ARTICLE TEMPLATE

Contextual Bandits-Guided Local Search for Solving Air Cargo Palletisation Problem

Fatima Ezzahra Achamrah[a] [b] and Sabine Limbourg[c]

[a] Sheffield University Management School, The University of Sheffield, Conduit Road, Sheffield, S10 1FL (United Kingdom) [b] Mines Paris, PSL University, Centre for management science (CGS), i3 UMR9217 CNRS, 75006 Paris (France) [c]HEC-University of Liege (ULiege), Rue Louvrex 14, 4000 Liege (Belgium)

ARTICLE HISTORY
Compiled May 13, 2025

ABSTRACT
In this paper, we address the challenges of efficiently assigning items to Unit Load Devices within the air cargo industry. We present a comprehensive formulation of the three-dimensional air cargo palletisation problem, focusing on cost minimisation and incorporating grouping, positioning, and compatibility constraints. We propose a set of 12 resolution approaches that utilise contextual bandits-guided local search heuristics. We conduct a thorough benchmark experiment to evaluate the performance of our proposed methods. Two objective functions, namely unused volume and costs are employed to underscore the significance of cost minimisation in air cargo palletisation. Furthermore, we address instances encompassing grouping, positioning, and compatibility constraints, enabling us to explore the managerial insights these constraints offer and assess the benefits of integrating cost-reduction strategies. The findings provide valuable insights for decision-makers involved in optimising air cargo palletisation operations.

KEYWORDS
Air transport; three-dimensional bin packing; complete shipment; contextual bandits; local search heuristic; SDG 12: Responsible consumption and production

1. Introduction

Air freight transport has demonstrated significant growth over the past few decades, despite encountering recessions, geopolitical instabilities, and a temporary decline of approximately 2% in 2020 due to the COVID-19 pandemic (Rodrigue 2020). The industry rebounded swiftly, with air cargo volumes experiencing an 18.7% year-on-year increase in 2021 (IATA 2023b).

Air cargo transportation encompasses general cargo and special cargo (IATA 2021). General cargo refers to items not requiring specific handling or additional precautions during air transport. On the other hand, special cargo entails goods subject to specific regulations and requiring special handling due to their nature. Examples of special cargo include dangerous goods, live animals, perishable goods, wet cargo, and time and temperature-sensitive products. General and special cargo can be transported using either full freighter aircraft or the belly space of passenger aircraft. Full freighters are

CONTACT S. Limbourg. Email:sabine.limbourg@uliege.be

dedicated exclusively to cargo transportation, offering a fixed and substantial capacity. In contrast, the cargo capacity of passenger aircraft is more limited and susceptible to unexpected changes (Tseremoglou, Bombelli, and Santos 2022).

Air cargo rates are determined by several factors, including the travel distance, the shipment's origin and destination, and the cargo's nature (e.g., hazardous materials or perishable goods). The pricing structure considers both the weight and volume of the shipments. In addition, the airlines typically offer significant quantity discounts (Huang and Chi 2007).

Shippers often rely on freight forwarders to facilitate the delivery of their goods (Bombelli and Fazi 2022). Freight forwarders play a crucial role as decision-makers, they capitalise on consolidating multiple shipments into a limited number of larger consignments (Van Asch 2022). By doing so, they meet the shippers' requirements while minimising the costs charged by the airlines.

In this study, we focus on investigating the perspective of a large freight forwarder regarding the utilisation of full freighter aircraft. Specifically, we examine the key factors and considerations related to reserving and managing unit load devices (ULDs), which encompass various shapes and sizes, such as air cargo pallets or containers (Li, Bookbinder, and Elhedhli 2012). The freight forwarder's decision-making process involves several important parameters associated with ULDs. These parameters include fixed reservation charges, pivot weights, unit pivot costs, maximum weights, and over-pivot rates. Fixed reservation charges pertain to the costs incurred when reserving a specific ULD for transportation. Pivot weights represent the threshold at which the weight of the cargo loaded onto a ULD becomes subject to additional charges or fees. Cargo carriers typically determine these pivot weights, which vary depending on the specific ULD utilised, as well as the origin and destination of the shipment. The under-pivot rate denotes the cost associated with transporting cargo in a ULD below the pivot weight. Over-pivot rates are the additional charges applied to cargo that exceeds the pivot weight for a given ULD. Maximum weights refer to the maximum permissible load that can be safely transported.

The problem of assigning items to ULDs and determining their placement inside them is commonly referred to as the Air Cargo Palletisation Problem (Brandt and Nickel 2019). Specifically, we focus on the static version of this problem, where all items are known in advance. This problem can be formulated as a three-dimensional Multiple Bin Size Bin Packing Problem (3D-MBSBPP), as defined by Wäscher, Haußner, and Schumann (2007), considering physical, regulatory, and operational requirements. In this problem, each item must be assigned to a suitable ULD. Since most items have a parallelepiped shape (boxes), only orthogonal rotations are typically considered. The goal is to minimise the unused volume within the ULDs while satisfying various constraints.

Paquay, Schyns, and Limbourg (2016) proposed a method that minimises unused volume while considering all physical requirements. These requirements include ensuring that each loaded item is fully positioned within the admissible contour of its assigned ULD, adhering to the maximum weight limit of the ULD after loading, maintaining orthogonal placement, avoiding overlap between items, respecting orientation constraints, accommodating the specific shape of the ULDs, ensuring vertical stability, accounting for fragility considerations, and achieving appropriate ULD weight distribution.

However, in addition to these physical requirements, air freight forwarders encounter three other loading constraints (Brandt and Nickel 2019), namely:

(1) *Item compatibility.* Some items may contain substances (e.g., dry ice, radiation, chemicals, explosives) subject to additional requirements. The *United Nations* (UN) provides recommendations on transporting dangerous goods to governments and international organisations. They have been regularly amended and updated in response to technological developments and needs of users (UN 2019). Moreover, the *International Civil Aviation Organisation* (ICAO 2017) produces the basic legal requirements for storing and loading dangerous goods. In addition, the *International Air Transport Association*'s Dangerous Goods Regulations considers three types of dangerous goods: goods too dangerous to be transported by air, goods transported with *cargo aircraft only* (called CAO shipments) and goods transported both with cargo and passenger aircraft. Some dangerous goods are subject to maximum weight or quantity limitations. Moreover, hazardous materials transportation must consider that some goods may react dangerously with others.
(2) *Item grouping.* To reduce handling costs, items of the same shipment should be split across as few ULDs as possible when a cargo aircraft carries out pick-up and delivery operations at intermediate airports (Lurkin and Schyns 2015). Grouping constraints related to standard and priority items must also be respected to save handling time (Brandt 2017). Temperature-sensitive products such as food, flowers, or pharmaceutical products must be transported into special temperature-controlled ULDs to maintain goods at the desired temperature (Baxter and Kourousis 2015). Moreover, cargo registered under the same house airway bill must be transported together to meet tax and customs requirements (Chan et al. 2006). Furthermore, item grouping can also ensure that release/clearance is coordinated and carried out with minimum delay since the nature of a shipment could attract the attention of different public authorities, e.g., the customs, veterinary or sanitary controllers (Abeyratne 2018).
(3) *Item positioning.* An item might need to be placed only on the periphery of the ULD for easy accessibility (Brandt and Nickel 2019).

Our research aims to enhance the 3D-MBSBPP by incorporating practical loading constraints. The main contribution of our study is to integrate these constraints into the 3D-MBSBPP framework, focusing on minimising the total cost under the pivot-weight scheme (Bookbinder, Elhedhli, and Li 2015). This scheme is characterised by a pivot weight and two unit costs: the under-pivot rate and the over-pivot rate. Indeed, minimising unused volume, while a common objective in the literature, assumes a fixed cost per ULD, suggesting larger ULDs incur proportionally greater fixed costs. However, this assumption does not align with the reality of the transportation industry. According to Bookbinder, Elhedhli, and Li (2015), in the airfreight business, the ULD reservation fee is an attribute independent of its capacity, meaning that it may or may not be correlated to the ULD's size. Furthermore, Crainic et al. (2011) acknowledge the independence between fixed reservation costs and bin capacity in various bin packing problems. Our second contribution lies in enhancing the resolution of the 3D-MBSBPP applied to air transport. To this end, we propose a learning-based algorithm by hybridising a contextual bandits-guided local search-based heuristic with machine learning techniques (Juan et al. 2021; Achamrah, Riane, and Limbourg 2021; Achamrah et al. 2022). This combination enables us to leverage the strengths of both approaches, leading to more effective and efficient solutions for the 3D-MBSBPP in the context of air transport.

The remainder of the paper is organised as follows. Section 2 provides the back-

ground of the problem, while in Section 3, we present the problem formulation. The resolution algorithm is described in Section 4, and the results of benchmark experiments are provided in Section 5. Managerial insights are discussed in Section 6 while closing remarks are given in the last section along with potential future perspectives.

2. Literature Review

This literature review explores three main research streams: the first concerns approaches to solving the multiple bin-size bin packing problem (MBSBPP), the second focuses on various constraints considered in bin packing, and the third is related to the integration of contextual bandits in local searches.

2.1. *Approaches to solving the MBSBPP*

Various solutions have been proposed to tackle the MBSBPP in two and three dimensions. Motivated by a real-world problem in which the customers' orders involving products packed into boxes of different sizes have to be put together into bins to be delivered, Alvarez-Valdés, Parreño, and Tamarit (2013) proposed a greedy randomised adaptive search procedure (GRASP)/Path relinking algorithm for two- and three- dimensional MBSBPP. The objective is to select the most appropriate bins to minimise transportation costs. Alvarez-Valdes, Parreño, and Tamarit (2015) provided further improvements on lower bounds for two- and three-dimensional MBSBPP.

Sun et al. (2022) concentrated on the 3D-MBSBPP within the context of Alibaba's warehouses. The objective is to minimise the total number of boxes, and if there are multiple solutions with the same number of boxes, the secondary objective is to minimise the total box material costs. Based on the heuristic algorithm developed by Alibaba, the authors proposed to anticipate and incorporate human deviations to reduce them and improve performance.

An algorithm proposed by Li et al. (2017) involves considering the positions and orientations of all items simultaneously and computing their optimal arrangement in variable box sizes for an e-fulfilment packaging system. The position of the box's centre of gravity is assumed to make the box more stable and easier to carry. Moreover, similar box sizes are preferred for multiple-box deliveries. The algorithm also considers the robots' and robot manipulators' real situations to ensure a smooth process. Indeed, as mentioned in Elhedhli, Gzara, and Yildiz (2019), the automated warehouse is expected to pack thousands of items into hundreds of industry-sized pallets daily. Further, the authors included bin stability through layers to enable a column-generation approach where the subproblem generates two-dimensional packings.

Tresca et al. (2022) addressed the problem of automating the definition of feasible pallet configurations that requires a fast solution of a three-dimensional Single Bin-Size Bin Packing Problem (3D-SBSBPP) with additional logistic specifications fundamental in real applications. Such specifications include items' grouping by logistic features, load bearing, stability, height homogeneity, overhang, weight limits, and robotised layer picking. The authors also proposed matheuristics combining a mixed integer linear programming formulation of 3D-SBSBPP and layer-building heuristics. However, the concept of layers does not fit in air transport since according to Lee et al. (2021), items are strongly heterogeneous in terms of their dimensions.

Paquay, Schyns, and Limbourg (2016) introduced a mixed integer linear program to address 3D-MBSBPP. Their main contribution is to consider the weight distribution

and centre of gravity of each container to maintain safety during air transportation. The work is further extended in Paquay et al. (2018) by providing three matheuristics: Relax-and-Fix, Insert-and-Fix and Fractional Relax-and-Fix; a heuristics is developed in Paquay, Limbourg, and Schyns (2018) to solve larger instances.

2.2. *Constraints in bin packing problem*

Research in the field of bin packing problem has also looked into constraints such as separation, stability, and grouping of items. Pollaris et al. (2015) discussed separation constraints in their review on vehicle routing problems, exploring articles, as in Battarra, Monaci, and Vigo (2009), where certain products are prevented from being shipped in the same container or vehicle. Indeed, a distinction is made between three types of commodities: vegetables, fresh products, and non-perishable items, in the framework of a minimum multiple-trip vehicle routing problem.

Regarding air transportation, the safe transport of dangerous goods is subject to several regulations, as highlighted by Brandt and Nickel (2019). In their work, they mentioned the segregation table, which outlines the incompatibilities between different shipment types, varying according to factors such as the type of aircraft and the location of ULD: on the main deck or lower deck, or close to the cockpit. Indeed, aside from hazmat, other materials, often transported by air due to its rapidity, need the same consideration. For instance, hatching eggs must not be loaded near dry ice, used as a refrigerant for perishable goods transportation; foodstuffs must not be loaded close to human remains, or magnetised materials must not be loaded with undeveloped films (Dangerous goods panel 1995; Abeyratne 2018). Nascimento, Alves de Queiroz, and Junqueira (2021) considered the conflicting items that cannot be placed inside the same container and separation constraints when items should keep a certain distance from each other in the same container. Whereas Hamdi-Dhaoui, Labadie, and Yalaoui (2014) qualified as partially conflicting when products can be loaded together, an additional constraint on the distance separating them must be respected. Finally, Lurkin and Schyns (2015) addressed the same issue for the weight and balance problem: optimally loading a set of containers and pallets into a compartmentalised cargo aircraft (Limbourg, Schyns, and Laporte 2012; Zhao et al. 2021; Desai et al. 2023).

