A Hybrid Approach for Pallet Loading in Ceramic Tile Industry

Marco Taccini[1][0009−0004−7257−473X], Matheus Aguilar de Oliveira[2][0000−0002−4671−2500], André Gustavo dos Santos[2][0000−0002−9700−9815], Thiago Alves de Queiroz[3][0000−0003−2674−3366], and Manuel Iori[1][0000−0003−2097−6572]

[1] Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Reggio Emilia, Italy.
{marco.taccini,manuel.iori}@unimore.it
[2] Department of Informatics, Federal University of Viçosa, Viçosa, MG, Brazil.
{matheus.a.aguilar,andresantos}@ufv.br
[3] Institute of Mathematics and Technology, Federal University of Catalão, Catalão, GO, Brazil.
taq@ufcat.edu.br

Abstract. Motivated by a real-world case study in ceramic tile production, this paper addresses the problem of determining the minimum number of pallets required to load a given set of boxes. The problem must be solved quickly to give customers an expectation of the transportation cost of their orders. In addition, not all constraints and instance data can be easily determined in advance, and the items are loaded onto the pallets by operators who mostly rely on their personal experience. Therefore, traditional model-based solution methods do not apply well, and data-driven approaches are preferable. To solve the problem, we propose a hybrid algorithm in which a machine learning technique is trained over a company dataset comprising two years of customer orders, with the aim of predicting the number of pallets required by an order. The accuracy of the machine learning technique is largely improved by including additional features, such as lower and upper bounds, in the dataset, obtained using quick optimization algorithms. The resulting hybrid algorithm has been compared with the model-based software currently used at the company, consistently providing better-quality results in shorter computing times.

Keywords: Cutting and packing problems · Distributor's pallet loading · Ceramic tile · Machine learning.

1 Introduction

The global ceramic tile market has grown significantly over the past decade, with worldwide production reaching 16.8 billion square meters [1]. Italy is a leading exporter, generating €7.2 billion in revenue, highlighting the sector's importance to the national economy. Efficient logistics optimization in the ceramic

tile industry, including loading of tiles on pallets and transportation of pallets, is crucial for cost reduction and company competitiveness. Loading operations involve transferring specified quantities of ceramic tile boxes, either of the same size or different sizes, from production pallets to shipping pallets to fulfill customers' orders. When loading pallets, the company needs to carefully consider weight and volume constraints and keep pallet stability and balance to ensure the safe transport of tiles while using the minimum number of shipping pallets.

In this paper, we handle a variant of the distributor's pallet loading problem (DPLP) [12] that emerges from an international ceramic tile company headquartered in Italy. The company has an e-commerce platform to receive customers' orders. As the company does not provide the transportation of the shipping pallets, customers need to collect their pallets by using their vehicles or third-party carriers. Consequently, the company should provide the number of shipping pallets associated with each order or at least predict this number accurately, especially for customers who need to contract a third-party carrier. This prediction is complex, even if using specialized software to address pallet loading problems [19], because the company faces specific constraints in its operational daily routine, necessitating extensive customization of such software.

In this paper, we propose a hybrid approach to predict the number of shipping pallets required for a given customer order. The approach combines machine learning (ML) techniques to gain information from a large dataset of previously shipped pallets with techniques from the optimization field to enrich the dataset by quickly providing additional features such as lower and upper bounds on the number of pallets. The hybrid approach is then compared with solutions currently in use at the company, including manual operators' decisions and outputs from a given specialized software [14]. The computational experiments show that the proposed approach is a valuable tool for the company.

The remainder of the paper is organized as follows. Section 2 contains a brief review of the literature related to the DPLP. Section 3 describes the company problem with its constraints and requirements. Section 4 presents our proposed hybrid approach. Section 5 contains the numerical experiments on real-world instances. Finally, Section 6 concludes the paper and provides several directions for future research.

