Optimisation of Ceramic Kiln Loading Problem Using Multi-Objective Genetic Algorithm

Derya Deliktaş[a] and Ayşe Kaygısız[b]
Kütahya Dumlupınar University, Faculty of Engineering, Department of Industrial Engineering, Kütahya, Turkey

Keywords: Ceramic Kiln Loading Problem, Ceramic Industry, Genetic Algorithm, Multi-Objective Optimisation, Manufacturing Efficiency.

Abstract: Efficient resource utilisation is paramount for boosting productivity and competitiveness within industrial contexts. In ceramic manufacturing, the Ceramic Kiln Loading Problem is critical, wherein the optimal arrangement of ceramic products within kilns significantly influences production efficiency. This study aims to enhance efficiency by maximising the utilisation of the oven vehicle through optimal loading of the ordered products. To achieve this objective, the Genetic Algorithm has been integrated with weighted sum and conic scalarisation methods, and the results obtained from each method have been compared. Additionally, since the algorithm's parameters can significantly influence its performance, parameter tuning has been conducted using the **irace** method. The findings corroborate the superiority of results obtained by integrating the Genetic Algorithm with weighted sum scalarisation.

1 INTRODUCTION

Cutting and packing represents two classic combinatorial optimisation problems. Cutting problems involve optimising the use of materials such as wood, steel, and cloth, while packing problems focus on maximising the use of available packing space. Efficient utilisation of material and transport capacities is crucial in production and distribution processes, as it contributes to the economic use of natural resources. Cutting and packing problems share a common structure: a set of large objects and small items are provided, defined exhaustively in one, two, three, or more geometric dimensions. The goal is to select some or all of the small items, group them into one or more subsets, and assign each subset to a large object. This assignment must satisfy geometric conditions, ensuring that all small items in a subset lie completely within the large object without overlap, while optimising a given objective function (Wäscher et al., 2007; Sheng et al., 2016).

The loading problem entails the efficient placement of a maximum quantity of identical rectangular boxes onto a single rectangular pallet. This challenge is also occasionally referred to as the manufacturer's pallet loading problem (Silva et al., 2016). The container loading problem (CLP), also known as the packing problem, constitutes sub-problems of cutting and packing problems (Hodgson, 1982). There are two primary container loading problems (Dyckhoff, 1990). The first problem involves loading either the entire consignment or a portion thereof into a single container. The aim is to optimise volume utilisation by maximising the filled container volume or minimising any unused space within the container. The second problem, known as the multiple container loading problem, entails loading the entire consignment into one or more identical containers. The objective here is to minimise the total number of containers required for the consignment (Lim et al., 2012). When the primary aim is space utilisation, this problem essentially reflects the pallet packing problem, which involves loading boxes onto pallets rather than into containers. The primary distinction lies in the fact that pallets do not offer lateral support for the boxes. Given that the well-known knapsack problem is a component of the loading problem, the loading problem belongs to the class of NP-hard problems (Terno et al., 2000).

This study focuses on the real-life problem of correctly loading ceramic tableware into a ceramic kiln at specified temperatures within the ceramic industry. The ceramic kiln loading problem is a variant of the

[a] https://orcid.org/0000-0003-2676-1628
[b] https://orcid.org/0009-0002-9002-1229

Deliktaş, D. and Kaygısız, A.
Optimisation of Ceramic Kiln Loading Problem Using Multi-Objective Genetic Algorithm.
DOI: 10.5220/0012807100003753

In *Proceedings of the 19th International Conference on Software Technologies (ICSOFT 2024)*, pages 248-257
ISBN: 978-989-758-706-1; ISSN: 2184-2833

pallet loading problem. Due to the significant impact of loading on product quality, particularly in productions catering to the ceramic sector where the number and variety of items are extensive, the company takes the loading problem seriously. The optimal utilisation of shelves of various heights and dimensions placed on the pallet to be inserted into the kiln is crucial due to the extended baking time of the products in the kiln. Additionally, the complexity of the problem is exacerbated by the heterogeneous sizes of the products to be placed on the shelves, which prevent each product from being placed on every shelf due to these sizes. Each ceramic tableware item has priorities regarding the places they can be assigned to on the shelves. In this study, these assignment priorities have been considered to reduce deformations caused by products touching each other and being placed on the wrong shelf. Furthermore, maximising the total profits of the products placed in the kiln is also aimed. For this purpose, it is aimed to fully load the kiln tool (or pallet) by calculating the maximum score of the products to be assigned to the shelves on the kiln tool, considering their priorities, and maximising the total profit calculated by considering the unit profits of the products assigned to the kiln. A multi-objective genetic algorithm is proposed to solve this problem. The genetic algorithm (GA) is combined with the weighted-sum (WSM) and the conic scalarisation method (CSM). Since the parameters of the proposed algorithm will affect the results, the **irace** method is used for parameter tuning. The results obtained from GA with WSM are compared with the results obtained from GA with CSM. A decision support system (DSS) has also been designed.

The remainder of this paper is structured as follows: Section 2 offers an overview of the literature. In Section 3, the problem is described. Section 4 details the proposed approaches. Section 5 presents the findings, and finally, Section 6 provides a summary of the paper.

