

## COBEM-2023-0816

### A methodological proposal for the optimal solution of the 2-DPLP problem

**Fernanda de Oliveira Ferreira\***

fernanda\_o\_ferreira@hotmail.com

**João Victor Zanoni\***

jvzanoni111@gmail.com

**Aline Souza Teodoro\*\***

line\_tsa@hotmail.com

**Leonardo Mejia Rincon\*\***

leonardo.mejia.rincon@ufsc.br

\*Undergraduate program in Automation, Control, and Computation Engineering (CAC) Federal University of Santa Catarina - Campus Blumenau. Santa Catarina, Brazil.

\*\*Graduate Program in Textile Engineering (PGETEX) Federal University of Santa Catarina - Campus Blumenau. Blumenau, Santa Catarina, Brazil

**Abstract.** *The Pallet Loading Problem (PLP) constitutes a significant focus in the field of industrial engineering, with wide-ranging applications in multiple industries such as automated palletization and robotics. To solve the PLP, a variety of algorithms and techniques have been used, including heuristics, meta-heuristics, mathematical programming, and artificial intelligence. Palletization, in broad terms, pertains to efficiently loading and arranging products onto pallets for either transportation or storage purposes. Nevertheless, this represents a multifaceted challenge, given the wide array of factors that must be meticulously assessed to achieve an optimal solution. Parameter selection contributes to numerous PLP variations, with the two-dimensional Distributor's Pallet Loading Problem (2-DPLP) being one of them. To solve this problem, this paper proposes a novel methodology devised to optimize the distribution of items on a two-dimensional pallet. Our proposed strategy is separated into two main stages. The first consists of minimizing an objective function based on the definition of a computational tensor through the application of an evolutionary computation algorithm, and the second one consists of overlapping elimination through a new proposed algorithmic approach. Each tensor layer in our model corresponds to an item destined for placement on the pallet. This methodology has notable benefits, including the easy identification of overlapping items on the pallet. Our findings suggest that the computational solution optimizes pallet space utilization and provides precise placement information that can be used by a robotic system in order to execute the palletizing process.*

**Keywords:** *Two-dimensional pallet loading problem, Evolutionary computation, Tensor, Methodological proposal, Robotic System.*

#### 1. INTRODUCTION

The Pallet Loading Problem (PLP) stands as a critical research area in industrial engineering, with its influence pervading a variety of industries (Arun Prasad and Krishnakumar, 2021). Notably, its impact extends to logistics and supply chain management domains, where operational success hinges upon the efficient spatial and resource utilization (Ahn *et al.*, 2015). The role of logistics in the textile industry cannot be understated, spanning the entire supply chain from sourcing manufacturing inputs to the distribution of the final product.

A common simplification of the PLP is its two-dimensional variant, which revolves around packing identical rectangular items (boxes) onto a larger, rectangular pallet. Items rotation is permitted in steps of 90 degrees, with the caveat that only orthogonal packages are allowed. The overarching goal for this simplification is to maximize the number of packed items onto the pallet while precluding any overlap (Lu and Cha, 2014).

To solve this problem, this paper proposes a new methodology aiming at an optimal distribution of items in a two-dimensional pallet. This methodology aims to minimize an objective function based on a tensorial structure where each layer of the tensor represents an object in the interface. The dimension of each tensor's layer is equal to the dimension of the pallet and, each tensor layer corresponds to an item that will be placed on the pallet. To minimize the proposed objective function, the Evolution Differential algorithm of Storn and Price was used. Our approach provides the significant benefit of allowing straightforward verification of pallet overlap, along with immediate rejection of overlapping cases. Upon the methodology's full development, we anticipate our computational solution to significantly enhance pallet space utilization and provide precise placement information to a robotic system.

Historically, the Distributor's Pallet Loading Problem (DPLP) has been the focal point of numerous studies, yielding an array of methods designed to address this challenge. Various procedures have been developed to obtain the most

effective arrangement of boxes on a pallet. Such strategies encompass methods like Higher-Order Non-Guillotine Block Heuristics (Arun Prasad and Krishnakumar, 2021), meta-heuristic approaches (Hopper and Turton, 2001), and even hybrid reinforcement learning algorithms (Fang *et al.*, 2023). By using those techniques, numerous studies have addressed pallet loading and packing problems, focusing on efficient space utilization and optimization. They offer valuable insights applicable to the 2-Dimensional Distributor's Pallet Loading Problem (2-DPLP).

For instance, (Gzara *et al.*, 2020) explored three-dimensional pallet loading with practical constraints, using clustering and sorting techniques. (Ribeiro and Lorena, 2007) applied Lagrangean relaxation and column generation to solve a similar problem. The Pallet Loading Problem (PLP) has also been investigated using an exact depth-first search algorithm introducing novel efficiency-improving concepts (Bhattacharya *et al.*, 1998), while (Nelien, 1995) proposed an upper bound method based on structural constraints. Optimizing pallet loading and space utilization within the framework of multiple containers and customer order fulfillment was addressed by (Gimenez-Palacios *et al.*, 2023), and (Morabito *et al.*, 2000) focused on maximizing distribution efficiency in truck loading. Loading irregular and non-homogeneous items onto pallets has been a challenging area addressed through heuristic approaches and exact algorithms (Bischoff *et al.*, 1995; Mascarenhas, 2005).

Also, several studies have aimed at optimizing pallet loading patterns to maximize packed items while minimizing transfer operations, applying a range of techniques, including heuristics and graph-theory-based models (Dowland, 1990). (Dyckhoff, 1990) provided a comprehensive typology for cutting and packing problems. Further studies have shown the relevance of rectangle packing and irregular nesting problems to pallet loading. These have employed heuristics and rasterization methods to minimize overlap and ensure efficient containment of irregular items (Scheithauer and Sommerweib, 1998; Terno *et al.*, 2000; Martinez-Sykora *et al.*, 2017). Lastly, loading multiple containers while meeting customer orders with minimal truck usage, and efficiently filling containers with product boxes were explored (Alonso *et al.*, 2017; Sheng *et al.*, 2016).