2 Literature Review

The literature on cutting and packing problems is extensive. According to [20], the first scientific papers on these problems date back to the 1940s. The most recent surveys concern problems in one-, two-, and three-dimensions and contain detailed discussions on solution methods based on exact and heuristic approaches (see, e.g., [10] and [22]). An essential characteristic of these problems is given by the number of practical constraints they may include, such as complete shipment, conflicting items, cargo stability, load-bearing, multi-drop, and load-balancing [13]. Concerning pallet loading problems, [19] surveyed the manufacturer's variant, in which all items are identical, and the objective is to pack the maximum

number of items onto a single pallet. Besides commenting on the different data sets and solution methods, the authors emphasized that the computational complexity of this problem is still an open question. The DPLP, on the other hand, assumes that items are not necessarily identical and is an NP-hard problem. Its objective is to find the minimum number of pallets necessary to load all items. The DPLP is also referenced as the multi-pallet loading problem [21] and the pallet building problem [6].

One of the first authors to handle the DPLP was [9], considering the transport of palletized cargo by the U.S. Air Force. The author proposed a dynamic programming-based algorithm and tested it on instances having 30 boxes. In [21], the DPLP was solved using an algorithm based on branch-and-bound. The authors considered practical constraints such as weight limits, load-bearing, stability, and grouping of items. In [4], the problem had no limit on the number of boxes to load, and boxes could have multiple orientations. The authors developed a recursive partitioning algorithm, which integrated a recursive five-block heuristic and an L-approach. The authors could not find an instance for which this algorithm failed in finding an optimal solution (such an instance was later found in [17]). In [6], the DPLP emerged from a robotized task of loading boxes into layers, which were then stacked while meeting load-bearing requirements. Constraints like contiguity, visibility, and multiple orientations were considered. The authors proposed a mathematical model and a reactive greedy randomized adaptive search procedure for solving realistic, large-sized instances. A new DPLP variant in which pallets may have different sizes was handled in [12]. The problem emerged from a lamp and lighting manufacturing company that needed to load many carton boxes onto pallets of multiple sizes. Constraints such as load-bearing and multiple orientations were taken into consideration.

Using artificial intelligence techniques to handle combinatorial optimization problems is a promising area of research. The survey in [3] discussed recent advances in combining ML techniques with optimization methods to efficiently handle hard combinatorial optimization problems. The authors also commented on the benefits (e.g., speed and generalization) and challenges (e.g., training and accuracy) of using ML purely. Concerning cutting and packing problems, an increasing number of publications combine ML or artificial intelligence techniques with exact or heuristic algorithms to handle, e.g., bin packing [8] and container loading problems [16].

3 Problem Description

The problem under investigation is a variant of the DPLP with specific operational and product-related constraints. After being produced, the ceramic tiles are first loaded on production pallets, which are only used for inbound logistics. An automated system then delivers the production pallets to the loading area. The company operators load the products onto shipping pallets using hydraulic machines. The automated system is also responsible for other logistics activities inside the warehouse. As a result, products are not available simultaneously,

as their arrival sequence depends on the system decisions and the warehouse's current state. For instance, machinery downtime may alter the order in which production pallets are sent to the loading areas. Moreover, the loading area cannot have more than four production pallets at a time. Due to this space limitation, operators need to load the products onto the pallets as soon as possible. As other products arrive, operators continue loading until the shipping pallets of that order are completed before moving to the next order. When loading a shipping pallet, operators must also satisfy load requirements such as vertical stability, load-bearing, and restricted box orientations.

