Automating Bin Packing: A Layer Building Matheuristics for Cost Effective Logistics

Giulia Tresca, *Graduate Student Member, IEEE*, Graziana Cavone, *Member, IEEE*, Raffaele Carli, *Senior Member, IEEE*, Antonio Cerviotti, and Mariagrazia Dotoli, *Senior Member, IEEE*

***Abstract*— In this paper, we address the problem of automating the definition of feasible pallets configurations. This issue is crucial for the competitiveness of logistic companies and is still one of the most difficult problems in internal logistics. In fact, it requires the fast solution of a three-dimensional Bin Packing Problem (3D-BPP) with additional logistic specifications that are fundamental in real applications. To this aim, we propose a matheuristics that, given a set of items, provides feasible pallets configurations that satisfy the practical requirements of items' grouping by logistic features, load bearing, stability, height homogeneity, overhang as well as weight limits, and robotized layer picking. The proposed matheuristics combines a mixed integer linear programming (MILP) formulation of the 3D-Single Bin-Size BPP (3D-SBSBPP) and a layer building heuristics. In particular, the feasible pallets configurations are obtained by sequentially solving two MILP sub-problems: the first, given the set of items to be packed, aims at minimizing the unused space in each layer and thus the number of layers; the latter aims at minimizing the number of shipping bins given the set of layers obtained from the first problem. The approach is extensively tested and compared with existing approaches. For its validation we use both realistic data-sets drawn from the literature and real data-sets, obtained from an Italian logistics leader. The resulting outcomes show the effectiveness of the method in providing high-quality bin configurations in short computational times.**

***Note to Practitioners*—This work is motivated by the intention of facilitating the transition from Logistics 3.0 to Logistics 4.0 by providing an effective tool to automate bin packing, suitable for automated warehouses. On the one hand, the proposed technique provides stable and compact bin configurations in less than half a minute per bin on average, despite the high computational complexity of the 3D-SBSBPP. On the other hand, the approach allows to consider compatibility constraints for the items (e.g., final customer and category of the items), and the use of robotized layer picking in automated warehouses. In effect, layers composed by only one type of items (i.e., monoitem layers) can be directly picked and placed on the pallet by a robotic arm without the intervention of any operator. Consequently, the adoption of this approach in warehouses could drastically improve the efficiency of the packing process.**

Manuscript received 24 December 2021; revised 30 March 2022; accepted 5 May 2022. Date of publication 3 June 2022; date of current version 5 July 2022. This article was recommended for publication by Associate Editor L. Moench and Editor F.-T. Cheng upon evaluation of the reviewers' comments. *(Corresponding author: Graziana Cavone.)*

Giulia Tresca, Graziana Cavone, Raffaele Carli, and Mariagrazia Dotoli are with the Department of Electrical and Information Engineering, Politecnico di Bari, 70125 Bari, Italy (e-mail: giulia.tresca@poliba.it; graziana.cavone@poliba.it; raffaele.carli@poliba.it; mariagrazia.dotoli@poliba.it).

Antonio Cerviotti is with Elettric80, 42030 Reggio Emilia, Italy (e-mail: cerviotti.a@elettric80.it).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASE.2022.3177422.

Digital Object Identifier 10.1109/TASE.2022.3177422

***Index Terms*— Logistics 4.0, bin packing, pallets configuration, matheuristics.**

I. Introduction

The industrial sector is experiencing the so-called fourth industrial revolution, based on the pervasive use of the enabling technologies of *Industry 4.0* [1] to increase productivity and quality standards of production processes [2]. In this context, the novel concept of *Logistics 4.0* has been introduced, aiming at adapting the general *Industry 4.0* pillars to logistics (e.g., automated warehouse, digitization of freight and information flows, and transport tracing) [3]–[8].

Among the various internal logistic activities, the packing of items on pallets is one of the most complex and resource consuming. It requires the resolution of the three-dimensional Bin Packing Problem (3D-BPP) with additional logistic specifications that ensure the stability and safety of the cargo. Its automatic solution can allow the warehouse manager to rapidly obtain feasible configurations of pallets with respect to specific cost functions and packing constraints, to reduce the number of composed pallets and the time required for their packing, and to increase the throughput of the company. Although commercial Warehouse Management Systems (WMSs) offer a large variety of functionalities for the management of internal logistics, effective algorithms that automate the definition of feasible pallets configurations are still lacking. In particular, the available commercial solutions mainly offer 3D graphical tools that allow simulating the manual composition of items into bins from a visual perspective only. In addition, in the related literature, as will be detailed in Section II, contributions to the definition and resolution of the 3D-BPP for internal logistic applications are limited, especially when compared with their two-dimensional counterparts, and they only include a restricted set of constraints that partially represent the business rules adopted in real logistic systems. Furthermore, the few existing literature contributions mainly address the 3D-BPP for simplified simulation scenarios and a recent survey highlights the lack of realistic benchmark data-sets [9].

With the aim of facilitating the transition from Logistics 3.0 to Logistics 4.0, in this paper we propose a matheuristic algorithm [10] that can be integrated in the WMS to automatically and efficiently provide feasible pallets configurations. The proposed matheuristics allows defining feasible pallets configurations while taking into account the practical requirements of items' grouping by logistic features, load bearing, stability, height homogeneity, overhang as well as weight limits, and robotized layer picking. The method consists in a layer building heuristics that is based on the sequential solution of two Mixed Integer Linear Programming (MILP) sub-problems: the former, given the set of items to be packed, aims at minimizing the unused space in each layer and thus the number of layers; the latter aims at minimizing the number of shipping bins given the set of layers obtained from the first problem resolution.

We highlight that the proposed matheuristics allows packing homogeneous and heterogeneous items into multiple single sized bins and properly including particular logistic business rules, such as the *robotized layer picking* (i.e., the picking of entire layers composed by only one type of item by robotic manipulators) and the grouping of items by specific logistic features (i.e., delivery number, identification number, and category of product). The matheuristics is extensively tested to evaluate its performance in terms of quality of results and computational efficiency. Specifically, the first set of tests compares the outcomes and computational time of the 3D-Single Bin-Size BPP (3D-SBSBPP) with the ones of the proposed algorithm including only geometric, weight, and overhang constraints. The second set of tests provides a comparison between our layer-building matheuristics and the literature algorithm proposed in [11], which is one of the most recent and most complete matheuristics in terms of logistic requirements. The third regards the performance analysis of the layer-building algorithm with a real data-set obtained by an Italian industry leader of the logistic sector. The obtained outcomes show the ability of the method to provide high quality results in short computational times.

The remainder of this paper is organized as follows. Section II discusses the literature related to the 3D-BPP in logistics and positions the proposed approach in the related context. Section III describes the basic concepts and assumptions relevant to the 3D-SBSBPP. Section IV introduces the MILP-based layer building formulation of the 3D-SBSBPP. Section V describes the proposed matheuristics to efficiently solve the 3D-SBSBPP with logistic requirements. Section VI shows and discusses the experimental results obtained with the proposed method. Finally, Section VII provides conclusions and future research perspectives. Appendix A details a basic exact formulation of the 3D-BPP based on MILP.

II. Literature Review and Paper Positioning

A. Related Works on 3D-BPP

The bin packing problem definition relies on the Cutting-Stock problem firstly proposed in [12] and in [13]. In particular, the authors discussed the problem of cutting standard pieces of stock material into pieces of specified sizes while minimizing material waste and the related costs. This problem was then extended to various applications and redefined leading to the broad category of Cutting and Packing (C&P) problems. According to the widely accepted typology for C&P problems presented in [14], the classic or one-dimensional BPP belongs to the class of one-dimensional Single Bin-Size BPP (1D-SBSBPP) and consists in packing a set of strongly heterogeneous items of given weights to a minimal number of bins of identical capacity (i.e., only one bin type) such that for each bin the total capacity of the small items does not exceed the bins' capacity. Further types of BPP identified by [14] are the two- and the three-dimensional Single Bin-Size BPP (2D- and 3D-SBSBPP), the Multiple Bin-Size BPP (MBSBPP) if the bins are weakly heterogeneous, and the Residual BPP (RBPP) if the assortment of bins is strongly heterogeneous.

In this paper we focus on the 3D-SBSBPP for internal logistic applications, for which several formulations are presented in the literature. The classic ones consider only geometric conditions, i.e., items must not overlap and must lie inside the bins; while more recent formulations include additional conditions of practical utility, such as the stability of the configuration and the family or category of items (i.e., items of the same family/category have common fundamental characteristics and can be grouped in the same bin; for example, chemical products should not be combined with food). In Table I, similarly to the early classification presented in [15], we specify for each article discussed in this section the implemented logistic requirements (i.e., rotation of the items, stability, load bearing, weight, family/category, shape, overhang, robotized layer picking). As it emerges from the following review of the state of the art, the problem is typically formulated by employing linear programming (LP), integer linear programming, and MILP (see, e.g., the reviews [16]–[18]). Nevertheless, the contributions on real logistic applications of the 3D-SBSBPP mainly consider the MILP formulation since it properly allows to represent geometric and safety constraints, see e.g., [19]–[21]. As for the resolution of the problem, due to its NP-hard nature, the majority of contributions consider heuristic or matheuristic approaches that allow to obtain good quality solutions.

The first attempts to define and solve the 3D-SBSBPP as an optimization problem were proposed in [16] and [27], both aiming at minimizing the number of filled bins given a set of items. Since both these preliminary approaches are impractical for companies, due to long computational times and the presented geometric constraints, several studies have proposed extended formulations and advanced heuristics or adaptive approaches to produce sub-optimal feasible solutions in a reasonable computational time. In this perspective, the authors of [22] propose a greedy randomized adaptive search procedure, based on the *wall building* algorithm (firstly proposed in [28]) without formulating the 3D-SBSBPP as a programming problem, but only providing a list of instructions to be executed by operators. The algorithm allows to decompose the 3D-SBSBPP into two sub-problems. The first aims at composing the items into vertical walls, while the second aims at composing the obtained walls into bins; both minimize the free space in the bin. The authors focus on the resolution of the problem for only one container type and multiple heterogeneous items, and consider geometric, stability, and rotation constraints. Differently from [22], the authors of [29] assume the 3D-SBSBPP to be similar to the parallel-machine scheduling problem. The problem is modeled as a MILP problem including only geometric constraints. The authors consider a heuristic procedure to define a new lower bound based on

TABLE I
LOGISTIC REQUIREMENTS CONSIDERED IN PUBLICATIONS ON PACKING PROBLEMS

Publication	Rotation	Stability	Load Bearing	Weight	Family/Category	Shape	Overhang	Robotized layer picking
Moura and Oliveira (2005) [22]	X	X						
Saraiva *et al.* (2015) [19]			X	X				
Trivella and Pisinger (2016) [20]			X					
Paquay *et al.* (2018) [23]	X	X	X	X		X		
Elhedhli *et al.* (2019) [24]		X	X	X	X	X		
Gzara *et al.* (2020) [11]		X	X	X	X	X		
Zaho *et al.* (2020) [25]		X	X					
Jiang *et al.* (2021) [26]	X							
Proposed matheuristics	X	X	X	X	X		X	X

an LP-relaxation of the MILP problem, which is improved by including valid inequalities based on some similarities with the parallel-machine scheduling problems. Three further important literature contributions about the 3D-BPP are [30], [31], and [32]. The first two present a two-level tabu search where the first-level aims at reducing the number of bins, while the second one optimizes the packing of bins following a procedure based on the interval graph representation of the packing, which reduces the size of the search space. The last contribution aims at solving large BPPs in two and three dimensions. The authors propose a straightforward heuristics based on the container loading problem method following a wall building approach and on a one-dimensional BPP approach applied then for 2D and 3D-BPP taking into account several logistic constraints such as stability, load bearing, and weight.