Other constraints have also been investigated, such as the requirement for items from the same group or customer to be placed as close as possible to each other inside a container. Nascimento, Alves de Queiroz, and Junqueira (2021) provided comprehensive formulations and exact algorithms for these practical constraints in the container loading problem. Indeed, they considered packing rectangular boxes into rectangular containers to maximise their occupied volume (or maximise the total profit). They also defined the grouping of items constraint when items belonging to the same group (or customer) should be placed as close as possible to each other inside the container. However, it is worth noting that their grouping constraints differ from ours, as items can be placed in multiple ULDs.

2.3. *Application of contextual bandit methodology in local searches*

Bandit problems, introduced by Thompson (1933), balance exploration and exploitation in decision-making. The Contextual Multi-Armed Bandit (CMAB) framework incorporates external contextual information (Lu, Pál, and Pál 2010), with applications across various fields and machine learning processes (Bouneffouf, Rish, and Aggarwal

2020). CMAB combined with local search has shown success in several domains. Yu, Kveton, and Mengshoel (2017) introduced the Stochastic Local Search Bandit for tuning multiple parameters simultaneously, improving regret bounds and efficiency. Chen, Wang, and Zhou (2020) used CMAB for dynamic assortment optimisation, incorporating local search and handling time-varying contextual information. Jun, Choi, and Lee (2022) employed a contextual bandit-enhanced local search algorithm for scheduling and routing autonomous mobile robots, reducing average total tardiness. Zheng et al. (2022) proposed a multi-armed bandit local search solver for MaxSAT problems, using the bandit model to escape suboptimal solutions. To the best of our knowledge, no existing papers have applied the combination of CMAB and local search techniques to solve 3D-MBSBPP.

3. Problem formulation

A ULD is charged depending on whether the total weight exceeds a certain threshold, which is called the pivot weight. Shipments are charged the under-pivot rate up to the pivot weight. Additional weight is charged at the over-pivot rate (Bookbinder, Elhedhli, and Li 2015).

Using the same notation as in Paquay, Schyns, and Limbourg (2016), the problem can be formulated as follows.

3.1. *Parameters*

A set of n rectangular boxes of dimensions $l_i \times w_i \times h_i$ and weight m_i ($i \in \{1, ..., n\}$) has to be packed into $\mathfrak{m}$ available ULDs of dimensions $L_j \times W_j \times H_j$, a maximal capacity, also called maximum gross weight, C_j and a volume V_j ($j \in \{1, ..., m\}$). All these numbers are assumed integers. Without loss of generality, we place, as in Paquay, Schyns, and Limbourg (2016), the axes of the coordinate system so that the length L_j (resp. width W_j, height H_j) of the ULD j lies on the x-axis (resp. y-axis, z-axis) $\forall j \in \{1, ..., \mathfrak{m}\}$.

The origin of this coordinate system lies on the front left bottom corner of the containers. We also assume that the ULD can be opened in $x = L$, $y = W$, or $z = H$. Each ULD has a fixed reservation cost f_j, a pivot weight $\bar{C}_j$, an under-pivot fixed cost $\bar{c}_j$ and an over-pivot rate c_j. These parameters are referred to as: $\forall i \in \{1, ..., n\},\ j \in \{1, ..., \mathfrak{m}\}$

n	Total number of boxes to be packed,	
$l_i \times w_i \times h_i$	Length $\times$ width $\times$ height of box i,	$\forall i$,
m_i	Weight of box i,	$\forall i$,
$\mathfrak{m}$	Total number of available ULDs,	
$L_j \times W_j \times H_j$	Length $\times$ width $\times$ height of ULD j,	$\forall j$,
C_j	Maximum gross weight of ULD j,	$\forall j$,
V_j	Volume of ULD j,	$\forall j$.
f_j	Fixed reservation cost of ULD j,	$\forall j$.
$\bar{C}_j$	The pivot weight of ULD j,	$\forall j$.
$\bar{c}_j$	The under-pivot rate of ULD j,	$\forall j$.

c_j The over-pivot rate per weight unit of ULD j, $\forall j$.

3.2. *Objective function*

We use binary decision variables:

$$p_{ij} = \begin{cases} 1 & \text{if box } i \text{ is in ULD } j, \\ 0 & \text{otherwise,} \end{cases} \qquad \forall i \in \{1, ..., n\},\ j \in \{1, ..., \mathfrak{m}\},$$

$$u_j = \begin{cases} 1 & \text{if ULD } j \text{ is used,} \\ 0 & \text{otherwise,} \end{cases} \qquad \forall j \in \{1, ..., \mathfrak{m}\}.$$

In Paquay, Schyns, and Limbourg (2016), the objective function consists in minimising the unused volume of the selected ULDs

$$\sum_{j=1}^{m} u_j V_j - \sum_{i=1}^{n} l_i\ w_i\ h_i. \tag{1}$$

Since l_i, w_i, h_i are parameters, the term $\sum_{i=1}^{n} l_i\ w_i\ h_i$ is a constant. Therefore, the volume of the used ULDs is minimised:

$$\sum_{j=1}^{m} u_j V_j. \tag{2}$$

As in Bookbinder, Elhedhli, and Li (2015), we also consider the objective function that consists in minimising the total cost, that is, the fixed reservation cost plus the under-pivot and over-pivot costs. While the fixed reservation costs in our formulation are similar to those in Bookbinder, Elhedhli, and Li (2015), the treatment of under-pivot and over-pivot costs differs. In Bookbinder, Elhedhli, and Li (2015), the under-pivot cost is computed by multiplying the total weight of all boxes i loaded in ULD j by the under-pivot rate. In contrast, following IATA (2023a), we define the under-pivot cost as the basic unit load device charge, meaning that if at least one box is loaded in ULD j, the under-pivot rate is applied, regardless of the total weight of the boxes. Regarding over-pivot costs, in Bookbinder, Elhedhli, and Li (2015), employ continuous variables to denote the additional capacity beyond the pivot weight. In our formulation, the weight beyond the pivot weight is given by: $\max(0, \sum_{i=1}^{n} m_i p_{ij} - \bar{C}_j)$, this ensures that the over-pivot rate is only applied to the weight that exceeds the pivot threshold.

$$\sum_{j=1}^{\mathfrak{m}} f_j u_j + \sum_{j=1}^{\mathfrak{m}} \bar{c}_j u_j + \sum_{j=1}^{\mathfrak{m}} c_j \max(0, \sum_{i=1}^{n} m_i p_{ij} - \bar{C}_j). \tag{3}$$

We can replace $\max(0, \sum_{i=1}^{n} m_i p_{ij} - \bar{C}_j)$ with $\rho_j\ \forall j \in \{1, ..., \mathfrak{m}\}$, subject to:

$$\rho_j \geq 0 \tag{4}$$

and

$$\rho_j \geq \sum_{i=1}^{n} m_i p_{ij} - \bar{C}_j. \tag{5}$$

In the context of air cargo transport, the filling rate is used to measure the extent to which the available cargo capacity is being filled. Typically, the filling rate is calculated based on either weight (load factor) or volume, and in most cases, cargo space fills up based on volume before reaching the aircraft's maximum weight capacity (Loadstar 2021; The International Air Cargo Association (TIACA) 2020). The filling rate, as used in Paquay, Schyns, and Limbourg (2016), serves as a Key Performance Indicator (KPI) for evaluating the efficiency of ULD usage, thereby complementing the cost minimisation objective.

3.3. *Operational constraints*

The objective function (3) is also subject to the following constraints:

- the maximum capacity of each ULD cannot be exceeded.
- each box is allocated to exactly one ULD.
- the boxes do not exceed their ULD size.
- there is no overlap, i.e., two boxes cannot occupy the same portion of the space.
- the boxes can rotate orthogonally in all directions allowed by their content. For instance, if the boxes contain fragile or liquid items, or items that must remain upright, their ability to rotate in specific directions might be restricted to prevent damage or spillage.
- each box lies inside the special shapes of the ULD.
- the vertical stability, that is, the bottom side of each box needs to be supported by the top face of other boxes or by the ULD floor, as mentioned in Paquay, Schyns, and Limbourg (2016). The horizontal stability is not considered since it can be fixed by adding a special sheet increasing the friction coefficient, or by bounding the unstable boxes.
- fragility: some boxes cannot support boxes on their top face.
- weight distribution: the centre of gravity (CG) of the ULDs must lie within a specific area, horizontally defined around the geometric centre of the ULD basis, and vertically, the CG must be lower than a given ceiling.

The formulations of those constraints are described in Paquay, Schyns, and Limbourg (2016).

3.4. *Compatibility*

To deal with the segregation of incompatible goods, we first define h categories of goods according to their specificity. We assume the first category is for neutral products that can be set close to others. We define a $h \times h$ segregation matrix S, element $\bar{s}_{\lambda\lambda'} = 0$ if goods belonging to category $\lambda \in \{1, ..., h\}$ and goods belonging to category $\lambda' \in \{1, ..., h\}$ can be loaded in the same ULD and 1 otherwise. S is symmetrical, and the elements on the main diagonal are equal to zero. To express that a box i belongs

to the category $\lambda \in \{1, ..., h\}$, a new set of parameters is introduced:

$$\Lambda_{i\lambda} = \begin{cases} 1 & \text{if the box } i \text{ belongs to the category } \lambda, \\ 0 & \text{otherwise,} \end{cases} \qquad \forall i \in \{1, ..., n\}.$$

Constraints (6) state that two incompatible boxes i and k cannot be in the same ULD j.

$$p_{ij} + \sum_{k=1}^{n} p_{kj} \sum_{\lambda,\lambda'=1}^{h} \Lambda_{k\lambda}\Lambda_{i\lambda'}\bar{s}_{\lambda\lambda'} \leq 1 \qquad \forall i \in \{1, ..., n\},\ \forall j \in \{1, ..., \mathfrak{m}\}. \tag{6}$$

3.5. *Grouping*

To express that a box i belongs to the group $\omega \in \{1, ..., g\}$, a new set of parameters is introduced:

$$\Omega_{i\omega} = \begin{cases} 1 & \text{if the box } i \text{ belongs to the group } \omega, \\ 0 & \text{otherwise,} \end{cases} \qquad \forall i \in \{1, ..., n\}, \omega \in \{1, ..., g\},$$

and a new set of variables:

$$\mu_{j\omega} = \begin{cases} 1 & \text{if the ULD } j \text{ is used and contains at least one box of the group } \omega, \\ 0 & \text{otherwise,} \end{cases}$$
$$\forall j \in \{1, ..., \mathfrak{m}\}, \omega \in \{1, ..., g\}.$$

3.5.1. Exclusive grouping

Exclusive grouping is used to refer to homogeneous grouping; that is, boxes that belong to a specific group cannot be mixed with other boxes. For instance, when an aircraft stops at intermediate airports for pickup and delivery, we want to minimise the number of ULDs used by grouping boxes headed to the same destination together, avoiding mixing them with boxes intended for different destinations.

In this specific case, each box should belong to one group, and two sets of constraints should be added. Constraints (7) guaranty that when a box i belonging to group ω is allocated to ULD j, $\mu_{j\omega} = 1$. Meanwhile, Constraints (8) stipulate that if ULD j is utilised, it must exclusively contain boxes from a single group.

$$p_{ij}\Omega_{i\omega} \leq \mu_{j\omega} \qquad \forall i \in \{1, ..., n\}, j \in \{1, ..., \mathfrak{m}\}, \omega \in \{1, ..., g\}. \tag{7}$$

$$\sum_{\omega=1}^{g} \mu_{j\omega} = u_j \qquad \forall j \in \{1, ..., \mathfrak{m}\}. \tag{8}$$

This formulation is also applicable for boxes containing temperature-sensitive items such as deep freeze, frozen, chill or pharmaceutical, and for organising items to facili-

tate coordinated release or clearance. In this scenario, boxes that do not have specific constraints are classified under a default group.

3.5.2. Inclusive grouping

In this situation, boxes from different groups can be assigned to the same ULD. For example, priority boxes must be loaded into the fewest possible ULDs. However, if the capacity of these ULDs permits, standard boxes can be combined with priority boxes. This approach allows for efficient use of space, mixing priority and standard boxes where feasible.

Let us assume that priority boxes belong to group 1 and that standard boxes belong to group 2. Constraints (8) should be modified as

$$u_j \leq \mu_{j1} + \mu_{j2} \leq 2u_j \qquad \forall j \in \{1, ..., \mathfrak{m}\}. \tag{6.bis}$$

Moreover, the objective function should include penalty costs to split priority boxes of across as few ULDs as possible.

$$\sum_{j=1}^{\mathfrak{m}} f_j u_j + \sum_{j=1}^{\mathfrak{m}} \bar{c}_j u_j + \sum_{j=1}^{\mathfrak{m}} c_j \rho_j + \gamma \sum_{j=1}^{\mathfrak{m}} \mu_{j1}. \tag{9}$$

where γ is a penalty cost per ULD used for priority boxes.

