BİTLİS EREN ÜNİVERSİTESİ
FEN BİLİMLERİ DERGİSİ

ISSN: 2147-3129 / e-ISSN: 2147-3188

Article Type : Research Article		**Year**	: 2025
Received	: June 5, 2025	**Volume**	: 14
Revised	: July 14, 2025	**Issue**	: 3
Accepted	: July 30, 2025	**Pages**	: 1772-1786
DOI	: 10.17798/bitlisfen.1714876		

CORRELATED SKU ASSIGNMENT IN WAREHOUSES USING THE JOINT DEMAND PROBABILITY DISTRIBUTION: A METAHEURISTIC ALGORITHM APPROACH

Bayram DÜNDAR [1]

[1] *Bartın University, Department of Computer Engineering, Bartın, Türkiye, bdundar@bartin.edu.tr*

ABSTRACT

In warehouse management, picking orders from storage locations quickly and in the shortest time has become even more important with the development of e-commerce. Thus, efficiently assigning affined products to storage locations within the warehouses is crucial in reducing operational costs and preserving product quality. In this study, a Mixed-Integer Linear Programming model (MILP) is developed to minimize in-warehouse picking distances. Based on demand data, inter-product relationships are analyzed, and correlation coefficients are estimated for product pairs with a high tendency to be ordered together. These correlation values are then integrated into the objective function to optimize storage location decisions. To obtain faster and near-optimal solutions from the MILP model on large-scale data sets, a genetic algorithm (GA)-based approach has been developed. A set of computational experiments conducted on medium and large-scale instances compares the performance of the proposed GA approach with the Random-Based Correlated Skus Assignment Model (RBC-SAM). The GA approach under different scenarios shows an improvement of up to 22%.

Keywords: SKUs assignment problem, Demand correlation, Genetic algorithm, Mathematical modeling.

1 INTRODUCTION

With the rapid development of e-commerce and international supply chains today, warehouse management has become one of the most critical stages of the supply chain. Stocking products within a specific plan provides significant convenience in placing and picking products after the order. A Stock Keeping Unit, or SKU, is the smallest physical unit of a product to facilitate tracking in warehouse management [2]. Optimal assignment of SKUs allows warehouse management to offer significant cost advantages to businesses in terms of both time and distance during the process of picking orders from storage locations. As stated in the study by [2], the order-picking process consists of several key sub-activities, including traveling, searching, and extracting. Among these, traveling accounts for approximately 55\% of the total order-picking time, making it the most time-consuming activity. Therefore, warehouse management systems should primarily focus on reducing traveling time to expedite the fulfillment of any given order. Moreover, since traveling is a labor-intensive activity, it constitutes one of the most significant cost components in warehouse operations. To the end, optimizing the traveling activity is crucial for enhancing warehouse efficiency and reducing operational costs.

Effectively managing space and time is among the most critical elements in warehouse operations. An analysis of customer orders often reveals meaningful relationships between certain SKUs. In other words, strong correlations can be observed between specific SKU pairs. This strategy aims to optimize both time and space efficiency by leveraging the inherent relationships between frequently co-ordered items. In the literature, this problem is generally known as the SKU assignment problem. There are various studies addressing the SKUs assignment problem. For example, [6] examined the assignment of SKUs to areas in area-based carton collection distribution centers (DCs) where they decide which products to collect. In their study, a simulated annealing-based heuristic approach was used due to the large-scale nature of the problem. There are also some studies addressing the storage assignment problem within the framework of the Quadratic Assignment Problem (QAP). In this study, the problem of allocating related SKUs to storage locations is addressed. Some SKUs are frequently ordered together; therefore, the relationship between such SKUs should be taken into account when assigning them to available storage locations. Placing related SKUs close together could reduce the total picking distance and, therefore, the total cost of warehouse operations. This highlights the importance of considering proximity relationships between SKUs. In this study, unlike the approaches in the literature, the correlation value between two SKUs is calculated using the 'joint distribution' methodology. Furthermore, since the correlated SKU assignment problem exhibits a combinatorial structure, a genetic algorithm (GA) based metaheuristic approach is proposed.

The remaining sections of this study are planned as follows: In the second section, the literature on modeling and solution approaches for the correlated SKU assignment problem is reviewed. In the third section, the modeling details of the SKU assignment problem are presented, and the details of the Genetic Algorithm (GA) based solution approach used are given. The developed GA approach is subjected to computational tests under various scenarios in the fourth section. The last section presents a general evaluation and a brief summary of the study.

2 LITERATURE REVIEW

This section reviews studies in the literature that focus on the assignment of correlated SKUs to storage locations. Particular attention has been paid to how these problems are modeled and which solution methods are employed.