Subsequently, [19] considers items as physical boxes and therefore pays particular attention to their physical characteristics, including the weight and material of the considered items. Similarly to [22], the authors do not provide a mathematical formulation of the 3D-BPP, but they present an algorithm for the composition of the bins that aims at minimizing the free space in the bins and the total number of composed bins. The approach first builds horizontal layers of identical items and, then, generates packing schemes by greedily loading layers according to a selection criterion. Further, [20] considers the problem of 3D bin packing including both geometric and load bearing constraints. The goal is to find the minimum number of bins ensuring that the loaded items' average mass center falls as close as possible to an ideal point, for example the center of the bin. The authors propose a MILP formulation of the problem and a multi-level local search heuristics for its resolution. The method leads to good results in terms of bin balancing and computational time on literature examples tests.

In two recent works [21] and [23] the authors succeed in providing a complete formulation of the problem and efficient resolution methods for industrial applications. More in detail, [23] defines a mixed integer programming problem able to find a solution to the three-dimensional Multiple Bin Size Bin Packing Problem with bins of different shapes to fit inside an aircraft. For the resolution of this specific problem, three heuristics are adopted, i.e., Relax-and-Fix, Insert-and-Fix and Fractional Relax-and-Fix. Some tests are performed on test cases specifically designed for this type of problem, showing good results of the heuristics. Nevertheless, this paper disregards some important aspects such as the compatibility among the different families/categories of items to be packed and the possibility to create configurations with only one type of items both for layers and bins. Such precautions are fundamental in a real company environment and can further reduce the computational time and simplify the picking process of the items from the warehouse.

Further recent contributions [25], [33] and [26] introduce the adoption of new resolution approaches derived from the Internet technology field. The first one uses a hybrid genetic approach for the resolution of the heterogeneous BPP transportation and distribution to various locations by satisfying practical constraints, such as box rotation, fragility, container stability, weight, overlapping, and shipment placement. The second one uses a constrained deep reinforcement learning method for the resolution of an online 3D-BPP formulated as a constrained Markov decision process which includes physical stability and rotation constraints, while the last one exploits multimodal deep reinforcement learning in order to reduce the computational complexity of the classical version of the BPP (i.e., without the logistic constraints) and solve medium-scale instances of 100 items while most existing methods are only able to handle up to 50 boxes in short computational times.

Further particularly recent approaches are presented in [24] and [11], which address the mixed-case pallettization problem, i.e., an extension of the classic 3D-BPP that incorporates practical logistic features, such as bin stability, item support, family/category groupings, isle friendliness, and load bearing. To solve such a problem, a column-generation solution approach is proposed, where the pricing sub-problem is a two-dimensional layer-generation problem.

B. Paper Contributions

As it emerges from the discussion of the literature review, only a few contributions address the 3D-SBSBPP from the general perspective of the real logistic sector and present applications to real case studies.

In this paper we propose a matheuristics of practical applicability for the automatic and efficient resolution of the 3D-SBSBPP. In particular, we define a novel MILP-based layer building algorithm that efficiently determines feasible pallet configurations, minimizing the unused space and fulfilling a set of geometric and logistic specifications, i.e., items' grouping by logistic features, load bearing, stability, height homogeneity, overhang and weight limits, and robotized layer picking. In contrast with the contributions analyzed in Section II-A, we propose an algorithm that allows the automatic management of the packing process, starting from the analysis of the shipment list and allowing the definition of the most suitable configuration of bins to be delivered to the final customers in a short computational time. A synthetic comparison of our approach with the related literature is reported in Table I. In particular, differently from [22] and [29], the proposed algorithm allows the definition of multiple bins of homogeneous and heterogeneous items. With respect to [23] and [25], our algorithm takes into account also compatibility constraints among items, e.g., food is not packed with chemical products, and fragility constraints. Moreover, similarly to [24], our work considers the decomposition of the problem into the two Layer Building and Bin Building phases, but we additionally provide a detailed mathematical formulation of the practical constraints such as item support, bin stability, and load bearing for the application of the method in the logistic field. The contributions of this paper can be then summarized as follows:

- we formulate the 3D-SBSBPP as a MILP-based layer building BPP that first optimizes the layers composition and then the bin configuration. The formulation includes not only geometric constraints, but also logistic requirements, i.e., load bearing, stability, height homogeneity, overhang and weight limits (see Table I);
- we define a three-phase matheuristics based on the layer building formulation of the 3D-SBSBPP that allows, in a short computational time, the proper and automatic management of the packing process from the analysis of the shipment list to the definition of the most suitable configuration of bins to be delivered to the final customer. The method allows fulfilling further logistic constraints that are not included in the mathematical formulation of the problem, i.e, the grouping of items by logistic features, such as, delivery number, identification number, and category of product, and the robotized layer picking function (see Table I);
- we carry out extensive computational tests to compare the performance of the proposed algorithm with respect to a reference method using both realistic data-sets drawn from the literature and real data-sets. In particular, differently from related works that consider testing instances with geometrical features only, we show the effectiveness of the proposed methodology on industry-size scenarios including practical constraints required by logistic companies.

III. Problem Statement and Assumptions

Before introducing the mathematical formulation of the logistic 3D-SBSBPP and the matheuristics for its resolution, we describe some basic concepts and assumptions. In general, the logistic 3D-SBSBPP consists in properly packing a set of items to be delivered to customers inside the minimum number of identical shipping bins while fulfilling geometric and safety requirements.

We assume that items are goods to be delivered and packed in basic rectangular unit loads, i.e., packages. Differently, bins or shipping bins are a set of items packed in second level unit loads, i.e., pallet loads. The bin can be composed manually by expert operators, or automatically by anthropomorphic robots that can handle single items or homogeneous sets of items organized in layers (i.e., the so-called robotized layer picking), or both automatically and manually. With the aim of facilitating the bin assembly procedure by both operators and robots and reducing economical losses for the company, here we assume that a bin is composed by stacked layers, which can be either homogeneous (i.e., containing only one type of items) or heterogeneous (i.e., containing different types of items). Homogeneous layers are further distinguished into monoitem layers, i.e., composed by items with the same identification number (ID), and monocategory layers, i.e., composed by the same category of goods. Conversely, heterogeneous layers are defined as mixed layers that can include items with different ID and category, given that the categories are compatible. At each item a stability index is associated that ranges from 1 to 100: the higher the value, the more stable the item (note that the value is computed based on the geometrical features of the item). In order to have a stable and compact configuration of the pallet we impose that items are stacked by a decreasing value of the stability index starting from the bottom layer up to the top one.

TABLE II

Nomenclature of the Logistic 3D-SBSBPP

Name	Description
Item	object representing goods packed in a basic unit load
Layer	set of items assembled on the same plane
Monoitem layer	layer composed by items with the same ID
Monocategory layer	layer composed by items with the same category
Mixed layer	layer composed by items with different IDs and categories
Bin	set of stacked layers packed in a pallet load
Pallet	wooden platform where items are stacked to form a bin
Delivery	set of items to be transported to one single customer
Shipment	the entire set of deliveries
Category	class of items sharing specific characteristics (e.g., liquids, food, chemical products)

The dimensions of the base of bins depend on the pallet dimensions plus a tolerance excess band both in width and length called overhang, while the bins' maximum height depends on the height of the unit load used to transport the bins, namely the transport unit. A bin generally contains multiple items, here assumed related to a single delivery, while one or multiple deliveries compose a shipment order.

The algorithm proposed in this work for solving the logistic 3D-SBSBPP takes as input a shipment list and the type of pallet to be used as base for the bin, while it returns as output the most efficient configurations of the input items in bins. We assume that input items are associated to a single shipment order but they may actually be associated to different delivery orders. Actually, depending on the type of shipment, this can include one or more customer deliveries, while each delivery is associated only to a single customer. Consequently, the algorithm efficiently groups the items by delivery, ID, and category. The nomenclature of the problem and the corresponding meanings are summarized in Table II.

TABLE III
LIST OF PARAMETERS FOR THE LAYER BUILDING AND BIN BUILDING SUB-PROBLEMS

Name	**Description**
Common	
L	an arbitrary large number
s_i	stability index of item i
c_i	maximum load supported by an item i
γ_S	weight factor to estimate the stability of layer j
γ_C	weight factor to estimate the maximum load supported by a layer j
H_j	height of layer j
A_j	area of layer j
S_j	stability index of layer j
W_j	weight of layer j
C_j	maximum load supported by layer j
Layer Building	
M_{max}	maximum number of available bins to be composed
N	total number of items
Θ	width of pallets
Λ	length of pallets
Ψ	maximum height of bins
O	maximum width overhang
Q	maximum length overhang
θ_i	width of item i
λ_i	length of item i
ψ_i	height of item i
ω_i	weight of item i
Bin Building	
F	maximum load supported by a pallet
V	total number of layers
G	maximum height gap among items of the same layer
B	maximum area gap among two consecutive layers

IV. THE 3D-SBSBPP FORMULATION

In this section we present the mathematical models on which the proposed matheuristics relies, which is detailed in Section V. In particular, the logistic 3D-SBSBPP is formulated with two sub-problems based on [34], namely:

- Layer building sub-problem: given the set of items to be packed, this sub-problem aims at minimizing the unused space in each layer and thus the number of items' layers, while fulfilling geometric constraints.
- Bin building sub-problem: given the set of layers obtained from the Layer Building sub-problem, this sub-problem aims at minimizing the number of shipping bins, while fulfilling geometric and safety constraints.

Table III shows all the parameters used in the proposed formulation, while the variables are presented in Table IV.