3.6. *Positioning*

To express that the box i must be placed on the periphery of the ULD j for easy accessibility, new parameters are introduced:
$\bar{x_j} = 1$ if ULD j can be opened in $x = L_j$, 0 otherwise
$\bar{y_j} = 1$ if ULD j can be opened in $y = W_j$, 0 otherwise
$\bar{z_j} = 1$ if ULD j can be opened in $z = H_j$, 0 otherwise
with $\bar{x_j} + \bar{y_j} + \bar{z_j} = 1$, and the set of parameters

$$\pi_i = \begin{cases} 1 & \text{if the boxes } i \text{ must be placed on the periphery,} \\ 0 & \text{otherwise,} \end{cases} \qquad \forall i \in \{1, ..., n\},$$

As in Paquay, Schyns, and Limbourg (2016), $(x'_i; y'_i; z'_i)$ is the location of the rear right top corner of the box i. Since the boxes can rotate orthogonally, variables r_{iab} describe the orientation of box i inside a ULD. Index a indicates the axis, i.e., $a \in \{x := 1, y := 2, z := 3\}$, and index b indicates the side of the box, i.e., $b \in \{l := 1, w := 2, h := 3\}$. Variables r_{iab} specify which side of box i is along which axis.

$$x'_i - x_i = r_{i11}\ l_i + r_{i12}\ w_i + r_{i13}\ h_i, \qquad \forall i, \tag{10}$$
$$y'_i - y_i = r_{i21}\ l_i + r_{i22}\ w_i + r_{i23}\ h_i, \qquad \forall i, \tag{11}$$
$$z'_i - z_i = r_{i31}\ l_i + r_{i32}\ w_i + r_{i33}\ h_i, \qquad \forall i, \tag{12}$$
$$\sum_{a=1}^{3} r_{iab} = 1, \qquad \forall i, b, \tag{13}$$

$$\sum_{b=1}^{3} r_{iab} = 1, \qquad \forall i, a. \tag{14}$$

Constraints (10)-(14) describe that the boxes can rotate orthogonally in the ULD. Note that (10)-(12) imply $x_i < x'_i,\ y_i < y'_i,\ z_i < z'_i$.

Constraints (15) ensure that the box i is placed on the periphery if required.

$$\pi_i\big(\bar{x_j}(x'_i - L_j p_{ij}) + \bar{y_j}(y'_i - W_j p_{ij}) + \bar{z_j}(z'_i - H_j p_{ij})\big) \leq (1 - p_{ij}) \max(L_j, W_j, H_j)$$
$$\forall i \in \{1, ..., n\}, j \in \{1, ..., \mathfrak{m}\} \tag{15}$$

In practice, while ULDs are typically designed to be accessible from multiple sides, in actual loading they are often positioned close to the aircraft's fuselage or adjacent ULDs. Consequently, we assume that only one side of a ULD remains reachable for access. It is important to note that during the aircraft loading process, some ULDs might become completely inaccessible. However, this will only be determined when the ULDs are loaded onto the aircraft.

4. Resolution Approach

Our research introduces an advanced solution method tailored to the complexity of 3D-MBSBPP. This method intricately weaves together sophisticated local search technique and machine learning algorithms. The local search algorithm is designed to iteratively improve upon an initial solution. This is achieved through a series of refinements where each step involves selecting and applying a local search operator to modify the current solution. Further, the selection of operators is not random; it is guided by the CMAB algorithm, which evaluates the context of the current solution and historical performance data to choose the most appropriate operator, effectively guiding it towards more promising regions of the solution space.

4.1. *Local Search-Based Heuristic*

A local search-based heuristic begins with an initial solution to the problem and iteratively improves upon this solution using a set of search operators. The algorithm continues to explore potential solutions until it meets a termination criterion, such as reaching a maximum number of iterations or a specified computational time limit. One of the key strengths of local search methods is their ability to find high-quality solutions within a reasonable time frame, even when dealing with complex and large-scale problems. Next, we will describe the process of constructing an initial solution for the problem at hand.

4.1.1. Solution construction

All non-fragile boxes belonging to each group ω and each category λ, are sorted based on their weight and dimensions. For instance, heavy boxes should be placed at the base of a ULD j, to improve the position of the centre of gravity. In this paper, a box i is considered heavy if $m_i \geq \epsilon m_j$; with $0.3 \leq \epsilon \leq 1$, and j the assigned ULD. Also, boxes with large bases should be placed first. The initialisation starts by rotating each box to give the retained orientation minimum height. Similarly, for the given orientation,

a box i is considered large if $l_i w_i \geq \beta L_j W_j$; with $0.3 \leq \beta \leq 1$. In summary, using *Sorting* procedure (see Algorithm 1), boxes in each category and group are sorted in a decreasing order regarding their weight. Heavy boxes are then sorted in decreasing order with respect to their dimension and orientation. The same logic holds for fragile boxes.

Algorithm 1 Sorting Procedure

```
Input: List of boxes, groups, and categories
Output: Sorted list of boxes
procedure SortBoxes(boxes, groups, categories)
    Initialise an empty list sortedBoxes
    for each group and category combination do
        Collect all boxes matching the current group and category
        Sort these boxes by weight and dimensions
        Add sorted boxes to sortedBoxes
    return sortedBoxes
```

By using a straightforward sorting procedure, we aim to evaluate how specific sorting criteria, such as weight and dimensions, interact and influence the solution quality. Also, we can more easily assess these relationships without introducing unnecessary complexity. This approach is consistent with previous studies addressing similar three-dimensional bin packing problems, such as Angelelli, Archetti, and Peirano (2022). Moreover, the main focus of our work is on the integration of the CMAB algorithm to guide the local search process. The CMAB component intelligently selects operators based on the context and historical performance, making the overall optimisation process more efficient and adaptive.

To determine a box position in a ULD, we used, as in Paquay, Limbourg, and Schyns (2018), Extreme Point (EPs) methodology introduced by Crainic, Perboli, and Tadei (2008) for the 3D Bin Packing problem. EPs are defined as the positions where the boxes can be accommodated. The EPs list is initialised with a point $(0, 0, 0)$ if ULD is open, or $(\theta, 0, 0)$ if its down corner is cut. If a box is placed at this point, it is removed from the sorting list, and the EP is removed from the EPs list.

As in Paquay, Limbourg, and Schyns (2018), the EPs list is updated using an *EP-Generation* procedure. Indeed, for a given box i that is placed with its front left bottom vertex on the point (x_i, y_i, z_i) with its opposite vertex on (x'_i, y'_i, z'_i). Three initial points (x'_i, y_i, z_i), (x_i, y'_i, z_i), and (x_i, y_i, z'_i) represent the new EPs. Each point is projected along two directions (x and y) until reaching a previously packed box or a ULD side to create new EPs.

To construct an initial solution to the problem, we use *Greedy-Solution* procedure (see Algorithm 2). For a given ULD, selected randomly, and for each group and category, the first and non-fragile box in the sorting list is picked and placed at the base and initial EP if and only if its dimensions are lower than the maximum dimensions of ULD, and its weight is lower than the maximum weight allowed for ULD. Otherwise, the next box in the sorting list is picked, and the same conditions are checked. Once the first box is placed, the EPs list is updated using *EP-Generation* procedure, and the box is removed from the sorting list. Another box is picked from the list, and its dimensions and weight are checked with respect to the capacity and limit constraints of ULD. Also, the best orientation is chosen so that the box is placed on an EP located on the remaining space of the ULD base. Otherwise, the next feasible orientation is selected along with the corresponding EP where the box will be placed so that:

- maximum capacity and limit of each ULD are not exceeded;

- no overlapping is allowed;
- fragile box, if suitable and can be packed, has no other boxes on its top face;
- each box is allocated to exactly one ULD.

The previous steps are applied to fragile boxes if they are still unpacked. If other boxes are still unpacked, they are assigned to the biggest and empty ULD.

Algorithm 2 Greedy-Solution Procedure

```
Input: ULDs, sorted boxes, groups, categories
Output: Placement of boxes in ULDs
procedure GreedySolution(ULDs, sortedBoxes, groups, categories)
    Initialise an empty solution set
    for each ULD in ULDs do
        Randomly select a ULD for loading
        Initialise EPs in the selected ULD
        for each combination of group and category do
            Attempt to place boxes from sortedBoxes into the ULD
            Update EPs after each insertion
        Place any remaining fragile boxes
    Handle any boxes that could not be accommodated
    return the final placement of boxes in ULDs
```

To measure how far the projection of the centre of gravity of a ULD, noted CG_j, from the centre of the base, we use, as in Angelelli, Archetti, and Peirano (2022), the following formulas:

$$CG_j = (X_j^{CG}; Y_j^{CG}; Z_j^{CG}) = \frac{\sum_{i|p_{ij}=1} CG_i}{\sum_{i|p_{ij}=1} m_i} = \frac{\sum_{i|p_{ij}=1} (x_i^{CG}, y_i^{CG}, z_i^{CG}) m_i}{\sum_{i|p_{ij}=1} m_i} \tag{16}$$

Since we assume a homogeneous density for the box, its centre of gravity is $(x_i^{CG}, y_i^{CG}, z_i^{CG}) = (x_i, y_i, z_i) + \frac{1}{2}(l_i, w_i, h_i)$.

We also define:

$$M_j^x = |\frac{2X_j^{CG} - L_j}{L_j}| \tag{17}$$

$$M_j^y = |\frac{2Y_j^{CG} - W_j}{W_j}| \tag{18}$$

$$M_j^z = |\frac{Z_j^{CG}}{H_j}| \tag{19}$$

Ideally, the perfect layout has $M_j^{xy} = M_j^x + M_j^y = 0$. This paper assumes a ULD has an acceptable uniform weight distribution if $0 \leq M_j^{xy} \leq 0.5$. Moreover, we adopt the formulation for M_j^z as presented in (Paquay, Schyns, and Limbourg 2016), ensuring consistency with prior research in this field.

The following presents local search operators used to enhance the solution quality and ensure uniform weight distributions.

4.1.2. Local search operators

Local search operators for optimisation problems are classified into exploitation and exploration. Exploitation operators improve the current solution by making small, local changes. They aim at converging quickly to a local optimum based on the initial solution. On the other hand, exploration operators investigate new regions of the search space in order to find potentially better solutions. They are used to escape from local optima by increasing the diversity of the initial solution at the expense of the current local improvement.

We use nine operators, based on the literature to improve the initial solution, which are widely used in various local-search techniques. These local search operators are designed to optimise the number of ULDs used and unused space and improve weight distribution. The operators are the following:

- Exchange operator:
  - Weight-based exchange: one box on the top of the greatest weight ULD is picked and moved to (1) the top of the ULD having the smallest weight or (2) a randomly selected ULD among ULDs that still have space and weight capacities to pack the box.
  - Height-based exchange: one box on the top of the greatest height ULD is picked and moved to (1) the top of the ULD having the smallest height or (2) a randomly ULD among ULDs that still have height and weight capacities to pack the box.
  - Random exchange 1: if feasible, two boxes on top of two ULDs are randomly selected and switched.
  - Random exchange 2: if feasible, two boxes on top of a ULD are randomly selected and switched.
- Removal operator
  - A roulette wheel is used to select ULDs to be removed. This procedure gives high probabilities to remove ULDs having the greatest cost or the most unbalanced ones (i.e., deviation regarding the centre of gravity). An initial solution for this sub-problem (with the boxes contained in the removed ULDs) is constructed following the steps described above.
  - A roulette wheel selects unbalanced ULDs to be removed. This procedure gives high probabilities to remove ULDs having weight distribution issues $M_j^{xy} \geq 0.5$. An initial solution for this sub-problem (with the boxes contained in the removed ULDs) is constructed following the same steps described previously.
  - Random ULDs are removed. Again, an initial solution to the resulting sub-problem is constructed.

It is worth noting that the local search method manages weight distribution while strictly satisfying the CG position constraints.

Finally, the algorithm incorporates the positioning constraint by prioritising the placement of boxes requiring peripheral placement and selecting appropriate placement points on the periphery of the ULDs. It also considers the compatibility and grouping constraints by updating the respective constraints after placing each box. Algorithm 3 summarises the steps involved in constructing the solution.

Algorithm 3 Local Search Algorithm

```
Input: boxes, ULDs, categories, groups, positioning
Output: Constructed solution
procedure ConstructSolution(boxes, ULDs, categories, groups, positioning)
    currentSolution ← GreedySolution(boxes, ULDs, categories, groups, positioning)
    Initialize operator index i = 0
    while stopping criterion not met do
        for each ULD type do
            Select operator using Round-Robin: operator = operators[i mod 4]
            i = i + 1
            if operator is Weight-based exchange then
                Select box from ULD with greatest weight
                Move to ULD with smallest weight or random ULD with capacity
            else if operator is Height-based exchange then
                Select box from ULD with greatest height
                Move to ULD with smallest height or random ULD with capacity
            else if operator is Random exchange then
                Randomly select two boxes from tops of ULDs
                Swap the selected boxes if feasible
            else if operator is Removal then
                Select ULDs to remove using roulette wheel
                Remove selected ULDs
                SortBoxes(removedboxes, groups, categories)
                Repack sorted boxes
            Apply selected operator
            Update solution considering all constraints
        Evaluate new solution
        if new solution is better then
            Update currentSolution
    return Constructed solution with boxes placed in ULDs
```

4.2. *Selecting the best operator using CMAB*

In general, the performance of local search algorithms largely depends on the characteristics of the problem's search space and the design of the search operators. Both exploitation and exploration operators are important for effective optimisation. Thus, to select the best operator considering each context, we use, as employed in Jun, Choi, and Lee (2022) for solving a scheduling problem of autonomous mobile robots, a new Local Search algorithm based on the Contextual multi-armed bandit approach (LSC).

4.2.1. CMAB: introduction

The contextual bandit problem, also known as associative reinforcement learning or multi-armed bandits (Auer 2002), involves balancing exploring new options with exploiting known good options to maximise a reward. It is an iterative process that works as follows (Cortes 2018): an agent that selects from a variety of options (referred to as "arms") that include stochastic rewards. The environment generates a set of covariates of fixed dimensionality, called context, for each arm tied to the covariates at the start of each round. An agent selects one arm for that round; the environment reveals the related reward. The objective is for the agent to use the history of his prior actions to maximise the rewards that are gained over the long run. In the literature, several approaches, including upper confidence bounds (Auer 2002), Thompson sampling (Chapelle and Li 2011) and Bayesian optimisation (Sui and Yu 2020), have been

examined to balance between the exploration of unknown alternatives and exploitation of known good ones.

