouse planning”, Transportation Research Part E 140 (2020) 102003

[8] A. Andjelkovic, M. Radosavljevic, “Improving order-picking process through implementation of warehouse management system”, Strategic Management 23(2):3-10. January 2018

[9] The warehouse of tomorrow, today: Ubiquitous Computing, Internet of Things, Machine Learning. https://www2.deloitte.com/ accessed Oct,2020.

[10] G. Baruffaldi, R. Accorsi and R. Manzini, “Warehouse management system customization and information availability in 3pl companies A decision-support tool”, Industrial Management & Data Systems, Vol. 119 No. 2, 2019 pp. 251-273, 2019.

[11] F.H. Staudt, G. Alpan,M. Di. Mascolo, C. M. T. Rodriguez, “Warehouse performance measurement: a

literature review", International Journal of Production Research, Vol. 53 No. 18, pp. 5524-5544, 2015.

[12] D. Kojek, R. Varbic, B. Rihtrasic, P. Butala. Big data analytics for operations management in engineer-to-order manufacturing. Procedia CIRP volume 72, 2018, pages 209-214.

[13] Y. Y., Hui, K. Choy, G. Ho, K. Leung, & H. Lam, "A cloud-based location assignment system for packaged food allocation in e-fulfillment warehouse", Hong Kong, International Journal of Engineering Business Management, Volume 8: 1–15, 2016.

[14] J. Li, M. Moghaddam, and S.Y. Nof, "Dynamic storage assignment with product affinity and ABC classification—a case study", The International Journal of Advanced Manufacturing Technology, Vol. 84 No. 9–12, pp. 2179–2194, 2016.

[15] M. Kofler, A. Beham, S. Wagner, and M. Affenzeller. Affinity Based Slotting in Warehouses with Dynamic Order Patterns. Chapter 7 of the book Advanced Methods and Applications in Computational Intelligence, Topics in Intelligent Engineering and Informatics Volume 6, 2014, pp 123-143. DOI: 10.1007/978-3-319-01436-4_7.

[16] M. Syafrudin, G. Alfian, N. L. Fitriyani,and J. Rhee, "Performance Analysis of IoT-Based Sensor, "Big Data Processing, and Machine Learning Model for Real-Time Monitoring System in Automotive Manufacturing". Sensors, VOL 18, 2946, 2018.

[17] S. Alyahya, Q. Wang, and N. Bennett, "Application and integration of an RFID enabled warehousing management system – a feasibility study", Journal of Industrial Information Integration, Elsevier Inc., Vol. 4, pp. 15-25, 2016

[18] D. Goldberg, Genetic Algorithms In Search, Optimization And Machine Learning, 1st Edition, Addison-Wesley, Reading, MA, USA, 1989.