A. Layer Building Sub-Problem

In this sub-problem the goal is to arrange the given items into a minimum number V of layers having the highest fill ratio and including items with similar heights. The layers are composed in accordance with an iterative procedure. Initially, the first layer is composed extracting the optimal subset from all the N items, which maximizes the fill ratio while fulfilling geometric constraints, overhang limits and height homogeneity requirements, i.e., ensuring that the height difference between the selected items is lower than a given threshold. For the second layer, the optimal selection procedure is applied to the remaining available items. The optimization steps are then iterated until all items are paired with a layer. For the composition of a generic layer j, given the set $\mathcal{N}_j \subseteq \mathcal{N}$ of N_j available items, the layer building model is formulated as follows:

$$\min\left((\theta+\mathrm{O})(\lambda+\mathrm{Q}) - \sum_{i\in\mathcal{N}_j} p_i\theta_i\lambda_i\right) \tag{1}$$

subject to:

$$x_i+\theta_i l_{xi}+\lambda_i(1-l_{xi}) \leq \theta+\mathrm{O}+(1-p_i)\mathrm{L}, \quad \forall i \tag{2}$$

$$y_i+\lambda_i(1-l_{yi})+\theta_i l_{yi} \leq \lambda+\mathrm{Q}+(1-p_i)\mathrm{L}, \quad \forall i \tag{3}$$

$$le_{(i,i')}+r_{(i,i')}+b_{(i,i')}+f_{(i,i')} \geq p_i+p_{i'}-1, \quad \forall i,i',i'<i \tag{4}$$

$$x_i+\theta_i l_{xi}+\lambda_i(1-l_{xi}) \leq x_{i'}+(1-le_{(i,i')})\mathrm{L}, \quad \forall i,i',i'<i \tag{5}$$

$$x_{i'}+\theta_{i'} l_{xi'}+\lambda_{i'}(1-l_{xi'}) \leq x_i+(1-r_{(i,i')})\mathrm{L}, \quad \forall i,i',i'<i \tag{6}$$

$$y_i+\lambda_i(1-l_{yi})+\theta_i l_{yi} \leq y_{i'}+(1-b_{(i,i')})\mathrm{L}, \quad \forall i',i'<i \tag{7}$$

$$y_{i'}+\lambda_{i'}(1-l_{yi'})+\theta_{i'} l_{yi'} \leq y_i+(1-f_{(i,i')})\mathrm{L}, \quad \forall i,i',i'<i \tag{8}$$

$$(\psi_i-\psi_{i'})\big(le_{(i,i')}+r_{(i,i')}+f_{(i,i')}+b_{(i,i')}\big) \leq \mathrm{G}, \quad \forall i,i',i'<i \tag{9}$$

$$(\psi_i-\psi_{i'})\big(le_{(i,i')}+r_{(i,i')}+f_{(i,i')}+b_{(i,i')}\big) \geq \text{-G}, \quad \forall i,i',i'<i \tag{10}$$

$$l_{xi}+l_{yi}=1, \quad \forall i \tag{11}$$

$$0 \leq x_i \leq \theta, \quad \forall i \tag{12}$$

$$0 \leq y_i \leq \lambda, \quad \forall i \tag{13}$$

$$le_{(i,i')}, r_{(i,i')}, b_{(i,i')}, f_{(i,i')} \in \{0,1\}, \quad \forall i,i',i'<i \tag{14}$$

$$p_i \in \{0,1\}, \quad \forall i \tag{15}$$

$$l_{xi}, l_{yi} \in \{0,1\}, \quad \forall i. \tag{16}$$

The objective in (1) is to maximize the fill ratio of the layer (i.e., to minimize the horizontal area of the pallet that is not occupied by selected items). Constraints (2)-(3) guarantee that each item is contained in the dimensions of the layer allowing a overhang tolerance for the x and y axis; moreover, they allow the rotation of the item by 90 degrees along the vertical axis. Constraints (4)-(8) ensure the assignment of the relative position of two items without overlapping, allowing the combination of the positions front, back, left, and right. Constraints (9)-(10) ensure that the maximum gap between the items height inside one layer is lower than a threshold, thus keeping layers as homogeneous as possible and contributing to the stability of the overall configuration. Constraints (11) guarantee the unique assignment of the orientation of each item i. Finally, constraints (12)-(13) and (14)-(16) specify the bounding and the integrality conditions on the defined real and binary decision variables, respectively.

Summing up, the resulting MILP problem (1)-(16) consists in determining the $2\mathrm{N}_j$ real and $\mathrm{N}_j(2\mathrm{N}_j+1)$ binary variables characterizing the layers and listed in the first part of Table IV, which minimize the objective function in (1) and meet the N_j equality constraints (11), the $\frac{7}{2}\mathrm{N}_j(\mathrm{N}_j+3)$ inequality constraints (2)-(10), the $4\mathrm{N}_j$ bounding constraints (12)-(13), and the $\mathrm{N}_j(2\mathrm{N}_j+1)$ integrality constraints in (14)-(16).

The iterative resolution of (1)-(16) allows determining the composition of layers (the items allocated to each layer and their location in terms of coordinates and vertical rotation) and, after executing the above MILP problem, the parameters representing the physical features of layers are then determined

TABLE IV

LIST OF VARIABLES FOR THE LAYER BUILDING AN BIN BUILDING SUB-PROBLEMS

Name	Description	Value
Layer Building		
i, i'	indices of items	$[1, \mathrm{N}_j]$
p_i	binary variable indicating if item i is inside (1) or outside (0) the layer	{0,1}
x_i	x-axis coordinate of the left-bottom corner of item i in the layer	$\mathbb{R}^+$
y_i	y-axis coordinate of the left-bottom corner of item i in the layer	$\mathbb{R}^+$
$l_{\mathrm{x}i}$	binary variable indicating whether the length of item i is parallel to the x-axis (1) or not (0)	{0,1}
$l_{\mathrm{y}i}$	binary variable indicating whether the length of item i is parallel to the y-axis (1) or not (0)	{0,1}
$le_{(i,i')}$	binary variable indicating whether item i is on the left (1) of item i' or not (0)	{0,1}
$r_{(i,i')}$	binary variable indicating whether item i is on the right (1) of item i' or not (0)	{0,1}
$f_{(i,i')}$	binary variable indicating whether item i is in front (1) of item i' or not (0)	{0,1}
$b_{(i,i')}$	binary variable indicating whether item i is behind (1) item i' or not (0)	{0,1}
Bin Building		
j, j'	indices of layers	$[1, \mathrm{V}]$
k	index of bin	$[1, \mathrm{M}_{\max}]$
n_k	binary variable indicating whether bin k is empty (0) or not (1)	{0,1}
$\upsilon_{j,k}$	binary variable indicating whether layer j is inside (1) or outside (0) bin k	{0,1}
z_j	z-axis coordinate of the left-bottom corner of layer j	$\mathbb{R}^+$
$o_{(j,j')}$	binary variable indicating whether layer j is above (1) layer j' or not (0)	{0,1}
$u_{(j,j')}$	binary variable indicating whether layer j is below (1) layer j' or not (0)	{0,1}

as follows:

$$H_j = \max_{i \in \mathcal{N}_j} \psi_i, \quad \forall j \tag{17}$$

$$A_j = \sum_{i \in \mathcal{N}_j} \lambda_i \theta_i, \quad \forall j \tag{18}$$

$$S_j = \gamma_S \sum_{i \in \mathcal{N}_j} \mathrm{s}_i, \quad \forall j \tag{19}$$

$$C_j = \gamma_C \sum_{i \in \mathcal{N}_j} \mathrm{c}_i, \quad \forall j \tag{20}$$

$$W_j = \sum_{i \in \mathcal{N}_j} \omega_i, \quad \forall j. \tag{21}$$

In particular, the height of a given layer is set equal to the height of the highest selected item; the layer occupied area is the sum of the areas occupied by all the selected items; the layer stability is defined as the sum (scaled by factor γ_S) of the values associated to the selected items; the maximum load supported by the given layer is estimated as the sum (scaled by factor γ_C) of the values associated to the selected items; the weight of the layer is given by the sum of the weights of all the items contained in it.

B. Bin Building Sub-Problem

In this sub-problem, given the set of the V layers obtained by solving the optimization problem (1)-(16) and the corresponding parameters computed by (17)-(21), the aim is to minimize the number of bins composed by the given layers, while fulfilling geometric and safety requirements.

The bin building problem is formulated as follows:

$$\min \sum_{k=1}^{\mathrm{M}_{\max}} n_k \tag{22}$$

subject to:

$$\sum_{j=1}^{\mathrm{V}} \upsilon_{j,k} \leq \mathrm{V}\, \mathrm{n}_k, \quad \forall k \tag{23}$$

$$\sum_{k=1}^{\mathrm{M}_{\max}} \upsilon_{j,k} = 1, \quad \forall j \tag{24}$$

$$z_j + H_j \leq \psi + (1 - \upsilon_{j,k})\mathrm{L}, \quad \forall k, j \tag{25}$$

$$o_{(j,j')} + u_{(j,j')} \geq \upsilon_{j,k} + \upsilon_{j',k} - 1, \quad \forall k, j, j', j' < j \tag{26}$$

$$z_j + \psi_j \leq z_{j'} + (1 - o_{(j,j')})\mathrm{L}, \quad \forall j, j', j' < j \tag{27}$$

$$z_{j'} + \psi_{j'} \leq z_j + (1 - u_{(j,j')})\mathrm{L}, \quad \forall j, j', j' < j \tag{28}$$

$$\sum_{j'=1}^{\mathrm{V}} W_{j'} o_{(j',j)} \leq C_j, \quad \forall j \tag{29}$$

$$\sum_{j=1}^{\mathrm{V}} \upsilon_{j,k} W_j \leq \mathrm{F}, \quad \forall k \tag{30}$$

$$S_j o_{(j,j')} \leq S_{j'} u_{(j',j)}, \quad \forall j, j', j' < j \tag{31}$$

$$S_j u_{(j,j')} \geq S_{j'} o_{(j',j)}, \quad \forall j, j', j' < j \tag{32}$$

$$(A_j - A_{j'}) o_{(j,j')} \leq \mathrm{B}, \quad \forall j, j', j' < j \tag{33}$$

$$(A_j - A_{j'}) u_{(j,j')} \leq \mathrm{B}, \quad \forall j, j', j' < j \tag{34}$$

$$0 \leq z_j \leq \psi, \quad \forall j \tag{35}$$

$$n_k \geq n_{k+1}, \quad \forall k \in \{1, \ldots, \mathrm{M}_{\max} - 1\} \tag{36}$$

$$o_{(j,j')}, u_{(j,j')} \in \{0, 1\}, \quad \forall j, j', j' < j \tag{37}$$

$$\upsilon_{j,k} \in \{0, 1\}, \quad \forall k, j \tag{38}$$

$$n_k \in \{0, 1\}, \quad \forall k. \tag{39}$$

The objective in (22) is to minimize the number of bins to be composed, given the set of layers. Constraints (23) ensure the consistency between binary variables $\upsilon_{j,k}$ $(\forall j)$ and n_k for each bin k, i.e., if any layer is assigned to a bin, the bin is considered not empty. Constraints (24) make sure that each layer can be part at most of one bin. Constraints (25) guarantee that the placement of each layer does not exceed the maximum height of the bin. Constraints (26) concern the relative position that two consecutive layers can assume inside the bin (i.e., on top or below). Constraints (27) - (28) are related to the non-overlapping of two layers placed in the same bin. Constraints (29) limit the maximum load that a single layer can withstand. Constraints (30) limit the maximum weight that a single pallet can withstand. The safety constraints (31) and (32) ensure that the stability index of each layer is higher than or equal to the stability index of the respective above layers, while (33) and (34) impose that the gap of area between two consecutive layers must not be greater than the given threshold B. Finally, constraints (35), (36) and (37)-(39) specify the bounding, the validity, and integrality conditions on the defined real and binary decision variables, respectively.

Summing up, the resulting MILP problem (22)-(39) consists in determining the V real and $2\mathrm{M}_{\max}(1+\mathrm{V})+\mathrm{V}(\mathrm{V}-1)$ binary variables characterizing the bins and listed in the second part of Table IV, which minimize the objective function in (22) and meet the V equality constraints (24), the $\mathrm{M}_{\max}(2+\frac{1}{2}\mathrm{V}(\mathrm{V}+1))+\mathrm{V}(2+3(\mathrm{V}-1))$ inequality constraints (23) and (26)-(39), the 2V bounding constraints (35), the $\frac{1}{2}(\mathrm{M}_{\max}(\mathrm{M}_{\max}-1))$ validity constraints (36), and the $\mathrm{M}_{\max}(1+\mathrm{V})+\mathrm{V}(\mathrm{V}-1)$ integrality constraints (37)-(39).

V. The Matheuristics for the Automated 3D-SBSBPP

In this section we describe the proposed matheuristics for the logistic 3D-SBSBPP, which is based on the mathematical models described in section IV. We highlight that, as described in Section III, the algorithm takes as input a shipment list that includes items with different IDs and different categories, to be delivered to different clients. Thus, the algorithm is in charge of creating monoitem, monocategory, and mixed layers. As shown in Fig. 1, the proposed matheuristics is composed by 3 different phases, namely, Grouping, Layer Building, and Bin Building, which are represented in the flowchart by dashed boxes and are executed sequentially.