4.2.2. CMAB: exploitation

The integration of the CMAB algorithm within our LSC framework dynamically and intelligently balances exploration with exploitation, a critical duality in optimisation scenarios characterised by inherent uncertainty.

Fundamentally, the CMAB algorithm addresses the sequential decision-making process under conditions of uncertainty. Central to our study, this involves the selection of an action, or "arm", from a set of local search operators, each choice contingent upon the prevailing context. The CMAB enriches this decision-making process by integrating contextual data, thereby enabling the algorithm to utilise a more extensive array of environmental variables. This calibration is dynamic, evolving in response to the algorithm's accumulating experiential data, thereby ensuring a solution approach that is both adaptable and robust.

In practice, a set of neighbourhoods is generated repeatedly by choosing and applying the aforementioned local search operators to the current solution. Using a M-dimensional context vector $\zeta_t \in R^M$ as input, CMAB aims to select one of the K possible local search operators at trial t to maximise the expected improvement. ζ_t contains various features of the current solution and factors that may affect the improvement of operators. This higher-level context vector is designed to offer a generalised overview of the problem space, enhancing robustness and adaptability. This abstraction level is crucial for preventing overfitting to specific geometries, particularly important in dynamic and diverse scenarios. The context vector's features are also selected to provide sufficient insight for informed operator selection, considering the system's overall state, thus balancing detail with computational efficiency.

Further, we use cumulative improvement, $\sum_{\tau=1}^{t-1} r_{\tau,a_\tau}$, previous improvement, $r_{t-1,a_{t-1}}$, previous operator, a_{t-1}, current total cost $\mathcal{F}_{t-1}$, and M_j^{xy} for each ULD. $a_t \in 1, ..., K$ denotes the chosen operator at trial t by the algorithm. Let $\mathcal{F}_0$ and $\mathcal{F}_t$ denote the total cost of the solution at the beginning and trial t, respectively. After checking the total cost of the best neighbourhood solution, we can observe the reward r_{t,a_t} associated with the chosen operator a_t and context ζ_t as shown below.

$$r_{t,a_t} = 1 - \frac{\mathcal{F}_0}{\mathcal{F}_t} - \sum_{\tau=1}^{t-1} r_{\tau,a_\tau} \tag{20}$$

The reward r_{t,a_t} represents the improvement by choosing at trial t excluding the previous improvements.

In addressing the exploration challenge, we carefully selected our models and hyperparameters, taking into account their suitability for our problem's unique characteristics. Our approach involves an in-depth application of various combinations of estimators and exploration algorithms.

4.2.3. Vowpal Wabbit framework

The Vowpal Wabbit approach (VW), an open-source machine-learning library, is integral to our LSC method for addressing the contextual bandit problem. Originally developed at Yahoo! Research and later at Microsoft Research, VW facilitates fast,

scalable learning on large datasets (Bietti, Agarwal, and Langford 2018; Cortes 2018).

The utilisation of VW is justified by several critical factors pertinent to the contextual bandits. Primarily, VW's capability to process large datasets expeditiously is aligned with the exigencies of contextual bandit problems, which requires rapid integration of data and immediate decision-making. The efficiency of VW extends beyond mere processing speed; it encompasses the capacity for quick, informed decision-making, an essential element in the optimisation of local search operators.

Furthermore, VW supports a spectrum of exploration strategies, integral to the contextual bandit framework, namely:

- ϵ-greedy: chooses the best-known action with probability $1-\epsilon$, and a random action with probability ϵ, allowing for some exploration of less-frequently sampled actions (Cortes 2018).
- Bag (Online Bootstrap Thompson Sampling): is a variant of Thompson sampling which maintains a distribution over models of the data-generating process and selects actions according to a model sampled from this distribution, allowing for exploration of different possible models of the environment (Agrawal and Goyal 2013).
- Online Cover: maintains a collection of policies meant to approximate a covering distribution over policies that are good for exploration and exploitation (Bietti, Agarwal, and Langford 2018).
- Softmax Explorer: predicts both the best action and a score indicating the quality of each possible action and selects an action probabilistically based on these scores, allowing for exploration of lower-scoring action (Cortes 2018).

Finally, to optimise policy, VW computes an estimate of the full feedback using the observed reward. The following estimators can be used:

- Inverse Propensity Score (IPS): unbiased and makes no assumptions about how rewards might depend on context and actions. When such information is available, it is natural to posit a parametric or non-parametric model and fit it on the logged data to obtain a reward estimator (Bietti, Agarwal, and Langford 2018).
- Direct Method (DM): which estimates the reward function from a given data and uses this estimate in place of actual reward to evaluate the policy value on a set of contexts (Wang, Agarwal, and Dudık 2017).
- Doubly Robust (DR): is a statistical approach for estimation from incomplete data with a noteworthy property: if either of the two estimators (in DM and IPS) is correct, then the estimation is unbiased. This method thus increases the chances of drawing reliable inference (Dudik, Langford, and Li 2011).

4.2.4. Hyperparameters tuning

The tuning of hyperparameters in our solution method is a crucial step to enhance the effectiveness and efficiency of the local search optimisation process. This involved a meticulous adjustment of various parameters within the VW framework, specifically tailored to our problem's unique characteristics and objective.

The learning rate is a fundamental hyperparameter in machine learning models, particularly in the context of iterative optimisation algorithms like ours. It determines the step size at each iteration while moving toward a minimum of a loss function.

In our case, we began with the default learning rate suggested by VW and conducted a series of experiments to evaluate its impact on the convergence speed and quality

of the solution. Through a process of trial and error, we adjusted the learning rate incrementally. A lower learning rate was tested to see if it improved the model's ability to converge more smoothly to a global minimum, while a higher learning rate was examined for its potential to speed up convergence but with the risk of overshooting the minimum. This iterative process was guided by a balance between the speed of convergence and the quality of the solution, aiming to avoid both slow convergence and oscillation around the minimum.

Further, we employed statistical measures to validate the effectiveness of our hyperparameter tuning. We observed metrics such as the average improvement in the objective function, the variability in the solutions obtained across different runs, and the computational efficiency. These measures helped us in understanding the stability of the model under different hyperparameter settings and in identifying the configurations that offered the best trade-off between solution quality and computational effort.

Finally, Algorithm 4 outlines the steps for selecting the best operator using the CMAB approach within the LSC framework. It includes initialisation, iterative selection and application of operators, reward computation, and model updating. The stopping criteria are deemed met if either an improved solution is found within a predetermined number of consecutive iterations, or the maximum allowable computation time has elapsed.

Algorithm 4 Selecting the best operator within LSC framework

Input: Set of local search operators, K; Context vector ζ_t; Learning rate parameters; Exploration strategies
Output: Best solution using best operator
Initialise VW framework
Set initial context ζ_0 and total cost $\mathcal{F}_0$
for each trial $t = 1, 2, \ldots, T$ **do**
    Generate set of neighbourhoods using local search operators
    Update context vector ζ_t based on current solution features
    Select operator a_t using CMAB strategy within VW ▷ Selection based on learned policy
    Apply a_t to current solution
    Compute reward r_{t,a_t} using:

$$r_{t,a_t} = 1 - \frac{\mathcal{F}_0}{\mathcal{F}_t} - \sum_{\tau=1}^{t-1} r_{\tau,a_\tau}$$

    Update VW model with new data (context, action, reward)
    Adjust hyperparameters based on performance metrics
return Best solution found

For the sake of illustration, Algorithm 5 provides a comprehensive description of the DR/Softmax variant, detailing the composition of the context vector, the selection and application of operators, and the handling of constraints in the solution construction process.

Algorithm 5 LSC Algorithm: DR/Softmax

1: **Input:** Problem instance, set of local search operators
2: **Output:** Best solution using DR/Softmax
3: Initialize context ζ_0 and total cost F_0 of initial solution
4: Initialize DR/Softmax model
5: **for** each iteration $t = 1, 2, \ldots, T$ **do**
6: Generate feasible moves for all operators using GenerateMove($op, problem$)
7: Select best move a_t using Softmax Explorer based on context ζ_t
8: Apply selected move to generate a new solution using ApplyMove($a_t, problem$)
9: Update total cost F_t and compute reward r_{t,a_t}
10: Update context vector ζ_t with $\sum_{\tau=1}^{t-1} r_{\tau,a_\tau}, r_{t-1,a_{t-1}}, a_{t-1}, F_{t-1}, M_j^{xy}$
11: Update DR/Softmax model with (ζ_t, a_t, rt, a_t)
12: **return** Best solution found
13: **procedure** GenerateMove($op, problem$)
14: **if** *op* is Weight-based exchange **then**
15: Select a box from the top of the ULD with the greatest weight
16: Move the box to the top of the ULD with the smallest weight or a random ULD
17: **else if** *op* is Height-based exchange **then**
18: Select a box from the top of the ULD with the greatest height
19: Move the box to the top of the ULD with the smallest height or a random ULD
20: **else if** *op* is Random exchange **then**
21: Randomly select two boxes from the top of different ULDs or the same ULD
22: Swap the selected boxes
23: **else if** *op* is Removal based on cost or balance **then**
24: Use roulette wheel to select ULDs to remove based on cost or balance
25: Remove the selected ULDs and repack their boxes
26: **else if** *op* is Removal of unbalanced ULDs **then**
27: Use roulette wheel to select unbalanced ULDs to remove
28: Remove the selected ULDs and repack their boxes
29: **else if** *op* is Random removal **then**
30: Randomly select ULDs to remove
31: Remove the selected ULDs and repack their boxes
32: **return** *move*
33: **procedure** ApplyMove($a_t, problem$)
34: $newSolution \leftarrow$ ConstructSolution($problem, a_t$)
35: **while** *newSolution* violates constraints **do**
36: Adjust *newSolution* to satisfy constraints (e.g., repack boxes, adjust ULD assignments)
37: **return** *newSolution*

It is worth noting that to maintain solution feasibility throughout the local search process, we employ a multi-tiered approach (Algorithm 5, Line 36). First, before applying any move, we perform constraint checking to ensure it does not violate ULD capacity, compatibility, grouping, or positioning constraints. If a move would result in an infeasible solution, it is typically rejected, and an alternative move is generated or another operator is selected. In some cases, minor adjustments are made to maintain feasibility. For instance, if a box placement violates the weight distribution constraint, we may shift other boxes within the ULD to rebalance the load. Similarly, if a grouping constraint is violated, we may swap the newly placed box with another box from the correct group. This approach ensures that our algorithm consistently works with feasible solutions, avoiding the need for complex repair mechanisms.

4.3. *Training, validation, and testing process*

In this study, we adopted a comprehensive and systematic methodology for training, validating, and testing our LSC algorithm to ensure its effectiveness and wide applicability.

We began by constructing a dataset, comprising 1000 instances, tailored to reflect the intricate nature of the problem we aimed to address. This dataset was developed in accordance with the standards established by Paquay, Limbourg, and Schyns (2018). Their work involved the creation of several box instances derived from a real-world dataset, effectively addressing a significant shortfall in benchmark resources for the MBSBPP. Additionally, we used this dataset to implement the LSC method, incorporating various estimators and exploration strategies. During the training phase, the algorithm underwent an iterative learning process, continuously adjusting its parameters to optimise the objective function.

To mitigate the risk of overfitting and to enhance the model's ability to generalise to novel data, we employed a k-fold cross-validation method. This approach entailed segmenting the dataset into k distinct subsets. The model was iteratively trained on $k-1$ of these subsets, with the remaining subset serving for validation. This cycle was repeated k times, ensuring that each subset was used precisely once for validation.

Upon completion of the training and validation stages, the model underwent testing using an independent test set, distinct from the data utilised in the training and validation phases. This test set was instrumental in evaluating the algorithm's performance across various combinations of estimators and exploration strategies. We benchmarked its performance against both the heuristic model proposed by Paquay, Limbourg, and Schyns (2018) and a modified version of the LSC algorithm that does not incorporate CMAB. In the following, we present the findings from these comparative experiments.