For the development of this study, it is important to consider an approach that is connected with both scientific research and technological research. In this sense, in addition to scientific studies, it is believed that technological research, such as the ones conducted by the authors Granemann (2020), Ikenohuchi (2022), and Pegoretti (2021), who proposed robotic palletizing systems for different specific products, such as dairy products, rotors, and motor carcasses, respectively, is necessary for a comprehensive understanding of the subject. In their studies, they highlight the benefits of implementing a robotic system and the challenges faced when adapting their solutions to physical constraints, such as collision prevention. The main objective in all cases is to automate the palletizing process, replacing manual and repetitive tasks through the use of industrial robots.

For the development of this study, it is important to consider an approach that is connected with both scientific research and technological research. In this sense, in addition to scientific studies, it is believed that technological research, such as the ones conducted by the authors Granemann (2020), Ikenohuchi (2022), and Pegoretti (2021), who proposed robotic palletizing systems for different specific products, such as dairy products, rotors, and motor carcasses, respectively, is necessary for a comprehensive understanding of the subject. In their studies, they highlight the benefits of implementing a robotic system and the challenges faced when adapting their solutions to physical constraints, such as collision prevention. The main objective in all cases is to automate the palletizing process, replacing manual and repetitive tasks through the use of industrial robots.

In summary, all the studies presented in this section contributed to the broader understanding of pallet loading and packing problems, sharing valuable insights, optimization techniques, and practical solutions relevant to the 2-DPLP, providing a solid foundation for further research in the field.

## 2. DESCRIPTION OF THE PROBLEM

The main aim of this study is to accurately determine the placement of a set of boxes on a pallet while simultaneously avoiding overlap. This information would then be used for a robotic palletizing system in order to place the boxes on the pallet. It is assumed that the boxes have a uniform height. From a technical point of view, this research looks to answer the question: "Given the dimensions of a pallet and a set of rectangular boxes, what are the precise placements for the boxes on the pallet such that overlap is avoided and the vacant space on the pallet is minimized?" The input for our PLP problem is given as a set of variables ( $L$ ,  $W$ ,  $x_i$ ,  $y_i$ ,  $h_i$ ). Where  $L$  is the length of the pallet;  $W$  is the width of the pallet;  $(x_i, y_i)$  is the ordered pair representing the coordinate position of the box and  $h_i$  is the orientation of the box.  $h_i = 1$  means vertical orientation and  $h_i = 0$  means horizontal orientation.

## 3. PROPOSED METHODOLOGY

This section introduces a novel method to address the aforementioned problem, involving the arrangement of a set of boxes on a pallet in a manner that minimizes the unoccupied space. Though initially the problem can be perceived as unique, the proposed method bifurcates it into two interrelated questions: First, where should boxes be positioned on the pallet to maximize space utilization? And second, how can we circumvent the potential overlapping of the placed

boxes? Given the dual nature of the research question, the proposed methodology is likewise divided into two phases. The initial phase aims to discover the optimal position and orientation of a downscaled set of boxes. The subsequent phase endeavors to eliminate possible overlap when the boxes are rescaled to their original size. Both stages hinge on the concept of tensors, where each layer mirrors the dimensions of the pallet and represents a distinct box to be palletized. In the subsections below, there are presented and discussed the stages that compose the proposed method.

### 3.1 Stage 1: Optimum placement of boxes

Optimization is a fundamental field in mathematics and computer science that focuses on finding the best possible solution to a problem within a given set of constraints or objectives. It plays a crucial role in numerous applications across diverse domains, such as engineering, logistics, finance, and data science. Various algorithms have been proposed to solve optimization problems. These algorithms range from traditional mathematical programming methods, such as linear programming and integer programming, to more advanced techniques like evolutionary algorithms, particle swarm optimization, and simulated annealing.

Specifically, in this work, Evolutionary algorithms were chosen to solve the PLP problem once they offer flexibility, global optimization capability, robustness to uncertainties, parallelizability, exploratory capability, and domain independence, making them a powerful choice for solving complex optimization problems across diverse domains.

Evolutionary computing algorithms are a class of optimization techniques inspired by principles from biological evolution and natural selection. These algorithms mimic the process of evolution, iteratively searching for optimal solutions to complex problems. The main idea behind evolutionary computing is the creation of a population of potential solutions, represented as individuals or chromosomes. These individuals undergo genetic operations such as crossover, selection, and mutation, simulating the mechanisms of variation and inheritance found in natural populations. Over successive generations, the best individuals are more likely to survive and produce offspring, while the worse are gradually replaced. This iterative process drives the population towards increasingly better solutions over time.

#### 3.1.1 Objective function

In the field of optimization, every problem can be expressed mathematically by defining its objective function. This objective function plays a crucial role as it quantifies and expresses the specific goal or target that must be achieved in the optimization problem at hand. By evaluating and optimizing this objective function, one can effectively measure how well a particular solution or set of decisions meets the requirements and criteria of the problem. The objective function guides the search for the optimal solution that best aligns with the desired outcome or objective of the optimization problem, ensuring the attainment of the most favorable results.

In this stage, it is aimed to minimize the overall empty area within a pallet of dimensions  $L \times W$ , upon which a collection of  $N$  boxes with dimensions  $a_i \times b_i \quad \forall i = 1 \dots N$  is placed. The optimization problem is formulated through the utilization of a mathematical entity that represents a set of data organized in a multidimensional structure called a tensor. In this stage is used a tensor ( $T$ ) comprising  $N$  layers, where each layer represents a binary matrix denoting the position and orientation of an individual scaled object. The scaling process is proposed in order to reduce the computing cost during the optimization process and It is suggested to use values within the interval 80% to 100% of the original size for each box. Lower values to these proposed could result in a local minimum for the proposed objective function. By adding the layers of the tensor, we obtain a comprehensive map illustrating the distribution of all the placed boxes, thus yielding the total occupied area. Consequently, the total empty area is computed by subtracting the total occupied area  $\left( \sum_{i=1}^N T_i(a_i, b_i, x_i, y_i) \quad \forall i = 1 \dots N \right)$  from the pallet area ( $L \cdot W$ ), as illustrated in Equation 1a. This method offers a precise quantification of the vacant space remaining within the pallet, facilitating the determination of an optimized arrangement of the boxes.