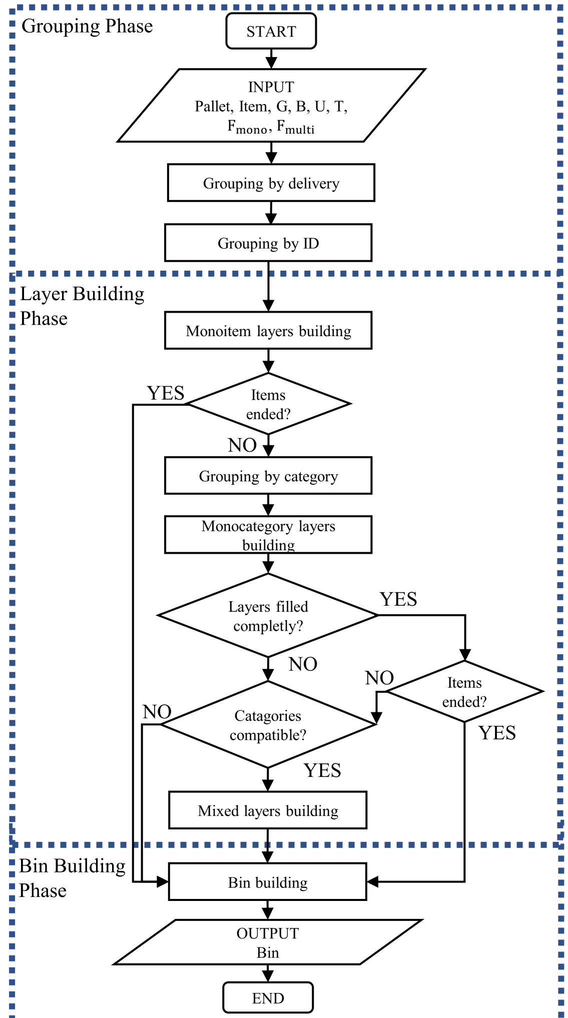

Fig. 1. High-level flow chart of the proposed matheuristics.

A. Grouping

this is a pre-processing phase aimed mainly at grouping items by delivery and ID. This procedure receives as input the data related to a shipment including the list of deliveries associated to various customers. First, this phase initializes all the parameters of the problem, i.e., the parameters related to items (N, θ_i, λ_i, ψ_i, s_i, a_i, ω_i), the class of parameters related to pallets (V, θ, λ, ψ, O, Q), and the parameters related to the creation of layers (G, B, U, T, $\mathrm{FR}_{\mathrm{mono}}$, $\mathrm{FR}_{\mathrm{multi}}$), where U and T are the maximum allowable layer picking weight and height, and $\mathrm{FR}_{\mathrm{mono}}$ and $\mathrm{FR}_{\mathrm{multi}}$ are the minimum admitted values of the fill ratio of monoitem layers and of monocategory and mixed layers. The fill ratio indicates the percentage of the layers' area occupied by the respective items. We also highlight that parameters O and Q are set according to the dimensions of the input items, ensuring that at least the half base of the smallest item is placed inside the bin. They are calculated as follows:

$$\mathrm{O}=\mathrm{Q}<\frac{\min_i\{\theta_i,\lambda_i\}}{2}. \tag{40}$$

Moreover, with the aim of facilitating the delivery and the loading/unloading of bins from transport units to customers, we impose that items belonging to a specific delivery are grouped, so that such items can be packed together inside the bins. Subsequently, given a delivery, the algorithm further groups the items by ID.

B. Layer Building

the main purpose of this phase is to obtain feasible configurations of monoitem layers, monocategory layers, and mixed layers. In particular, monoitem layers are created starting from each monoitem set by means of a heuristic procedure that creates strips of items inside of the layer by starting from its left bottom corner. A monoitem layer is created if the area of the items fully covers the area of the layer (i.e., the fill ratio of the j-th layer is $\mathrm{FR}_j \geq \mathrm{FR}_{\mathrm{mono}}$) and also both the length and width of the items fit the length and width of the layer. In particular, the items are sequentially positioned in the layer by iteratively assigning the coordinates of the left bottom corner of each item. For the monoitem layers, the robotized layer picking is enabled only if the total weight and the height of the layer are lower than or equal to the maximum allowable layer picking weight (U) and maximum allowable layer picking height (T). If all the items of the shipment are inserted in monoitem layers, the algorithm executes the subsequent Bin Building phase, otherwise the remaining items are processed for creating monocategory and mixed layers. Specifically, the items are first grouped by category. For each category, the items are grouped in subsets $\mathcal{N}_j$ for which the summation of the items' base area satisfies the relation $\mathrm{FR}_j \geq \mathrm{FR}_{\mathrm{multi}}$. The optimization problem (1)-(16) is then iteratively solved over the subsets $\mathcal{N}_j$, until all items are assigned to a layer. Note that at each step a MILP solver iterates until a solution is found or the computational time is greater than a given threshold Δ_t (i.e., an approximate solution is found). At each iteration, the obtained layer configuration is saved if it satisfies the requirement on the fill ratio, i.e., $\mathrm{FR}_j \geq \mathrm{FR}_{\mathrm{multi}}$, otherwise the corresponding items are recombined with the remaining items and novel subsets $\mathcal{N}_j$, containing all the items that are still not assigned to any bin, are defined. The eventual residual items of different but compatible categories are combined in mixed layers following the same steps described for monocategory layers. In the eventuality of residual compatible items, further mixed layers will be composed by relaxing the height homogeneity and fill ratio requirements.

The algorithm then moves to the Bin Building phase in the following cases: (1) all the items included in the monocategory sets are assigned to layers (both monocategory and mixed layers); (2) not all items are assigned to a layer, but these residual items belong to incompatible categories (i.e., categories that cannot be combined in the same layer, e.g., food and chemical products).

C. Bin Building

this phase aims at properly combining the layers computed in the previous phases (i.e., monoitem, monocategory, and mixed layers) into a minimum number of bins. In particular, this procedure is executed on a delivery basis, i.e., all the layers related to a given delivery are grouped by compatible categories and used to solve the bin building problem (22)-(39) presented in Section IV-B.

We highlight that the proposed matheuristic algorithm provides fesible solutions in a short computational time. The quality of the results and the scalability of the method is obtained empirically. In the experimental results section we discuss both aspects and compare the results of the presented algorithm with those of the exact resolution. Moreover, we evaluate the computational time required by both the exact and proposed methods, proving the advantages of our matheuristic algorithm.

VI. Experimental Results

This section is devoted to the validation of the performance of the proposed matheuristics. All tests are performed on a laptop equipped with a 2.20 GHz Intel Core i7-8750H CPU and 32 GB RAM using C# language [35], combined with the Glop linear solver [36]. In particular, due to the nature of the elements characterizing the 3D-SBSBPP, object-oriented programming has been employed, in order to properly represent the structure and relations of the problem data.

In the following sub-sections the proposed matheuristics is tested first on a small data-set where the results and performance of the algorithm are compared with the ones of the 3D-SBSBPP basic formulation presented in Appendix A. Then the method is compared with a reference method using both realistic data-sets drawn from the literature and realistic industrial data-sets [11]. Finally, the algorithm is tested on real data provided by the Italian logistic company Elettric80 Ltd [37] to further investigate its performance.

The following performance indicators are considered to analyse the performance of the algorithm.

Number of created layers and bins:

- M: total number of obtained bins;
- V: total number of obtained layers;

Fill ratio [%]:

- $\mathrm{AvgFR}_{\mathrm{V}}$: layers' average fill ratio;

$$\mathrm{AvgFR}_{\mathrm{V}} = \frac{100}{\mathrm{V}} \sum_{j=1}^{\mathrm{V}} (1 - \frac{(\theta + \mathrm{O})(\lambda + \mathrm{Q}) - \sum_{i=1}^{\mathrm{N}_j} \mathrm{a}_i}{(\theta + \mathrm{O})(\lambda + \mathrm{Q})})$$

where a_i is the base area of the $i-$th item obtained as $\theta_i \lambda_i$.

- $\mathrm{AvgFR}_{\mathrm{M}}$: bins' average fill ratio;

$$\mathrm{AvgFR}_{\mathrm{M}} = \frac{100}{\mathrm{M}} \sum_{k=1}^{\mathrm{M}} (1 - \frac{(\theta + \mathrm{O})(\lambda + \mathrm{Q})\psi - \sum_{j=1}^{\mathrm{V}_k} \sum_{i=1}^{\mathrm{N}_{j,k}} \mathrm{d}_i}{(\theta + \mathrm{O})(\lambda + \mathrm{Q})\psi})$$

where d_i is the volume of the $i-$th item obtained as $\theta_i \lambda_i \psi_i$.

Computational time [s]:

- T_{ex}: total computational time of the algorithm;

Stability indices $\mathrm{S1}_k$ and $\mathrm{S2}_k$, firstly qualitatively described by [38] and used also by [22], are formalized as follows:

- $\mathrm{S1}_k$: average number of items positioned below each item, in case this is not positioned directly on the pallet, i.e., not considering the lowest layer, formulated as follows:

$$\mathrm{S1}_k = \frac{1}{\mathrm{V}_k - 1} \sum_{j=1}^{\mathrm{V}_k - 1} \left(\frac{\sum_{i'=1}^{\mathrm{N}_{j+1,k}} \sum_{i=1}^{\mathrm{N}_{j,k}} \mu_{(i,i')}}{\mathrm{N}_{j+1,k}} \right)$$

where $\mu_{(i,i')}$ is equal to 1 if item i is under item i' otherwise is equal to 0

- $\mathrm{S2}_k$: average percentage of items which are not surrounded by other items in at least 3 sides inside of bin k, formulated as follows:

$$\begin{aligned}\mathrm{S2}_k = \frac{100}{\mathrm{N}_k} \sum_{j=1}^{\mathrm{V}_k} \sum_{i=1}^{\mathrm{N}_{j,k}} \min\Bigg\{1, \max\Bigg\{0, 3 \\ -\Bigg(\min\Bigg\{1, \sum_{i'=1}^{\mathrm{N}_{j,k}} b_{(i,i'),j}\Bigg\} + \min\Bigg\{1, \sum_{i'=1}^{\mathrm{N}_{j,k}} f_{(i,i'),j}\Bigg\} \\ +\min\Bigg\{1, \sum_{i'=1}^{\mathrm{N}_{j,k}} le_{(i,i'),j}\Bigg\} + \min\Bigg\{1, \sum_{i'=1}^{\mathrm{N}_{j,k}} r_{(i,i'),j}\Bigg\}\Bigg)\Bigg\}\Bigg\}\end{aligned}$$

where $b_{(i,i'),j}$ is equal to 1 if item i is behind item i' in layer j otherwise is 0, $f_{(i,i'),j}$ is equal to 1 if item i is in front of item i' in layer j otherwise is 0, $le_{(i,i'),j}$ is equal to 1 if item i is on the left of item i' in layer j otherwise is 0, and $r_{(i,i'),j}$ is equal to 1 if item i is on the right of item i' in layer j otherwise is 0. As for the parameters, N_j is the number of items in layer j, V_k is the number of layers of bin k, and N_k is the number of items of bin k.