5. Benchmark experiments

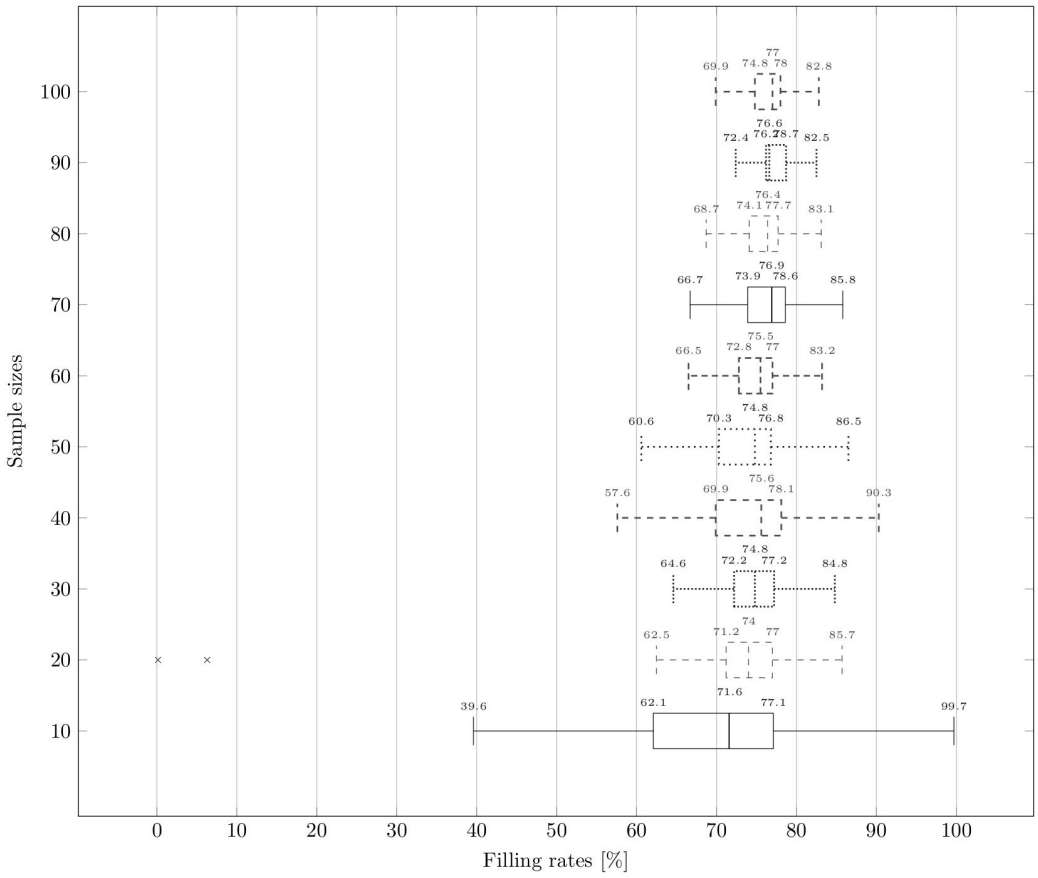
In this paper, the LSC combines three estimators (IPS, DM, and DR), with four distinct exploration algorithms (ϵ-greedy, Bag, Online Cover, and Softmax Explorer). This combination of estimators and exploration algorithms yields 12 resolution methods that are compared based on their filling rate (%) with the existing values reported in Paquay, Limbourg, and Schyns (2018) in Table 1. Further, to ensure a thorough analysis of the resolution methods, we evaluate them based on objective function values, i.e., unused volume, and CPU times. We also provide several key statistical measures for our analysis, including the median, the lower and upper quartiles, as well as the lower and upper whiskers. Additionally, we calculate the InterQuartile Range (IQR), which measures the spread between the lower and upper quartiles, offering insight into the variability of the dataset.

Moreover, the experimentation includes a comparison with a variant of the LSC algorithm that does not incorporate the CMAB component. This comparison demonstrates the impact of the learning process on the algorithm's performance. Indeed, in this variant, the same local search operators are utilised, while employing the Round-Robin technique for selection. In this method, operators are selected in a fixed sequence, cycling through all available operators before repeating the cycle. This ensures a systematic exploration where each operator is applied periodically, providing a balanced use of different search strategies.

Further, we compare our solution method with the two-phase local search-based

method proposed by Paquay, Limbourg, and Schyns (2018), which employs two operators (jump and swap) in a random manner to enhance solutions. In contrast, our LSC algorithm utilises a broader set of operators used in the literature, which we have adapted to address the complexity of our specific problem.

The results of this comprehensive evaluation, including detailed statistical analyses for each method, are presented in Table 1. Additionally, Figure 1 showcases the best results found achieved using the DR/Softmax combination. It is worth noting, as in Paquay, Limbourg, and Schyns (2018), the primary objective in these experiments is to minimise unused space.



Boxplots Illustrating Filling Rates Achieved by the DR/Softmax Variant Across Different Sample Sizes, Ranging from 10 to 100.

Figure 1. Filling rates in percent per sample size. Variant: DR/Softmax.

The results derived from employing the LSC method without the CMAB underscore the significant role that the learning component plays in enhancing the performance of our solution method. This impact is observed consistently across various used estimators and exploration strategies.

The results also highlight that the DR estimator generally provides robust solutions by combining the advantages of both DM and IPS estimators. As a result, it can produce unbiased estimates even if one of the two estimators is inaccurately specified. Nevertheless, it is worth noting that, for smaller instances, the DR estimator may be unnecessarily computationally more demanding due to the need to fit both reward

and probability models. While, both the DM and IPS methods, when used individually, require fitting only one model. This makes them computationally less demanding compared to the DR estimator. In practical terms, for smaller instances, where the complexity and variability of the data might be lower, DM or IPS estimator might perform adequately well. This is because the potential for model inaccuracies (which the DR estimator guards against by combining two models) might be less of an issue in simpler or smaller-scale problems.

Additionally, the IPS estimator tends to be computationally faster, similar to findings from Paquay, Limbourg, and Schyns (2018), and more efficient than DM, as it only requires weighting each observed reward by the inverse probability of the action taken. On the other hand, combining the DM estimator with the Softmax Explorer or Online Cover algorithms tends to yield better solutions compared to IPS. The combination of Softmax Explorer and DM allows for exploring the search space with a probability proportional to the estimated quality. Meanwhile, the combination of Online Cover enables exploration by maintaining a collection of policies that approximates a covering distribution over policies suitable for both exploration and exploitation.

Table 1. Comparison summary (over the 1010 instances).

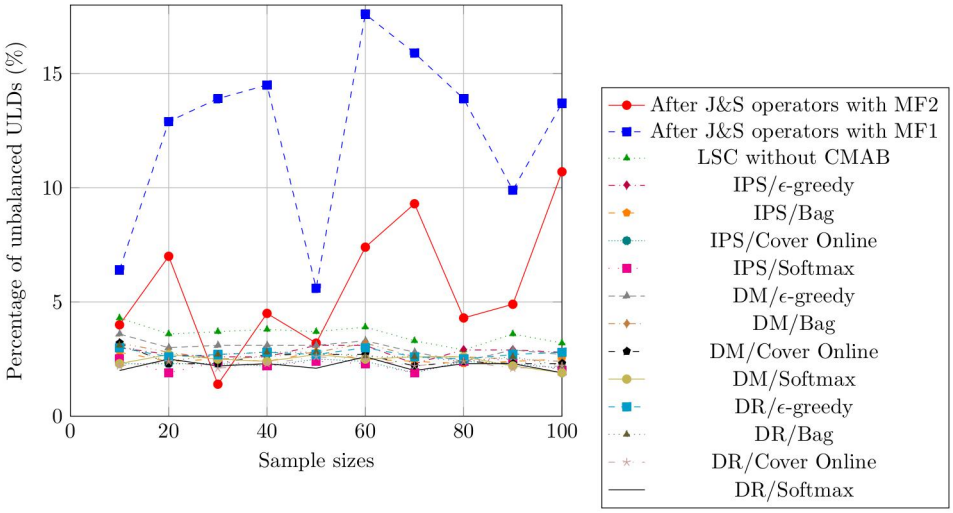
	Filling rate (%)						**Average**	
Resolution method	Median	Lower quartile	Upper quartile	Lower whisker	Upper whisker	IQR	Unused volume (m^3)	CPU times (s)
Paquay, Limbourg, and Schyns (2018)	41.6	29.6	50.1	5.5	66.6	20.5	65.7	13.3*
LSC without CMAB	62.5	58,2	66.7	45.4	79.5	8,5	58.5	11.1
IPS/ϵ-greedy	71.8	66.1	75.4	52.2	89.3	9.3	46.3	12.1
IPS/Bag	72.0	66.7	75.3	53.8	88.3	8.6	45.7	13.8
IPS/Cover Online	73.8	69.5	77.0	58.2	88.2	7.5	44.2	11.6
IPS/Softmax	73.5	68.5	76.2	57.1	87.6	7.6	42.4	13.4
DM/ϵ-greedy	68.8	64.1	73.2	50.6	86.7	9.0	49.3	15.9
DM/Bag	69.0	63.2	73.3	48.2	88.3	10.0	48.4	13.0
DM/Cover Online	70.8	65.3	74.3	51.8	87.8	9.0	47.3	18.8
DM/Softmax	72.5	67.0	76.0	53.5	89.6	9.0	45.1	19.1
DR/ϵ-greedy	73.8	69.5	77.0	58.4	88.1	7.4	44.9	36.4
DR/Bag	74.4	69.5	76.8	58.5	87.8	7.3	41.7	37.9
DR/Cover Online	75.1	72.2	77.9	63.7	86.4	5.7	41.5	32.4
DR/Softmax	75.3	71.7	77.6	62.9	86.4	5.9	40.2	35.7

* Scaled up to comply with the performance of the computer used in Paquay (2017) (`www.cpubenchmark.net`).

Figure 2 displays the average percentage of unbalanced ULDs for each sample size, with 30 instances included for each size.

The results of the 12 resolution methods are compared to the obtained results after applying Jump and Swap (J&S) operators with MF2 and after applying J&S operators with MF1: two variants of the method presented in Paquay, Limbourg, and Schyns (2018). In addition, we compared our results with those achieved using the solution method without CMAB. These comparisons reveal that the average percentage of unbalanced ULDs has been notably reduced to less than 4% across all 12 resolution methods examined. This reduction demonstrates a marked enhancement in both the balance and stability of the ULDs, thereby underscoring the efficacy of our newly developed methods, as well as the integral role of the learning component.

Figure 3 represents the average percentage deviation between the centre of gravity and the allowable area among the 1010 used ULDs. The deviation patterns observed with our new resolution methods, as well as the variant excluding CMAB, are more consistent than those reported in Paquay, Limbourg, and Schyns (2018). This is evidenced by the similar results we achieved for deviations to the left, right, front, and rear, indicating enhanced stability in all directions.



Unbalanced ULDs: Comparisons of the 12 resolution methods, method without CMAB, after applying J&S operators with MF2 and after applying J&S operators with MF1.

Figure 2. Average percentage of unbalanced ULDs per sample size (each sample size has 30 instances).

Furthermore, our algorithms are specifically designed to ensure that the CG remains within an acceptable area, which is a necessary condition for the safety and stability of the ULDs. The seemingly unbalanced states reported in the results represent configurations where the CG remains within permissible limits, even though packages may appear concentrated on one side. This is due to the dimensions, weights, and distribution of packages allowing for certain placement flexibility as long as the CG constraints are satisfied.

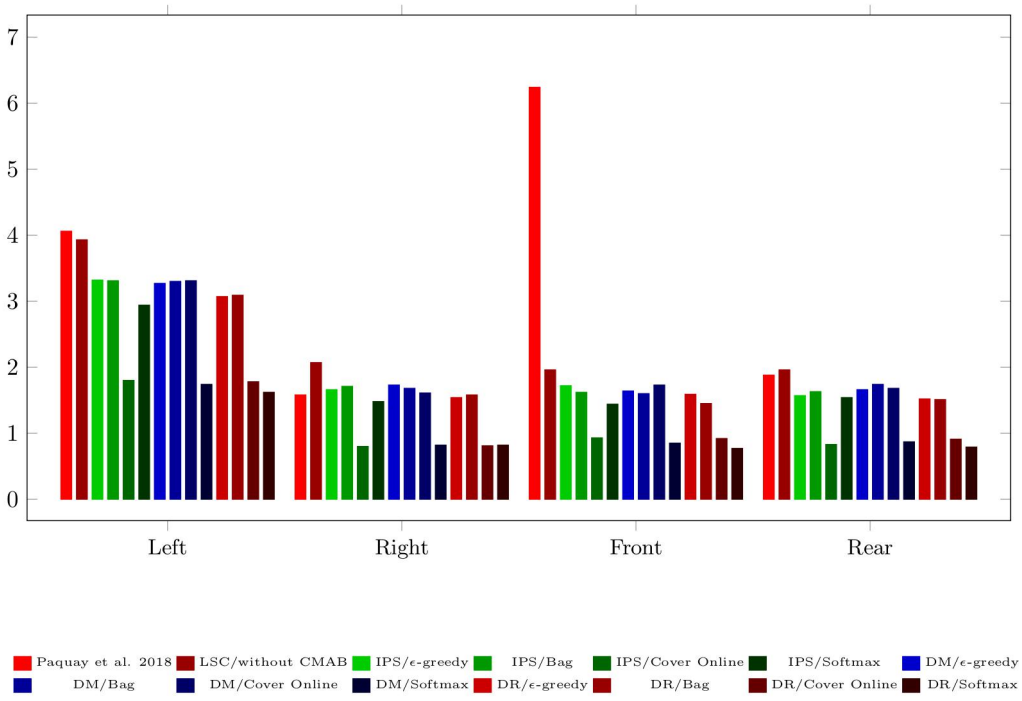
In the following experiments, DR/Softmax variant is used since it is generally more efficient in terms of filling rate, balancing, and deviations.

6. Managerial insights

In this section, we adapt the instances from Paquay, Limbourg, and Schyns (2018) to consider the objective function of minimising costs (3) and incorporate the new constraints. As in Bookbinder, Elhedhli, and Li (2015), we assume that the fixed cost of a ULD is $f_j = 4000 + 0.5(\bar{C}_j - 1000)$ and the over pivot rate is generated uniformly between $[7, 8]$ for $\bar{C}_j < 3000$ and between $[6.5, 7.5]$ for $\bar{C}_j \geq 3000$. The pivot weight is considered to be 90% of the maximum weight of a ULD.

Regarding the segregation matrix, the compatibility between nine classes is defined as reported in Wong and Ling (2020). In this paper, the authors utilised data from a cargo airline specialising in international air freight transportation services, including charter services, courier services, transportation of dangerous goods, express shipping, forwarding services, and live animal transportation. The airline operates a fleet of five A330-200F cargo freighters.

