Computers in Industry 164 (2025) 104202

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.sciencedirect.com/journal/computers-in-industry

A deep reinforcement learning approach for online and concurrent 3D bin packing optimisation with bin replacement strategies

Y.P. Tsang [a,*], D.Y. Mo [b], K.T. Chung [a], C.K.M. Lee [a]

[a] Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong
[b] Department of Supply Chain and Information Management, The Hang Seng University of Hong Kong, Shatin, Hong Kong

ARTICLE INFO

Keywords:
Deep reinforcement learning
Online optimisation
3D bin packing problem
Dual bin strategy
Robotic warehouse

ABSTRACT

In the realm of robotic palletisation, the quest for optimal space utilization remains vital but also presents a critical challenge, particularly due to the constraints of decision complexity and the need for real-time decision-making without complete prior information. The widely adopted rule-based heuristics approaches were ease to use, but failed to adapt dynamically to the complex and changing landscape of online 3D bin packing. This study is motivated by the need for a system that is both more agile and intelligent, capable of managing the intricacies of dual-bin scenarios and the variable inflow of items. This study introduces a novel deep reinforcement learning (DRL) optimiser, employing a double deep Q-network (DDQN) to obtain optimal packing policies in an online environment with two proposed bin replacement strategies. This approach surpasses the limitations of previous methods by facilitating the simultaneous management of multiple bins and enabling on-the-fly adjustments to decisions based on limited prior knowledge. In a case study involving a logistics company, the proposed optimizer demonstrated a significant improvement in average space utilization across various lookahead scenarios, outperforming traditional heuristics in simulation experiments. The proposed optimiser contributes significantly to the economic and environmental sustainability of robotic warehouses, positioning itself as a cornerstone for the future of smart logistics.

1. Introduction

In recent decades, the research on three-dimensional (3D) bin packing optimization has garnered considerable attention for determining the optimal strategy for packing items into a finite number of 3D spaces. Such research underpins numerous practical industrial applications, including container loading, palletization, and the packaging of items throughout the supply chain journey (Paquay et al., 2018; Kurpel et al., 2020; Gajda et al., 2022). By maximizing space utilization, packed goods occupy less space, enabling more economical transportation and consequently reducing material use and energy consumption. In essence, effective 3D bin packing optimization emerges as a pivotal driver for enhancing the sustainability of supply chains. Although various software packages address 3D bin packing optimization, most of them, such as 3DBinPacking, primarily provide offline optimal solutions with the requirement of pre-configured item sizes and shapes. In contrast, certain practical situations, notably within fully robotic warehouses, necessitate online 3D bin packing that can adapt with limited knowledge of inflow items, thereby engendering a more resilient and smooth operational process (Yang et al., 2023). Through the integration of dimension measurement systems, the dimensions of goods can be automatically gauged to buttress the optimization process (Yang et al., 2023). Consequently, online 3D bin packing can eliminate the need for a staging area, thereby streamlining operations within robotic e-fulfilment process. In this regard, online 3D bin packing optimization is fundamentally critical for the advancement of robotic warehouses. However, even in the offline environment, the 3D bin packing problem is a combinatorial optimization problem, and the solution time increases exponentially as the number of items grows. This presents an additional challenge for real-time decision-making in a dynamic online environment. To tackle this complex combinatorial optimization challenge, deep reinforcement learning (DRL)-based methods have been actively investigated and show promise for performing online 3D bin packing with a limited look-ahead at upcoming items (Zhao et al., 2022; Yang et al., 2023). These studies demonstrate that DRL agents can effectively learn optimal packing strategies to maximize space utilization in the online optimisation process (Zhu et al., 2024). However, the majority of existing research concentrates on online 3D bin packing with handling a

* Correspondence to: The Hong Kong Polytechnic University, Room CF407, 4/F, Core C, Kowloon, Hong Kong.
E-mail address: yungpo.tsang@polyu.edu.hk (Y.P. Tsang).

https://doi.org/10.1016/j.compind.2024.104202
Received 4 July 2024; Received in revised form 22 September 2024; Accepted 21 October 2024
Available online 4 November 2024

bin with fixed dimensions at a time, which limits the throughput in the robotic packing process. The dual bin environment, as depicted in Fig. 1, where two robotic arms operate concurrently for bin packing, poses a novel challenge. Preliminary evidence suggests that the efficiency of bin packing could be significantly improved in a dual bin setup. Yet, the development of an optimal bin packing strategy to manage two independent bins in tandem remains insufficiently explored. Further research is imperative to unveil strategies that can exploit the potential synergies and coordination between the robotic arms, thus optimizing the packing process in a dual bin scenario.

In this study, the online 3D dual multi-bin packing problem (3D-DBPP) is formulated which is then solved by the deep reinforcement learning approach. Two bin replacement strategies are developed and implemented in the DRL agent to smoothly manage two bins concurrently. To investigate the viability and performance, a set of simulation experiments are conducted, where the configurations of simulation environment follow the operations of a third-party logistics company in Hong Kong. Furthermore, the performance of the proposed method is benchmarked with four heuristic approaches, namely (i) bottom left (BL) (Chazelle, 1983), (ii) best volume fit (BVF) extended from the best area fit (Hassan and Pillay, 2019), (iii) best short side fit (BSSF) (Yang et al., 2023) and (iv) best long side fit (BLSF) (Martinez et al., 2021). Overall, it is found that the proposed DRL solver for the 3D-DBPP averagely outperforms the aforementioned heuristic approaches, where the performance gap is larger when the number of lookahead items is smaller. Consequently, the research provides solid theoretical support to the facility and operational design in the autonomous robotic warehouses.