Our DPLP can be formally described as follows. A customer order consists of a set B of three-dimensional boxes, each containing a given number of tiles. Each box belongs to a type $t \in T = \{1, 2, ..., m\}$, corresponding to a specific tile size. A box of type t has width $w_t \in \mathbb{Z}_+^*$, depth $l_t \in \mathbb{Z}_+^*$, height $h_t \in \mathbb{Z}_+^*$, and weight $p_t \in \mathbb{Z}_+^*$. Let $d_t \in \mathbb{N}$ denote the number of boxes of type t required by the order. Let P represent the set of identical shipping pallets, each with a two-dimensional loading surface of width $W \in \mathbb{Z}_+^*$ and depth $L \in \mathbb{Z}_+^*$. The DPLP's objective is to load all boxes requested in the order by using the minimum number of pallets. Constraints on the maximum pallet height $H \in \mathbb{Z}_+^*$ and weight $W_{\max} \in \mathbb{Z}_+^*$ need to be satisfied.

Due to these complex company constraints, obtaining a solution for this DPLP variant is not easy, especially for large-sized and heterogeneous orders (i.e., composed of boxes of different sizes). Even if a DPLP solution satisfying all constraints is available, it could suffer from warehouse uncertainties, possibly making it impractical for the company operators. Besides that, the company should give the number of pallets associated with each customer's order at the moment of purchase on its e-commerce platform. Therefore, the company is currently interested in a fast and accurate tool capable of predicting the number of shipping pallets, rather than solving a dedicated but time-consuming optimization problem.

4 Hybrid Approach

We propose a hybrid approach to the DPLP under investigation that considers optimization algorithms to enhance the predictive performance of an ML model. The company provided data on two years of sales and the corresponding number of pallets generated for each order. This data includes the maximum number of boxes per shipping pallet for each box size t, denoted by $F_t^{\max}$, and the maximum weight ($W_{\max}$) and volume (WLH) a shipping pallet can support.

An initial training dataset is created from the company-provided data. We start calculating the frequency $F_t = d_t$ (i.e., the number of boxes of type t), the weight $W_t = d_t p_t$, and the volume $V_t = w_t l_t h_t d_t$ for each type $t \in T$. The dataset is labeled with the number of shipping pallets Z_j generated by each order $j \in \{1, ..., n\}$. Since training on the initial dataset can take a long time, all intermediate training steps (Sections 4.1 - 4.3) are conducted using a reduced dataset. The complete dataset is only used on the final computational

experiments (Section 5). The reduced dataset is created by randomly selecting a subset of orders from the initial dataset, resulting in a training set with about $\frac{1}{8}$ of the total number of orders.

4.1 Machine Learning Model Selection

We developed a pipeline to train and test four different machine learning models. These models, which were chosen based on the literature review and our experience, consist of: XGBoost [7], LightGBM (LGBM) [11], Random Forest, and Support Vector Machine (SVM) from scikit-learn [15]. Each model passes through a hyperparameter tuning phase on the Mean Squared Error (MSE) and Mean Absolute Error (MAE) metrics, obtained by performing a standard grid search. Table 1 reports the hyperparameters and their possible values, besides the best values found along with MSE and MAE metrics. The possible values were defined using a trial and error approach and following the indications in each model documentation and the literature (see, e.g., [2]).

Table 1. Grid search results with the hyperparameter values and errors.

Model	Hyperparameter	Values	Best Value	MSE	MAE
XGBoost	learning_rate	0.1, 0.3	0.3	4.78	1.17
	max_depth	4, 5, 6	4		
	gamma	0, 0.25, 1	0.25		
LGBM	learning_rate	0.005, 0.01	0.01	16.49	2.58
	num_leaves	6, 8, 12, 16	16		
Random Forest	max_features	sqrt, log2, None	None	13.61	2.26
	max_depth	3, 6, 9	9		
	max_leaf_nodes	3, 6, 9	9		
SVM	C	0.1, 1, 10, 100, 1000	100	35.89	3.45
	gamma	1, 0.1, 0.01, 0.001, 0.0001	0.0001		

The results in Table 1 show that XGBoost outperforms the other models regarding the MSE and MAE values. Consequently, we decided to use only XGBoost in the following steps. We also decided to perform a deeper tuning of this model's hyperparameters. Table 2 contains the outcome of this additional tuning test, which leads to better MSE and MAE values for the XGBoost model.