First, we solve a 10-box instance with grouping, positioning, and compatibility constraints and analyse the solution. We then use the 100-box sample with a size of 30



Percentages of deviations: Average of the percentages of deviations (%), between the centre gravity and the allowable area for the 12 variants.

Figure 3. Average of the percentages of deviations (%), between the centre gravity and the allowable area among the 1010 used ULDs.

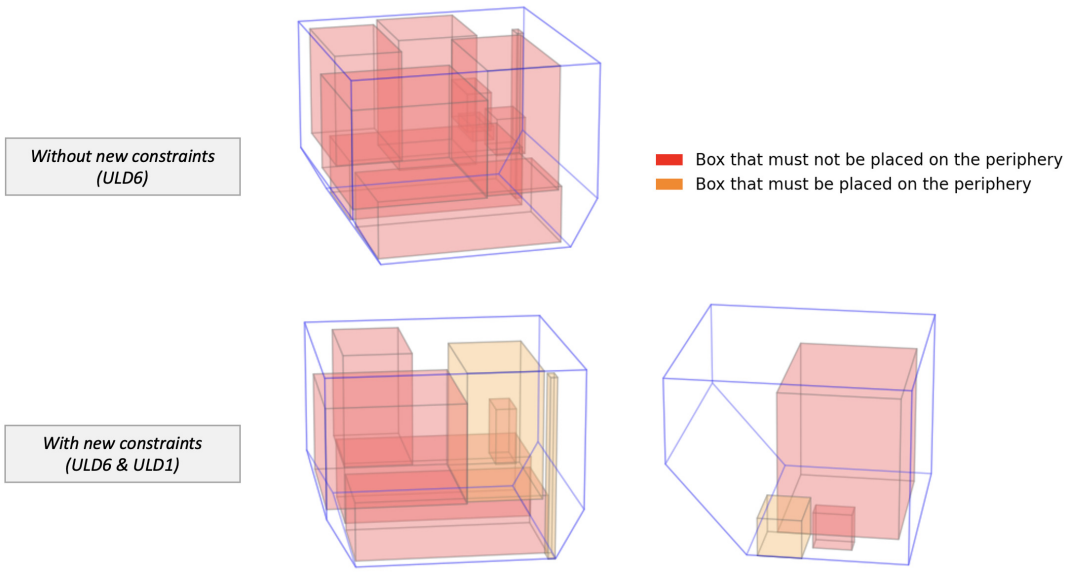
instances to assess the impact of the objective function, the number of groups, and the number of categories. Finally, we conduct computational experiments to determine how our findings can be generalised.

6.1. *Addressing Grouping, Positioning, and Compatibility Constraints*

In this experiment, we solved the model presented in Section 3. The characteristics of the considered instances are structured around introducing new parameters. The instance involves a set of 10 boxes inclusively grouped. Additionally, there are three boxes that exhibit incompatibility with the remainder and another three that should be placed on the periphery. This incompatibility necessitates using two separate ULDs (Figure 4) compared to one ULD observed in the solution provided by the basic model.

Before implementing the new constraints, all boxes could be loaded into the same ULD, as shown in Figure 4, achieving a filling rate of 91% and a cost of 6971. However, with the new constraints, two ULDs become necessary, altering the filling rate to 65% with a cost of using two ULDs of 10879.

Finally, to gain further insights into the performance of CMAB, we solved the mathematical model using CPLEX and compared its results with those of our CMAB heuristic on a set of 30 instances, each with 10 boxes. Table 2 presents the GAP values calculated between the lower bound and the solutions obtained by CPLEX and CMAB, respectively. As expected for an NP-hard problem, CPLEX provides optimal solutions (0% GAP) but at the expense of longer computational times. In contrast, our



3D visualization: When new constraints are not considered, all boxes are loaded into a single ULD; when they are considered, the boxes are loaded into two separate ULDs.

Figure 4. 3D visualisation of the results for the 10-box sample, with and without the new constraints.

CMAB heuristic achieves near-optimal solutions with GAP values consistently below 0.01%, while significantly reducing computational time.

Table 2. Comparison of CPLEX and CMAB performance for 10-box instances

	CPLEX						CMAB			
	10s limit		2min limit		3min limit		4s limit		10s limit	
# Instance	Opt.Gap(%)	RPD(%)	Opt.Gap(%)	RPD(%)	Opt.Gap(%)	RPD(%)	Gap(%)	RPD(%)	Gap(%)	RPD(%)
0	0.0156	0.245	0.0078	0.195	0	0	0.0052	0.208	0.0034	0.203
1	0.0170	0.263	0.0085	0.213	0	0	0.0068	0.227	0.0044	0.221
2	0.0182	0.278	0.0091	0.228	0	0	0.0073	0.243	0.0047	0.236
3	0.0158	0.248	0.0079	0.198	0	0	0.0061	0.211	0.0040	0.205
4	0.0152	0.240	0.0076	0.190	0	0	0.0059	0.203	0.0038	0.197
5	0.0166	0.258	0.0083	0.208	0	0	0.0065	0.221	0.0042	0.215
6	0.0176	0.270	0.0088	0.220	0	0	0.0070	0.234	0.0036	0.228
7	0.0144	0.230	0.0072	0.180	0	0	0.0056	0.192	0.0045	0.187
8	0.0162	0.253	0.0081	0.203	0	0	0.0063	0.216	0.0041	0.210
9	0.0178	0.273	0.0089	0.223	0	0	0.0071	0.237	0.0046	0.231
10	0.0170	0.263	0.0085	0.213	0	0	0.0067	0.226	0.0035	0.220
11	0.0140	0.225	0.0070	0.175	0	0	0.0054	0.187	0.0043	0.182
12	0.0174	0.268	0.0087	0.218	0	0	0.0069	0.232	0.0039	0.226
13	0.0180	0.275	0.0090	0.225	0	0	0.0072	0.240	0.0047	0.233
14	0.0156	0.245	0.0078	0.195	0	0	0.0060	0.208	0.0043	0.202
15	0.0168	0.260	0.0084	0.210	0	0	0.0066	0.224	0.0042	0.218
16	0.0150	0.238	0.0075	0.188	0	0	0.0058	0.200	0.0028	0.195
17	0.0164	0.255	0.0082	0.205	0	0	0.0064	0.219	0.0040	0.213
18	0.0186	0.283	0.0093	0.233	0	0	0.0075	0.248	0.0037	0.241
19	0.0160	0.250	0.0080	0.200	0	0	0.0062	0.213	0.0048	0.207
20	0.0148	0.235	0.0074	0.185	0	0	0.0057	0.197	0.0037	0.192
21	0.0184	0.280	0.0092	0.230	0	0	0.0074	0.245	0.0048	0.238
22	0.0142	0.228	0.0071	0.178	0	0	0.0055	0.190	0.0026	0.184
23	0.0178	0.273	0.0089	0.223	0	0	0.0071	0.237	0.0041	0.231
24	0.0162	0.253	0.0081	0.203	0	0	0.0063	0.216	0.0049	0.210
25	0.0174	0.268	0.0087	0.218	0	0	0.0069	0.232	0.0043	0.226
26	0.0188	0.285	0.0094	0.235	0	0	0.0076	0.250	0.0047	0.244
27	0.0156	0.245	0.0078	0.195	0	0	0.0060	0.208	0.0043	0.202
28	0.0170	0.263	0.0085	0.213	0	0	0.0067	0.226	0.0047	0.220
29	0.0180	0.275	0.0090	0.225	0	0	0.0072	0.240	0.0043	0.234
Mean	0.0166	0.258	0.0083	0.208	0	0	0.0066	0.221	0.0041	0.216
Std Dev	0.0014	0.018	0.0007	0.018	0	0	0.0007	0.019	0.0006	0.018

Opt.Gap = (Upper Bound - Lower Bound) / Upper Bound × 100
Gap = (Solution- Lower Bound) / Solution × 100
RPD = (Solution - Best Solution) / Best Solution × 100
Best Solution: optimal solution found by CPLEX after 3 minutes

6.2. *Impact of the objective function*

We show the impact of different objective functions: minimising unused space versus minimising costs. Additionally, we explore variations in the cost parameters to assess their influence on these objectives.

6.2.1. Impact of density

To analyse the impact of density on the two objective functions, minimising unused space (based on volume) and minimising costs (based on weight), we conducted experiments across four scenarios. In these scenarios, the total volume of boxes remained constant at 20 m^3, while the total weight of boxes to be loaded into ULDs varied, ranging from 7800 kg to 23000 kg. The ULDs' characteristics are provided in Table 3.

Table 3. Description of the parameters of the proposed ULDs.

IATA code	ULD type (j)	Max. cap. [kg]	Vol. [m^3]	f_j	$\bar{c}_j$	c_j [/kg]
LD1	1	1518	5	4183	1159	7.1
LD6	2	2945	9.1	4846	1362	7.6
LD11	3	2991	7.4	8378	2101	6.6
PA	4	5890	20	6151	1471	7.2
PG	5	10840	31.1	4825	1528	7.0
PM	6	6680	18.9	3500	1885	6.7

In each scenario, the total number of boxes remained constant and were divided into three weight categories to represent different densities:

- Light Boxes: Weight $<$ 100 kg
- Medium Boxes: 100 kg $\leq$ Weight $<$ 1000 kg
- Heavy Boxes: Weight $\geq$ 1000 kg

By varying the total weight and weight distribution across these categories, we aimed to evaluate the performance of the proposed methods under different density conditions, which can have a significant impact on the packing solutions and associated costs. The results are reported in Table 4.

Table 4. Comparison of objective functions based on various density scenarios

Scenario	Total weight [kg]	Objective function	Costs	Filling rate (%)	ULD type
1	7800	Unused volume	11158	92	1,6
		Costs	10770	51	6,6
2	13000	Unused volume	16833	87	3,5
		Costs	12117	57	1,6
3	19000	Unused volume	29828	75	3,3
		Costs	11738	67	5,6
4	23000	Unused volume	35671	89	1,3,3,6
		Costs	23537	70	2,4,4

From Table 4, we can observe that minimising unused space leads to higher filling rates but can result in higher costs due to the use of a larger number of ULDs and potential costs from over-pivot weights. On the other hand, minimising costs often involves consolidating boxes into a smaller number of larger ULDs, which can result in more unused space and lower filling rates, but typically lower overall costs.

6.2.2. Impact of costs

The total cost is influenced by the fixed reservation cost of ULDs, the under-pivot rate, and the over-pivot rate per weight unit (c_j). Specifically, we sought to understand how varying the over-pivot rate c_j across different pivot weight thresholds (C_j) impacts the filling rate and the total cost.

Table 5. Comparison of objective functions based on various over-pivot rate scenarios

c_j	Minimise unused volume			Minimise cost		
	Total cost	Filling rate (%)	ULD type	Total cost	Filling rate (%)	ULD type
6.5 for $C_j \geq 3000$; 7.5 for $C_j < 3000$	18052	94	2, 5, 6	14753	62	3
7 for $C_j \geq 3000$; 8 for $C_j < 3000$	25210	77	2, 2, 5, 6	18652	47	3, 4
7.5 for $C_j \geq 3000$; 8.5 for $C_j < 3000$	30022	71	2, 2, 5, 6, 6	24628	44	3, 4, 5

The experiment demonstrates the impact of the objective function on the results. Table 5 underscores that higher c_j values discourage exceeding the pivot weight and decrease filling rates. By thoroughly analysing these results, we can derive valuable insights into the optimal c_j value ranges that best align with the specific priorities and constraints of a given air cargo operation. These insights can guide the development of adaptive strategies that dynamically adjust the over-pivot rate based on the underlying objectives, resource availability, and operational demands, ultimately enabling more efficient and cost-effective air cargo palletisation solutions.

6.3. *Impact of the number of groups*

The impact of grouping is represented in batches, i.e., the number of boxes per group. The objective is to see how filling rates and costs evolve when 10% to 100% of boxes belong to a batch. Figure 5 provides the relative change for each KPI using the following formula:

$$\frac{KPI_{\omega} - KPI_{\omega=1}}{KPI_{\omega=1}} \tag{21}$$

As the percentage of boxes per batch increases, the size of each group also increases. This leads to potentially more efficient utilisation of ULD volume, as larger groups of homogeneous or compatible items can be more easily arranged to utilise the available space effectively.