2 LITERATURE SURVEY

The CLP involves determining the optimal arrangement and placement of cargo items within rectangular containers to maximise specific objectives. Constraints primarily revolve around container capacity in terms of volume, weight or dimensions, weight distribution, cargo orientation, loading order, and stability. The problem may pertain to a single container or multiple containers, which can be homogeneous or heterogeneous (Vélez-Gallego et al., 2020). Objectives include:

- maximising capacity utilisation (Dereli and Sena Das, 2010; Ramos et al., 2018),
- minimising cost (Tian et al., 2016),
- minimising container count (Toffolo et al., 2017),
- maximising profit (Liu et al., 2016),
- optimising space utilisation or replenishment time (Yüceer and Özakça, 2010).

Given its NP-hard complexity (Young-Gun and Kang, 2001; Zhou and Liu, 2017), exact methods face limitations in effectively addressing the CLP (Chen et al., 1995; Wu et al., 2010; Junqueira et al., 2012; Paquay et al., 2016). Consequently, there has been a proliferation of alternative methodologies such as heuristics (Toffolo et al., 2017), metaheuristics (Zhang et al., 2009; Zheng et al., 2015; Özdemir et al., 2022), and hybrid approaches (Dereli and Sena Das, 2010; Romão et al., 2012) in solving the CLP.

In this study, we consider all practical constraints mandated by ceramic industry and aim to furnish solutions within a computational time frame of average one minute. Experimental results show that the proposed multi-objective algorithm, which includes maximising both total priority score and total profit objectives, can efficiently solve the practical ceramic kiln loading problem.

3 PROBLEM DEFINITION

In a study conducted in a ceramic industry, an optimisation study was carried out by placing the products on shelves on pallets in the ceramic baking oven in the glazing unit, in order to respond to the demands in the most appropriate way. In this study, we can define the assignment of i ceramic products to the racks on the pallet to be loaded into the kiln as $P = p_1, ..., p_i$ and the j racks to be assigned as $R = r_1, ..., r_j$. The racks to which each product can be assigned and their assignment priorities are given in Table 3. According to this dataset, there are a total of twelve racks (or kiln trays) of three different types on the pallet to be loaded into a ceramic kiln, namely, 2 square racks (Figure 1(a)), 4 six-layer rectangular shelves (Figure 1(b)), and 6 five-layer rectangular racks (Figure 1(c)). In addition to these racks, the products can also be assigned on the floor (Figure 1(d)). The part shown with a red square in Figure 1(a) represents the top-side of the square rack. Some products can only be assigned to this part.

A visual of the pallet to be loaded into the kiln is provided in Figure 2. Pallet and racks on it used for the ceramic kiln loading problem was generated by using the ZW3D package (ZWCAD Software Co., Ltd., Guangzhou, China). Each product must be assigned to the specified racks in Table 3. The expression 'I' in the table implies assigning first priority to the corresponding rack; 'II' implies assigning second priority to the corresponding rack, and 'III' implies assigning third priority to the corresponding rack. While assigning products to specified racks is a hard constraint, assigning according to priority of racks is a soft constraint.

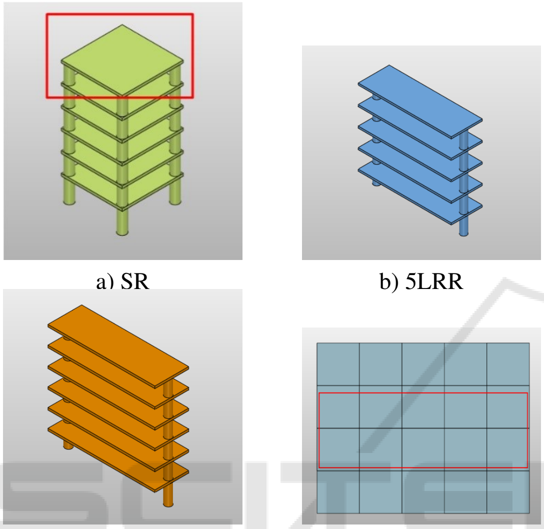

Figure 1: Racks on pallet loaded into kiln.

Primarily, it is preferred to assign the product to the first priority given in Table 3. However, if the rack belonging to the first priority is full, assignments are made according to other priorities in the table. In this study, 1000 points are assigned to the first priority, 300 points to the second priority, and 100 points to the third priority. This scoring was determined through brainstorming with company employees. In addition to priorities in Table 3, the values in parentheses indicate the maximum number of products that can be assigned from the relevant product to the appropriate rack.

For example, P_1 can be assigned to only two types of racks, namely six-layer rectangular (6LRR) and five-layer rectangular (5LRR) racks. It cannot be assigned to other rack types. A maximum of two P_1 can be assigned to these racks. Here, P_1 is expected to be assigned to 6LRR as the first priority and to 5LRR as the second priority. Since the dimensions of each product and rack are different, assignments were made considering the floor area of each product and the rack space. In addition, the unit profits of each product are also given in this table. In this study, it is assumed that 10 units of each product are ordered. Additionally, in this study, it was planned to assign the 169 products most ordered for the last six months to the kiln vehicle. These products are baked in the kiln at the same temperature. Finally, this study considers only *one* kiln pallet.