4.2 Features Optimization

After successfully fine-tuning the XGBoost model, we investigated the importance of each feature in the reduced dataset. This analysis reveals that columns W_t and V_t, for $t \in T$, are marginally contributing to the model's predictions, as shown in Figure 1. Furthermore, tile sizes that appear less frequently in the dataset exhibit higher error rates, as shown in Figure 2.

Table 2. XGBoost tuning with the best hyperparameter values and errors.

Hyperparameter	Values	Best Value	MSE	MAE
learning_rate	0.01, 0.1, 0.2, 0.3	0.3	1.11	0.43
gamma	0, 0.1, 0.5	0.1		
max_depth	3, 6, 10	6		
min_child_weight	0, 1, 3, 5	0		
subsample	0.5, 1	0.5		
colsample_bytree	0.5, 1	1		

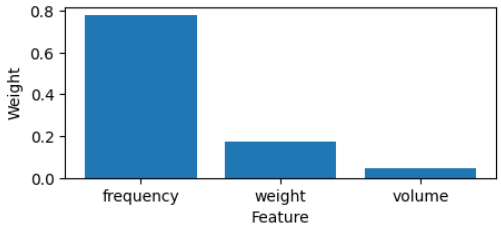

Fig. 1. Importance of the feature weights in the XGBoost model.

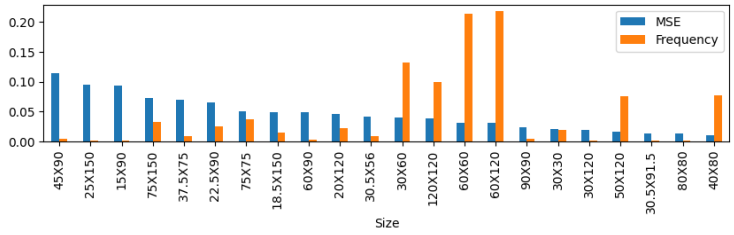

Fig. 2. MSE and frequency values (F_t), normalized and sorted by decreasing MSE.

These results allow the removal of less informative features, such as the size-grouped weight W_t and volume V_t. At the same time, we introduce more general order-aggregated features, as follows. Let O be the set of orders we need to process, and let $o \in O$ be a given order. We determine the total number of boxes $F_o = \sum_{t=1}^{m} F_t$, the total weight $W_o = \sum_{t=1}^{m} W_t$, the total volume $V_o = \sum_{t=1}^{m} V_t$, the average box weight $W_{avg} = \frac{W_o}{F_o}$, and the average box volume $V_{avg} = \frac{V_o}{F_o}$. Figure 3 shows how these new features impacted the predictive capability of the XGBoost model.

In the set O of orders, we observe that predicting the number of shipping pallets for homogeneous orders, defined O_h, which consist of only one type of box, and O_{full} ones, which require only production pallets regardless of the types, is straightforward for the model. In this way, we divided results into two groups: one containing all available orders O and another excluding these simpler

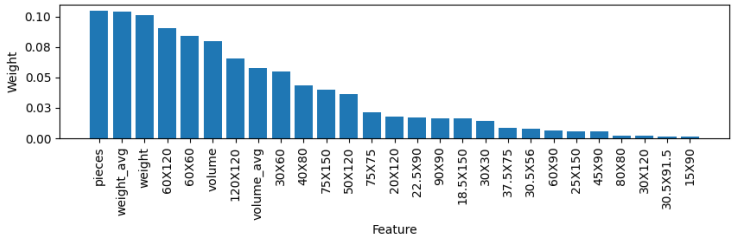

Fig. 3. Importance of the new order-aggregated features in the XGBoost model.

orders $O_{mix} = O - (O_h \cup O_{full})$. With this strategy, we evaluated the model in both scenarios, preventing simple orders to improve the metrics and resulting in premature conclusions. Table 3 summarizes the model's performance before and after including the new features.