Bottani et al. [3] solved a class-based assignment problem in warehouses using a genetic algorithm. However, their work did not take into account the correlation between products. Xiao and Zheng [13] utilized the product bill of materials to estimate demand correlation among the SKUs in the storage assignment problem. The authors proposed heuristic algorithms to solve the mathematical model. Wisittipanich and Kasemset [12] presented a mixed-integer linear programming (MILP) model for the storage location assignment problem and proposed two metaheuristic algorithms to solve large-scale instances. Li et al. [9] used data mining techniques for the dynamic storage allocation problem to calculate the "affinity" values of the products, and based on these values, they used a greedy genetic algorithm to assign the products to achieve the maximum possible total relatedness value. Zhang et al. [14] took into account the pattern of correlation of demand between products to assign items to storage areas in warehouses. The developed model was solved using simulated annealing and a heuristic algorithm. Ansari et al. [1] proposed a gravity model including a clustering method in order to optimize the SKUs assignment to storage locations. Kim et al. [7] took into account the heuristic algorithms when assigning the affined SKUs to storage locations. Squires et al. [11] has developed a genetic algorithm-based solution for the scheduling problem of medical treatment processes.

Lee et al. [8] proposed a bi-objective optimization model for the correlated assignment problem, taking into account traffic congestion in warehouses. The proposed model aims to improve the efficiency of the picking process. It was validated through a case study using data collected from an actual warehouse. The simulation results demonstrated significant improvements in travel time and delays. Mirzaei et al. [10] proposed a mixed integer model for the distribution center of personal care products by considering the correlation between the products. The aim of this model is to minimize the travel time of robots by assigning products to optimal storage locations. Gabellini et al. [4] proposed a genetic algorithm and a machine learning-based model to address a batch assignment problem in warehouses, with the aim of predicting the pick-up time of orders. Dündar [15] developed a robust counterpart of the correlated SKUs assignment model and solved it with a commercial solver. In that study, a limited number of SKUs could be assigned within an efficient computational time.

Islam et al. [5] have conducted a comprehensive literature review on the correlated assignment problem. The vast majority of studies in the literature aim to minimize travel time and distance within the warehouse. To this end, optimal or near-optimal solutions to large-scale problems have been proposed, usually using various heuristic and metaheuristic algorithms. However, no study has been found that specifically addresses the correlation between two SKUs using the marginal distribution method. The summary of reviewed studies is provided in Table 1.

Table 1 The summary of reviewed studies

Authors	Developed model	Solution method
Bottani et all.(2012)	Class based assignment problem	Genetic Algorithm
Xiao and Zheng (2012)	Bill of material based correlation assignment model	Heuristic Algorithm
Wisittipanich and Kasemset (2015)	Mixed integer programming model	Metaheuristic algorithms
Li et al. (2016)	Data mining technique based dynamic storage allocation problem	Greedy genetic algorithm
Zhang et al. (2019)	The storage location assignment problem	Simulated annealing and a heuristic algorithm
Ansari et al. (2020)	A gravity model based clustering method	A simulation model approach
Kim et al. (2020)	Storage location assignment model	Heuristic Algorithm
Lee et al. (2020)	A bi-objective optimization model	Multi-objective evolutionary algorithms
Mirzaei et al. (2022)	Correlated dispersed storage assignment model	Heuristic Algorithm
Gabellini et al. (2024)	A batch assignment problem	Genetic Algorithm and a machine learning-based approach
Dündar (2025)	Correlated SKUs assignment model with uncertainty	Gurobi solver

3 PROBLEM FORMULATIONS

A mixed-integer linear programming (MILP) model is proposed for assigning correlated SKUs to storage racks. The proposed model aims to minimize the total picking time in warehouses. The objective function depends on the correlation between each pair of SKUs, the distance between the racks to which they are assigned, and the associated decision variables. Let i and j represent the SKUs from the set I, and let t and r represent the stock locations from the set $\aleph$. δ_{tr} represents the distance between stock location t and stock location r. y_{ijtr} is a binary decision variable; if SKU i and SKU j are assigned to stock locations t and r, respectively, y_{ijtr} takes the value of 1. Otherwise, it takes the value of 0. Additionally, Q_i is a random variable representing the demand quantity for product i within a specific period. The details of sets, indices, parameters, and decision variables are provided in Table 2.