The rest of this paper is organised as follows. Section 2 reviews the theoretical preliminaries of online 3D bin packing problems (3D-BPP), and compare with the existing reinforcement learning applications for online 3D-BPP. Thus, the research gap of this study can be explicitly summarized. Section 3 presents the proposed research methodology, specifically for the solution architecture of the DRL-based 3D-DBPP optimisation, for achieving the optimal strategy to simultaneously handle two bin packing tasks. In Section 4, the case analysis is conducted to examine the system efficacy and performance in different scenarios, while Section 5 discusses the primary findings from the analyses. Eventually, conclusions and future directions are presented in Section 6.

2. Literature review

In this section, the foundational concepts of the 3D Bin Packing Problem (3D-BPP) are reviewed, and the existing studies contributing to the online solution approaches are compared to articulate the research motivation. The research gap of this study is highlighted based on the review.

2.1. Overview of 3D bin packing problems

The bin packing problem (BPP) is one of the classical engineering management problems, which aim to determine the minimal number of bins to pack all items (Abdou and Yang, 1994; Gonçalves and Resende, 2013). In the context of 3D bins, the optimization of BPP is used in various operational scenarios, such as palletization and container loading. Initially, the single 3D bin packing problem was solved by a mixed integer programming formulation to balance space utilization, cost, and load balancing issues (Baldi et al., 2012; Moon et al., 2014). However, the single bin packing optimization is limited by the number of items it can handle, leading to restricted applicability. Recent studies have begun exploring multi-bin packing optimization (Erbayrak et al., 2021), where multiple bins are considered in the optimization process, considering the number of bins an additional objective alongside space utilization.

Compared to other optimization problems, a particularly challenging issue in solving 3D-BPP is managing the data structure of used and free 3D spaces. Jylänki (2010) conducted a comprehensive survey of algorithms such as the shelf, guillotine, and maximal rectangles algorithms to develop data structures for 2D spaces. Of these, the shelf algorithm is considered the simplest, organizing free spaces into shelves for packing in a bottom-up and left-to-right manner (Xavier and Miyazawa, 2008). To reduce space wastage, the guillotine algorithm was proposed, which

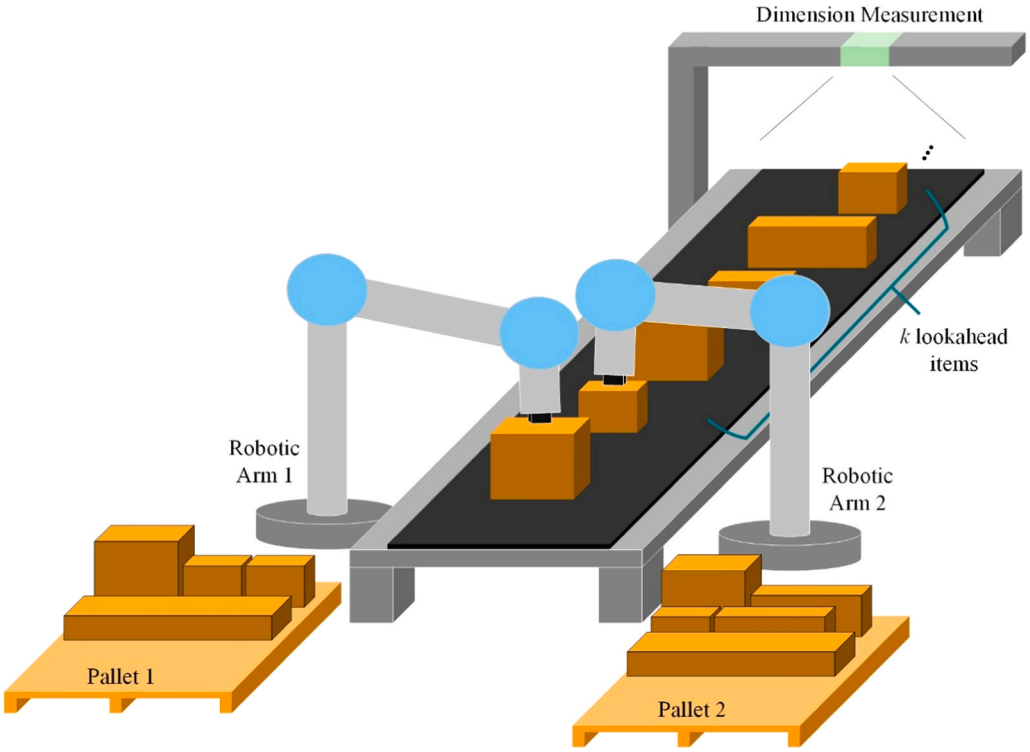


Fig. 1. Overview of the online 3D dual bin packing in the robotic palletisation process.