Table 3. MSE and MAE values for each group of orders and the dataset structures S_1 (original) and S_2 (enriched with additional features).

	MSE		MAE	
Dataset structure	O	O_{mix}	O	O_{mix}
S_1	1.01	1.38	0.42	0.53
S_2	0.36	0.47	0.24	0.31

4.3 Incorporation of Optimization-based Features

Aiming to reduce the errors further, three additional features (i.e., HS_1, HS_2, and HS_3) computed with quick algorithms are introduced into the training dataset. To ensure computational efficiency, the proposed algorithms have the worst-case time complexity of at most $O(F_o^2)$, where F_o is the total number of boxes in the customer's order. The proposed algorithms are as follows:

- HS_1: calculated as $HS_1 = \max\left(\left\lceil \frac{\sum_{t=1}^{m} W_t}{W_{\max}} \right\rceil, \left\lceil \frac{\sum_{t=1}^{m} V_t}{WLH} \right\rceil\right)$. This can be interpreted as the value of the linear relaxation lower bound (LB);
- HS_2: calculated as $HS_2 = \sum_{t=1}^{m} \left\lceil \frac{F_t}{F_t^{\max}} \right\rceil$, where $F_t^{\max}$ is the maximum number of boxes of type t that can be loaded onto a pallet. This is, instead, an upper bound (UB) for the DPLP instance, computed by considering that all pallets are built with only homogeneous tiles (and this can easily be done by satisfying vertical stability and load bearing constraints);
- HS_3: described in Algorithms 1 and 2. This algorithm loads pallets by stacking layers of same-sized boxes, starting with those having the largest area until the pallet limits are reached. Once full, it begins a new pallet and continues until all boxes are packed. This may be seen as a quick heuristic solution approach, but, in fact, it does not consider vertical stability and load

bearing constraints, and hence it only provides an approximation function of the minimum number of pallets.

Algorithm 1 HS3 - constructive heuristic to give the number of shipping pallets

$pallets \leftarrow 1$ ▷ Pallet estimation
for $t \in \text{SORTED}(T)$ **do** ▷ Iterate over box types sorted by decreasing base area
  while $d_t > 0$ **do** ▷ Boxes of type t left to pack
    if $w - p_t < 0$ **OR** $h - h_t < 0$ **OR** no space to pack one box **then**
      $x, y, w, h \leftarrow W, L, W_{\max}, H$ ▷ Restart using an empty pallet
      $pallets \leftarrow pallets + 1$
    end if
    $b \leftarrow \lfloor \frac{w}{p_t} \rfloor$ ▷ How many boxes can be packed based on weight
    $q, d_x, d_y \leftarrow \text{TRYPACK}(b, w_t, l_t, x, y)$ ▷ Try to pack directly
    $q_2, d_{x2}, d_{y2} \leftarrow \text{TRYPACK}(b, l_t, w_t, x, y)$ ▷ Try rotating to pack
    if $q < q_2$ **OR** $(q = q_2$ **AND** $p_x p_y < p_{x2} p_{y2})$ **then**
      $q, d_x, d_y \leftarrow q_2, d_{x2}, d_{y2}$ ▷ Choose to pack more boxes, then more area
    end if
    $d_t, w, h \leftarrow d_t - q, w - p_t q, h - h_t$ ▷ Pack boxes, update weight and height
    $x, y \leftarrow d_x, d_y$ ▷ Update dimensions for next layer
  end while
end for
return $pallets$