Table 2: The description of notations in the mathematical model

Sets and Indices	**Description**
$\aleph$	Set of SKUs
I	Set of stock locations
i, j	Represents SKUs in a set of $\aleph$
t, r	Denotes the stock locations in the set of I
Parameters	
δ_{tr}	Distance between the stock locations *t* and *r*
Q_i	Random variable for the number of SKUs ordered
Decision variables	
y_{ijtr}	If SKU i and SKU j are assigned to stock locations *t* and *r*, respectively, y_{ijtr} takes the value of 1, otherwise 0
z_{it}	A binary variable used in linearization that enforces the binary nature of y_{ijtr}. If SKU i is assigned to stock location *t,* the value of 1 is taken; otherwise, 0

The following assumptions were used in the correlated SKUs assignment problem. First, only one SKU is assigned to a single storage location. Second, there are as many available storage locations as SKUs to be assigned. The number of storage locations to which the SKUs will be assigned is fixed. Finally, the storage locations to which the SKUs will be assigned are large enough to accommodate large quantities of SKUs.

$$\min \sum_{\{(i,j)\in I|i<j\}} \sum_{t,r\in\aleph} \frac{\mathbb{E}[Q_iQ_j] - \mathbb{E}[Q_i]\mathbb{E}[Q_j]}{\sqrt{(\mathbb{E}[Q_i^2]\mathbb{E}[Q_i]^2)(\mathbb{E}[Q_j^2]\mathbb{E}[Q_j]^2)}} \delta_{tr} y_{ijtr} \tag{1}$$

The division operation in the objective function calculates the correlation value between SKUs. The expected value expressions used here are calculated with the help of marginal distributions obtained from the Q random variables. Assume that the joint probability distribution of two SKUs, i and j, is estimated based on the demand pattern. In this case, the marginal distributions for each SKU, denoted as $P(Q_i)$ and $P(Q_j)$, are obtained as follows:

$$P(Q_i = q_i) = \sum_{q_j} P_{(Q_i,Q_j)}(Q_i = q_i, Q_j = q_j) \tag{2}$$

In the objective function, $\mathbb{E}[Q_i]$ represents the expected value of SKU i (Q_i), which is calculated as $\mathbb{E}[Q_i] = \sum_q q\, P(Q = q)$. The term $\mathbb{E}[Q_iQ_j]$ denotes the expected value of the product of SKUs i and j, as given in the following equation $\mathbb{E}[Q_iQ_j] = \sum_{q_iq_j} P_{(Q_i,Q_j)}(Q_i = q_i, Q_j = q_j)$. The terms in the objective function denominator represent the product of the standard deviations of SKU i and SKU j. The objective function is subject to the following constraints. In the constraint (3), $SKU\ i$ can only be assigned to one storage location.

$$\sum_{k\in\aleph} z_{it} = 1, \qquad \forall t \in I \tag{3}$$

The constraint (4) stipulates that each storage location t can be assigned to only one $SKU\ i$.

$$\sum_{i\in I} z_{it} = 1, \qquad \forall t \in \aleph \tag{4}$$

The following constraints ensure that the variable y_{ijtr} takes binary values, i.e., either 0 or 1, within the linearized objective function. Similarly, z_{ij} is defined as a binary variable, $z_{it} \in \{0, 1\}$, that enforces the binary nature of y_{ijtr}.

$$y_{ijtr} \le z_{it}, \qquad \forall i,j \in I|i < j, \qquad \forall t,r \in \aleph \tag{5}$$

$$y_{ijtr} \le z_{jr}, \qquad \forall i,j \in I|i < j, \qquad \forall t,r \in \aleph \tag{6}$$

$$z_{it} + z_{jr} - 1 \le y_{ijtr}, \qquad \forall i,j \in I|i < j, \qquad \forall t,r \in \aleph \tag{7}$$

3.1 Genetic Algorithm Approach

The SKU assignment problem is inherently combinatorial and falls into the class of NP-hard problems. In solving large-scale versions of such problems, as widely emphasized in the literature, it is possible to reach optimal or near-optimal solutions within reasonable times through genetic algorithm (GA) based approaches. In this study, the solution of the correlation-based SKU assignment model is obtained using the genetic algorithm framework. The basic steps of the proposed genetic algorithm approach are outlined in Algorithm 1. In the following subsections, detailed explanations of the functions and procedures of the genetic algorithm are provided.

3.1.1 Chromosome Structure

The chromosome structure consists of genes equal in number to the SKUs and storage locations. Each gene in the chromosome represents an assigned SKU. In this context, a permutation-based encoding method is employed. The number of potential chromosomes corresponds to the total number of permutations of the SKUs.

3.1.2 Evaluation Function

This function is used to evaluate the extent to which each chromosome in the population, that is, the potential solution, is suitable according to the objective function. The evaluation function aims to minimize the product of the correlation between SKUs and the distance between the locations to which these SKUs are assigned.

Algorithm 1 Genetic algorithm

```
[H] Input: C, |I|, φ, ξ, number of generation(n),S
function EvaluationFunc(C)
    return obj
end function
function Crossover(f_candidate,s_candidate,|I|,S)
    Crossover procedure
    return best candidate solution
end function
function Mutation(chromozom)
    Mutation procedure
    return mutated chromozom
end function
function Selection(C)
    return parent 1, parent 2
end function
function CandidateSolution(C)
    Mutation
    Selection
    Immigration
    return New candidate solution
end function
function GeneticAlgorithm(C, |I|, φ, ξ, n, S)
    for i in range(n):
        if New candidate solution ≤ incumbent solution then
         incumbent solution = New candidate solution
end function
return Assigned SKUs
```

3.1.3 Selection Procedure

This function contains the procedures that define how any two solutions are obtained in order to generate new candidate solutions from the population. In this study, within the scope of the selection procedure, two candidate solutions are randomly selected from the existing population.