performs a guillotine split placement, dividing an L-shaped free space into two disjoint spaces after placing a rectangular item (Amossen & Pisinger, 2010; Martin et al., 2021). While the guillotine algorithm addresses the shelf algorithm's space wastage issue, its fixed split lines, or space sub-boundaries, limit its practical performance. Consequently, the maximal rectangles algorithm was developed to store and update a list of maximal free spaces following guillotine split placements. With a well-defined data structure for spaces, various packing rules such as bottom-left (BL), best-area-fit (BAF), best-short-side-fit (BSSF), and best-long-side-fit (BLSF) can be implemented to create heuristics for multiple bin packing problems. These algorithms and rules can be further extended to 3D environments to establish different 3D bin packing heuristics. The online 3D Bin Packing Problem (3D-BPP) emerges as a critical area of study when the packing process operates with incomplete information about the characteristics of incoming items. This scenario is particularly relevant in various real-world automation contexts, such as robotic container loading (Ha et al., 2017). While offline 3D-BPP has been extensively studied, with approaches often leveraging local search and meta-heuristic methods to refine solutions generated by constructive heuristics, these techniques prove inadequate in online environments. The fundamental challenge lies in the necessity for immediate decision-making regarding item placement in online scenarios (Ali et al., 2022). This constraint renders traditional offline optimisation methods ineffective, as they typically rely on full knowledge of all items to be packed. Consequently, there is a compelling need to develop sophisticated online models capable of addressing the nuances of online 3D-BPP in practical applications. Such models should demonstrate the ability to make rapid, near-optimal decisions in the absence of complete information, thereby more accurately reflecting and resolving the challenges inherent in dynamic, real-time packing environments.

Since 3D-BPP is inherently a combinatorial optimization problem characterized by non-deterministic polynomial-time (NP) hardness (Puche and Lee, 2022), developing an efficient optimization process is crucial to obtain an optimal loading plan within a reasonable timeframe. This imposes an additional challenge for online optimization, where prior knowledge of all items to be packed is unavailable, and the optimal strategy for making informed decisions remains under-explored in the existing literature.

2.2. Reinforcement learning for online optimisation

Recent advancements in machine learning have highlighted deep reinforcement learning (DRL) as a promising method for sequential decision-making in online and dynamic optimization contexts (Panzer and Bender, 2022). Zhao et al. (2021) and Zhao et al. (2022) pioneered a constrained DRL architecture, a model-free approach designed to learn policies aimed at maximizing cumulative discounted rewards. These rewards are defined in terms of volumetric occupancy and the quantity of safe loading positions within the bin. The 3D bin environment is parameterized by discretization and a 2D integer height map, coupled with a feasibility mask to denote placement feasibility. Notably, a convolutional neural network is employed to encode the height map, feasibility mask, and a sequence of lookahead items for placement. This facilitates the interaction between the actor and critic within the DRL framework, enabling the learning of an optimal policy.

Conversely, Yang et al. (2023) integrated three heuristic rules into a DRL-based 3D bin packing optimiser to enhance physical stability, action variation, and space utilization. Their findings suggest that the inclusion of heuristic rules leads to improved bin utilization compared to existing methodologies. These studies collectively affirm the effectiveness of DRL in the online 3D-BPP landscape, significantly augmenting space utilization. Nevertheless, current research predominantly concentrates on executing online 3D-BPP one bin at a time, which constrains facility design and throughput rates in robotic order packing systems. Furthermore, beyond discretization, alternative algorithms for representing the data structure of 3D spaces, such as the maximal rectangles algorithm, could be investigated to refine the DRL framework's formulation.

2.3. Position of this research

The existing studies in literature indicate that the successful implementation of online 3D-BPP can catalyse numerous business applications, particularly in robotic palletization and container loading, thereby enhancing smart manufacturing and logistics processes. While the extant literature substantiates the feasibility and advantage of the DRL approach in online 3D-BPP, the studies focus narrowly on single-bin packing operations, which limit the productivity of robotic packing processes. The method proposed in this research addresses the research gap in online 3D-BPP involving dual bins by employing DRL to elevate operational flexibility and productivity in robotic warehouses.

3. Research Methodology

In order to enhance the aforementioned operational capability, the DRL-based optimiser for the online 3D-DBPP is designed and developed in this section, covering the problem description, model formulation, system architecture and DRL formulation.

3.1. Problem description of online 3D-DBPP

The online 3D dual bin packing problem (3D-DBPP) represents a novel advancement in the field of logistics optimization, extending the classic bin packing problem into a more complex and practically relevant domain. This variant incorporates three key elements that set it apart from traditional formulations: three-dimensional space consideration, online optimization processes, and simultaneous packing into two bins. In modern automated warehouses and distribution centres, robotic arms are increasingly employed for palletization tasks. The 3D-DBPP mirrors this real-world scenario by tasking these robotic systems with optimally packing a sequence of items with known dimensions. The online nature is reflected in the lookahead mechanism, where the dimensions of the next k items in the queue are known in advance. This feature closely aligns with practical scenarios where items are scanned before reaching the packing station, providing a limited preview of upcoming items. The dual bin aspect of the problem introduces a novel decision-making challenge. The robotic system should make real-time decisions about which of the two available bins to place each item in, aiming to maximize overall space utilization. This dual-bin approach offers increased flexibility compared to single-bin models, as it allows for more packing strategies and accommodation of varied item sizes. A critical practical consideration in this problem is the physical stability of the packed items. The requirement that at least 50 % of the base area of any item should overlap with the item beneath it ensures a stable and secure stack. This constraint is crucial for real-world applications, as it mitigates the risk of collapse during packing and transit, addressing a key concern in automated logistics operations.