Furthermore, our optimization problem incorporates two distinct sets of constraints to ensure the absence of overlapping between the boxes and to prohibit the placement of any object beyond the pallet's boundaries. Equation 1b rigorously defines the constraint pertaining to overlapping, while equations 1c, 1d, 1e, and 1f introduce penalty terms accounting for instances where boxes cross the borders of the pallet. These constraints effectively safeguard the integrity of the packing arrangement, ensuring that each box occupies a unique and non-overlapping space while remaining within the prescribed confines of the pallet. The utilization of these equations facilitates the establishment of a comprehensive and efficient optimization framework that not only optimizes the empty space but also respects the spatial boundaries and arrangement constraints imposed by the packing problem.

$$\underset{a_i, b_i}{\text{minimize}} \quad L \cdot W - \sum_{i=1}^N a_i \cdot b_i \quad (1a)$$

$$\text{subject to} \quad \max \left( \sum_{i=1}^N T_i(a_i, b_i, x_i, y_i) \right) \leq 1, \quad (1b)$$

$$x_i + a_i \leq L, \quad (1c)$$

$$y_i + b_i \leq W, \quad (1d)$$

$$x_i \geq 1, \quad (1e)$$

$$y_i \geq 1 \quad (1f)$$

### 3.2 Stage 2: Overlap Correction

The optimization process previously discussed culminates in a tensor. Each layer within this tensor is a matrix representing a distinct, scaled object. Areas where the object is present are marked with 1's, while regions of absence are filled with 0's. This situation is illustrated in Figure 1, where the yellow regions designate the areas occupied by the boxes, and the blue regions denote the vacant areas. This visualization is presented for a scenario involving a pallet of dimensions 100 [l.u.] by 100 [l.u.] housing a total of 11 layers (each representing a box). From this optimized tensor, it is feasible to extract the final coordinate position (x, y) for any box within the system. This example illustrates the usefulness of the tensor in order to obtain the final coordinates as an essential feature to describe the position of the object. This information will be lastly used to rearrange the re-scaled objects onto the pallets.

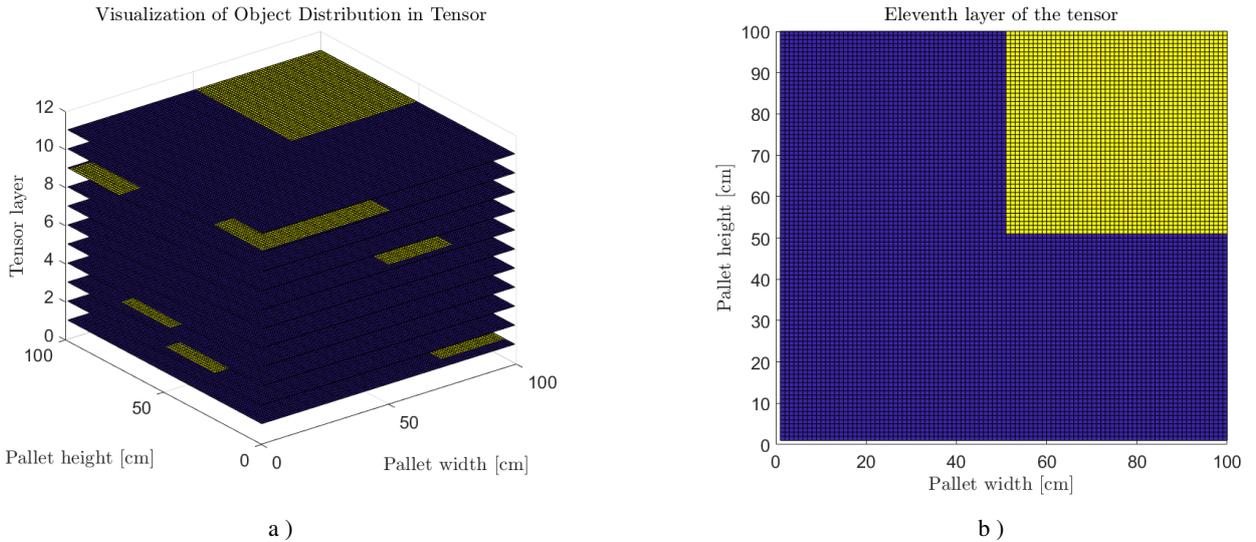


Figure 1. Visualization of Object Distribution in Tensor.

The subsequent phase in this stage needs re-scaling the boxes to their original sizes. This action may potentially lead to overlapping. To rectify this, the algorithm needs the formulation of a virtual pallet to help in the box rearrangement process. This step encompasses a virtual expansion in the height and width of the pallet, resulting in a two-dimensional augmentation of the tensor dimensions. Illustrations of the enlargement process are depicted in Figures 2a and 2b, where the expanded region is highlighted in green. The tensor's expansion affords increased flexibility during the box movement process, thereby facilitating efficient overlap rectification.

Following this, the virtually expanded tensor feeds an overlap correction function, along with the vector of box position coordinates and the desired movement direction for each object (with options being left, right, up, and down). These parameters are crucial in the process of overlap correction and box movements. Hereafter, we present a detailed explanation of the overlap correction function's operation.

Subsequent to the preliminary corrections involving rescaling and virtual expansion, the next phase involves transforming the tensor into a two-dimensional representation, as shown in Figure 2c. A meticulous examination of this figure uncovers that the rescaling of boxes results in multiple overlaps, with the most notorious case manifesting when four boxes overlap. However, it is paramount to emphasize that this outcome pertains to a specific case, as the maximum number of overlaps can change depending on the number of boxes to be placed.

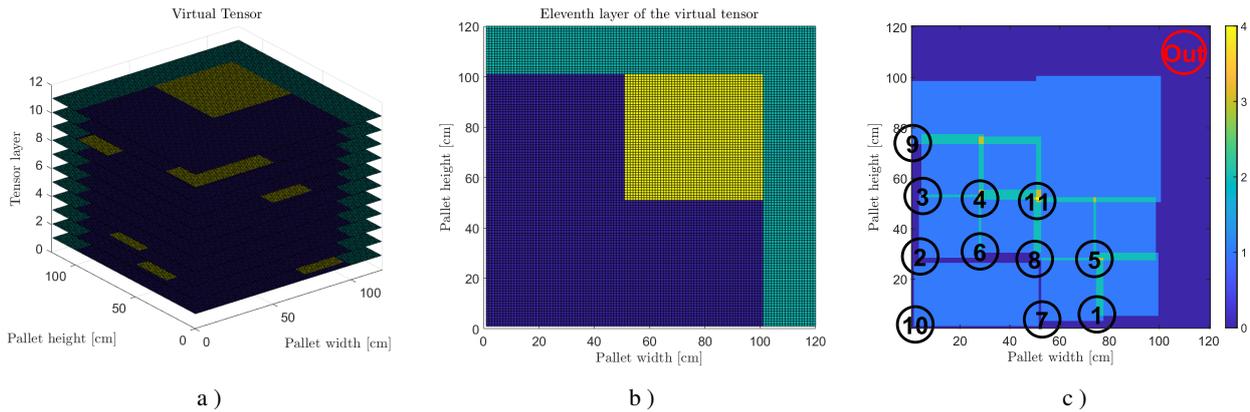


Figure 2. Visualization of Object Distribution in Virtual Tensor.