3.1.4 Crossover Procedure

The crossover procedure defines the rules by which a new solution is produced from the two selected parent chromosomes. In this method, the genes on both chromosomes are compared based on the index. If the genes in the same index are different, the value of the gene on the first chromosome is determined by the index it is located on the second chromosome. Then, the relevant gene on the first chromosome is replaced with the corresponding value on the second chromosome. As a result of these operations, the first candidate solution is obtained. Similarly, similar changes are made on the second chromosome, and an alternative candidate solution is produced.

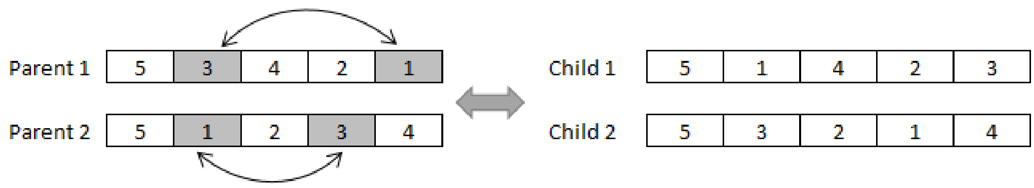

Figure 1. Illustration of the crossover procedure for generating new candidate solution

3.1.5 Mutation

It is the process of exchanging two randomly selected genes on a chromosome with each other in order to increase genetic diversity. This mutation process is performed on the new candidate solution obtained as a result of the crossover process. Whether or not the mutation will be applied is decided according to whether a randomly generated number in the range [0,1] is smaller than the predefined mutation rate ξ.

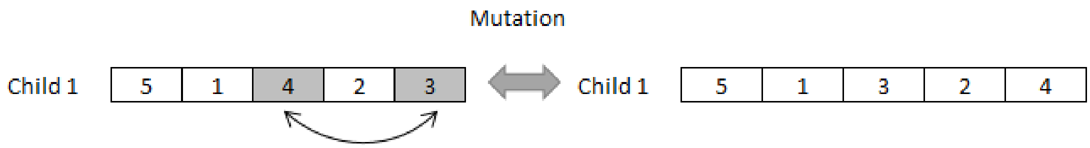

Figure 2. Mutation procedure for generating more diverse candidate solutions

3.1.6 Generation of A New Candidate Solution

A new candidate solution is generated on the basis of the crossover and mutation procedures previously explained. The steps related to this process are presented in detail in Algorithm 2. The predefined chromosome set C, the number of SKUs (|I|) to be assigned, the crossover rate (φ), the mutation rate (ξ) and the set S, which includes the correlation values between SKUs, are used as input parameters. If the two candidate solutions determined as a result of the selection procedure are identical, two different individuals from the chromosome set C are selected again. As stated in the third line of the algorithm, if a randomly generated number in the range [0,1] is smaller than the crossover rate (φ), the crossover procedure is applied. Similarly, as stated in the eighth line, if a random number generated in the range [0,1] is smaller than the mutation rate (ξ), the mutation procedure is activated. This mechanism increases genetic diversity, resulting in a more diverse candidate solution.

Algorithm 2 A function of generating a candidate solution

Input: $\mathcal{C}$, $|\mathcal{I}|$, φ, ξ
Output: A new candidate solution

```
Select f_candidate and s_canditate from C using Selection procedure while
  f_candidate.Sequence = s_candidate.Sequence do
    Select f_candidate and s_candidate again from C
if random.random() ≤ φ then
    Candidate solution ← crossover procedure
else
    Candidate solution ← copy of f_candidate ()
if random.random() ≤ ξ then
    Candidate solution ← mutation procedure
return Candidate solution
```

4 COMPUTATIONAL RESULTS

In order to test the GA model, an artificial distance data set was created with a distance of 4 meters between storage locations. The correlation data expressing the relationship between SKUs is also created artificially. As aforementioned in the objective function of the MILP model, the correlation is calculated based on the context of the marginal distribution of demand for SKUs. Identifying the most suitable combination of these parameters ensures that the algorithm converges on the optimal solution quickly and accurately. In this context, the GA model developed is evaluated through a series of experimental tests on a data set consisting of 15 SKUs and 15 storage locations, with the aim of determining the optimal parameter settings. As shown in Figure 3, four different crossover rate values, 0.50, 0.75, 0.90, and 0.99, are tested. As a result of the analyses, the crossover rate of 0.99 led to the fastest convergence to the optimal solution. In this study, we assume that the number of SKUs to be assigned and the number of stock locations are equal to each other. If the number of products to be assigned is less than the number of available storage locations, a routine assignment is performed. Conversely, if the number of products to be assigned is greater than the number of available storage locations, a routine assignment is performed by grouping products according to the number of storage locations, taking into account the correlation value between products.