The complexity of the 3D-DBPP is further compounded by its classification as an NP-hard problem. This highly decisional complexity means that solution time increases exponentially as the problem size grows, even in offline environments where the full item list is known in advance. In the online context of the 3D-DBPP, this complexity is exacerbated by the need for real-time decision-making. The practical value of studying the 3D-DBPP lies in its direct applicability to modern logistics challenges. As e-commerce and just-in-time manufacturing continue to grow, efficient packing and space utilization become increasingly critical. The dual-bin approach offers potential improvements in throughput and resource utilization, while the online nature of the problem reflects the dynamic, real-time decision-making required in modern warehouses. Moreover, this problem variant serves as a bridge between theoretical optimization models and practical logistics

operations. By incorporating realistic constraints such as stability requirements and limited lookahead, solutions to the 3D-DBPP can be more readily adapted to real-world automated packing systems. This makes this study a stepping stone towards more efficient and flexible logistics operations in the era of smart warehouses and Industry 4.0.

3.2. Decision model of 3D bin packing problem

As online 3D dual bin packing problem (3D-DBPP) is extension of 3D bin packing problem (3D-BPP), some common definitions and assumptions of 3D-DBPP would be based on the general notation used in 3D-BP. The offline 3D bin packing problem could be formulated as a mixed integer programming model as a basic model, which assumes the available complete information of item list. To deal with the partial information of item lists, the decisions of packing items into dual multi-bins would be formulated into the Markov Decision Process (MDP).

3.2.1. Basic model of 3D-BPP

A set of n three-dimensional cubic items i is considered, where item I is of width w_i, length l_i, height h_i, and weight m_i, $i = 1, \ldots, n$. To represent the coordination of an item in a bin for three-dimensional bin packing problem, the Cartesian coordinate system is adopted, where the origin O (0,0,0) refers to the lower back left corner of the bin. The length, width and height are oriented with the x, y, z axes. Each item is packed into the bin a which its lower back left corner is located at the selection position. Each 3D cubic item can be packed into a bin with the allowed orientations with six possible rotations, r. The objective is to maximize the volume utilization by packing all items into a bin. The definitions of indices, parameters and decision variables used to formulate the model of 3D-BP as the basic model are presented in Appendix A. The MIP model for 3D bin packing is formulated as follows:

$$Max \sum_{i=1}^{n} v_i x_i \tag{1}$$

Subject to:

$$\sum_{i=1}^{n} v_i x_i \le V \tag{2}$$

$$C_{i,d} \le Z_d x_i \quad \forall i = 1, \ldots, n; \forall d = 1, \ldots, 3 \tag{3}$$

$$C_{i,d} \ge 0 \quad \forall i = 1, \ldots, n; \forall d = 1, \ldots, 3 \tag{4}$$

$$\sum_{r=1}^{6} R_{i,r} = x_i \quad \forall i = 1, \ldots, n \tag{5}$$

$$C_{i,d} + \sum_{r=1}^{6} z_{i,r,d} R_{i,r} \le Z_d \quad \forall i = 1, \ldots, n; \forall d = 1, \ldots, 3 \tag{6}$$

$$C_{i,d} + \sum_{r=1}^{6} z_{i,r,d} R_{i,r} - Z_d(1 - y_{i,j,d}) \le C_{j,d} \forall i = 1, \ldots, n; \forall i \ne j; \quad \forall d = 1, \ldots, 3 \tag{7}$$

$$x_i + x_j - 1 \le \sum_{d=1}^{3} y_{i,j,d} \quad \forall i = 1, \ldots, n; \forall i \ne j \tag{8}$$

$$y_{i,j,d} \le x_i \quad \forall i = 1, \ldots, n; \forall i \ne j; \quad \forall d = 1, \ldots, 3 \tag{9}$$

$$x_i \in \{0, 1\} \quad \forall i = 1, \ldots, n \tag{10}$$

$$R_{i,r} \in \{0, 1\} \quad \forall i = 1, \ldots, n; \forall r = 1, \ldots, 6 \tag{11}$$

$$y_{i,j,d} \in \{0, 1\} \forall i = 1, \ldots, n; \forall i \ne j; \quad \forall d = 1, \ldots, 3 \tag{12}$$

The objective function (1) is to maximize the total volume of the packed items in a bin. Constraint (2) refers to the capacity constraint that the total volume of the packed items is within the volume of bin. Constraint (3) state that if an item is not packed into the bin its coordinates are zero. Constraint (4) indicates that the coordinates of the position of packed items must be non-negative. Constraint (5) states that an item only has one value of packed rotation if an item is packed into the bin; otherwise, the value of rotation is not assigned. Constraint (6) ensures that items must lie inside the bin. Constraints (7)-(9) are used to prevent from overlapping in the feasible solutions. Constraints (10)-(12) refer to binary value of decision variables.

The basic model of 3D-BPP indicates the decision complexity of such the NP-hard problem, in the environment with the complete information of item lists. However, when the inflow of items is not completely known in the beginning, the applicability of solving the basic model is limited. Instead, an online 3D-DBPP optimiser is proposed based on the Markov decision process (MDP). The objective and operational considerations in the 3D-BPP would be transformed to MDP, which also serves as the fundamental model of Double Deep Q Network (Double DQN) under reinforcement learning framework.