Each coordinate of interest associated with each object is measured, and the distance vector is created through a simple Euclidean distance operation. The points of interest were defined in such a way that it is impossible for any point to be exactly located in the coordinates (pallet size in  $x+1$ , pallet size in  $y+1$ ). Consequently, the object with the shortest distance from the origin is moved first and, once moved, must not be changed further. Therefore, the invalid distance corresponding to the original position is assigned instead of the new position.

When a moving operation is applied, all the boxes are tried to be moved, but once at a time. For example, choosing the option to move to the right, the algorithm tries to move all the boxes to the right. But, it can only move each of them if it does not cause a new overlap. This routine will move the boxes up to the limits of the virtual pallet or up to a new overlap being identified. Figures 3a to 3f indicate the sequence of the complete cycle of movement to the right, with some jumps to facilitate the representation.

Note that in the example the first object to be moved is shown in Figure 3a, but it does not move, as it generates a new overlap to the right side when trying to move. In Figure 3b it is possible to observe that moving the object in the desired direction will not cause overlap, so it will be moved until it finds the limits of the virtual pallet as can be observed in Figure 3c. The same situation occurs for the object shown in Figure 3d. As other movements are not allowed, the process ends. Finally, the complete process for this routine which started at the instant shown in Figure 3a ends at the instant shown in Figure 3f.

Similar to the last case where a box was moved to the right, it is also possible to perform left, up, and down movements through a “*shift*” function. The shift function accepts the following inputs: a) the index of the layer, b) the matrix corresponding to that layer, c) the tensor’s two-dimensional representation, d) the tensor itself, and e) the specified operation. There exist four operations, each limited by a termination condition established by the boundaries of the virtual pallet. These operations are as follows: shift left (termination condition at minimum  $x$ -size of the pallet), shift right (termination condition at maximum  $x$ -size of the pallet), shift up (termination condition at maximum  $y$ -size of the pallet), and shift down (termination condition at the maximum  $y$ -size of the pallet). These conditions avoid the objects cross the pallet boundaries, thereby precluding their displacement beyond the pallet’s dimensions.

The overlap’s correction through the shift function occurs in a sequential and cyclical way, moving iteratively the objects along the four possible directions, the number of iterations is arbitrarily determined. This approach guarantees that all boxes are adjusted appropriately, resulting in an overlap-free final representation.

After the displacement of the boxes, a series of iterative updates are conducted for each object. These updates encompass refreshing the object’s layer, denoting with a zero where it is absent, and with a one where it is present. Other updates include the revision of the tensor’s two-dimensional representation, updating the vector representing object positions, and designating invalid distances in the distance vector. All these updates are obtained from the output of the shift function.

## 4. RESULTS

### 4.1 Algorithmic approach to the proposed methodology

In this section, the results of five case studies are presented. The pallet’s dimensions were set at 100 *cm* by 100 *cm* and were kept constant for all the study cases. Three different types of boxes were used: **Box 1**, with dimensions of 25 *cm* by 25 *cm*; **Box 2**, with dimensions of 25 *cm* by 50 *cm* and **Box 3**, with dimensions of 50 *cm* by 50 *cm*. All boxes have the same height.

The first case study is the simplest, in which the positions of 16 type-1 boxes are optimized. Initially, the dimensions of these boxes are reduced by 80%, that is, they become 20 *cm* by 20 *cm*. Then, the evolutionary computation algorithm is tasked with finding the optimal positions for these 16 elements without overlap. With this preliminary result (see Figure