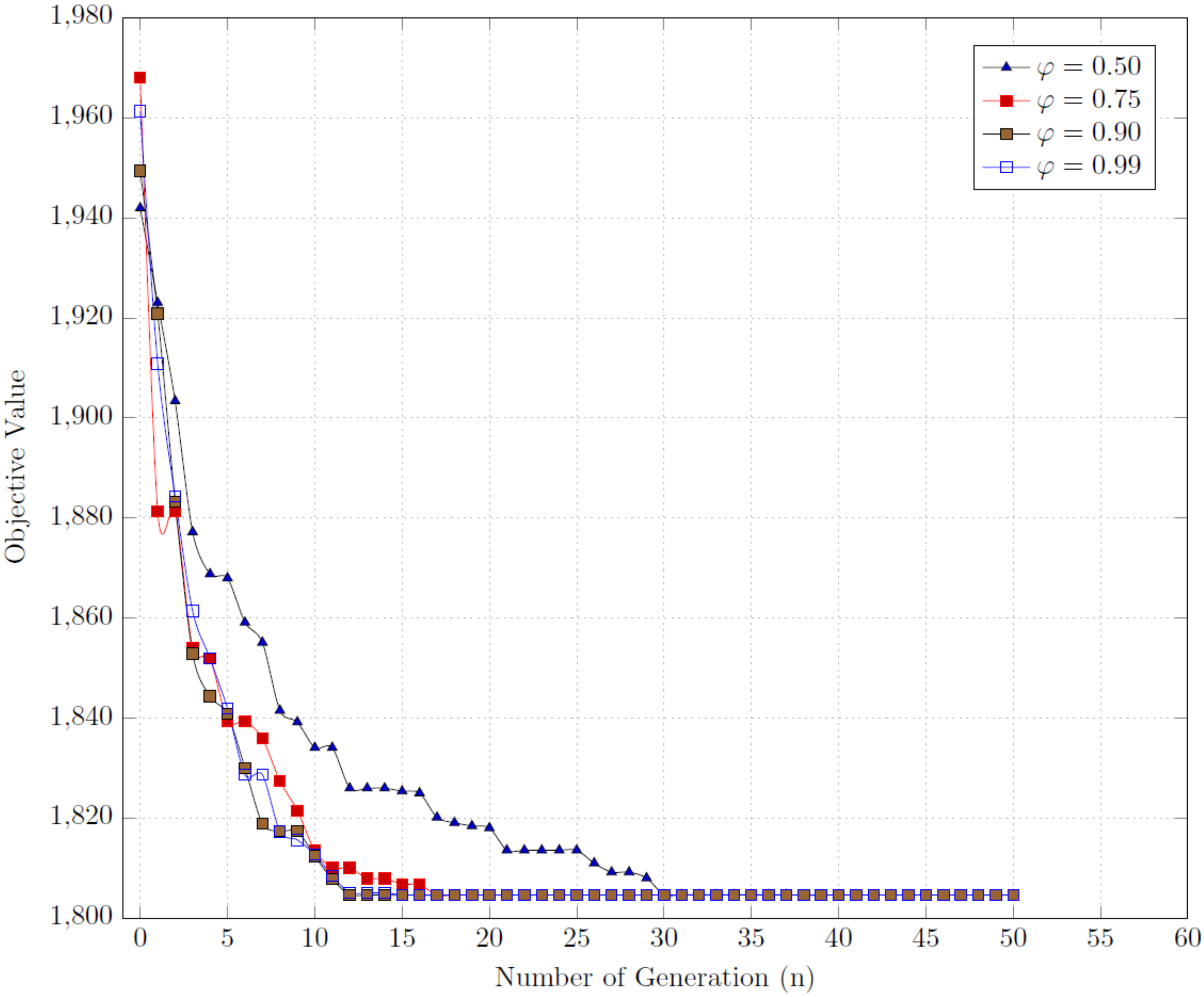

Figure 3. The convergence of objective value for various crossover rates

One of the key parameters that enhances the diversity of candidate optimum solutions in the GA model is the mutation rate. In this context, in order to evaluate the effect of the mutation rate, the GA algorithm was tested for four different values, 0.25, 0.50, 0.75, and 0.90, as shown in Figure 4. The experimental test remarked that the mutation rate of 0.5 led to the fastest convergence to the optimal or near-optimal solution.