3.2.2. Markov decision process (MDP)

To formulate the online 3D-DBPP optimiser, an markov decision process (MDP) is defined as a tuple (S, A, R, P), where S is a state space ($s_t \in S$, representing the observable state at timestep t), A is an action space ($a_t \in A$, representing the action taken at timestep t). Also, R is a reward function ($R : S \times A \rightarrow \mathbb{R}$), where $r_{t+1} = R(s_t, a_t)$ mapping the immediate reward after performing action a_t in state s_t. Lastly, P is a state transition probability, $P(s_{t+1}|s_t, a_t)$, representing the probability of transferring the state, s_t from to s_{t+1} when action a_t is taken in state s_t at timestep t. The detailed definitions of state, action and reward are depicted as follows.

3.2.2.1. State space. In the state representation, there are two primary elements: (i) a set of maximal cuboid spaces $\mathscr{M}$ and (ii) dimensions of k items (w_i, l_i, h_i), where $i \in (1, 2, \ldots, k)$. M refers to cuboid spaces for packing k items with dimensions, w_i, l_i, h_i.

Fig. 2 shows the set of three maximal cuboid spaces for one item in the bin. When multiple items are considered, the determination of the maximal cuboid spaces M would be a complex task. An algorithm will be proposed to obtain the set of maximal cuboids. Additionally, for as an input to the MDP environment, the height map $\mathscr{H}$ from the set of maximal spaces is used and retrieved. To be specific, the height map is defined to build a top-view map showing discretized grids as shown in Fig. 4(a), where each pixel has a normalised value ranging from 0 to 1 representing the proportion of the highest point at a position and the bin's height.

Hence, the environment state $\dot{s}_t$ at timestep t includes the maximal cuboid spaces $\mathscr{M}_t$ and the dimension of item i =1,…, k in terms of width, length and height: $(w_{t,i}, l_{t,i}, h_{t,i})$. It is expressed in Eq. (13).

$$\dot{s}_t = [\mathscr{M}_t, (w_{t,1}, l_{t,1}, d_{t,1}), (w_{t,2}, l_{t,2}, d_{t,2}), \ldots, (w_{t,k}, l_{t,k}, d_{t,k})] \tag{13}$$

3.2.2.2. Action space. Regarding the action space of MDP for 3D-DBPP, there are three primary elements: (i) the item selection, x_i, from the set of lookahead items, (ii) item orientation, $R_{i,r}$, and (iii) location of packed item, $C_{i,d}$. For the item orientation, each cuboid has six possible rotations, namely (w, l, h), (w, h, l), (l, h, w), (l, w, h), (l, w, d) and (l, d, w). When some of the item dimensions are identical, for example $w = l$, the dimensions of some orientations are the same. To eliminate the number of redundant actions, only the unique orientations are considered, while the items are placed in one of the maximal cuboids.