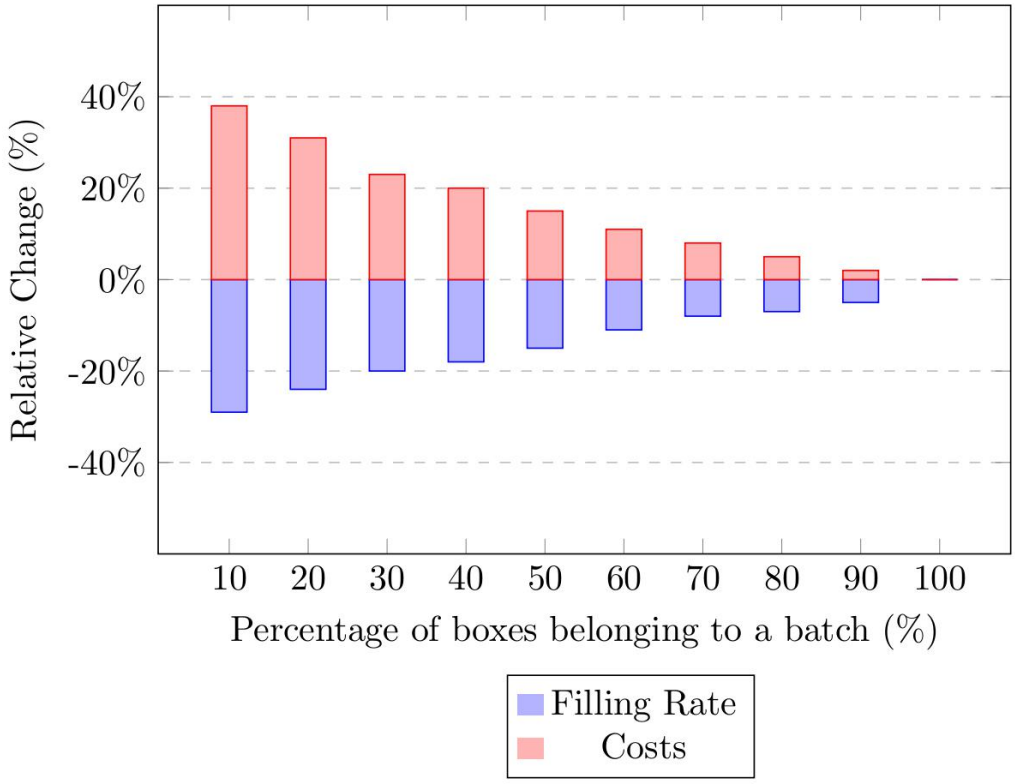
6.4. *Impact of the number of categories*

In this experiment, for the sake of theoretical assessment of the heuristic's performance, we considered a number of categories λ varying between 10 to 100. Also, we computed the average filling rate and costs. We calculated the relative change for each KPI using the following formula:

$$\frac{KPI_{\lambda} - KPI_{\lambda=1}}{KPI_{\lambda=1}} \tag{22}$$

Separating boxes into multiple categories inherently reduces packing flexibility. Each category may have specific handling and storage requirements or compatibility issues that restrict how and with which other items they can be co-located. This limitation often necessitates using additional ULDs, which increases costs and decreases overall filling rates. In particular, if each box is categorised uniquely (i.e., one category per box), this represents the least flexible packing scenario.

While effective grouping strategies can significantly enhance filling rates and reduce costs, over-categorisation can lead to increased complexity, reduced flexibility, and higher operational costs. Therefore, while it is crucial to respect safety and compatibility requirements necessitated by categorisation, reducing the number of categories



Impact of groups: Relative change of the filling rate and costs according to the number of groups.

Figure 5. Impact of the number of groups on KPIs.

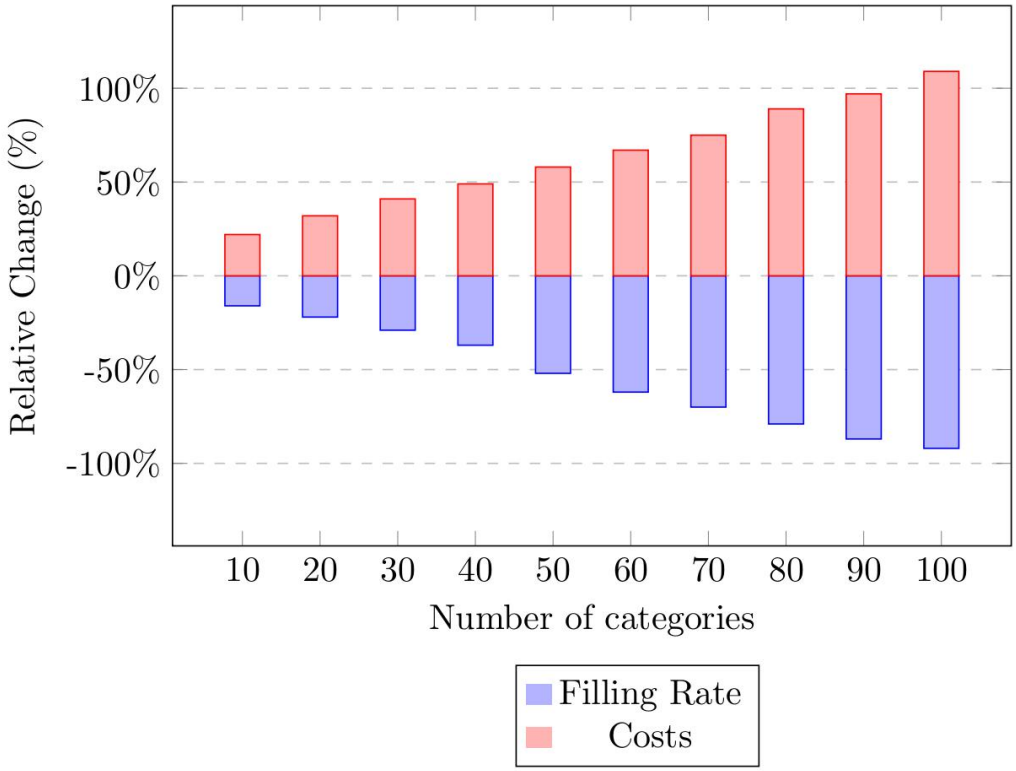
is also pertinent to avoid overly restrictive packing conditions.

It is worth noting that the number of categories is generally quite limited. Dangerous goods are typically classified into different hazard classes, and there are specific regulations and requirements for their handling and transportation (Feng, Li, and Shen 2015). Moreover, the segregation of incompatible dangerous goods can vary depending on factors such as the type of aircraft and the location of the ULD within the aircraft (Brandt and Nickel 2019). However, even with a limited number of categories, the impact on packing flexibility and costs can be significant, as demonstrated in our experiments.

6.5. *Impact of instance sizes on CPU times*

Table 6 displays the CPU times of DR/Softmax under varying settings. The number of boxes influences the CPU times, the specifications of groups and categories, and any periphery requirements.

In the most straightforward scenarios, the algorithm can handle up to 100,000 boxes assigned to a single group and category. Furthermore, it can efficiently solve configurations with a small to medium number of boxes, even with increased group, category, and periphery constraints. The complexity markedly increases for configurations incorporating 100,000 boxes along with an elevated number of groups and categories.



Impact of categories: Relative change of the filling rate and costs according to the number of categories.

Figure 6. Impact of the number of categories on KPIs.

Table 6. Impact of instance sizes on CPU times.

# boxes	# groups	# categories	# periphery	CPU (s)
100	1	1	0	35
500	1	1	0	83
1000	1	1	0	190
100000	1	1	0	365
100	20	1	0	48
500	20	1	0	158
1000	20	1	0	443
100000	20	1	0	1618
100	40	1	0	42
500	40	1	0	225
1000	40	1	0	508
100000	40	1	0	2770
100	60	1	0	63
500	60	1	0	279
1000	60	1	0	870
100000	60	1	0	3431
100	1	20	0	41
500	1	20	0	160
1000	1	20	0	549
100000	1	20	0	1834
100	1	40	0	57
500	1	40	0	284
1000	1	40	0	809
100000	1	40	0	2859
100	1	60	0	71
500	1	60	0	309
1000	1	60	0	1085
100000	1	60	0	3924
100	1	1	10	31
500	1	1	10	164
1000	1	1	10	234
100000	1	1	10	613
100	1	1	20	38
500	1	1	20	222
1000	1	1	20	417
100000	1	1	20	1293
100	20	20	20	481
500	20	20	20	2085
1000	20	20	20	3522
100000	20	20	20	6319

This showcases the intricate interplay between the variables in determining the computational demand of the problem. However, the CPU time remains less than two hours.

7. Conclusion

In this paper, we conducted experiments employing 12 contextual bandits-guided local search heuristics to address the air cargo palletization problem. To ensure comprehensive analysis, we evaluated the performance of our methods using three different estimators (IPS, DM, and DR) and four distinct exploration algorithms (ϵ-greedy, Bag, Online Cover, and Softmax Explorer). Through rigorous validation, we benchmarked our approaches against a variant that does not incorporate the learning component and against the results achieved by previous researchers. Our findings demonstrated the effectiveness of our methods. We showed that the LSC outperformed the other heuristics. Further, the LSC with the DR estimator generally provided robust solutions by combining the advantages of both DM and IPS estimators, yet it may be computationally more demanding.

We also conducted experiments utilising two distinct objective functions, namely the unused volume and the costs, to emphasise the importance of cost minimisation. We successfully tackled instances incorporating grouping, positioning, and compatibility constraints, aiming to evaluate the influence of these constraints on our study and to evaluate the broader applicability of our findings.

As a research outlook, this work could be extended by exploring the trade-off between the rising costs of grouping items of the same shipment and the handling costs involved in disassembling certain ULDs. Considering this trade-off, a more comprehensive understanding of the cost dynamics in cargo operations can be achieved. Nevertheless, depicting such an aspect requires a drastic mathematical transformation of the actual framework, as several stakeholders might be involved. Furthermore, the segregation constraints can vary depending on factors such as the ULD's position within the aircraft, the specific aircraft type, and the nature of the service provided (e.g., cargo only). To address these complexities comprehensively, an integrated model that incorporates multiple decision-makers could be developed, considering both the air cargo palletisation problem and the weight and balance problem. Such a model would enable a holistic approach to optimising air cargo operations.

Moreover, this study can be extended to incorporate additional metaheuristic comparisons and further sensitivity analyses on an expanded set of constraints. This will ensure a comprehensive benchmarking across a wider spectrum of operational scenarios, further validating the LSC's adaptability and efficiency. Additionally, it could be interesting to incorporate more complex operators that might bring additional benefits to the solution quality.

Finally, in our current work, we assume a priori knowledge of the boxes to be packed. However, exploring an online version of the problem is important, where the boxes need to be packed without prior knowledge of the subsequent boxes. In this online scenario, each box can only be allocated to a ULD with a build-up period that is at least equal to the time when the item becomes available for build-up. By considering this online variant, we can simulate real-world packing scenarios more accurately, where decisions must be made in real-time without complete knowledge of future items, leading to a more practical and robust solution for the air cargo palletisation problem.

Acknowledgments

The authors thank the editor-in-chief, the associate editor and three anonymous referees for their constructive comments and encouragements that have helped improve our paper greatly.

Disclosure Statement

The authors have no conflicts of interest to disclose.

Data availability statement

The data that support the findings of this study are openly available in ORBi at `https://hdl.handle.net/2268/310078`.

References

Abeyratne, Ruwantissa. 2018. *Regulation of Air Cargo*, 129–155. Cham: Springer International Publishing. `https://doi.org/10.1007/978-3-319-92489-2_4`.

Achamrah, Fatima Ezzahra, Fouad Riane, and Sabine Limbourg. 2021. "Solving inventory routing with transshipment and substitution under dynamic and stochastic demands using genetic algorithm and deep reinforcement learning." *International Journal of Production Research* 1–18.

Achamrah, Fatima Ezzahra, Fouad Riane, Evren Sahin, and Sabine Limbourg. 2022. "An Artificial-Immune-System-Based Algorithm Enhanced with Deep Reinforcement Learning for Solving Returnable Transport Item Problems." *Sustainability* 14 (10). https://doi.org/10.3390/su14105805, `https://www.mdpi.com/2071-1050/14/10/5805`.

Agrawal, Shipra, and Navin Goyal. 2013. "Thompson sampling for contextual bandits with linear payoffs." In *International conference on machine learning*, 127–135. PMLR.

Alvarez-Valdes, R., F. Parreño, and J. M. Tamarit. 2015. "Lower bounds for three-dimensional multiple-bin-size bin packing problems." *OR Spectrum* 37 (1): 49–74. https://doi.org/10.1007/s00291-013-0347-2.

Alvarez-Valdés, Ramón, Francisco Parreño, and José Manuel Tamarit. 2013. "A GRASP/Path relinking algorithm for two-and three-dimensional multiple bin-size bin packing problems." *Computers & Operations Research* 40 (12): 3081–3090.

Angelelli, Enrico, Claudia Archetti, and Lorenzo Peirano. 2022. "A heuristic algorithm for the Air Transport Unit Consolidation Problem." *arXiv* `https://arxiv.org/abs/2210.06004`.

Auer, Peter. 2002. "Using confidence bounds for exploitation-exploration trade-offs." *Journal of Machine Learning Research* 3 (Nov): 397–422.

Battarra, Maria, Michele Monaci, and Daniele Vigo. 2009. "An adaptive guidance approach for the heuristic solution of a minimum multiple trip vehicle routing problem." *Computers & Operations Research* 36 (11): 3041–3050.

Baxter, Glenn, and Kyriakos Kourousis. 2015. "Temperature Controlled Aircraft Unit Load Devices: The Technological Response to Growing Global Air Cargo Cool Chain Requirements." *Journal of technology management & innovation* 10 (1): 157–172. https://doi.org/10.4067/S0718-27242015000100012.

Bietti, Alberto, Alekh Agarwal, and John Langford. 2018. "A Contextual Bandit Bake-off." *arXiv* https://doi.org/10.48550/ARXIV.1802.04064, `https://arxiv.org/abs/1802.04064`.

Bombelli, Alessandro, and Stefano Fazi. 2022. "The ground handler dock capacitated pickup and delivery problem with time windows: A collaborative framework for air cargo operations." *Transportation Research Part E: Logistics and Transportation Review* 159: 102603. https://doi.org/https://doi.org/10.1016/j.tre.2022.102603, https://www.sciencedirect.com/science/article/pii/S1366554522000011.

Bookbinder, James H., Samir Elhedhli, and Zichao Li. 2015. "The air-cargo consolidation problem with pivot weight: Models and solution methods." *Computers & Operations Research* 59: 22–32. https://doi.org/https://doi.org/10.1016/j.cor.2014.11.015, https://www.sciencedirect.com/science/article/pii/S0305054814003190.