The GA model was tested for different immigration probabilities, 0.04, 0.05, 0.50, and 0.90, as shown in Figure 5. According to the experimental results, the immigration probability of 0.05 is identified as the most appropriate optimum rate that provides the optimal solution.

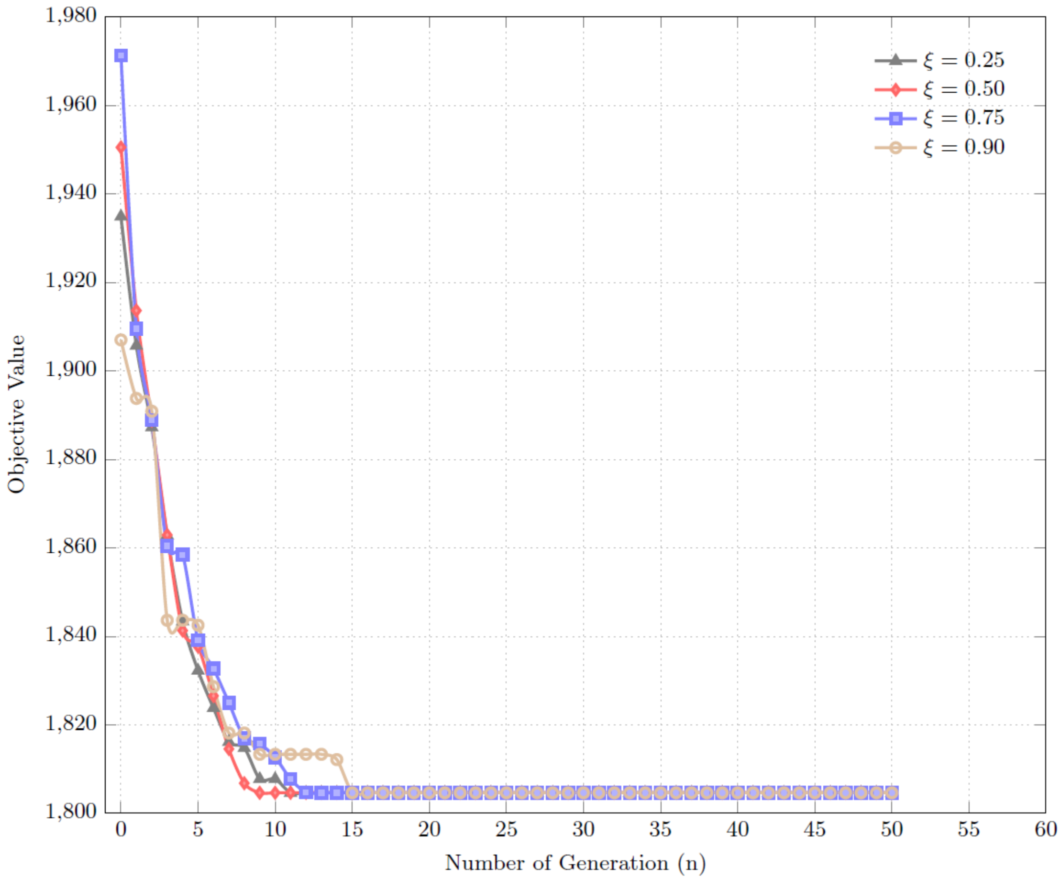

Figure 4. The convergence of objective value for various mutation rates

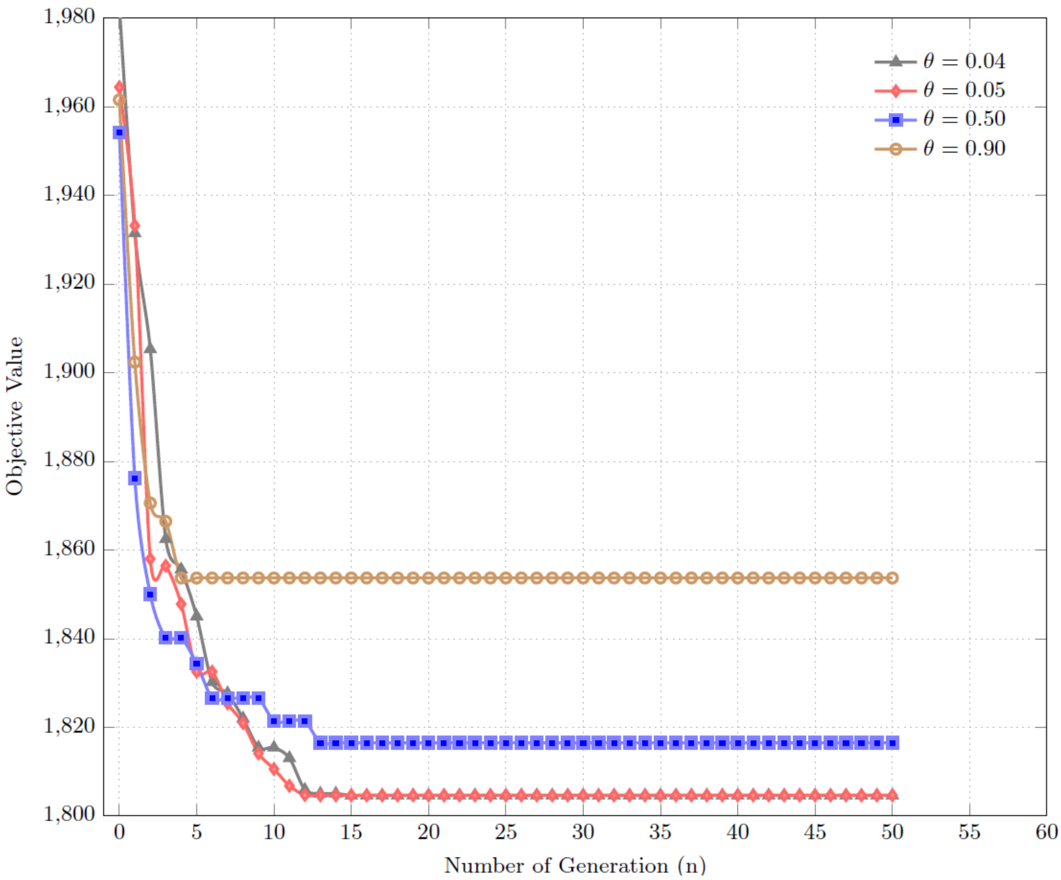

Figure 5. The convergence of the objective value for various immigration rates

As seen in Table 3, the GA model was tested for a range of scenarios that start with 15 SKUs and storage locations and increase to 100 SKUs, according to the GA parameters specified previously. Each scenario was run 50 times, and the best, average, and worst objective function values obtained in each run were recorded. For example, in the assignment scenario with 15 SKUs, the objective function value was the same in all runs and was measured as 1825.7. The average CPU time was only 1.9 seconds, which shows that the GA model can produce a high-quality and consistent solution in a short time. When the number of SKUs was increased to 20, the best objective function value was obtained as 4554.4, and the deviation between the best and worst results was only 0.48 %. Notice that in the case of assigning 50 SKUs, as can be estimated from Table 3, the 1.27 % deviation occurred compared to the best solution. The average CPU time per run increased to 67.68 seconds. When the GA model was run to assign 100 SKUs, there was a difference of approximately 1.05% between the best and worst solutions obtained. With an average CPU time of 768.92 seconds, the GA model provided a valid solution for 100 correlated SKUs.