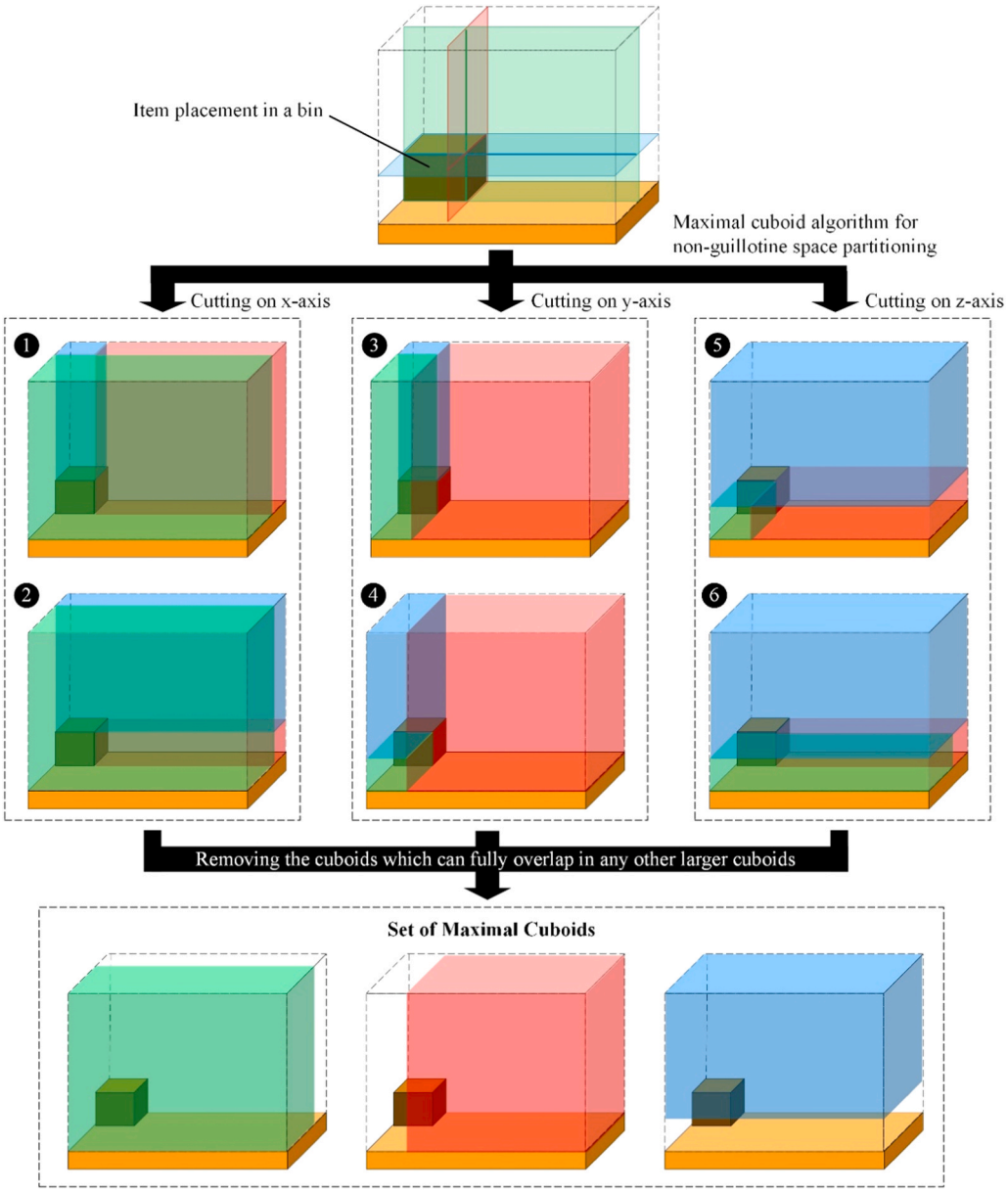


Fig. 2. Maximal cuboids spaces for an item in the bin.

Furthermore, the feasibility of the action to be made is checked by considering the following two conditions. First, the base of the item to be placed must be at least 50 % supported by the top of another item or the floor (i.e. y = 0) so as to guarantee the stability (τ), expressed as in Eq. (14). When an item of (w, l, h) is placed in a maximal cuboid at the minimum corner of ($x_{\min}$, $y_{\min}$, $z_{\min}$) in the action map as shown in Fig. 4 (b), the function $CL()$ is defined to check whether the points in area of ($x_{\min}+w$) and ($z_{\min}+l$) in the height map is consistent to the value $y_{\min}$. The function $CL()$ returns 1 if consistent, and vice versa. On the other hand, the placement of the item must be aligned in any axes with the maximal free cuboid. In other words, the items cannot be placed in the middle of the free spaces so as to reduce the size of action spaces.

$$\tau = \frac{\sum_{i=0}^{w}\sum_{j=0}^{l} CL(\mathscr{H}_{(x_{\min}+i,\quad z_{\min}+j)},y_{\min})}{w \bullet l} \geq 0.5 \tag{14}$$

Based on the above concepts, the decision variables representing the actions can be defined by: (i) i-th choice of a rotated item and (ii) j-th choice of item placement in a maximal free cuboid at the minimum corner of ($x_{\min}$, $y_{\min}$, $z_{\min}$). Subsequently, the discrete action space is represented as a jagged array $\mathscr{A} = \{A_1, A_2, \ldots, A_{6\bullet k\bullet|\mathscr{M}|}\}$, where some duplicate and infeasible actions are filtered out.

3.2.2.3. Reward. In order to guide the DRL agent in solving the Online 3D-DBPP, the maximisation of space utilisation is a sole direction, but it is inefficient to set the overall space utilisation as the reward after making a sequence of actions. Consequently, the goal of the maximisation of space utilisation is refined into two components called as $R_{pyramid}$ and $R_{compactness}$, as in Eqs. (15) and (16), so as to evaluate immediate reward values throughout the MDP. In (15) and (16), N denotes the total number of packed items in the bin; (W,L,H) denotes the size of the container space; (w_i, l_i,h_i) denotes the size of a packed item i. On one hand, the reward $R_{pyramid}$ evaluates the proportion of the volume of packed items and occupied volume such that the interspace in bin packing can be discouraged. On the other hand, the reward $R_{compactness}$ measures the proportion of the volume of packed items and the volume of $W \bullet I \bullet \mathscr{H}_{\max}$, which facilitate the items to be packed in the current layer. In this DRL-based optimiser, the above two reward components