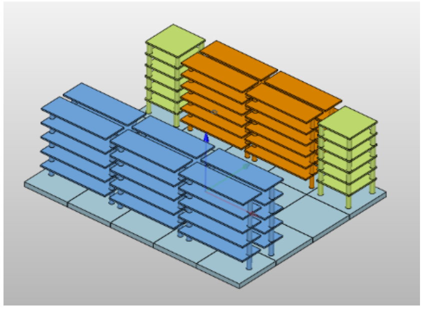

Figure 2: A visual of the pallet to be loaded into the kiln.

4 MULTI-OBJECTIVE GENETIC ALGORITHM

GA is an evolutionary algorithm developed by Holland that seeks solutions to problems through a stochastic approach (Holland, 1973). GA can concurrently explore solutions or sets of solutions across the solution space. As GA is an intuitive approach, it may not find exact solutions to the problem; however, it can yield solutions near the optimum. Therefore, it can provide acceptable solutions for loading and optimisation problems falling under the NP-hard class in a reasonable time frame (Panchal and Panchal, 2015).

The objective functions of the examined problem are: maximising the total score of the products assigned to the racks on the pallet ($f_1(x)$) and maximising the total of the unit profits of these products ($f_2(x)$). Solving multi-objective problems necessitates the utilisation of multi-objective optimisation techniques. Among these methods, scalarisation methods are prominent. Scalarisation involves converting a multi-objective problem into a single-objective problem. In this study, we proposed multi-objective genetic algorithm by integrating weighted-sum and conic scalarisation methods to solve the ceramic kiln loading problem.

Although the *weighted-sum method* (WSM) is recognised as the most popular, it does not ensure the acquisition of all Pareto-optimal solutions in non-convex regions owing to the linear combination of its objectives (Miettinen, 2012). The *conic scalarisation method* (CSM) outlined by Gasimov facilitates Pareto-front via cones instead of the hyperplanes employed in WSM, without necessitating convexity assumptions (Gasimov, 2001; Kasimbeyli, 2013). The proposed multi-objective model, scalarised using WSM (Eq. (1)) and CSM (Eq. (2)) respectively, uses the following calculation:

$$\max \ \text{WSM}(x) = w_1 f_1(x) + w_2 f_2(x) \tag{1}$$

$$\max \ \text{CSM}(x) = w_1(f_1(x) - Rf_1) + w_2(f_2(x) - Rf_2) + \alpha[\, |f_1(x) - Rf_1| + |f_2(x) - Rf_2|| \,] \tag{2}$$

where $w_1, w_2 > 0$ are the importance degrees of the total score ($f_1(x)$) and the total unit profit ($f_2(x)$), respectively. α is utilised to form a cone by adjusting the angle of a cone surface, $0 \leqslant \alpha < min\{w_1, w_2\}$. Rf_1 and Rf_2 denote reference points that can be determined by a decision maker for the objective values in this study.

The objective function values are normalised by dividing them by the maximum value of the corresponding objective function because different objective functions may have varying magnitudes (Marler and Arora, 2005). The normalised objective function values are utilised as the fitness value in the proposed multi-objective genetic algorithm with the scalarisation methods. In these equations, the *ideal value* of each objective represents the maximum value among all solutions examined so far. The maximum value of the corresponding objective is established as the ideal point based on the 31 runs by setting $w_1 = 1$ ($w_2 = 0$) and $w_2 = 1$ ($w_1 = 0$), respectively. The *nadir value* is determined as the maximum value of each objective among the obtained results for the same objective weights. I_k denotes the ideal value of the kth objective ($k = 1, 2$). The fitness values scalarised by WSM and CSM are normalised as depicted in Eqs. (3)-(4):

$$\max \ \text{WSM}^{Norm}(x) = \left[w_1 \frac{f_1(x)}{I_1} + w_2 \frac{f_2(x)}{I_2} \right] \tag{3}$$

$$\max \ \text{CSM}^{Norm}(x) = \left[w_1 \left(\frac{(f_1(x) - Rf_1}{I_1} \right) + w_2 \left(\frac{(f_2(x) - Rf_2}{I_2} \right) \right] + \alpha \left[\frac{|(f_1(x) - Rf_1|}{I_1} + \frac{|(f_2(x) - Rf_2|}{I_2} \right] \tag{4}$$

Algorithm 1 illustrates the solution of the multi-objective GA approach for the ceramic kiln loading problem.

Data: Table 3
Result: The best solution
Randomly generate an initial population;
Compute the fitness of each individual (see Eqs. (3-4));
while *termination criteria not satisfied* **do**
  Choose parents from the population ← (*PopSize*, *TourSize*);
  Perform crossover to produce offspring ← (OOX, parents);
  Perform mutation operator ← (SPM, offspring);
  Compute the fitness of each individual (see Eqs. (3-4));
  Apply elitism operator;
end

Algorithm 1: Pseudo-code of the proposed multi-objective algorithm.

4.1 Chromosome Representation and Fitness Function Evaluation

A chromosome fundamentally represents a sequence of genes, serving as a candidate solution for a given problem. Depending on the problem, it can possess a specialized structure such as one-dimensional, multi-dimensional, or a tree data structure.

$p1$	p_2	p_3	p_4	p_5	p_6	...	p_{168}	p_{169}
1	5	16	19	21	24	...	72	84

Figure 3: An illustration of the chromosome representation.