Algorithm 2 TRYPACK

Input: b, b_x, b_y, l_x, l_y ▷ Boxes to pack, box and layer dimensions
$X \leftarrow \min(b, \lfloor \frac{l_x}{b_x} \rfloor)$ ▷ Maximum number of boxes in x-dimension
$Y \leftarrow \min(b, \lfloor \frac{l_y}{b_y} \rfloor)$ ▷ Maximum number of boxes in y-dimension
$q, d_x, d_y \leftarrow 0, 0, 0$ ▷ Best quantity and new layer dimensions
for $x \in \{1 \dots X\}$ **do** ▷ Test all number of boxes in x-dimension
  $y \leftarrow \min(Y, \lceil \frac{b}{x} \rceil)$ ▷ How many rows in y-dimension will be needed
  if $q < \min(b, xy)$ **OR** $(q = \min(b, xy)$ **AND** $d_x d_y < b_x b_y xy)$ **then**
    $q, d_x, d_y \leftarrow \min(b, xy), b_x x, b_y y$
  end if
end for
return q, d_x, d_y

Table 4 describes the final structure S_3 of the training dataset, which now includes the three additional optimization-based features. Compared with Table 3, the MSE and MAE values for the dataset structure S_3 are 0.24 and 0.19 for O, and 0.32 and 0.26 for O_{mix}, respectively, representing a considerable improvement in the accuracy of the model.

Table 4. Final training dataset structure S_3.

j	F_1	...	F_m	F_o	W_o	V_o	W_{avg}	V_{avg}	HS_1	HS_2	HS_3	Z_j
1	10	...	3	25	250	500	10	20	5	9	8	7
...	...	...	...	...	...	...	...	...	...	...	...	...
n	7	...	0	13	195	65	15	5	2	4	3	3

5 Computational Experiments

All the codes were implemented in Python 3.10 and executed in a computer with an Intel Core i5-1135G7 2.40 GHz processor, 8 GB of RAM, and Linux Mint 21 Cinnamon as the operating system. We evaluate the XGBoost model considering the structure S_3 (i.e., the one enriched with the additional optimization-based features) and the complete two-year dataset with all orders. This corresponds to our hybrid approach. In Table 5, we observe the impact of the two groups O_h and O_{full} on the training phase. Consequently, we train the model using the structure S_3, first including all orders O and then keeping only mixed orders O_{mix} (which we recall is equal to $O - (O_h \cup O_{full})$).

Table 5. MSE and MAE values for each group of orders and the dataset structure S_3.

			MSE		MAE	
Groups of orders	Orders	Training Time (min)	O	O_{mix}	O	O_{mix}
O	166391	54.79	0.22	0.73	0.11	0.33
O_{mix}	40533	26.90	0.25	0.76	0.12	0.33

As reported in Table 5, applying the model to all orders O yields slightly better results compared to using O_{mix}. However, using only O_{mix} can reduce the training time by around 51% while keeping satisfactory predictive results.

We next compare in Table 6 the proposed hybrid approach with PackVol [14], a specialized software the company uses to solve the DPLP heuristically. PackVol was configured by the company employees to satisfy the specific operational constraints of the DPLP as closely as possible. This means that PackVol is not able to capture some constraints. For example, as mentioned in Section 3, ceramic tiles of a given order are not all available simultaneously due to the warehouse's current state. As a result, PackVol could underestimate the number of pallets. We also present the number of pallets obtained from the company operators for comparison purposes. This experiment considers 30 real instances, consisting of 30 orders randomly selected among the heterogeneous orders made in 2024 (which are not included in the training set). The real instances I are all heterogeneous, and are sorted by increasing number of box types m. Its maximum value is 6 since 99.32% of the orders in the training dataset contain six or fewer types. Table 6 contains the instance number and the solution value (i.e., number of shipping pallets) obtained with PackVol, the proposed hybrid approach,

and the company operators. We assume the operators' solution is the basis for comparison, so the rows "Diff." contain the difference with such a solution for each instance. Positive values indicate the usage of more pallets, which could be acceptable. Negative values indicate fewer pallets in the solution and could represent additional logistic issues for customers (e.g., the need for contracting "on the flight" an extra vehicle).

Table 6. Results for the 30 real instances and comparison among the different methods.