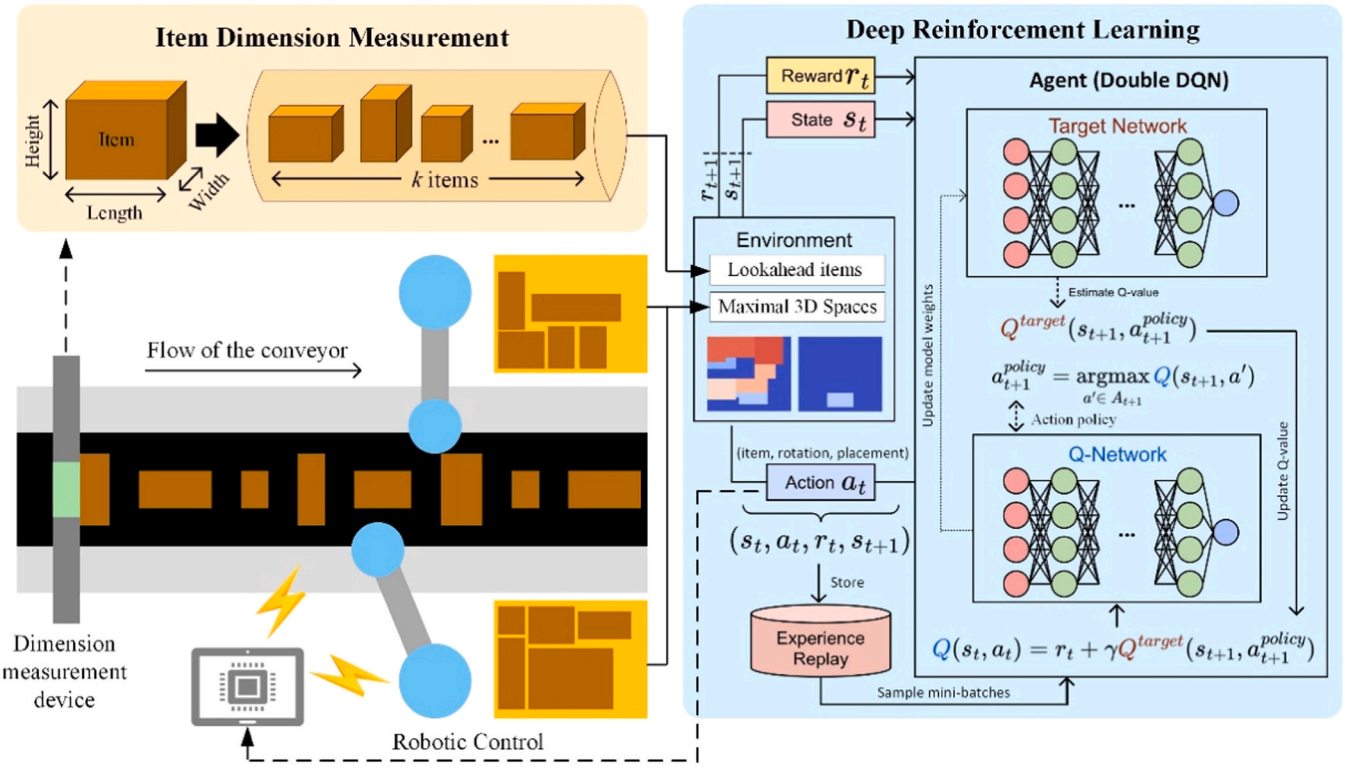


Fig. 4. Framework of the online 3D-DBPP optimiser.

are equally weighted to formulate a composite reward function R, expressed as $R = (R_{pyramid} + R_{compactness})/2$.

$$R_{pyramid} = \frac{\sum_{i \in N} w_i \bullet l_i \bullet h_i}{\sum_{j \in W} \sum_{k \in D} \mathscr{H}_{j,k}} \tag{15}$$

$$R_{compactness} = \frac{\sum_{i \in N} w_i \bullet l_i \bullet h_i}{W \bullet D \bullet \mathscr{H}_{\max}} \tag{16}$$

3.3. System architecture of the online 3D-DBPP optimiser

Based on the model formulation of the online 3D-DBPP, the proposed optimiser is designed to achieve automated dual bin packing in robotic warehouses, while the system architecture is illustrated as in Fig. 3. On the conveyor belt, the dimension measurement device is deployed to capture the information of item dimensions, including width, length and height. Subsequently, a set of lookahead items with the dimensions is obtained. Under the framework of DRL, the dimensions of those lookahead items and the maximal cuboid spaces become the inputs of state variables in the environment by the maximal cuboids algorithm (MCA). The informed decisions are made based on the Double Deep Q Network (Double DQN) to control the conveyor and robotic arms to perform palletisation in optimising the space utilisation.

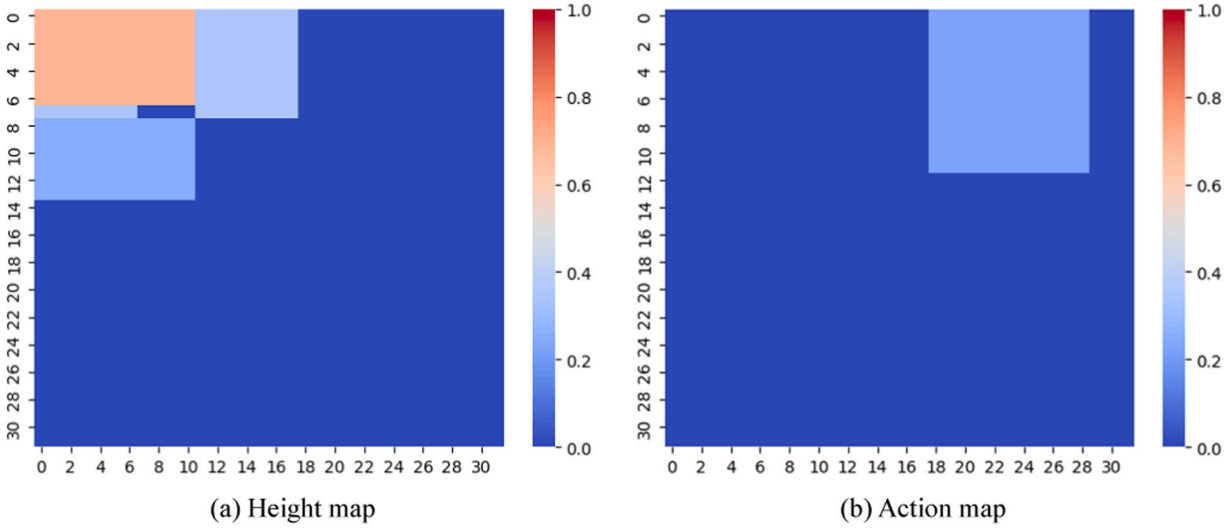


Fig. 3. Samples of (a) height map and (b) action map.