The overhang index is formulated as:

- $S3_k$: average value of the overhanging ratios over all the layers assigned to the $k-$th bin. For each layer the overhanging ratio is computed dividing the actual layer overhanging area by the maximum pallet overhanging area (i.e., $(\theta + \mathrm{O})(\lambda + \mathrm{Q}) - \theta\lambda$).

In the set-up of each implemented scenario the parameters for the 3D-SBSBPP are set as follows:

- satisfactory monoitem layer fill ratio $\mathrm{FR}_{\mathrm{mono}} = 99\%$;
- satisfactory monocategory or mixed layer fill ratio $\mathrm{FR}_{\mathrm{multi}} = 90\%$;
- maximum height gap among items of the same layer $\mathrm{G} = 20$ mm;
- maximum area gap among two consecutive layers $\mathrm{B} = 10000$ mm^2;
- maximum load supported by the layer picking U=1000 Kg;
- maximum height supported by the layer picking $\mathrm{T} = 1000$ mm.

Finally, the only remaining set-up parameter for the proposed algorithm is the maximum execution time in which the solver of Layer Building Phase has to find the best configuration for the creation of each layer, that we set $\Delta_t = 90$ seconds.

A. Comparison With an Exact Method

In this subsection the proposed matheuristic algorithm is compared with the exact solution of the logistic 3D-SBSBPP. Note that, as highlighted in the comparative review in [9], only a few contributions are available in the literature that apply exact methods to the 3D-SBSBPP. This is mainly due to the difficulties in representing patterns or practical packing constraints. For this reason, in Appendix A we report a MILP formulation of the 3D-SBSBPP based on the literature formulation by Chen *et al.* [34] that includes geometric, overhang, rotation, and weight constraints and we compare it with our approach including the same requirements. In the tests the pallet dimensions are equal to the standard EUR1 Euro pallet ones, i.e., $\theta = 800$ mm (width of the pallet), $\lambda = 1200$ mm (length of the pallet). The maximum admissible height for each bin is $\psi = 1800$ mm, while the maximum weight supported by the pallet is $\mathrm{F} = 1200$ kg. These assumptions are realistic, especially for logistic companies that handle high quantities of goods.

We first perform a scalability analysis testing both methods (i.e., for the matheuristic algorithm, indicated as $\mathcal{A}$, and for the exact method, indicated as $\widehat{\mathcal{A}}$) over instances with an increasing number of identical items. The items' dimensions are $\theta = 300$ mm (width), $\lambda = 200$ mm (length), $\psi = 600$ mm (height) and $\omega = 10.8$ kg (weight).

Starting with an instance including 10 items and increasing at each test the number of items by one unit, the computed results reveal that in 10% of scenarios the exact solution provides a higher number of bins with respect to the matheuristics. On the contrary, the matheuristics succeeds in including all items in a single bin. Moreover, as shown in Fig. 2, the computational time of the matheuristics presents a growth rate significantly lower with respect to the exact solution. In particular, with 32 items the computational time of the matheuristics is 699.0% lower than the exact solution, thus demonstrating the efficiency of the proposed algorithm.

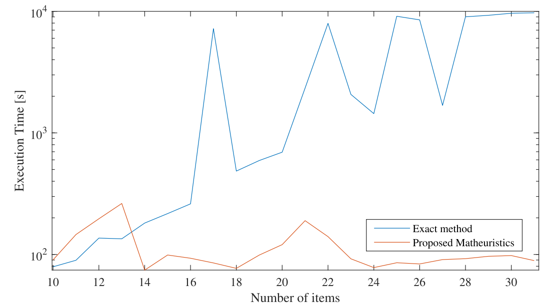

Fig. 2. Execution time of the proposed algorithm ($\mathcal{A}$) and the exact method in Appendix A ($\widehat{\mathcal{A}}$) as a function of the items number.

We further test and compare the performance of the proposed matheuristics with the ones of the exact method considering two additional scenarios, i.e., ScA and ScB. The two scenarios respectively include two and three types of different items, whose dimensions and weights are reported in Table V. It is important to mention that in both scenarios the considered items can fit in a single bin. The obtained results, reported in Table V are evaluated in terms of computational time T_{ex}, average fill ratio $\mathrm{AvgFR}_{\mathrm{M}}$, and number of obtained bins $\mathbf{M}$ for the matheuristic algorithm $\mathcal{A}$, and for the exact method $\widehat{\mathcal{A}}$. The outcomes show that in both cases the matheuristics can provide a solution with $\mathrm{AvgFR}_{\mathrm{M}} = 100\%$ in shorter computational times with respect to the exact method, and both assign all items to a single bin.

B. Comparison With a Literature Reference Matheuristics

In this subsection, we compare the results achieved with our matheuristics with the ones obtained by in [11] using a set of randomly generated industrial instances. We highlight that, in the related literature, the principal well-known datasets used to test the efficiency of 3D-SBSBPP algorithms lack most of the logistic data considered in this work (i.e., items' weight, stability index, supported load, overhang, categories, IDs, height and area gap, layer picking) that are necessary to practically implement the problem resolution. On the contrary, the work in [11] presents a matheuristics using a layer based column generation approach combined with second order cone programming and graphs, and includes all the requirements considered in this work except for the robotized layer picking. Moreover, this approach, considered as literature benchmark, is the extension of the one proposed by the same authors in [24], which proves that the algorithm largely outperforms the best performing literature's algorithms identified by [9], such as the ones in [16] and [30].

The data-set used for the comparison is obtained with the instance generator of [11] and is composed of 4 classes with 7 different instances including 100, 150, 200, 500, 1000, 1500, and 2000 items. The 4 classes contain different percentages of small to large volume items as specified in [24]. For what concerns the setup parameters, the pallet dimensions are

TABLE V

SET-UP AND RESULTS FOR THE COMPARISON OF THE PROPOSED ALGORITHM ($\mathcal{A}$) WITH THE EXACT METHOD OF APPENDIX A ($\widehat{\mathcal{A}}$)

	Input						Output					
Scenario	**IDs**	**N**	λ_i **[mm]**	θ_i **[mm]**	ψ_i **[mm]**	ω_i **[kg]**	$\mathbf{T_{ex}}$		$\mathbf{AvgFR_M}$		**M**	
							$\mathcal{A}$	$\widehat{\mathcal{A}}$	$\mathcal{A}$	$\widehat{\mathcal{A}}$	$\mathcal{A}$	$\widehat{\mathcal{A}}$
ScA	2	18	400	{400,600}	{300,900}	{5.6,10.8}	2.6	851.84	100	100	1	1
ScB	3	26	{200,400}	{400,600}	{300,900}	{5.6,10.8}	5.4	1250.80	100	100	1	1

TABLE VI

RESULTS FOR THE COMPARISON OF THE PROPOSED ALGORITHM ($\mathcal{A}$) WITH THE LITERATURE BENCHMARK [11] ($\mathcal{A}*$)

	N	**M**						$\mathbf{AvgT_{ex}}$ **[s]**		$\mathbf{AvgFR_M}$ **[%]**					
		min		**max**		**avg**				**min**		**max**		**avg**	
		$\mathcal{A}$	$\mathcal{A}*$	$\mathcal{A}$	$\mathcal{A}*$	$\mathcal{A}$	$\mathcal{A}*$	$\mathcal{A}$	$\mathcal{A}*$	$\mathcal{A}$	$\mathcal{A}*$	$\mathcal{A}$	$\mathcal{A}*$	$\mathcal{A}$	$\mathcal{A}*$
Class 1	100	1	1	2	2	1.4	1.8	21.90	6.02	78.2	28.47	89.23	59.40	80.5	35.37
	150	2	2	2	2	2	2	20.50	21.83	28.78	41.92	80.43	45.60	58.38	44.27
	200	3	3	5	3	3.3	3	3.50	40.89	16.51	37.88	83.44	39.71	53.69	39.07
	500	5	5	6	6	5.2	5.4	204.10	545.08	15.35	49.34	82.03	59.80	48.83	55.62
	1000	9	9	9	10	9	9.4	399.51	871.07	11.02	59.68	85.60	66.71	69.62	64.09
	1500	13	13	14	14	13.5	13.8	1254.19	2456.04	27,87	63.81	94,16	68.55	69,12	64.89
	2000	18	18	19	19	18.7	18.6	1744.52	3908.56	40.8	63.01	88.71	65.94	70.12	64.28
Class 2	100	2	2	2	2	2	2	21.02	3.27	26.4	29.05	85.4	30.46	75.4	29.74
	150	2	2	3	2	2.3	2	32.09	22.93	25.62	41.59	86.21	45.93	76.23	43.83
	200	2	2	3	3	2.5	2.8	29.13	47.25	47.32	38.16	81.32	59.93	68.46	43.29
	500	5	5	7	6	6.5	6.1	103.98	392.30	34.96	49.87	89.56	60.04	58.02	57.24
	1000	9	9	13	10	10.3	9.4	279.13	1080.03	14.56	58.9	82.33	65.56	57,84	62.97
	1500	14	13	14	14	14	13.4	1384.21	2158.65	37.31	63.81	85.23	68.55	70.45	66.68
	2000	18	18	19	19	18.6	18.4	1964.62	5201.74	20.54	62.87	80.32	66.46	68.58	64.76
Class 3	100	2	2	3	2	2.3	2	30.65	11.01	26.4	22.57	84.5	24.62	76.43	23.62
	150	2	2	3	2	2.3	2	61.32	10.45	43.25	34.4	87.43	36.53	63.36	35.57
	200	2	2	3	3	2.8	2.6	63.32	37.51	24.32	38.16	88.34	59.93	61.32	43.29
	500	5	4	7	5	6.1	4.2	206.41	392.30	20.52	47.27	80.79	60.67	66.68	57.16
	1000	8	8	9	8	8.4	8	212.23	1080.03	16.34	57.99	98,23	60.97	54.32	59.72
	1500	11	11	12	12	11.6	11.2	517.25	2661.04	44,54	59.58	88.23	65.84	52.77	63.9
	2000	15	15	16	16	15.7	15.8	1893.23	5201.74	19.00	59.48	80.91	63.42	53.63	60.62
Class 4	100	1	1	2	2	1.8	1.8	22.41	7.90	43.2	21.6	92.3	40.23	79.51	25.61
	150	2	2	3	2	2.5	2	36.01	17.07	49.9	31.07	87.2	34.18	64.4	32.47
	200	2	2	3	2	2.7	2	44.09	47.86	26.1	42.74	85.34	46.89	75.3	44.63
	500	4	4	5	5	4.6	4.4	114.33	451.83	15.98	44.42	83.63	56.01	57.06	51.36
	1000	8	8	9	10	9.1	8	577.21	1867.12	15.98	54.61	83.63	56.16	62.68	55.32
	1500	10	10	11	11	10.5	10.6	875.34	2158.65	37.31	60.34	85.23	67.31	70.45	63.35
	2000	15	15	16	16	15.6	15.4	2034.76	5201.74	20.49	55.23	84.91	59.08	79.63	57.4

$\theta = 1240$ mm, $\lambda = 840$ mm and $\psi = 2200$ mm, and F = 1500 kg. Table VI reports the average results achieved by testing each combination of class and number of items over five instances; we use the symbols $\mathcal{A}$ and $\mathcal{A}*$ to respectively indicate our algorithm and the literature one. Columns I and II report the class of the instance (i.e., Classes 1 to 4) and the corresponding number of items (N); column III reports the minimum, the maximum, and the average number of bins (M) obtained with the two algorithms for the five instances; column IV reports the average CPU times obtained over the instances ($\mathrm{AvgT_{ex}}$), and column V reports the minimum the maximum and the average value of the average bins' fill ratio ($\mathrm{AvgFR_M}$) achieved over five instances. The achieved results show that our algorithm provides a higher fill ratio in terms of the average and the maximum values, while, in some cases, it provides a lower value of the minimum value with respect to the reference algorithm. In general, it is possible to notice that, on the one hand, the difference between the minimum and the maximum fill ratio values is higher for our algorithm, meaning that it provides some very full configurations, and some other emptier because they are filled with the few remaining items. On the other hand, the number of filled bins is similar in the two methods, while the execution time is almost 70% higher with our method in the small scale instances, while it is notably 170% lower in the large scale instances. Concluding, the developed comparison demonstrates that the proposed matheuristics can efficiently provide feasible bins' configurations with different set of instances both in terms of number of items and features heterogeneity. Moreover, the proposed method generally outperforms the one in [11] both in terms of computational time and fill ratio in large size industrial scenarios and consequently also the principal literature algorithms.