In this study, the chromosome structure utilised is implemented as permutation encoding as illustrated in Figure 3. The length of the chromosome corresponds to the total number of products. As the constraint of the problem arises from the total area of the racks on the pallet, the algorithm is designed to calculate the number of products that will fill the rack with the highest priority to optimise the objective function. Once the chromosome structure is established, allocation begins by considering the first priorities of the racks from which each product can be assigned under the constraint of the rack area and product area. If the rack with the first priority for a product has already been occupied due to a previously assigned product, a similar process is applied to other priorities. This process is repeated for each order quantity of each product. If all the racks to which a product can be assigned, according to the dataset in Table 3, are occupied, then the remaining order of that product may not be assigned or only partially assigned. Subsequently, a similar process begins for the following gene representing a product. Thus, this process continues until the total rack area is filled. If the total area is filled, other products in the chromosome structure are not considered. Up to that point, the unit profits of the listed products are calculated, resulting in the total unit profit calculation. The total score is computed considering the priority scores of the racks assigned to the products in the same list. This process is completed the calculation of each objective.

4.2 Genetic Operators

Genetic operators play a crucial role in promoting diversity within the population and are fundamental in addressing the research problem. Specifically, mutation, elitism, selection, and crossover operators are tailored to provide viable solutions.

Crossover operator: the crossover operator facilitates the generation of improved chromosomes by enabling gene exchange. This study applied *order-based one-point crossover (OOX) operator* to the proposed algorithm. Firstly, a random crossover point is selected on the chromosome. Subsequently, all genetic material in the parents up to the crossover point is duplicated into two offspring. The products previously allocated are removed from the other parent. Then, the remaining genetic material from each parent is copied into the offspring without altering their sequence (Ruiz et al., 2006; Deliktaş et al., 2021).

Mutation operator: the primary objective of the mutation operator is to maintain population diversity. The proposed algorithm incorporates a *Swap (SPM) operator* (Deliktaş, 2022; Deliktaş and Ustun, 2023). SPM involves exchanging two operations and their machine assignments randomly selected while preserving the routing of operations across machines.

Selection operator: the selection operator plays a crucial role in ensuring the survival probability of the best individuals. Various standard selection operators, including roulette wheel selection, rank-based selection, tournament selection, and seed selection, are documented in the literature. At this stage, tournament selection is preferred due to its superior convergence properties and manageable computational complexity.

Elitism operator: it guarantees the preservation of the fittest chromosomes from one generation to the next, thereby safeguarding their advantageous characteristics following crossover and mutation operations.

5 COMPUTATIONAL RESULTS

This section presents the experimental results obtained from the proposed multi-objective algorithm to illustrate its robustness and effectiveness. We conduct 31 independent runs to compute the statistical outcomes of the algorithms. All methodologies are implemented using C# 2022, and the experiments are conducted on a laptop equipped with a 2.1 GHz Core i7 CPU and 16.00 GB RAM. The algorithm terminates once the maximum number of fitness evaluations, which is a factor of the population size, is exceeded.

5.1 Parameter Tuning

The proposed approach employs the Iterated F-Race algorithm (**irace**), an automated parameter configuration tool (López-Ibáñez et al., 2016). This algorithm is an iterated version of the F-Race parameter setting approach (Birattari et al., 2010), which operates on a competition-based principle. This tool refines parameter values by conducting successive races, where each race evaluates various candidate configurations across different problem instances. Through the application of the Friedman test, configurations demonstrating significantly poorer performance are identified and eliminated iteratively. Winners from each race contribute to generating improved candidate configurations for subsequent iterations. The outcome of this iterative process yields the optimal parameter configuration for the algorithm under examination.

Table 1: Ranges of the parameters for the proposed multi-objective algorithm and the best parameter configuration obtained with **irace**.

Parameter	Type	Range	Selected value
PopSize	Integer	$[20,100]$	79
TourSize	Integer	$[2,10]$	9
CrossProb	Real	$[0.7,1]$	0.74
MutaProb	Real	$[0.01,0.5]$	0.01

In order to facilitate parameter adjustment utilising **irace**, it is imperative to delineate the parameters slated for tuning, establish their permissible ranges, and delineate the set of instances earmarked for the tuning process. Table 1 outlines the selected parameters earmarked for tuning and their respective ranges tailored for the proposed algorithm. The proposed algorithm based on scalarisation methods continues running until the termination criterion is satisfied. This study defines the termination criterion as the maximum fitness evaluation number. This value is derived by multiplying the total number of parts by a constant. As shown in Figure 4, it is sufficient to set this value to 1000.

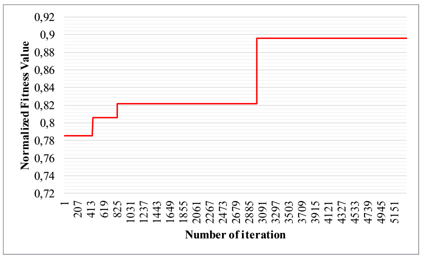

Figure 4: The convergence graph.