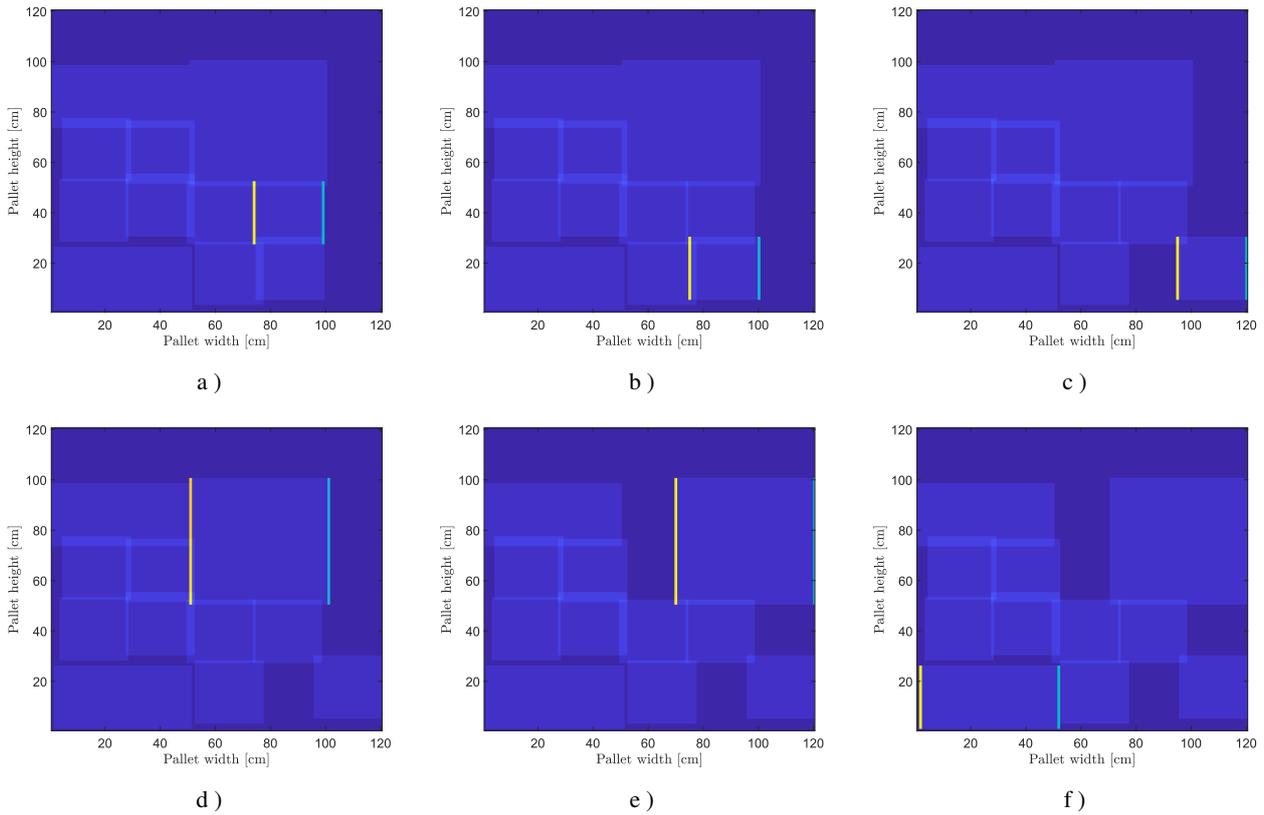


Figure 3. Visualization of Object Distribution in Virtual Tensor.

4a), the boxes are scaled back from 80% to 100%, generating overlap, as can be seen in Figure 4b. Thus, it is necessary to correct this overlap using the overlap correction algorithm, obtaining the final result, as can be seen in Figure 4c.

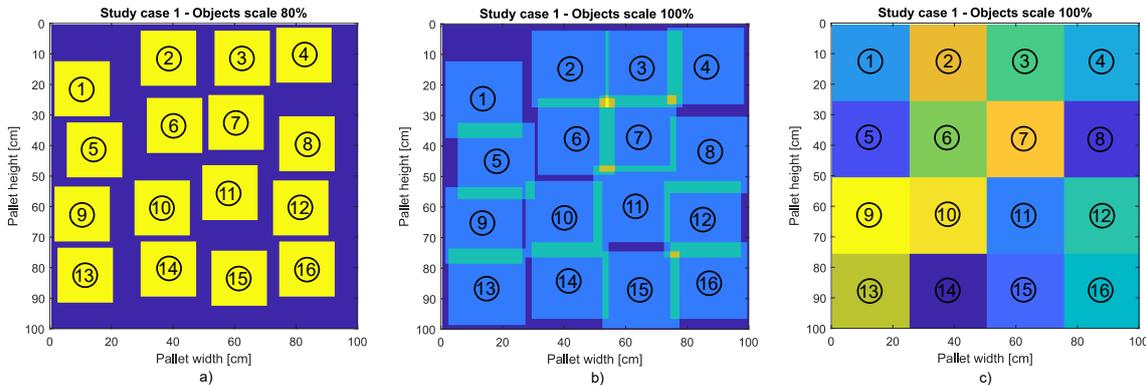


Figure 4. Study case 1 - 16 boxes of type 1.

In the second case study, type-2 boxes were added and were forced to be oriented vertically. In this case, 8 type-1 boxes along with 4 type-2 boxes in a vertical orientation were used. For this specific case, only the type-1 boxes undergo an 80% scaling, while the type-2 boxes maintain 100% of their original dimensions. The optimal positioning results, achieved with no overlaps, are demonstrated in Figure 5a. After the optimization process, the rescaling process is executed, inevitably causing overlap as seen in Figure 5b. Lastly, the overlap correction process is applied and the overlap is solved, The final disposition is shown in Figure 5c.

The third case study is a variation of the second study case, where the same number of boxes is considered, but now, the 4 type-2 boxes are forced to be horizontally orientated. The result of the optimization can be observed in Figure 6a. Next, the result presented in Figure 6a undergoes the rescaling process, generating the overlaps presented in Figure 6b, which are corrected by the overlap correction algorithm, resulting in the final result presented in Figure 6c.

In the fourth case study, a variation combining the conditions of the second case with the third case is considered. In this case, 8 type-1 boxes along with 4 type-2 boxes were used, but this time, the algorithm freely choose the optimum

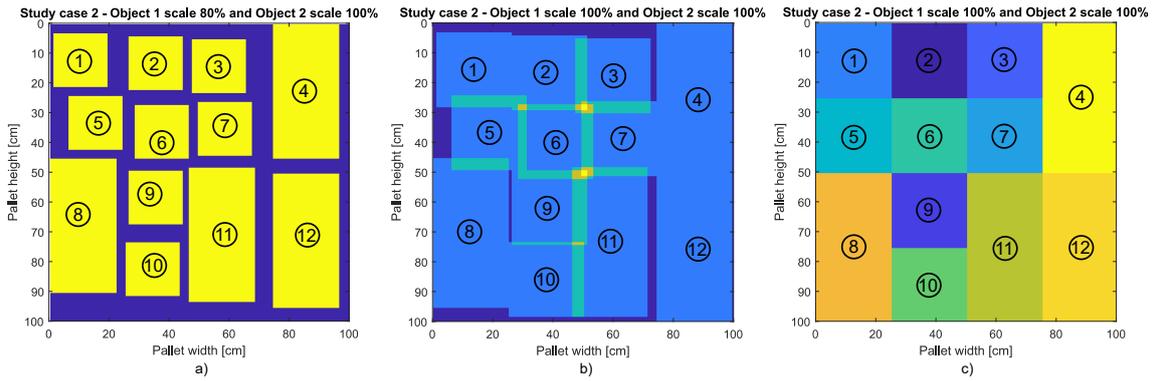


Figure 5. Study case 2 - 12 boxes: 8 boxes of type 1 and 4 boxes of type 2 in a horizontal orientation.