C. Tests on Industrial Data

In this subsection, our matheuristic method is further tested on more complex scenarios based on real data provided by the Italian logistic company Elettric80 Ltd [37]. These data are related to the three scenarios Sc1, Sc2, and Sc3 –described in Table VII– corresponding to logistic shipments with different level of item heterogeneity. The set-up of all scenarios is reported in Table VII, where column I specifies the scenario, column II the number of corresponding IDs, columns III-VI the interval for the length, width, height, and weight of the items included in the scenario. Each scenario is tested on 3 different instances of size 84, 486, and 522, corresponding

TABLE VII
SET-UP FOR THE TEST ON INDUSTRIAL DATA

Scenario	IDs	λ_i [mm]	θ_i [mm]	ψ_i [mm]	ω_i [kg]
Sc1	80	[240,400]	[170,300]	[150,305]	[5.6,17]
Sc2	30	[240,400]	[170,300]	[150,305]	[5.6,17]
Sc3	11	[240,400]	[170,300]	[150,305]	[5.6,17]

TABLE VIII
RESULTS FOR THE TEST ON INDUSTRIAL DATA

Scenario	N	V	M	T_{ex}[s]	$AvgFR_V$ [%]	$AvgFR_M$ [%]
Sc1	84	9	3	52	83.2	85.1
	486	69	13	306	84.1	54.3
	522	70	18	584.3	86.7	68.8
Sc2	84	10	2	91.1	88.8	63.7
	486	57	11	50.4	94.5	82.6
	522	62	16	66	99	99
Sc3	84	11	3	62.7	85.1	57.5
	486	59	16	26	97.5	89.3
	522	66	17	78.3	99	99

to the three most common sets of deliveries. For the tests the pallet dimensions are assumed equal to the standard EUR1 Euro pallet ones, i.e., $\theta = 800$ mm (width of the pallet), $\lambda = 1200$ mm (length of the pallet). The maximum admissible height for each bin is $\psi = 1800$ mm, while the maximum weight supported by the pallet is F = 1200 kg.

Table VIII shows the performance evaluation of the matheuristics for each scenario: columns I and II report the scenario (i.e., Sc1, Sc2, Sc3) and the corresponding number of items (N); columns III and IV report the obtained number of layers (V) and bins (M); column V reports the total computational time (T_{ex}); column VI reports the average computational time needed by the algorithm to compute each bin ($AvgT_M$); column VII reports the average value of the average bins' fill ratio ($AvgFR_M$). It is apparent that the higher the item heterogeneity, the higher the computational time. In particular, the worst performance in terms of computational time is obtained in Sc1 that contains the lowest level of homogeneous items. Actually, in Sc1 the largest part of the execution time is spent in the Layer Building Phase of the matheuristics, since a high percentage of mixed layers is required to be computed.

As for the quality of the solutions, first, the geometrical features of the composed bins are evaluated in terms of fill ratio. From Table VIII it can be noticed that the average fill ratio of the medium and large instances (i.e., with 486 and 522 items) is generally higher than the smallest one (i.e., with 84 items); in particular, for scenarios Sc2 and Sc3 it is up to the 99% of the total volume of the volume. In addition, it can be also noted that the average layers' fill ratio ($AvgFR_V$) is always higher than the average bins' fill ratio ($AvgFR_M$). Both results are due to two different reasons. First, the higher the number of items and the lower the number of IDs, the higher the number of created monoitem and monocategory layers (and consequently bins). Since these layers include items with the same geometric features and weight, the maximization of the fill ratio, i.e., eq. (1), can be more easily achieved with respect to the case of mixed items. Moreover, the algorithm is set so that the minimum fill ratio for the monoitem and monocategory layers must be higher than 90% and, in case this condition is not satisfied, the obtained layers are rejected and their items are used to create mixed layers. On the other hand, even if the average layers' fill ratio is higher due to weight, height, and safety constraints of the bins' building problem (Section IV-B), there can be limits to the possible configurations admissible for the bins' composition, especially in the case of mixed bins, thus the average bins' fill ratio can be lower than the average layers' fill ratio, thus leading to lower values in the average bins' fill ratio. To further highlight the difference in bins' fill ratio in the analyzed scenarios, Fig. 3 reports the 3D configuration of four illustrative examples that respectively represent: (a) a bin with residual mixed layers, (b) a full bin with mixed layers, (c) a full bin with heterogeneous monoitems layers, and (d) a full bin with homogeneous monoitem layers. The more homogeneous the items inside of the bin, the higher the fill ratio: the monoitem layers are generally the fullest ones (i.e., FR_{mono} is equal to 99% as specified in the initial set-up of the algorithm). Additionally, Fig. 4 reports examples respectively of mixed layers –i.e., (a) and (b)– and monoitem layers –i.e., (c)– configurations. Note that the number reported inside each item represents the item ID.

TABLE IX
STABILITY AND OVERHANGING INDICES

Scenario	N	AvgS1	AvgS2 [%]	AvgS3
Sc1	84	1.41	21.08	0.47
	486	1.31	18.83	0.44
	522	1.23	17.21	0.38
Sc2	84	1.17	3.93	0.29
	486	1.13	3.24	0.23
	522	1.08	3.01	0.20
Sc3	84	1.05	2.50	0.18
	486	1.02	2.00	0.16
	522	1.01	1.90	0.13

Finally, the obtained results are assessed in terms of logistic requirements. In particular, the stability indices (i.e., $S1_k$ and $S2_k$) and the overhang index (i.e., $S3_k$) are evaluated for each layer of the composed bins. Table IX shows AvgS1, AvgS2 and AvgS3 respectively representing the average value of $S1_k$, $S2_k$, and $S3_k$ over all the composed bins in the considered scenarios. According to [38] and [22], the higher the value of AvgS1 the higher the stability, while the lower the value of AvgS2 the higher the stability. For scenarios Sc2 and Sc3, very low values of AvgS2 are obtained, against a higher value in Sc1 that includes more mixed layers than the other two scenarios. As for AvgS1, it can be noticed that the corresponding values are particularly low, especially in scenarios Sc2 and Sc3. As a matter of fact, both in scenario Sc2 and Sc3 the bins' configurations are composed mainly by monoitem and monocategory layers, which are identical layers. Consequently, each item lies only on a totally full layer, thus not compromising the stability of the overall configuration (which has a high fill ratio and hence is more compact). Lastly, the values obtained by AvgS3 demonstrate that the higher the number of IDs the higher the overhang index. This implies that the overhang feature remarkably contributes to improve the

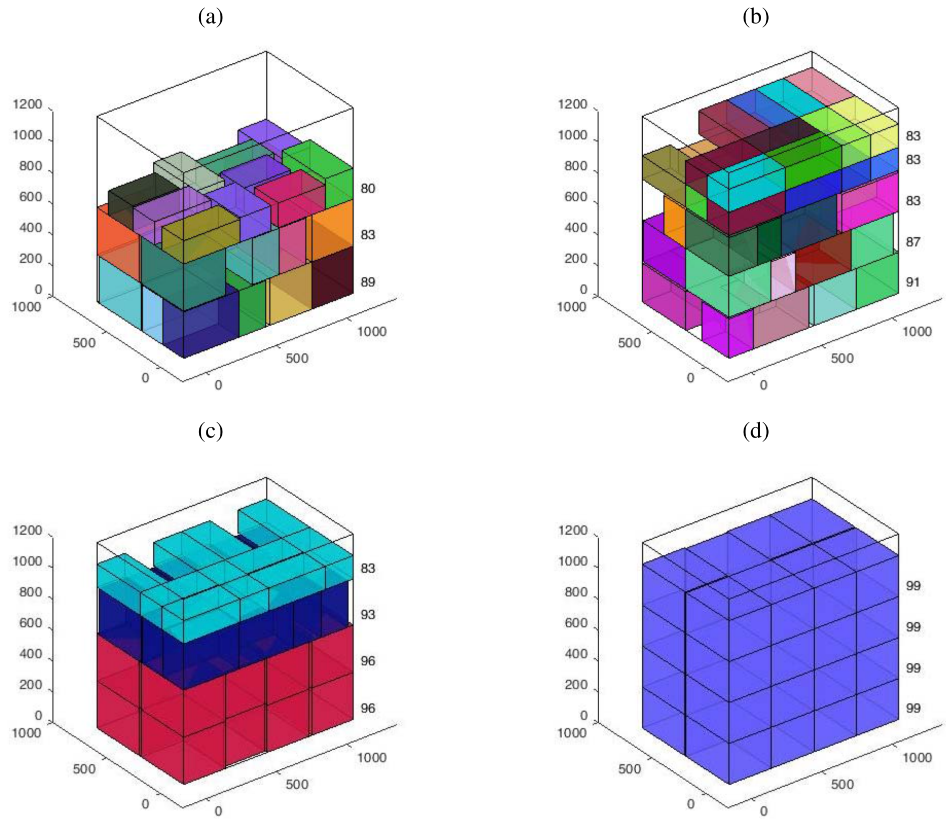

Fig. 3. Examples of: mixed bin with few items (a), mixed bin with heterogeneous items (b), mixed bin with homogeneous items (c), monoitem bin (d).

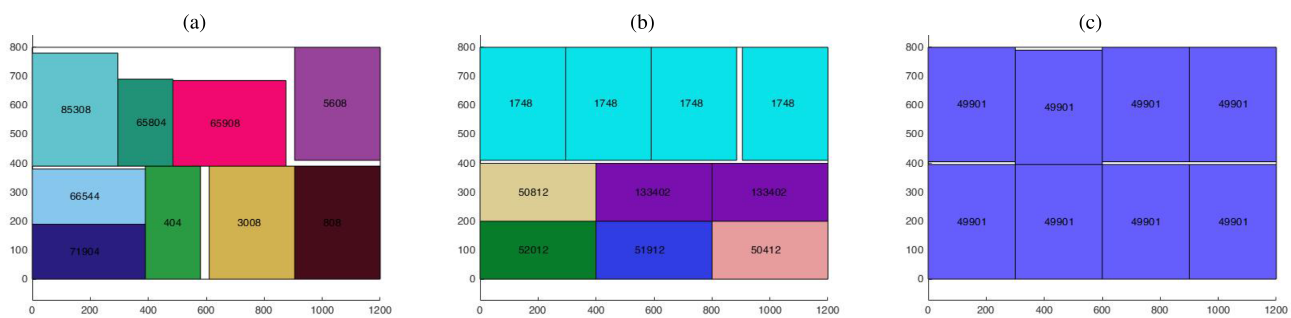

Fig. 4. Examples of mixed layers (a, b) and a monoitem layer (c).

fill ratio of scenarios with item heterogeneity, thus providing benefit to the bin packing in realistic scenarios.