5.2 Results and Discussion

Utilising the aforementioned parameters, the performance of the proposed multi-objective algorithm is analysed and tested. The reference points employed in CSM integrated with the GA are computed using the ideal points and nadir points of each objective. For the computation of the reference points, a convex combination technique (Kim and Kim, 2006) is utilised to ensure well-distributed reference points (Deliktaş and Ustun, 2023). Hence, different reference point combinations are employed in this study to compare the performance of each scalarisation method. Additionally, this study considers nine different combinations of objective weights: $[w_1 = 0.1; w_2 = 0.9]$, $[w_1 = 0.2; w_2 = 0.8]$, $[w_1 = 0.3; w_2 = 0.7]$, $[w_1 = 0.4; w_2 = 0.6]$, $[w_1 = 0.5; w_2 = 0.5]$, $[w_1 = 0.6; w_2 = 0.4]$, $[w_1 = 0.7; w_2 = 0.3]$, $[w_1 = 0.8; w_2 = 0.2]$, and $[w_1 = 0.9; w_2 = 0.1]$, respectively.

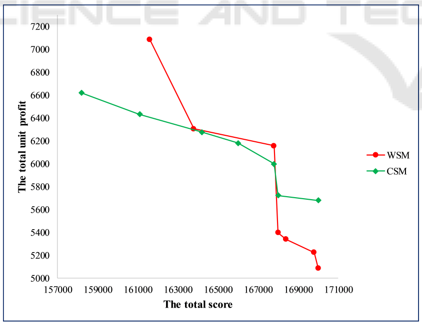

Figure 5: Pareto plot of the results obtained from WSM and CSM.

We generated plots illustrating the Pareto fronts, representing the set of non-dominated trade-off solutions attained by each method. As shown in Figure 5, the GA with WSM mostly produces better results when compared to the GA with CSM.

The best solution obtained from the proposed multi-objective algorithm is presented in Table 2. Based on this table, a total of 185 tableware items from 20 different products have been assigned to the racks on the pallet. The occupancy rate of the kiln pallet is 100%. The total objective value for the assigned items on this pallet is 164,000, while the total unit profit objective value is 5205.5. Table 2 provides information about each product assigned to each rack and the order quantity of each product. According to the results in Table 2, a sample visual of the fully occupied kiln pallet in Figure 6 was illustrated.

Table 2: An illustration of the products assigned to the racks on the pallet and the amount of assigned products according to the result obtained from the best solution.

Products	SR	SR (TS)	6LRR	5LRR	FR
P_{1}	—	—	10	—	—
P_{5}	—	—	10	—	—
P_{16}	—	—	10	—	—
P_{19}	—	—	3	—	—
P_{21}	—	—	—	10	—
P_{24}	—	—	—	10	—
P_{31}	—	—	—	10	—
P_{36}	—	—	—	10	—
P_{38}	—	—	—	10	—
P_{45}	—	—	—	10	—
P_{48}	—	2	—	—	—
P_{58}	—	—	—	—	10
P_{60}	—	—	—	—	10
P_{94}	—	—	—	—	10
P_{112}	10	—	—	—	—
P_{125}	10	—	—	—	—
P_{126}	10	—	—	—	—
P_{158}	—	—	10	—	—
P_{159}	—	—	10	—	—
P_{164}	—	—	10	—	—

SR: Square rack,
SR(TS): Square rack (top-side),
6LRR: Six-layer rectangular rack,
5LRR: Five-layer rectangular rack,
FR: Floor rack.

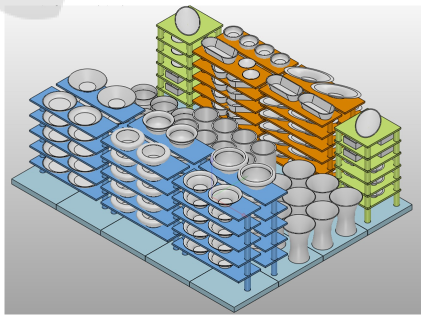

Figure 6: An illustration of the ceramic kiln loading according to the best solution based on the proposed multi-objective algorithm.

The conceptual design and enhancement of a Decision Support System (DSS) for strategic planning are crucial for overseeing operational activities within a ceramic industry system. Figure 7 illustrates the application of the proposed DSS through a real-world problem within the ceramic industry.

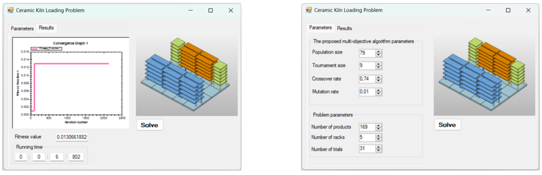

Figure 7: Screenshots of the decision support system.

6 CONCLUSIONS

This study investigated the Ceramic Kiln Loading Problem. Given the NP-hardness of the proposed model, we proposed a multi-objective genetic algorithm approach to load the pallet, aiming to maximise both total score and total unit profit simultaneously. We developed simple DSS software based on the multi-objective GA approach to obtain prompt and high-quality solutions to problems. Furthermore, we utilised an **irace** method to fine-tune the parameters of the GA.

There are numerous avenues for future research. Various heuristic or metaheuristic approaches, including ant colony optimisation, genetic algorithms, simulated annealing, and tabu search, can be applied to this model. Enhancements could be made by incorporating multi-objective considerations into the research. Additionally, future studies may explore the inclusion of multiple-kiln pallets.