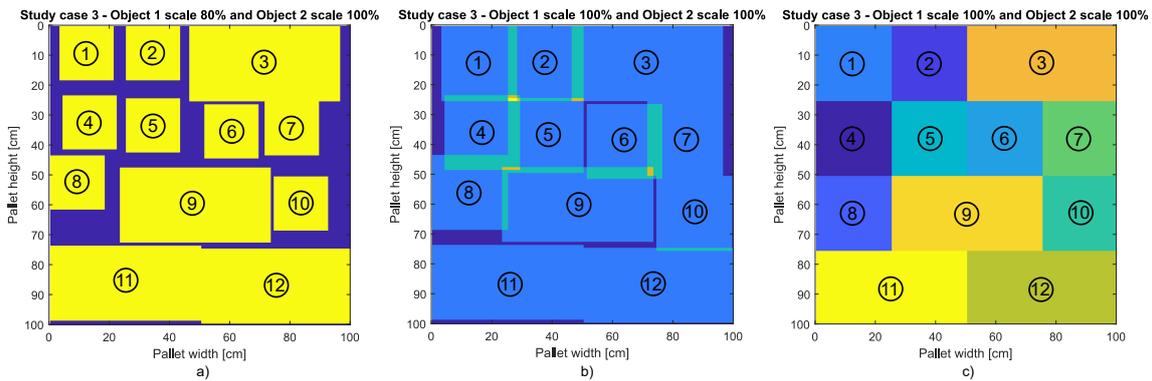


Figure 6. Study case 3 - 12 boxes: 8 boxes of type 1 and 4 boxes of type 2 in a vertical orientation.

orientation for the type-2 boxes. Here, the type-1 boxes undergo an 80% scaling while type-2 boxes undergo a 90% scaling. The initial optimization is presented in Figure 7a. After applying a resizing process, overlap occurs, as can be observed in Figure 7b. The overlap is easily corrected by the overlap correction algorithm as is presented in Figure 7c.

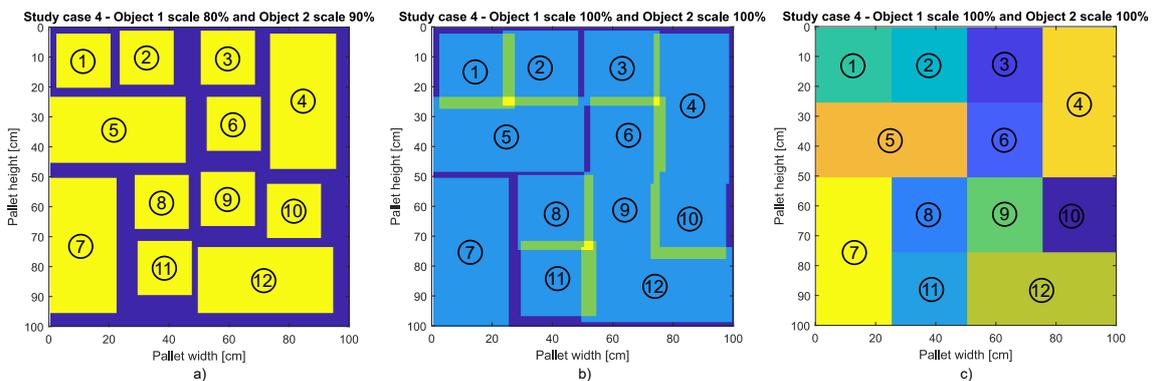


Figure 7. Study case 4 - 12 boxes: 8 boxes of type 1, 2 boxes of type 2 in a horizontal orientation and 2 boxes of type 2 in a vertical orientation.

The last case study uses 8 type-1 boxes, 2 type-2 boxes, and 1 type-3 box. Here, the type-1 boxes undergo an 80% scaling and type-2 and type-3 boxes undergo a 90% scaling before the optimization process. In Figure 8a, we observe the optimal solution for this case. Overlap occurs when the rescaling is applied and can be seen in Figure 8b. Finally, the results obtained from the overlap correction are presented in Figure 8c.

To end this section, the spent time on each simulation is shown in Table 1. This table presents individually the time taken for the application of the optimization algorithm (Stage 1), for the overlap correction algorithm (Stage 2), and finally presents the total spent time (Stage 1 + Stage 2).

#### 4.2 Robotized System

An inherent result of the introduced methodology is the necessity to place the boxes on a pallet once their final positions have been determined. It is suggested to utilize a robot manipulator for this operation, given its ability to place the boxes

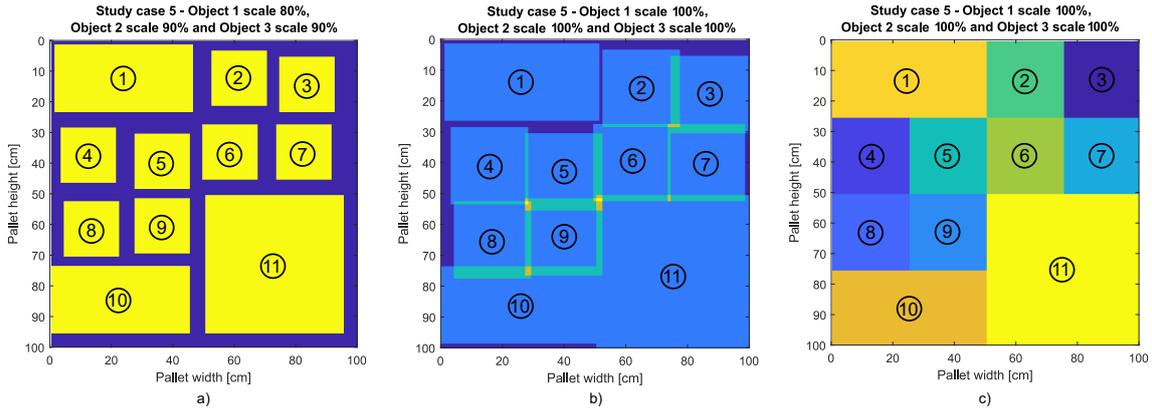


Figure 8. Study case 5 - 11 boxes: 8 boxes of type 1, 2 boxes of type 2 in a horizontal orientation and 1 box of type 3.

Table 1. Results Time Summary.

Study cases	Simulation time for Stage 1 (s)	Simulation time for Stage 2 (s)	Total time (s)
Study case 1	14.20	3.36	17.57
Study case 2	19.78	3.31	23.09
Study case 3	21.78	3.14	24.92
Study case 4	23.50	3.12	26.62
Study case 5	24.21	2.87	27.09

accurately. Verification of the resultant configuration is facilitated through the employment of a simulator. In this context, the CoppeliaSim simulator has been specifically employed. This simulator, and others like it, is widely accepted within the robotics and scientific communities. They provide virtual environments mirroring real-world scenarios, enabling the testing of algorithms and designs without the need for physical prototypes. CoppeliaSim, in particular, is acclaimed for its multifaceted features, user-friendly interface, and comprehensive library of pre-assembled models. It not only expedites research with its precise physics simulation and sensor emulation but also curtails costs and promotes innovation within the robotics field.