Table 3. Computational performance of the GA

# of runs	# of SKUs	Objective function value			Avg. CPU Time (sec)
		Best	Average	Worst	
50	15	1825.7	1825.7	1825.7	1.9
	20	4554.4	4558.7	4576.3	4.6
	30	14542.3	14542.8	14544.1	15.3
	40	35948.2	35968.0	36012.7	36.3
	50	72277.0	72452.7	73195.0	67.7
	70	207742.4	208842.0	211304.9	295.2
	90	449931.5	452993.6	457804.2	599.8
	100	622699.0	626176.3	629223.9	768.9

Within the framework of the parameters mentioned above, the GA model was tested for different numbers of SKUs and storage location scenarios in terms of both computation time and optimal solution quality. In this framework, a comparison was made between randomly assigned SKUs and GA based SKU assignment.

As seen in Table 4, the GA-based assignment model and the RBC-SAM were compared. In other words, the effect of an assignment made without considering the correlation relationship between SKUs on the objective function in terms of minimizing the total picking distance was examined. According to the results in Table 4, in the scenario where 15 SKUs were assigned, the GA-based model produced a 22.65% better solution in terms of the objective function value than the RBC-SAM. When the number of SKUs increased to 40, the GA model performed 20.24% better than the RBC-SAM model. In the scenario where 100 SKUs were assigned, the GA model produced an 8.83% better result compared to the RBC-SAM model in terms of the objective function value. Hence, reducing the total picking distance in warehouse management plays an important role in terms of requiring less labor in operational activities within the warehouse, reducing internal traffic density, maintaining product quality and generally reducing total operational costs.