ACKNOWLEDGEMENTS

This work was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under the BIDEB 2209-B Industry Oriented Research Project Support Programme for Undergraduate Students with grant number 1139B412102110.

7 CRediT AUTHORSHIP CONTRIBUTION STATEMENT

Derya Deliktaş: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - Original draft, Reviewing and Editing, Project administration, Visualization. **Ayşe Kaygısız:** Data acquisition, Validation, Visualization.

REFERENCES

Birattari, M., Yuan, Z., Balaprakash, P., and Stützle, T. (2010). F-Race and iterated F-Race: An overview. *Experimental methods for the analysis of optimization algorithms*, pages 311–336.

Chen, C.-S., Lee, S.-M., and Shen, Q. (1995). An analytical model for the container loading problem. *European Journal of operational research*, 80(1):68–76.

Deliktaş, D. (2022). Self-adaptive memetic algorithms for multi-objective single machine learning-effect scheduling problems with release times. *Flexible Services and Manufacturing Journal*, 34(3):748–784.

Deliktaş, D., Özcan, E., Ustun, O., and Torkul, O. (2021). Evolutionary algorithms for multi-objective flexible job shop cell scheduling. *Applied Soft Computing*, 113:107890.

Deliktaş, D. and Ustun, O. (2023). Multi-objective genetic algorithm based on the fuzzy multimoora method for solving the cardinality constrained portfolio optimization. *Applied Intelligence*, 53(12):14717–14743.

Dereli, T. and Sena Das, G. (2010). A hybrid simulated annealing algorithm for solving multi-objective container-loading problems. *Applied Artificial Intelligence*, 24(5):463–486.

Dyckhoff, H. (1990). A typology of cutting and packing problems. *European journal of operational research*, 44(2):145–159.

Gasimov, R. N. (2001). Characterization of the benson proper efficiency and scalarization in nonconvex vector optimization. In *Multiple criteria decision making in the new millennium*, pages 189–198. Springer.

Hodgson, T. J. (1982). A combined approach to the pallet loading problem. *IIE Transactions*, 14(3):175–182.

Holland, J. H. (1973). Genetic algorithms and the optimal allocation of trials. *SIAM journal on computing*, 2(2):88–105.

Junqueira, L., Morabito, R., and Yamashita, D. S. (2012). Three-dimensional container loading models with cargo stability and load bearing constraints. *Computers & Operations Research*, 39(1):74–85.

Kasimbeyli, R. (2013). A conic scalarization method in multi-objective optimization. *Journal of Global Optimization*, 56(2):279–297.

Kim, J. and Kim, S.-K. (2006). A chim-based interactive tchebycheff procedure for multiple objective decision making. *Computers & Operations Research*, 33(6):1557–1574.

Lim, A., Ma, H., Xu, J., and Zhang, X. (2012). An iterated construction approach with dynamic prioritization for solving the container loading problems. *Expert Systems with Applications*, 39(4):4292–4305.

Liu, J., Smith, A. E., and Qian, D. (2016). The vehicle loading problem with a heterogeneous transport fleet. *Computers & Industrial Engineering*, 97:137–145.

López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L. P., Birattari, M., and Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58.

Marler, R. T. and Arora, J. S. (2005). Function-transformation methods for multi-objective optimization. *Engineering Optimization*, 37(6):551–570.

Miettinen, K. (2012). *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media.

Özdemir, D., Deliktaş, D., and Ermutaf, E. G. (2022). A decision support system for multi-objective porcelain container loading problem based on genetic algorithm. *Academic Platform Journal of Engineering and Smart Systems*, 10(2):78–85.

Panchal, G. and Panchal, D. (2015). Solving np hard problems using genetic algorithm. *Transportation*, 106:6–2.

Paquay, C., Schyns, M., and Limbourg, S. (2016). A mixed integer programming formulation for the three-dimensional bin packing problem deriving from an air cargo application. *International Transactions in Operational Research*, 23(1-2):187–213.

Ramos, A. G., Silva, E., and Oliveira, J. F. (2018). A new load balance methodology for container loading problem in road transportation. *European Journal of Operational Research*, 266(3):1140–1152.

Romão, O. C., dos Santos, A. G., and Arroyo, J. E. C. (2012). A two phase hybrid method to support 2d and 3d vehicle loading. In *2012 12th International Conference on Intelligent Systems Design and Applications (ISDA)*, pages 320–325. IEEE.

Ruiz, R., Maroto, C., and Alcaraz, J. (2006). Two new robust genetic algorithms for the flowshop scheduling problem. *Omega*, 34(5):461–476.

Sheng, L., Hongxia, Z., Xisong, D., and Changjian, C. (2016). A heuristic algorithm for container loading of pallets with infill boxes. *European Journal of Operational Research*, 252(3):728–736.

Silva, E., Oliveira, J. F., and Wäscher, G. (2016). The pallet loading problem: a review of solution methods and computational experiments. *International Transactions in Operational Research*, 23(1-2):147–172.

Terno, J., Scheithauer, G., Sommerweiß, U., and Riehme, J. (2000). An efficient approach for the multi-pallet loading problem. *European Journal of Operational Research*, 123(2):372–381.

Tian, T., Zhu, W., Lim, A., and Wei, L. (2016). The multiple container loading problem with preference. *European Journal of Operational Research*, 248(1):84–94.