Inst.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
# types	2	2	2	2	2	2	3	3	3	3	3	3	4	4	4	4	4	4	5	5	5	5	5	5	6	6	6	6	6	6
Company	2	6	4	1	6	2	22	5	5	8	10	7	5	8	9	7	15	6	26	9	28	2	13	10	22	10	22	21	10	16
PackVol	1	6	4	1	6	2	22	5	3	8	10	6	4	6	8	7	15	6	25	8	27	4	13	6	18	8	21	19	10	14
Diff.	-1	0	0	0	0	0	0	0	-2	0	0	-1	-1	-2	-1	0	0	0	-1	-1	-1	2	0	-4	-4	-2	-1	-2	0	-2
Hybrid	2	7	4	1	6	2	22	5	5	9	11	7	6	7	9	7	16	7	26	9	27	3	13	10	21	10	22	20	12	16
Diff.	0	1	0	0	0	0	0	0	0	1	1	0	1	-1	0	0	1	1	0	0	-1	1	0	0	-1	0	0	-1	2	0

Observing the results in Table 6, PackVol achieves an MSE of 2.13 and an MAE of 0.93, whereas our hybrid approach achieves significantly better results, with 0.50 and 0.43 values of MSE and MAE, respectively. These outcomes highlight that our approach outperforms PackVol in both metrics, demonstrating its capability to predict the number of pallets following the real context of the company with its operational requirements and workflows. Another important aspect is the computation time required by the proposed approach. Although the training phase may be time-consuming, after training, it solved all instances in just 0.016 seconds, while PackVol required around one second per instance. From Table 6, we can also observe that PackVol returns solutions different from those by the company in 16 cases, returning fewer pallets for 15 cases and achieving a largest deviation of -4 pallets. This may bring issues for customers. On the other hand, the hybrid approach returns solutions that differ from the company ones for just 12 instances, returning fewer pallets for only 4 of them and having the largest deviation of just 2 pallets. These values confirm that our approach, trained on real-world data, effectively models the problem constraints and results in accurate predictions.

6 Concluding Remarks

We proposed a hybrid approach composed of machine learning and optimization components to predict the number of pallets required to ship an order of ceramic tiles. We evaluated four machine learning models, and after fine-tuning, we selected XGBoost. Its performance was improved by adding new features that capture general information about the company's operational constraints. We also incorporated three measures obtained by quick optimization algorithms, leading to significant reductions in both MSE and MAE values. The resulting hybrid approach obtained better results than PackVol, the specialized model-based software currently used at the company.

The computational experiments highlight that our hybrid approach, once it is trained on real-world data, is fast and well-suited for the company-specific operational constraints. It allows more accurate and practical predictions than PackVol, resulting in more solutions equal to the ones obtained by the company operators and a reduced number of solutions requiring fewer pallets. This is remarkably achieved without feeding the approach with any direct information about the pallet loading constraints (i.e., stability, load-bearing requirements, and orientation), making the approach a very flexible tool.

Future research could propose new features to the hybrid approach (e.g., solutions obtained with heuristics based on local search). Another interesting research avenue is to work on a more general problem, considering pallets of different sizes, which will lead to a more challenging objective function. We finally plan to test the approach to other related pallet loading, cutting, and packing problems, as well as scheduling problems [5, 18].

Acknowledgments. We acknowledge financial support from the National Council for Scientific and Technological Development (CNPq) [grant n. 405369/2021-2, 408722/2023-1, 315555/2023-8, 444679/2024-3, and 407005/2024-2], the State of Goiás Research Foundation (FAPEG), and the National Recovery and Resilience Plan (NRRP), Mission 04 Component 2 Investment 1.5–NextGenerationEU, Call n. 3277 dated 30/12/2021, Award n.: 0001052 dated 23/06/2022. We are also grateful to the company, which has provided us with relevant information, feedback, and data.