Table 4. Comparison of objective values between the GA Model and RBC-SAM

# of SKUs	GA Model	GA Std. deviation	RBC-SAM	% of Difference
15	1,8250.7	0	2,239.14	22.65
20	4,554.4	7.42	5,392.47	18.40
30	14,542.3	7.50	17,051.83	17.26
40	35,948.2	16.24	43,223.76	20.24
50	72,277.0	344.91	81,824.62	13.21
70	207,742.4	827.75	230,417.04	10.91
90	449,931.5	1948.49	493,591.99	9.70
100	622,699.0	2165.45	677,688.26	8.83

5 CONCLUSION AND SUGGESTIONS

This study considers the problem of assigning correlated SKUs to the storage location. The correlation between SKUs is estimated by using the joint probability distribution function. In preliminary computational tests conducted with commercial solvers, especially in cases where the problem reaches large instances, namely, more than 13 SKUs, it became intractable to provide a solution. The Genetic Algorithm (GA) based approach is proposed to solve medium and large instances of SKUs. The GA model was tested on up to 100 SKUs with an artificial data set under various scenarios. The proposed algorithm was able to provide optimal solutions for small-scale problems in a shorter time. However, as the instances of the SKUs increased, it was not possible to test whether the solution found by the model was optimal or not, but it still produced high-quality solutions in a very short time. The GA-based assignment method provided 8.8% to 22.6% more efficient solutions in terms of total distance compared to the Random-Based Correlated SKUs Assignment Model (RBC-SAM). For future work, one could apply the robust counterpart of the correlated assignment model to larger-scale problems. An algorithm could be proposed that would enable this problem to be solved efficiently.

Statement of Research and Publication Ethics

The study is complied with research and publication ethics.

Artificial Intelligence (AI) Contribution Statement

The author used AI-based tools, ChatGP and Grammarly to assist with language editing of the manuscript. All revisions were reviewed and approved by the author.

REFERENCES

[1] M. Ansari and J. S. Smith, "Gravity clustering: A correlated storage location assignment problem approach," in *Proc. 2020 Winter Simulation Conf. (WSC)*, 2020, pp. 1288–1299.

[2] J. J. Bartholdi and S. T. Hackman, *Warehouse & Distribution Science: Release 0.96*. Atlanta, GA: The Supply Chain and Logistics Institute, 2014. [Online]. Available: https://www.warehouse-science.com

[3] E. Bottani, M. Cecconi, G. Vignali, and R. Montanari, "Optimisation of storage allocation in order picking operations through a genetic algorithm," *Int. J. Logist. Res. Appl.*, vol. 15, no. 2, pp. 127–146, 2012.

[4] M. Gabellini, F. Calabrese, A. Regattieri, D. Loske, and M. Klumpp, "A hybrid approach integrating genetic algorithm and machine learning to solve the order picking batch assignment problem considering learning and fatigue of pickers," *Comput. Ind. Eng.*, vol. 191, p. 110175, 2024.

[5] S. Islam and K. Uddin, "Correlated storage assignment approach in warehouses: A systematic literature review," *J. Ind. Eng. Manag.*, vol. 16, no. 2, pp. 294–318, 2023.

[6] B. S. Kim and J. S. Smith, "Slotting methodology using correlated improvement for a zone-based carton picking distribution system," *Comput. Ind. Eng.*, vol. 62, no. 1, pp. 286–295, 2012.

[7] J. Kim, F. Méndez, and J. Jimenez, "Storage location assignment heuristics based on slot selection and frequent itemset grouping for large distribution centers," *IEEE Access*, vol. 8, pp. 189025–189035, 2020.

[8] G. Lee, S. H. Chung, and S. W. Yoon, "Two-stage storage assignment to minimize travel time and congestion for warehouse order picking operations," *Comput. Ind. Eng.*, vol. 139, p. 106129, 2020.

[9] J. Li, M. Moghaddam, and S. Y. Nof, "Dynamic storage assignment with product affinity and ABC classification—A case study," *Int. J. Adv. Manuf. Technol.*, vol. 84, pp. 2179–2194, 2016.

[10] M. Mirzaei, N. Zaerpour, and R. B. de Koster, "How to benefit from order data: Correlated dispersed storage assignment in robotic warehouses," *Int. J. Prod. Res.*, vol. 60, no. 2, pp. 549–568, 2022.

[11] M. Squires, X. Tao, S. Elangovan, R. Gururajan, X. Zhou, and U. R. Acharya, "A novel genetic algorithm based system for the scheduling of medical treatments," *Expert Syst. Appl.*, vol. 195, p. 116464, 2022.

[12] W. Wisittipanich and C. Kasemset, "Metaheuristics for warehouse storage location assignment problems," *Chiang Mai Univ. J. Nat. Sci.*, vol. 14, no. 4, pp. 361–377, 2015.

[13] J. Xiao and L. Zheng, "Correlated storage assignment to minimize zone visits for BOM picking," *Int. J. Adv. Manuf. Technol.*, vol. 61, pp. 797–807, 2012.

[14] R.-Q. Zhang, M. Wang, and X. Pan, "New model of the storage location assignment problem considering demand correlation pattern," *Comput. Ind. Eng.*, vol. 129, pp. 210–219, 2019.

[15] B. Dündar, "A robust optimization approach to address correlation uncertainty in stock-keeping unit assignment in warehouses," *Alphanumeric J.*, vol. 13, no. 1, pp. 1–12, 2025.