Bouneffouf, Djallel, Irina Rish, and Charu Aggarwal. 2020. "Survey on applications of multi-armed and contextual bandits." In *2020 IEEE Congress on Evolutionary Computation (CEC)*, 1–8. IEEE.

Brandt, Felix. 2017. "The Air Cargo Load Planning Problem." [Online; accessed 12. Aug. 2022].

Brandt, Felix, and Stefan Nickel. 2019. "The air cargo load planning problem - a consolidated problem definition and literature review on related problems." *European Journal of Operational Research* 275 (2): 399–410. https://doi.org/https://doi.org/10.1016/j.ejor.2018.07.013, https://www.sciencedirect.com/science/article/pii/S0377221718306180.

Chan, Felix T.S., Rajat Bhagwat, N. Kumar, M.K. Tiwari, and Philip Lam. 2006. "Development of a decision support system for air-cargo pallets loading problem: A case study." *Expert Systems with Applications* 31 (3): 472–485. https://doi.org/https://doi.org/10.1016/j.eswa.2005.09.057, https://www.sciencedirect.com/science/article/pii/S0957417405002617.

Chapelle, Olivier, and Lihong Li. 2011. "An empirical evaluation of thompson sampling." *Advances in neural information processing systems* 24.

Chen, Xi, Yining Wang, and Yuan Zhou. 2020. "Dynamic assortment optimization with changing contextual information." *Journal of machine learning research* 21 (216): 1–44.

Cortes, David. 2018. "Adapting multi-armed bandits policies to contextual bandits scenarios." *arXiv* https://doi.org/10.48550/ARXIV.1811.04383, https://arxiv.org/abs/1811.04383.

Crainic, Teodor Gabriel, Guido Perboli, Walter Rei, and Roberto Tadei. 2011. "Efficient lower bounds and heuristics for the variable cost and size bin packing problem." *Computers & Operations Research* 38 (11): 1474–1482. https://doi.org/https://doi.org/10.1016/j.cor.2011.01.001, https://www.sciencedirect.com/science/article/pii/S0305054811000049.

Crainic, Teodor Gabriel, Guido Perboli, and Roberto Tadei. 2008. "Extreme Point-Based Heuristics for Three-Dimensional Bin Packing." *INFORMS Journal on Computing* 20 (3): 368–384. https://doi.org/10.1287/ijoc.1070.0250, https://doi.org/10.1287/ijoc.1070.0250.

Dangerous goods panel. 1995. "Development of recommendations for amendments to the Technical Instructions for the Safe Transport of Dangerous Goods by Air for incorporation in the 2007-2008 Edition." [Online; accessed 15. Aug. 2022], https://www.icao.int/safety/DangerousGoods/DGP20\%20Working\%20Papers/DGP.20.IP.001.pdf.

Desai, Jitamitra, Sandeep Srivathsan, Woen Yon Lai, Liqun Li, and Chuhang Yu. 2023. "An optimization-based decision support tool for air cargo loading." *Computers & Industrial Engineering* 175: 108816. https://doi.org/https://doi.org/10.1016/j.cie.2022.108816, https://www.sciencedirect.com/science/article/pii/S036083522200804X.

Dudik, Miroslav, John Langford, and Lihong Li. 2011. "Doubly Robust Policy Evaluation and Learning." https://doi.org/10.48550/ARXIV.1103.4601, https://arxiv.org/abs/1103.4601.

Elhedhli, Samir, Fatma Gzara, and Burak Yildiz. 2019. "Three-dimensional bin packing and mixed-case palletization." *INFORMS Journal on Optimization* 1 (4): 323–352.

Feng, Bo, Yanzhi Li, and Zuo-Jun Max Shen. 2015. "Air cargo operations: Literature review

and comparison with practices." *Transportation Research Part C: Emerging Technologies* 56: 263–280.

Hamdi-Dhaoui, Khaoula, Nacima Labadie, and Alice Yalaoui. 2014. "The bi-objective two-dimensional loading vehicle routing problem with partial conflicts." *International Journal of Production Research* 52 (19): 5565–5582. https://doi.org/10.1080/00207543.2014.885181, https://doi.org/10.1080/00207543.2014.885181.

Huang, Kuancheng, and Wenhou Chi. 2007. "A Lagrangian relaxation based heuristic for the consolidation problem of airfreight forwarders." *Transportation Research Part C: Emerging Technologies* 15 (4): 235–245. Modeling and Optimization for Transportation Logistics, https://doi.org/https://doi.org/10.1016/j.trc.2006.08.006, https://www.sciencedirect.com/science/article/pii/S0968090X06000593.

IATA. 2021. "Air Cargo Market Analysis." https://www.iata.org/en/iata-repository/publications/economic-reports/air-freight-monthly-analysis---december-2021/.

IATA. 2023a. *European Air Cargo Programme Handbook*. International Air Transport Association (IATA). Accessed: Aug 21, 2024, https://www.iata.org/contentassets/e9cb5a72b88f4f68a5cfc572a50b60c9/eacph-european-air-cargo-programme-handbook.pdf.

IATA. 2023b. "What Types of Cargo are Transported by Air?" May. [Online; accessed 31. May 2023], https://www.iata.org/en/publications/newsletters/iata-knowledge-hub/what-types-of-cargo-are-transported-by-air.

ICAO. 2017. "Dangerous Goods Panel, Working Group Meeting (DGP-WG/17)." https://www.icao.int/safety/DangerousGoods/WG17/DGPWG.17.WP.001.2.en.pdf.

Juan, Angel A, Peter Keenan, Rafael Martí, Seán McGarraghy, Javier Panadero, Paula Carroll, and Diego Oliva. 2021. "A review of the role of heuristics in stochastic optimisation: From metaheuristics to learnheuristics." *Annals of Operations Research* 1–31.

Jun, Sungbum, Chul Hun Choi, and Seokcheon Lee. 2022. "Scheduling of autonomous mobile robots with conflict-free routes utilising contextual-bandit-based local search." *International Journal of Production Research* 60 (13): 4090–4116. https://doi.org/10.1080/00207543.2022.2063085, https://doi.org/10.1080/00207543.2022.2063085.

Lee, No-San, Philipp Gabriel Mazur, Moritz Bittner, and Detlef Schoder. 2021. "An Intelligent Decision-Support System for Air Cargo Palletizing." January. [Online; accessed 10. Aug. 2022], https://scholarspace.manoa.hawaii.edu/items/85427938-716c-4d49-9dda-46deac276745.

Li, Tzuu-Hseng S., Chih-Yin Liu, Ping-Huan Kuo, Nien-Chu Fang, Cheng-Hui Li, Ching-Wen Cheng, Cheng-Ying Hsieh, Li-Fan Wu, Jie-Jhong Liang, and Chih-Yen Chen. 2017. "A Three-Dimensional Adaptive PSO-Based Packing Algorithm for an IoT-Based Automated e-Fulfillment Packaging System." *IEEE Access* 5: 9188–9205. https://doi.org/10.1109/ACCESS.2017.2702715.

Li, Zichao, James H. Bookbinder, and Samir Elhedhli. 2012. "Optimal shipment decisions for an airfreight forwarder: Formulation and solution methods." *Transportation Research Part C: Emerging Technologies* 21 (1): 17–30. https://doi.org/https://doi.org/10.1016/j.trc.2011.08.001, https://www.sciencedirect.com/science/article/pii/S0968090X11001112.

Limbourg, S, M Schyns, and G Laporte. 2012. "Automatic aircraft cargo load planning." *Journal of the Operational Research Society* 63 (9): 1271–1283. https://doi.org/10.1057/jors.2011.134, https://doi.org/10.1057/jors.2011.134.

Loadstar, The. 2021. "IATA's poorly determined cargo load factors leave observers stupefied." https://theloadstar.com/iatas-poorly-determined-cargo-load-factors-leave-observers-stupefied/.

Lu, Tyler, Dávid Pál, and Martin Pál. 2010. "Contextual multi-armed bandits." In *Proceedings of the Thirteenth international conference on Artificial Intelligence and Statistics*, 485–492. JMLR Workshop and Conference Proceedings.

Lurkin, Virginie, and Michaël Schyns. 2015. "The Airline Container Loading Problem with pickup and delivery." *European Journal of Operational Research* 244 (3): 955–965.

https://doi.org/https://doi.org/10.1016/j.ejor.2015.02.027, https://www.sciencedirect.com/science/article/pii/S0377221715001289.

Nascimento, Oliviana Xavier, Thiago Alves de Queiroz, and Leonardo Junqueira. 2021. "Practical constraints in the container loading problem: Comprehensive formulations and exact algorithm." *Computers & Operations Research* 128: 105186. https://doi.org/https://doi.org/10.1016/j.cor.2020.105186, https://www.sciencedirect.com/science/article/pii/S0305054820303038.

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

Paquay, Célia. 2017. "The three-dimensional rectangular Multiple Bin Size Bin Packing Problem with transportation constraints: A case study in the field of air transportation." PhD diss., ULiège - Université de Liège.

Paquay, Célia, Sabine Limbourg, and Michaël Schyns. 2018. "A tailored two-phase constructive heuristic for the three-dimensional Multiple Bin Size Bin Packing Problem with transportation constraints." *European Journal of Operational Research* 267 (1): 52–64. https://doi.org/https://doi.org/10.1016/j.ejor.2017.11.010, https://www.sciencedirect.com/science/article/pii/S0377221717310214.

Paquay, Célia, Sabine Limbourg, Michaël Schyns, and José Fernando Oliveira. 2018. "MIP-based constructive heuristics for the three-dimensional Bin Packing Problem with transportation constraints." *International Journal of Production Research* 56 (4): 1581–1592. https://doi.org/10.1080/00207543.2017.1355577, https://doi.org/10.1080/00207543.2017.1355577.

Pollaris, Hanne, Kris Braekers, An Caris, Gerrit K Janssens, and Sabine Limbourg. 2015. "Vehicle routing problems with loading constraints: state-of-the-art and future directions." *OR Spectrum* 37 (2): 297–330.

Rodrigue, Jean-Paul. 2020. *The Geography of Transport Systems*. Andover, England, UK: Taylor & Francis.

Sui, Guoxin, and Yong Yu. 2020. "Bayesian Contextual Bandits for Hyper Parameter Optimization." *IEEE Access* 8: 42971–42979. https://doi.org/10.1109/ACCESS.2020.2977129.

Sun, Jiankun, Dennis J Zhang, Haoyuan Hu, and Jan A Van Mieghem. 2022. "Predicting human discretion to adjust algorithmic prescription: A large-scale field experiment in warehouse operations." *Management Science* 68 (2): 846–865.

The International Air Cargo Association (TIACA). 2020. "TIACA Calls for the Use of the Dynamic Load Factor Methodology to Measure Air Cargo Performance." https://tiaca.org/tiaca-calls-for-the-use-of-the-dynamic-load-factor-methodology-to-measure-air-cargo-performance/.

Thompson, William R. 1933. "Biometrika trust." *Biometrika* 25 (3/4): 285–294.

Tresca, Giulia, Graziana Cavone, Raffaele Carli, Antonio Cerviotti, and Mariagrazia Dotoli. 2022. "Automating Bin Packing: A Layer Building Matheuristics for Cost Effective Logistics." *IEEE Transactions on Automation Science and Engineering* 19 (3): 1599–1613. https://doi.org/10.1109/TASE.2022.3177422.

Tseremoglou, Iordanis, Alessandro Bombelli, and Bruno F. Santos. 2022. "A combined forecasting and packing model for air cargo loading: A risk-averse framework." *Transportation Research Part E: Logistics and Transportation Review* 158: 102579. https://doi.org/https://doi.org/10.1016/j.tre.2021.102579, https://www.sciencedirect.com/science/article/pii/S1366554521003367.

UN. 2019. "Transport of Dangerous Goods ." https://unece.org/fileadmin/DAM/trans/danger/publi/unrec/rev21/ST-SG-AC10-1r21e_Vol1_WEB.pdf.

Van Asch, Thomas. 2022. "Chapter 16 - The forwarders' power play effect on competition in the air cargo industry." In *The Air Transportation Industry*, edited by Rosário Macário and Eddy Van de Voorde, Contemporary Issues in Air Transport, 361–381. Elsevier. https://www.sciencedirect.com/science/article/pii/B9780323915229000038.

Wang, Yu-Xiang, Alekh Agarwal, and Miroslav Dudık. 2017. "Optimal and adaptive off-policy

evaluation in contextual bandits." 3589–3597.

Wäscher, Gerhard, Heike Haußner, and Holger Schumann. 2007. "An improved typology of cutting and packing problems." *European Journal of Operational Research* 183 (3): 1109–1130.

Wong, Eugene Y.C., and Kev K. T. Ling. 2020. "A Mixed Integer Programming Approach to Air Cargo Load Planning with Multiple Aircraft Configurations and Dangerous Goods." 123–130. https://doi.org/10.1109/ICFIE50845.2020.9266727.

Yu, Tong, Branislav Kveton, and Ole J Mengshoel. 2017. "Thompson sampling for optimizing stochastic local search." In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 493–510. Springer.

Zhao, X., Y. Yuan, Y. Dong, and R. Zhao. 2021. "Optimization approach to the aircraft weight and balance problem with the centre of gravity envelope constraints." *IET Intelligent Transport Systems* 15 (10): 1269–1286.

Zheng, Jiongzhi, Kun He, Jianrong Zhou, Yan Jin, Chu-Min Li, and Felip Manya. 2022. "Bandmaxsat: A local search MaxSAT solver with multi-armed bandit." *arXiv preprint arXiv:2201.05544* .