VII. Conclusion and Future Developments

The 3D Bin Packing Problem (3D-BPP) has a crucial role in Industry 4.0 and in particular in the management of internal logistics, since it allows to save time and resources in the mobilization of goods. Consequently, the 3D-BPP is largely studied in the literature both because of its NP-hard nature and its high versatility in industrial applications. This work presents an innovative matheuristic algorithm based on a layer building approach that allows the automated resolution of the 3D-BPP in a short computational time and suitably for the industrial context. In particular, we propose a mixed integer non-linear programming problem to formulate the 3D-BPP including a complete set of industrial requirements and we present a matheuristic algorithm to efficiently solve the problem. Simulation results on both realistic and real data prove the efficiency and effectiveness of the proposed algorithm in terms of computational time, optimization of the bins' configuration and number, and stability of the bins. Furthermore, the proposed algorithm outperforms the results of the respective exact method. Future developments will consider the extension of the proposed approach to the multiple bin size bin packing problem, i.e., the case of multiple types of load aids with different sizes, and the inclusion of further logistic constraints that can improve the stability of the packed bins and shape constraints. Moreover, with the aim of implementing an approach even more suitable for the industrial sector, the implementation of a multi-objective optimization approach to generate Pareto efficient solutions will be considered, followed by a multi-criteria analysis to rank these solutions.

APPENDIX A
EXACT METHOD FOR THE BASIC FORMULATION OF THE 3D-BPP

This section presents a MILP formulation of the logistic 3D-SBSBPP that aims at minimizing the number of shipping bins used for packing a given set of items, while fulfilling a basic set of geometric and safety constraints. Differently from the classical 3D-BPP in [16], this formulation takes additional constraints into account. Specifically, a tolerance excess band (both in width and length) in the size of the bin base dynamically calculated according to the dimensions of the items (see Section V), rotation along the z axis, and weight limits for the pallet. For the notation and meaning of parameters and variables of the formulation we refer to Tables III and IV.

The mathematical model is defined as follows:

$$\min \sum_{k=1}^{\mathrm{M_{max}}} (\theta + \mathrm{O})(\lambda + \mathrm{Q})\psi n_k - \sum_{i=1}^{\mathrm{N}} \theta_i \lambda_i \psi_i \quad (41)$$

subject to:

$$\sum_{i=1}^{\mathrm{N}} p_{i,k} \le \mathrm{N}\, \mathrm{n}_k \quad \forall k \quad (42)$$

$$\sum_{k=1}^{\mathrm{M_{max}}} p_{i,k} = 1, \quad \forall i \quad (43)$$

$$\sum_{i=1}^{\mathrm{N}} p_{i,k}\, \omega_i \le \mathrm{F}, \quad \forall k \quad (44)$$

$$x_i + \theta_i l_{xi} + \lambda_i (1-l_{xi}) \le \theta + \mathrm{O} + (1-p_{i,k})\mathrm{L}, \quad \forall k, i \quad (45)$$

$$y_i + \lambda_i (1-l_{yi}) + \theta_i l_{yi} \le \lambda + \mathrm{Q} + (1-p_{i,k})\mathrm{L}, \quad \forall k, i \quad (46)$$

$$z_i + \psi_i \le \psi + (1-p_{i,k})\mathrm{L}, \quad \forall k, i \quad (47)$$

$$le_{(i,i')} + r_{(i,i')} + b_{(i,i')} + f_{(i,i')} + o_{(i,i')}$$
$$+u_{(i,i')} \ge p_{i,k} + p_{i',k} - 1, \quad \forall k, i, i', i' < i \quad (48)$$

$$x_i + \theta_i l_{xi} + \lambda_i (1-l_{xi}) \le x_{i'} + (1-le_{(i,i')})\mathrm{L}, \ \forall i, i', i' < i \quad (49)$$

$$x_{i'} + \theta_{i'} l_{xi'} + \lambda_{i'} (1-l_{xi'}) \le x_i + (1-r_{(i,i')})\mathrm{L}, \ \forall i, i', i' < i \quad (50)$$

$$y_i + \lambda_i (1-l_{yi}) + \theta_i l_{yi} \le y_{i'} + (1-b_{(i,i')})\mathrm{L}, \ \forall i, i', i' < i \quad (51)$$

$$y_{i'} + \lambda_{i'} (1-l_{yi'}) + \theta_{i'} l_{yi'} \le y_i + (1-f_{(i,i')})\mathrm{L}, \ \forall i, i', i' < i \quad (52)$$

$$z_i + \psi_i \le z_{i'} + (1-o_{(i,i')})\mathrm{L}, \quad \forall i, i', i' < i \quad (53)$$

$$z_{i'} + \psi_{i'} \le z_i + (1-u_{(i,i')})\mathrm{L}, \quad \forall i, i', i' < i \quad (54)$$

$$l_{xi} + l_{yi} = 1, \quad \forall i \quad (55)$$

$$0 \le x_i \le \theta, \quad \forall i \quad (56)$$

$$0 \le y_i \le \lambda, \quad \forall i \quad (57)$$

$$0 \le z_i \le \psi, \quad \forall i \quad (58)$$

$$n_k \ge n_{k+1}, \quad \forall k \in \{1, \ldots, \mathrm{M_{max-1}}\} \quad (59)$$

$$n_k \in \{0, 1\}, \quad \forall k \quad (60)$$

$$p_{i,k} \in \{0, 1\}, \quad \forall i, k \quad (61)$$

$$le_{(i,i')}, r_{(i,i')}, b_{(i,i')} \in \{0, 1\}, \quad \forall i, i', i' < i \quad (62)$$

$$f_{(i,i')}, o_{(i,i')}, u_{(i,i')} \in \{0, 1\}, \quad \forall i, i', i' < i \quad (63)$$

$$l_{xi}, l_{yi} \in \{0, 1\}, \quad \forall i. \quad (64)$$

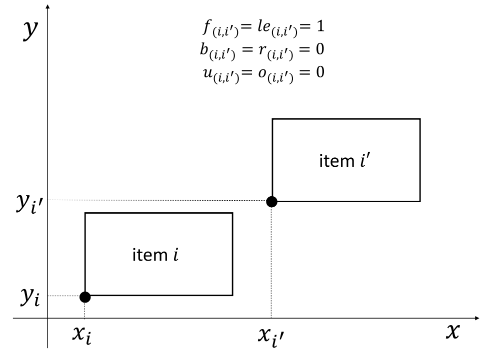

Fig. 5. Example of the relative position of two items laying in the x/y plane.

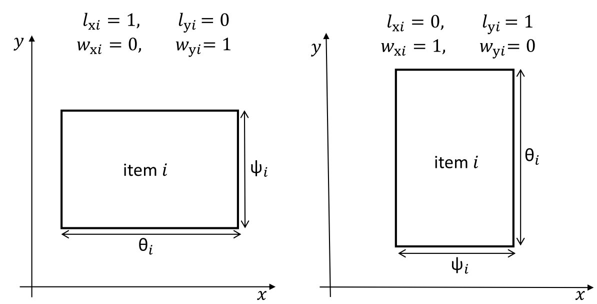

Fig. 6. Horizontal placement of an item in the rotated and not rotated configuration.

The objective in (41) is the minimization of the unoccupied space over the total number of used bins. In addition, constraints (42) ensure the consistency between binary variables $p_{i,k}$ $(\forall i)$ and n_k for each bin k, i.e., if any item is assigned to a bin, the bin is considered not empty. Constraints (43) make sure that each item can be part at most of one bin. Constraint (44) make sure that the overall weight of items allocated to each bin is not greater than the maximum weight supported by the pallet. Constraints (45)-(47) guarantee that each item is contained in the dimensions of the bin allowing a overhang tolerance for the x and y axis; moreover, they allow the rotation of the item by 90 degrees along the vertical axis. Constraints (48)-(54) are related to the relative positions that two items can assume inside the bin: (48) ensure the assignment of the relative position of two items allowing the combination of the positions front, back, left, right, over, and under, while (49)-(54) guarantee that those items do not overlap. Constraints (55) guarantee the unique assignment of the orientation of each item i. Finally, constraints (56)-(59) and (60)-(64) specify the bounding and integrality conditions on the defined real and binary decision variables, respectively.

Figure 5 shows an example of the relative position of two items (i and i') laying on the x/y plane, that is, item i is positioned on the front left side with respect to object i' in the same plane. Figure 6 represents the two different orientations that an item can assume with respect to the value of the variables l_{xi} and which may be 0 or 1 (for the sake of simplicity, the image is in 2D because the rotation is done on the x/y and so the height of the item is not influent).

Summing up, the resulting MILP problem (41)-(64) consists in determining 3N real and $\mathrm{M_{max}}(\mathrm{N}+1) + \mathrm{N}(3\mathrm{N}-1)$ binary variables, which minimize the objective function in (41) and meet the $\frac{1}{2}\mathrm{M}_{\max}(4+7\mathrm{N}+\mathrm{N}^2)+3\mathrm{N}(\mathrm{N}+1)$ inequality constraints (42) and (44)-(54), the 2N equality constraints (43) and (55), the 6N bounding constraints (56)-(58), the $\frac{1}{2}\mathrm{M}_{\max}(\mathrm{M}_{\max}-1)$ validity constraints (59), and the $\mathrm{M}_{\max}(\mathrm{N}+1)+\mathrm{N}(3\mathrm{N}+1)$ integrality constraints in (60)-(64).

References

[1] M. Rüßmann *et al.*, "Industry 4.0: The future of productivity and growth in manufacturing industries," *Boston Consulting Group*, vol. 9, no. 1, pp. 54–89, 2015.

[2] H. Lasi, P. Fettke, H. G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Bus. Inf. Syst. Eng.*, vol. 6, no. 4, pp. 239–242, 2014.

[3] J. O. Strandhagen, L. R. Vallandingham, G. Fragapane, J. W. Strandhagen, A. B. H. Stangeland, and N. Sharma, "Logistics 4.0 and emerging sustainable business models," *Adv. Manuf.*, vol. 5, no. 4, pp. 359–369, Dec. 2017.

[4] C. K. M. Lee, Y. Lv, K. K. H. Ng, W. Ho, and K. L. Choy, "Design and application of Internet of Things-based warehouse management system for smart logistics," *Int. J. Prod. Res.*, vol. 56, no. 8, pp. 2753–2768, Apr. 2018.

[5] F. Wang and A. Lim, "Effective neighborhood operators for solving the flexible demand assignment problem," *IEEE Trans. Autom. Sci. Eng.*vol. 5, no. 2, pp. 289–297, Apr. 2008.

[6] G. Cavone, R. Carli, G. Troccoli, G. Tresca, and M. Dotoli, "A MILP approach for the multi-drop container loading problem resolution in logistics 4.0," in *Proc. 29th Medit. Conf. Control Autom. (MED)*, Jun. 2021, pp. 687–692.

[7] G. Cavone, M. Dotoli, and C. Seatzu, "Management of intermodal freight terminals by first-order hybrid Petri nets," *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 2–9, Jan. 2015.

[8] G. Cavone, M. Dotoli, and C. Seatzu, "Resource planning of intermodal terminals using timed Petri nets," in *Proc. 13th Int. Workshop Discrete Event Syst. (WODES)*, May 2016, pp. 44–50.