The deployment of CoppeliaSim facilitates the construction of a simulated environment that can interact with the developed algorithms. This results in the determination of the palletizing coordinates for the boxes on a pallet and the simulation of an industrial robot. Initially, a robot model is established within an appropriate programming environment, considering its physical dimensions, movement limitations, and joint structures. Leveraging this model, inverse kinematics is implemented, enabling the acquisition of necessary joint configurations to reach the pallet's predetermined coordinates. Once the inverse kinematics algorithm and trajectory planning have been configured, the simulated system is prepared to conduct the comprehensive simulation.

Throughout the simulation, the industrial robot follows the planned trajectories and executes the palletizing operations based on the coordinates prescribed by the algorithm. This simulated process offers an evaluation of the algorithm's efficiency across varied scenarios and allows for the optimization of its performance before implementation on the physical robot. The intent is to integrate the algorithm into an environment analogous to that depicted in Figure 9.

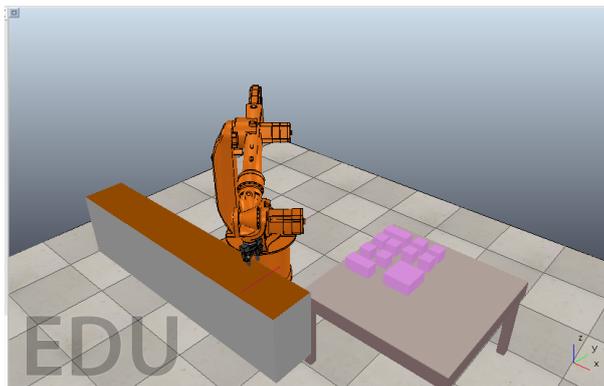


Figure 9. Industrial Robot Simulation Environment.

## 5. DISCUSSIONS

In the conducted experiments, the range of simulated boxes extended from 11 to 16 boxes. All cases studied were resolved within an acceptable interval of time, successfully fulfilling the goal of precisely ascertaining the positions of a collection of boxes to optimally fill the pallet space while avoiding overlap.

These experiments attest to the robustness of the solution in the face of diversity among the elements. Specifically, it maintains efficiency even when the elements present significant dimensional disparities.

The experiment comprising the largest number of boxes involved units of identical dimensions that entirely occupied the available space. Notably, this setup resulted in the shortest total time. Conversely, the scenario with fewer elements, in which one box encompassed a quarter of the total area, registered a longer total time due to the absence of size reduction for that particular object during the initial stage. However, this case marked the shortest duration in the second stage, underscoring the crucial role of the second stage in total time reduction and enhancing solution flexibility during the initial stage. Cases 3 and 4 recorded similar durations.

The bifurcation of the proposed methodology into two stages enables the reduction of time and the amalgamation of the optimal aspects of both stages. The first stage facilitates the inclusion of variables such as orientation and position and broadens the solution to encompass possible new variables like object weight and center of mass. The aim of this stage is to heighten the precision of the final fit and ensure the problem's solvability. The second stage constitutes a straightforward and efficient process that aids in the reduction of the time required to reach the objective.

## 6. CONCLUSIONS

The purpose of this investigation was to precisely ascertain the positioning of a collection of boxes within a two-dimensional space, devoid of overlapping, intended for application in a robotic system equipped with an industrial manipulator. The results indicated the successful attainment of this objective. Important characteristics of the proposed methodology are reflected in the flexibility and adaptability of the devised algorithm, enabling the introduction of new decision variables and efficiently rendering preliminary outcomes. The bifurcation of the proposed methodology into two stages also contributed to a significant reduction in time.

Despite the successful results, the study's potential expansion warrant consideration. To enhance the box positioning process further, the introduction of new decision variables such as the center of mass and the weight of the boxes is proposed. Moreover, an increase in the diversity of the types and sizes of boxes considered, incorporating scenarios with true-to-life dimensions, is suggested. These refinements could expand the algorithm's applicability and yield additional insights.

In summary, the proposed methodology fulfilled its goal of determining the precise positioning of boxes within a two-dimensional non-overlapped space for application in a robotic system. The research highlighted the flexibility and adaptability of the algorithm, along with the time efficiencies derived from the process division. Although potential improvements for the proposed methodology there exist, the current study lays a robust groundwork for subsequent investigations, facilitating the application of the algorithm in more complex and realistic scenarios.

## 7. ACKNOWLEDGEMENTS

The authors would like to extend their sincere gratitude to the Graduate Program in Textile Engineering (PGETEX) and the undergraduate program in Automation, Control, and Computation Engineering (CAC), both at the Federal University of Santa Catarina - Campus Blumenau for their invaluable support and collaboration in this project. We are especially grateful for their commitment to training and nurturing future masters and engineering professionals, which has greatly contributed to the success of this endeavor.

## 8. REFERENCES

- Ahn, S., Park, C. and Yoon, K., 2015. "An improved best-first branch and bound algorithm for the pallet-loading problem using a staircase structure". *Expert Systems with Applications*, Vol. 42, No. 21, pp. 7676–7683. ISSN 0957-4174. doi:<https://doi.org/10.1016/j.eswa.2015.05.045>.
- Alonso, M., Alvarez-Valdes, R., Iori, M., Parreño, F. and Tamarit, J., 2017. "Mathematical models for multicontainer loading problems". *Omega*, Vol. 66, pp. 106–117. ISSN 0305-0483. doi:<https://doi.org/10.1016/j.omega.2016.02.002>.
- Arun Prasad, S. and Krishnakumar, P., 2021. "Higher order block heuristics for 2d pallet loading problems with multiple box inputs". *Materials Today: Proceedings*, Vol. 46, pp. 4625–4633. ISSN 2214-7853. doi:<https://doi.org/10.1016/j.matpr.2020.10.280>. International Conference on Advances in Materials and Manufacturing Applications.
- Bhattacharya, S., Roy, R. and Bhattacharya, S., 1998. "An exact depth-first algorithm for the pallet loading problem". *European Journal of Operational Research*, Vol. 110, No. 3, pp. 610–625. ISSN 0377-2217. doi:

[https://doi.org/10.1016/S0377-2217\(97\)00272-5](https://doi.org/10.1016/S0377-2217(97)00272-5).