References

1. ACIMAC Research Department: World production and consumption of ceramic tiles. https://www.mecs.org/it/shop/cat/16-mercato-ceramico/prod/111-produzione-e-consumo-mondiale-di-piastrelle-ceramiche (2023)
2. Banerjee, P.: A guide on XGBoost hyperparameters tuning. https://www.kaggle.com/code/prashant111/a-guide-on-xgboost-hyperparameters-tuning (2020), accessed: 2024-09-26
3. Bengio, Y., Lodi, A., Prouvost, A.: Machine learning for combinatorial optimization: a methodological tour d'horizon. European Journal of Operational Research **290**(2), 405–421 (2021)
4. Birgin, E.G., Lobato, R.D., Morabito, R.: Generating unconstrained two-dimensional non-guillotine cutting patterns by a recursive partitioning algorithm. Journal of the Operational Research Society **63**(2), 183–200 (2012)
5. Borges, Y.G.F., de Lima, V.L., Miyazawa, F.K., Pedrosa, L.L.C., Queiroz, T.A., Schouery, R.C.S.: Algorithms for the bin packing problem with scenarios. Journal of Combinatorial Optimization **48**(4), 34 (Oct 2024)
6. Calzavara, G., Iori, M., Locatelli, M., Moreira, M.C.O., Silveira, T.: Mathematical models and heuristic algorithms for pallet building problems with practical constraints. Annals of Operations Research (2021)
7. Chen, T., Guestrin, C.: XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. pp. 785–794 (2016)

8. Fang, J., Rao, Y., Zhao, X., Du, B.: A hybrid reinforcement learning algorithm for 2D irregular packing problems. Mathematics **11**(2) (2023)
9. Hodgson, T.J.: A combined approach to the pallet loading problem. A I I E Transactions **14**(3), 175–182 (1982)
10. Iori, M., de Lima, V.L., Martello, S., Miyazawa, F.K., Monaci, M.: Exact solution techniques for two-dimensional cutting and packing. European Journal of Operational Research **289**(2), 399–415 (2021)
11. LightGBM: Light gradient boosting machine. https://github.com/microsoft/LightGBM (2024), accessed: 2024-12-10
12. Mungwattana, A., Piyachayawat, T., Janssens, G.K.: A two-step evolutionary algorithm for the distributor's pallet loading problem with multi-size pallets. Flexible Services and Manufacturing Journal **35**(4), 1256–1275 (2023)
13. Nascimento, O.X., Queiroz, T.A., Junqueira, L.: Practical constraints in the container loading problem: Comprehensive formulations and exact algorithm. Computers & Operations Research **128**, 105186 (2021)
14. PackVol: Optimization software for load planning. https://www.packvol.com/index.html (2024), accessed: 2024-12-10
15. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. the Journal of machine Learning research **12**, 2825–2830 (2011)
16. Que, Q., Yang, F., Zhang, D.: Solving 3D packing problem using transformer network and reinforcement learning. Expert Systems with Applications **214**, 119153 (2023)
17. Queiroz, T.A., Miyazawa, F.K., Wakabayashi, Y.: On the L-approach for generating unconstrained two-dimensional non-guillotine cutting patterns. 4OR **13**(2), 199–219 (Jun 2015)
18. Queiroz, T.A., Mundim, L.R.: Multiobjective pseudo-variable neighborhood descent for a bicriteria parallel machine scheduling problem with setup time. International Transactions in Operational Research **27**(3), 1478–1500 (2020)
19. Silva, E., Oliveira, J.F., Wäscher, G.: The pallet loading problem: a review of solution methods and computational experiments. International Transactions in Operational Research **23**(1-2), 147–172 (2016)
20. Sweeney, P.E., Paternoster, E.R.: Cutting and packing problems: A categorized, application-orientated research bibliography. Journal of the Operational Research Society **43**(7), 691–706 (1992)
21. Terno, J., Scheithauer, G., Sommerweiß, U., Riehme, J.: An efficient approach for the multi-pallet loading problem. European Journal of Operational Research **123**(2), 372–381 (2000)
22. Yagiura, M., Imahori, S., Hu, Y.: Cutting and Packing Problems. Springer Tokyo (2025)