Toffolo, T. A., Esprit, E., Wauters, T., and Berghe, G. V. (2017). A two-dimensional heuristic decomposition approach to a three-dimensional multiple container loading problem. *European Journal of Operational Research*, 257(2):526–538.

Vélez-Gallego, M. C., Teran-Somohano, A., and Smith, A. E. (2020). Minimizing late deliveries in a truck loading problem. *European Journal of Operational Research*, 286(3):919–928.

Wäscher, G., Haußner, H., and Schumann, H. (2007). An improved typology of cutting and packing problems. *European journal of operational research*, 183(3):1109–1130.

Wu, Y., Li, W., Goh, M., and De Souza, R. (2010). Three-dimensional bin packing problem with variable bin height. *European journal of operational research*, 202(2):347–355.

Young-Gun, G. and Kang, M.-K. (2001). A fast algorithm for two-dimensional pallet loading problems of large size. *European Journal of Operational Research*, 134(1):193–202.

Yüceer, Ü. and Özakça, A. (2010). A truck loading problem. *Computers & Industrial Engineering*, 58(4):766–773.

Zhang, D.-F., Peng, Y., Zhu, W.-X., and Chen, H.-W. (2009). A hybrid simulated annealing algorithm for the three-dimensional packing problem. *Chinese Journal of Computers*, 32(11):2147–2156.

Zheng, J.-N., Chien, C.-F., and Gen, M. (2015). Multi-objective multi-population biased random-key genetic algorithm for the 3-d container loading problem. *Computers & Industrial Engineering*, 89:80–87.

Zhou, Q. and Liu, X. (2017). A swarm optimization algorithm for practical container loading problem. In *IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society*, pages 5690–5695. IEEE.

APPENDIX

Table 3: Dataset of real-life ceramic kiln loading problem.

$i \backslash j$	R_1	R_2	R_3	R_4	R_5	Unit profit
	SR	SR (TS)	6LRR	5LRR	FR	
P_{1}			I (2)	II (2)		86.97
P_{2}			I (2)	II (2)		47.25
P_{3}			I (2)	II (2)		24.45
P_{4}			I (2)	II (2)		15.75
P_{5}			I (2)	II (2)		37.1
P_{6}			I (2)	II (2)		25.22
P_{7}			I (2)	II (2)		42.89
P_{8}			I (2)	II (2)		5.31
P_{9}			I (2)	II (2)		5
P_{10}			I (2)	II (2)		3.84
P_{11}			I (2)	II (2)		6.62
P_{12}			I (2)	II (2)		6.62
P_{13}			I (2)	II (2)		5.21
P_{14}			I (2)	II (2)		6.62
P_{15}			I (2)	II (2)		11.82
P_{16}			I (2)	II (2)		7.41
P_{17}			I (2)	II (2)		9.45
P_{18}			I (2)	II (2)		15.75
P_{19}			I (2)	II (2)		23.63
P_{20}				I (2)		3.84
P_{21}				I (2)		6.62
P_{22}				I (2)		5.21
P_{23}				I (2)		6.62
P_{24}				I (2)		23.63
P_{25}				I (2)		6.32
P_{26}				I (2)		5.85
P_{27}				I (2)		7.52
P_{28}				I (2)		17.18
P_{29}				I (2)		13.7
P_{30}				I (2)		9
P_{31}				I (2)		29.78
P_{32}				I (2)		22.94
P_{33}				I (2)		15.41
P_{34}			II (2)	I (2)		24.95
P_{35}			II (2)	I (2)		15.41
P_{36}			II (2)	I (2)		23.9
P_{37}	I (1)					23.9
P_{38}			II (2)	I (2)		37.86
P_{39}			II (2)	I (2)		42.24
P_{40}			II (2)	I (2)		6.93
P_{41}			II (2)	I (2)		5.09
P_{42}			II (2)	I (2)		3.66
P_{43}			II (2)	I (2)		5.06
P_{44}			II (2)	I (2)		6.27
P_{45}			II (2)	I (2)		7.49

Table 3: Dataset of real-life ceramic kiln loading problem (cont.).

$i \backslash j$	R_1	R_2	R_3	R_4	R_5	Unit profit
	SR	SR (TS)	6LRR	5LRR	FR	
P_{46}			II (2)	I (2)		7.83
P_{47}		I (1)				8.85
P_{48}		I (1)				11.61
P_{49}		I (1)				20.76
P_{50}		I (1)				31.01
P_{51}		I (1)				7.07
P_{52}					I (30)	13.26
P_{53}					I (30)	20.91
P_{54}					I (30)	10.95
P_{55}					I (30)	10.95
P_{56}					I (30)	23.06
P_{57}					I (30)	17.04
P_{58}					I (30)	16.25
P_{59}					I (30)	29.07
P_{60}					I (30)	39.95
P_{61}					I (30)	53.79
P_{62}					I (30)	58.32
P_{63}					I (30)	5.81
P_{64}					I (30)	4.43
P_{65}					I (30)	6.62
P_{66}					I (30)	6.62
P_{67}					I (30)	6.62
P_{68}					I (30)	25.19
P_{69}					I (30)	7.41
P_{70}					I (30)	30.02
P_{71}					I (30)	5.81
P_{72}					I (30)	4.02
P_{73}					I (30)	5.58
P_{74}					I (30)	4.02
P_{75}					I (30)	9.32
P_{76}					I (30)	5.55
P_{77}					I (30)	14.93
P_{78}					I (30)	7.85
P_{79}					I (30)	8.6
P_{80}					I (30)	10.35
P_{81}					I (30)	12.81
P_{82}					I (30)	22.86
P_{83}					I (30)	7.13
P_{84}					I (30)	7.83
P_{85}					I (30)	8.85
P_{86}					I (30)	10.11
P_{87}					I (30)	36.68
P_{88}					I (30)	7.5
P_{89}					I (30)	5.76
P_{90}					I (30)	2.6

Table 3: Dataset of real-life ceramic kiln loading problem (cont.).