- Bischoff, E., Janetz, F. and Ratcliff, M., 1995. "Loading pallets with non-identical items". *European Journal of Operational Research*, Vol. 84, No. 3, pp. 681–692. ISSN 0377-2217. doi:[https://doi.org/10.1016/0377-2217\(95\)00031-K](https://doi.org/10.1016/0377-2217(95)00031-K). Cutting and Packing.
- Dowland, K.A., 1990. "Efficient automated pallet loading". *European Journal of Operational Research*, Vol. 44, No. 2, pp. 232–238. ISSN 0377-2217. doi:[https://doi.org/10.1016/0377-2217\(90\)90358-I](https://doi.org/10.1016/0377-2217(90)90358-I). Cutting and Packing.
- Dyckhoff, H., 1990. "A typology of cutting and packing problems". *European Journal of Operational Research*, Vol. 44, No. 2, pp. 145–159. ISSN 0377-2217. doi:[https://doi.org/10.1016/0377-2217\(90\)90350-K](https://doi.org/10.1016/0377-2217(90)90350-K). Cutting and Packing.
- Fang, J., Rao, Y., Zhao, X. and Du, B., 2023. "A hybrid reinforcement learning algorithm for 2d irregular packing problems". *Mathematics*, Vol. 11, No. 2. ISSN 2227-7390. doi:10.3390/math11020327.
- Gimenez-Palacios, I., Alonso, M., Alvarez-Valdes, R. and Parreño, F., 2023. "Multi-container loading problems with multidrop and split delivery conditions". *Computers Industrial Engineering*, Vol. 175, p. 108844. ISSN 0360-8352. doi:<https://doi.org/10.1016/j.cie.2022.108844>.
- Granemann, A.L., 2020. *Development of a robotic system for packaging pots for a dairy products company*. Undergraduate thesis, Federal University of Santa Catarina, Blumenau. Presented to the Federal University of Santa Catarina as part of the requirements for the degree of Bachelor in Control and Automation Engineering.
- Gzara, F., Elhedhli, S. and Yildiz, B.C., 2020. "The pallet loading problem: Three-dimensional bin packing with practical constraints". *European Journal of Operational Research*, Vol. 287, No. 3, pp. 1062–1074. ISSN 0377-2217. doi:<https://doi.org/10.1016/j.ejor.2020.04.053>.
- Hopper, E. and Turton, B., 2001. "An empirical investigation of meta-heuristic and heuristic algorithms for a 2d packing problem". *European Journal of Operational Research*, Vol. 128, No. 1, pp. 34–57. ISSN 0377-2217. doi:[https://doi.org/10.1016/S0377-2217\(99\)00357-4](https://doi.org/10.1016/S0377-2217(99)00357-4).
- Ikenohuchi, J.Y.B., 2022. *Programming and simulation of an anthropomorphic robot for palletizing stamped rotors*. Undergraduate thesis, Federal University of Santa Catarina, Blumenau. Presented to the Federal University of Santa Catarina as part of the requirements for the degree of Bachelor in Control and Automation Engineering.
- Lu, Y. and Cha, J., 2014. "A fast algorithm for identifying minimum size instances of the equivalence classes of the pallet loading problem". *European Journal of Operational Research*, Vol. 237, No. 3, pp. 794–801. ISSN 0377-2217. doi:<https://doi.org/10.1016/j.ejor.2014.02.008>.
- Martinez-Sykora, A., Alvarez-Valdes, R., Bennell, J., Ruiz, R. and Tamarit, J., 2017. "Matheuristics for the irregular bin packing problem with free rotations". *European Journal of Operational Research*, Vol. 258, No. 2, pp. 440–455. ISSN 0377-2217. doi:<https://doi.org/10.1016/j.ejor.2016.09.043>.
- Mascarenhas, W.F., 2005. "Two aspects of the pallet loading problem". *Electronic Notes in Discrete Mathematics*, Vol. 19, pp. 381–387. ISSN 1571-0653. doi:<https://doi.org/10.1016/j.endm.2005.05.051>. Proceedings of GRACO2005.
- Morabito, R., Morales, S.R. and Widmer, J.A., 2000. "Loading optimization of palletized products on trucks". *Transportation Research Part E: Logistics and Transportation Review*, Vol. 36, No. 4, pp. 285–296. ISSN 1366-5545. doi:[https://doi.org/10.1016/S1366-5545\(00\)00003-X](https://doi.org/10.1016/S1366-5545(00)00003-X).
- Nelien, J., 1995. "How to use structural constraints to compute an upper bound for the pallet loading problem". *European Journal of Operational Research*, Vol. 84, No. 3, pp. 662–680. ISSN 0377-2217. doi:[https://doi.org/10.1016/0377-2217\(95\)00030-T](https://doi.org/10.1016/0377-2217(95)00030-T). Cutting and Packing.
- Pegoretti, F.R., 2021. *Programming and simulation of a robotic cell for the manipulation of electric motor carcasses*. Undergraduate thesis, Federal University of Santa Catarina, Blumenau. Department of Control and Automation Engineering and Computing, Technological Center, Exact Sciences and Education.
- Ribeiro, G.M. and Lorena, L.A.N., 2007. "Lagrangian relaxation with clusters and column generation for the manufacturer's pallet loading problem". *Computers Operations Research*, Vol. 34, No. 9, pp. 2695–2708. ISSN 0305-0548. doi:<https://doi.org/10.1016/j.cor.2005.10.008>.
- Scheithauer, G. and Sommerweib, U., 1998. "4-block heuristic for the rectangle packing problem". *European Journal of Operational Research*, Vol. 108, No. 3, pp. 509–526. ISSN 0377-2217. doi:[https://doi.org/10.1016/S0377-2217\(96\)00359-1](https://doi.org/10.1016/S0377-2217(96)00359-1).
- 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*, Vol. 252, No. 3, pp. 728–736. ISSN 0377-2217. doi:<https://doi.org/10.1016/j.ejor.2016.01.025>.
- Terno, J., Scheithauer, G., Sommerweiß, U. and Riehme, J., 2000. "An efficient approach for the multi-pallet loading problem". *European Journal of Operational Research*, Vol. 123, No. 2, pp. 372–381. ISSN 0377-2217. doi:[https://doi.org/10.1016/S0377-2217\(99\)00263-5](https://doi.org/10.1016/S0377-2217(99)00263-5).

## 9. RESPONSIBILITY NOTICE

The authors are solely responsible for the printed material included in this paper.