[9] X. Zhao, J. A. Bennell, T. Bektaş, and K. Dowsland, "A comparative review of 3D container loading algorithms," *Int. Trans. Oper. Res.*, vol. 23, nos. 1–2, pp. 287–320, Jan. 2016.

[10] M. Fischetti and M. Fischetti, "Matheuristics," in *Handbook Heuristics*. Cham, Switzerland: Springer, 2018, pp. 121–153.

[11] F. Gzara, S. Elhedhli, and B. C. Yildiz, "The pallet loading problem: Three-dimensional bin packing with practical constraints," *Eur. J. Oper. Res.*, vol. 287, no. 3, pp. 1062–1074, Dec. 2020.

[12] L. V. Kantorovich, "Mathematical methods of organizing and planning production," *Manage. Sci.*, vol. 6, pp. 366–422, Jul. 1960.

[13] P. C. Gilmore and R. E. Gomory, "A linear programming approach to the cutting-stock problem," *Oper. Res.*, vol. 9, no. 6, pp. 849–859, 1961.

[14] G. Wäscher, H. Haußner, and H. Schumann, "An improved typology of cutting and packing problems," *Eur. J. Oper. Res.* vol. 183, no. 3, pp. 1109–1130, Dec. 2007.

[15] A. Bortfeldt and G. Wäscher, "Constraints in container loading—A state-of-the-art review," *Eur. J. Oper. Res.*, vol. 229, no. 1, pp. 1–20, Aug. 2013.

[16] S. Martello, D. Pisinger, and D. Vigo, "The three-dimensional bin packing problem," *Oper. Res.*, vol. 48, no. 2, pp. 256–267, Apr. 2000.

[17] J. M. V. de Carvalho, "Using extra dual cuts to accelerate column generation," *INFORMS J. Comput.*, vol. 17, no. 2, pp. 175–182, May 2005.

[18] F. Brandão and J. P. Pedroso, "Bin packing and related problems: General arc-flow formulation with graph compression," *Comput. Oper. Res.*, vol. 69, pp. 56–67, May 2016.

[19] R. D. Saraiva, N. Nepomuceno, and P. R. Pinheiro, "A layer-building algorithm for the three-dimensional multiple bin packing problem: A case study in an automotive company," *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 490–495, 2015.

[20] A. Trivella and D. Pisinger, "The load-balanced multi-dimensional bin-packing problem," *Comput. Oper. Res.*, vol. 74, pp. 152–164, Oct. 2016.

[21] C. Paquay, M. Schyns, and S. Limbourg, "A mixed integer programming formulation for the three-dimensional bin packing problem deriving from an air cargo application," *Int. Trans. Oper. Res.*, vol. 23, nos. 1–2, pp. 187–213, Jan. 2016.

[22] A. Moura and J. F. Oliveira, "A GRASP approach to the container-loading problem," *IEEE Intell. Syst.*, vol. 20, no. 4, pp. 50–57, Jul. 2005.

[23] C. Paquay, S. Limbourg, M. Schyns, and J. F. Oliveira, "MIP-based constructive heuristics for the three-dimensional bin packing problem with transportation constraints," *Int. J. Prod. Res.*, vol. 56, no. 4, pp. 1581–1592, Feb. 2018.

[24] S. Elhedhli, F. Gzara, and B. Yildiz, "Three-dimensional bin packing and mixed-case palletization," *INFORMS J. Optim.*, vol. 1, no. 4, pp. 323–352, Oct. 2019.

[25] H. Zhao, Q. She, C. Zhu, Y. Yang, and K. Xu, "Online 3D bin packing with constrained deep reinforcement learning," 2020, *arXiv:2006.14978*.

[26] Y. Jiang, Z. Cao, and J. Zhang, "Solving 3D bin packing problem via multimodal deep reinforcement learning," in *Proc. 20th Int. Conf. Auton. Agents MultiAgent Syst.*, 2021, pp. 1548–1550.

[27] O. Faroe, D. Pisinger, and M. Zachariasen, "Guided local search for the three-dimensional bin-packing problem," *INFORMS J. Comput.*, vol. 15, no. 3, pp. 267–283, Aug. 2003.

[28] J. A. George and D. F. Robinson, "A heuristic for packing boxes into a container," *Comput. Oper. Res.*, vol. 7, no. 3, pp. 147–156, Jan. 1980.

[29] M. Hifi, I. Kacem, S. Nègre, and L. Wu, "A linear programming approach for the three-dimensional bin-packing problem," *Electron. Notes Discrete Math.*, vol. 36, pp. 993–1000, Aug. 2010.

[30] T. G. Crainic, G. Perboli, and R. Tadei, "TS2PACK: A two-level tabu search for the three-dimensional bin packing problem," *Eur. J. Oper. Res.*, vol. 195, no. 3, pp. 744–760, Jun. 2009.

[31] T. G. Crainic, G. Perboli, and R. Tadei, "Extreme point-based heuristics for three-dimensional bin packing," *INFORMS J. Comput.*, vol. 20, no. 3, pp. 368–384, Aug. 2008.

[32] D. Mack and A. Bortfeldt, "A heuristic for solving large bin packing problems in two and three dimensions," *Central Eur. J. Oper. Res.*, vol. 20, no. 2, pp. 337–354, Jun. 2012.

[33] R. Sridhar, M. Chandrasekaran, C. Sriramya, and T. Page, "Optimization of heterogeneous bin packing using adaptive genetic algorithm," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 183, no. 1, 2017, pp. 12–26.

[34] C. S. Chen, S. M. Lee, and Q. S. Shen, "An analytical model for the container loading problem," *Eur. J. Oper. Res.*, vol. 80, no. 1, pp. 68–76, Jan. 1995.

[35] Wikipedia. *C Sharp (Programming Language)*. Accessed: Sep. 2021. [Online]. Available: https://en.wikipedia.org/wiki/C_Sharp_(programming_language)

[36] Microsoft. *C # Documentation*. Accessed: Sep. 2021. [Online]. Available: https://docs.microsoft.com/en-us/dotnet/csharp/

[37] Elettic80. (2021). *Elettric80 Company*. Accessed: Sep. 2021. [Online]. Available: https://www.elettric80.com/en/

[38] E. E. Bischoff and M. S. W. Ratcliff, "Issues in the development of approaches to container loading," *Omega*, vol. 23, no. 4, pp. 377–390, Aug. 1995.

Giulia Tresca (Graduate Student Member, IEEE) received the bachelor's degree in computer science and automation engineering from the Politecnico di Bari in October 2018 and the master's degree (Hons.) in computer science engineering from the Public-Private Laboratory IoT 4.0, Politecnico di Bari, in July 2020, with a specialization in information systems with a thesis on "Smart Bin Packing Systems for Logistics 4.0," where she is currently pursuing the Ph.D. degree with the Department of Electrical and Information Engineering, with a scholarship funded by the Istituto Italiano Ricerca e Sviluppo (ISIRES)—Organismo di Ricerca S.r.l.

Her research topic includes logistics 4.0 automation, decision, and control techniques for transport management systems.

Graziana Cavone (Member, IEEE) received the master's degree *(summa cum laude)* in control engineering from the Politecnico di Bari, Italy, in 2013, and the Ph.D. degree (Hons.) in electronic and computer engineering from the University of Cagliari, Italy, in 2018. She was a Research Fellow with the Politecnico di Bari in 2014 and a visiting Ph.D. student with the Delft University of Technology, The Netherlands, from 2016 to 2017. She is currently a Post-Doctoral Research Fellow with the Politecnico di Bari. She is an author of over 30 printed international publications. Her research interests include definition, simulation, and application of decision and control techniques to discrete-event and hybrid systems, manufacturing systems, logistics, energy systems, and smart cities. She was a member of the international program committee of over 20 international conferences. She was awarded a research grant by the National Science Foundation of China for the year 2020. She was the Local Arrangements Chair of the 2021 Mediterranean Conference on Control and Automation. She was a guest editor for special issues on international journals. She is an Associate Editor of the international journal of *Results in Control and Optimization* (RICO).

Raffaele Carli (Senior Member, IEEE) received the Laurea degree (Hons.) in electronic engineering and the Ph.D. degree in electrical and information engineering from the Politecnico di Bari, Italy, in 2002 and 2016, respectively.

From 2003 to 2004, he was a Reserve Officer with Italian Navy. From 2004 to 2012, he worked as a Systems and Control Engineer and the Technical Manager for a space and defense multinational company. He is currently an Assistant Professor of Automatic Control with the Politecnico di Bari. He is an author of over 60 printed international publications. His research interests include the formalization, simulation, and implementation of decision and control systems, as well as modeling and optimization of complex systems. He was a member of the international program committee of over 20 international conferences and a guest editor for special issues on international journals.

Antonio Cerviotti received the degree in mechatronics engineering, robotics, and automation from the University of Modena and Reggio Emilia in 2006. He obtained the Industrial Engineer profession qualification in 2007. He began his career with Walvoil S.p.A. Company, Reggio Emilia, where he dealt with the study of new products in the hydraulic field with the role of Product Engineer Research and Development. Then, he was hired by the "Elettric80 S.p.A" Group in 2007 as a Programmer of Robotic Systems (a PLC Software Designer and an Application Engineer). In 2013, he moved to the PC Software Department as a Business Analyst, acquiring a high level of competence in the understanding and analysis of advanced intralogistics projects with important attention to all interactions, even complex ones with ERP systems. In 2017, he started as the Project Manager of the Elettric80 Field Department and he took the role of Project Manager and Systems Analyst at the Smart Integrated Logistics Department (SM.I.LE80) in the newly born business logic of Elettric80 in 2019. Currently, he plays the role of Product Owner at SM.I.LE80.

Mariagrazia Dotoli (Senior Member, IEEE) received the Laurea degree (Hons.) in electronic engineering and the Ph.D. degree in electrical engineering from the Politecnico di Bari, Italy, in 1995 and 1999, respectively.

She has been a Visiting Scholar with Paris 6 University and the Technical University of Denmark. She is an Expert Evaluator of the European Commission since the 6th Framework Program. She is a Full Professor in Automatic Control with the Politecnico di Bari, which she joined in 1999. She has been the Vice Rector for Research of the Politecnico di Bari. She is an author of over 200 publications, including one textbook (in Italian) and over 70 international journal articles. Her research interests include modeling, identification, management, control, and diagnosis of discrete event systems, manufacturing systems, logistic systems, traffic networks, smart grids, and networked systems.

Prof. Dotoli was a member of the international program committee of over 80 international conferences. She is a member elect of the Academic Senate. She was the Co-Chairperson of the Training and Education Committee of the European Commission Network of Excellence for Fuzzy Logic and Uncertainty Modeling in Information Technology (ERUDIT) and a Key Node Representative of the EUropean Network of Excellence on Intelligent Technologies (EUNITE). She was the General Chair of the 2021 29th Mediterranean Conference on Control and Automation, the Program Co-Chair of the 2020 and 2017 IEEE Conference on Automation Science and Engineering, the Workshop and Tutorial Chair of the 2015 IEEE Conference on Automation Science and Engineering, the Special Session Co-Chair of the 2013 IEEE Conference on Emerging Technology and Factory Automation, and the Chair of the National Committee of the 2009 IFAC Workshop on Dependable Control of Discrete Systems. She is a Senior Editor of the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING and an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS.

Open Access funding provided by 'Politecnico di Bari' within the CRUI CARE Agreement