$i \backslash j$	R_1	R_2	R_3	R_4	R_5	Unit profit
	SR	SR (TS)	6LRR	5LRR	FR	
P_{91}					I (30)	5.76
P_{92}					I (30)	5.76
P_{93}					I (30)	3.84
P_{94}					I (30)	6.05
P_{95}					I (30)	9.42
P_{96}					I (30)	11.4
P_{97}					I (30)	10.83
P_{98}					I (30)	6.32
P_{99}					I (30)	11.93
P_{100}					I (30)	8.57
P_{101}					I (30)	9.42
P_{102}					I (30)	5.49
P_{103}					I (30)	5.49
P_{104}					I (30)	2.97
P_{105}	I (3)	II (3)				10.37
P_{106}	I (3)	II (3)				11.4
P_{107}	I (3)	II (3)				10.37
P_{108}	I (3)	II (3)				11.4
P_{109}	I (3)	II (3)				6.65
P_{110}	I (3)	II (3)				10.37
P_{111}	I (3)	II (3)				8.46
P_{112}	I (3)	II (3)				8.57
P_{113}	I (3)	II (3)				5.49
P_{114}	I (3)	II (3)				9.42
P_{115}	I (3)	II (3)				6.05
P_{116}	I (3)	II (3)				6.05
P_{117}	I (3)	II (3)				11.4
P_{118}	I (3)	II (3)				13.19
P_{119}	I (3)	II (3)				7.7
P_{120}	I (3)	II (3)				7.68
P_{121}	I (3)	II (3)				11.15
P_{122}	I (3)	II (3)				10.35
P_{123}	I (3)	II (3)				11.04
P_{124}	I (3)	II (3)				45.09
P_{125}	I (3)	II (3)				51.8
P_{126}	I (3)	II (3)				95.55
P_{127}	I (2)	II (2)				19.62
P_{128}	I (2)	II (2)				27.32
P_{129}	I (2)	II (2)				24.23
P_{130}	I (2)	II (2)				34.05
P_{131}	I (2)	II (2)				14.39
P_{132}	I (2)	II (2)				32.58
P_{133}	I (2)	II (2)				90.21
P_{134}	I (2)	II (2)				50.43
P_{135}	I (2)	II (2)				22.92
P_{136}	I (2)	II (2)				27.32
P_{137}	I (2)	II (2)				42.06
P_{138}	I (2)	II (2)				45.86
P_{139}	I (2)	II (2)				17.42
P_{140}	I (1)	II (1)				20.84
P_{141}	I (1)	II (1)				41.67
P_{142}	I (1)	II (1)				11.82
P_{143}	I (1)	II (1)				23.63
P_{144}	I (1)	II (1)				18.48

Table 3: Dataset of real-life ceramic kiln loading problem (cont.).

$i \backslash j$	R_1	R_2	R_3	R_4	R_5	Unit profit
	SR	SR (TS)	6LRR	5LRR	FR	
P_{145}	I (1)	II (1)				57.95
P_{146}	I (1)	II (1)				15.63
P_{147}	I (1)	II (1)				19.14
P_{148}	I (1)	II (1)				50.97
P_{149}	I (1)	II (1)				68.15
P_{150}	I (1)	II (1)				7.22
P_{151}	I (1)	II (1)				5.21
P_{152}		I (1)			II (5)	5.21
P_{153}		I (1)			II (5)	7.41
P_{154}		I (1)			II (5)	9.53
P_{155}		I (1)			II (5)	22.83
P_{156}		I (1)			II (5)	15.03
P_{157}		I (1)			II (5)	46.86
P_{158}	I (3)		II (4)		III (30)	20.58
P_{159}	I (3)		II (4)		III (30)	5.21
P_{160}	I (3)		II (4)		III (30)	15.75
P_{161}	I (3)		II (4)		III (30)	7.41
P_{162}	I (3)		II (4)		III (30)	5.21
P_{163}	I (3)		II (4)		III (30)	8.15
P_{164}	I (3)		II (4)		III (30)	6.48
P_{165}	I (3)		II (4)		III (30)	5.9
P_{166}	I (3)		II (4)		III (30)	7.41
P_{167}	I (3)		II (4)		III (30)	15.75
P_{168}	I (3)		II (4)		III (30)	7.41
P_{169}	I (3)		II (4)		III (30)	5.21

SR: Square rack,
SR(TS): Square rack (top-side),
6LRR: Six-layer rectangular rack,
5LRR: Five-layer rectangular rack,
FR: Floor rack.