3.3.1. Maximal cuboids algorithm (MCA)

To partition the cuboid spaces and maintain a set of maximal cuboids as the state variables in the MDP, the proposed 3D-DBPP optimiser utilizes MCA. MCA is extended from the maximal rectangles algorithm, which introduces the MAXRECTS data structure to represent the available space post-placement (Jylänki, 2010; Elhedhli et al., 2019), as a reference for non-guillotine space partitioning. However, the existing MAXRECTS data structure is tailored for 2D spaces. Consequently, this research extends the algorithm to 3D spaces, resulting in the development of MCA, as depicted in Fig. 2 and Algorithm 1. In contrast to traditional guillotine space partitioning that necessitates straight cuts, non-guillotine space partitioning does not require such constraints. Items can be placed in any orientation, provided they do not overlap with other items and remain within the bin's boundaries, leading to enhanced flexibility and packing efficiency.

Algorithm 1. Maximal Cuboids Algorithm (MCA)

```
Initialise:
Set 𝓜 = {(W, I, H)}
Pack:
foreach item I = (w, l, h) in the sequence do
    Decide the free cuboid M_i ∈ 𝓜 to pack the item I
    Create a new cuboid if no such a cuboid is found
    Decide the orientation of the item
    Place the item at the bottom-left of M_i
    Denote by ℬ the bounding cuboid of I in the bin after packing the item
    Split M_i into M', M'', M'''
    Set 𝓜 ← 𝓜 ∪ {M', M'', M'''}\{M_i}
    foreach free cuboid M ∈ 𝓜 do
        Compute M\ℬ
        Subdivide at most 18 new cuboids G_1, ..., G_18
        Set 𝓜 ← 𝓜 ∪ {G_1, ..., G_18}\{M}
    end
    foreach ordered pair of free cuboids M_i, M_j ∈ 𝓜 do
        if M_i contains M_j then
            Set 𝓜 ← 𝓜 ∪ M_j
        end
    end
end
```

3.3.2. Double deep Q network (Double DQN) with experience replay

According to the above definitions of state space, action space and reward, the DRL-based optimiser can be formulated based on the double deep Q network (Double DQN) with the experience replay (Liu et al., 2023; Tian et al., 2023). To be specific, the DQN plays the role of a function approximator of $Q_\pi(s,a)$, where π denotes the optimal policy learnt from the environment.

Due to the fact that the action space has a variable length throughout the episodes, the representation of the DQN is modified in this study. As shown in Fig. 5(a), the typical DQN can be used to generate the Q values for all actions at a single inference, which cannot deal with the variable-length action spaces. Therefore, the proposed DQN process considers the state and action together as the input to the DQN so as to obtain the Q values one by one, as shown in Fig. 5(b). Technically speaking, an observation of the state s_t is created as in Eq. (17), based on the environment state $\dot{s}_t$, which includes the height map H, action map $A_{i\in\{0,\,|A|-1\}}$ and item list $I_{i\in\{0,\,|A|-1\}\,\{i\}}$, where $\max(|A|) = 6 \bullet k \bullet |\mathscr{M}|$. Subsequently, the proposed DQN estimates the state-action value for action a_t given a state s_t at the timestep t, in which the network architecture of the proposed DQN per bin is illustrated in Fig. 6. In other words, the proposed optimiser aims to learn a deterministic policy $\pi(s) = a$. For each feasible action, an inference of DQN is needed to determine the Q-value of this state and action, and the total inference

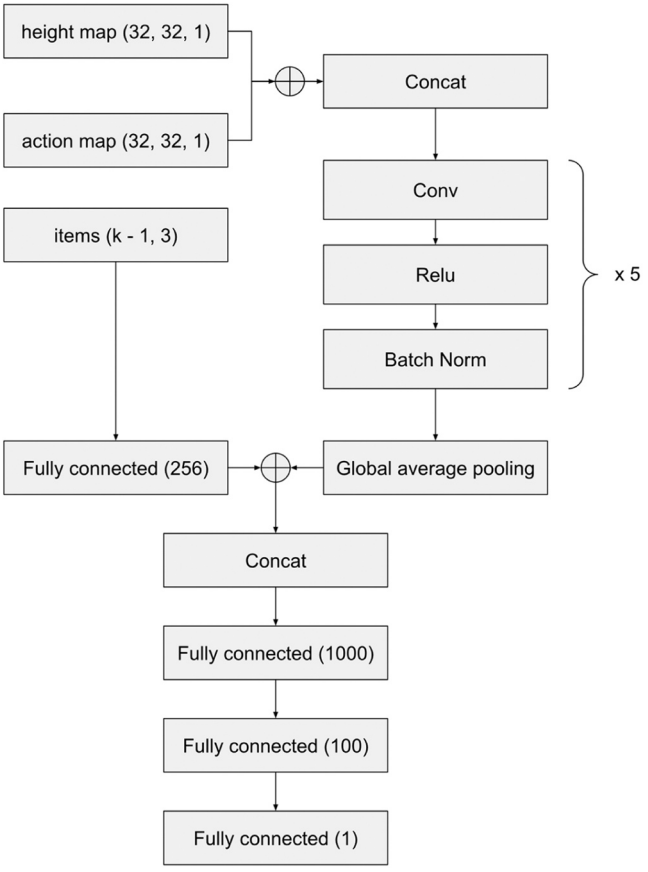
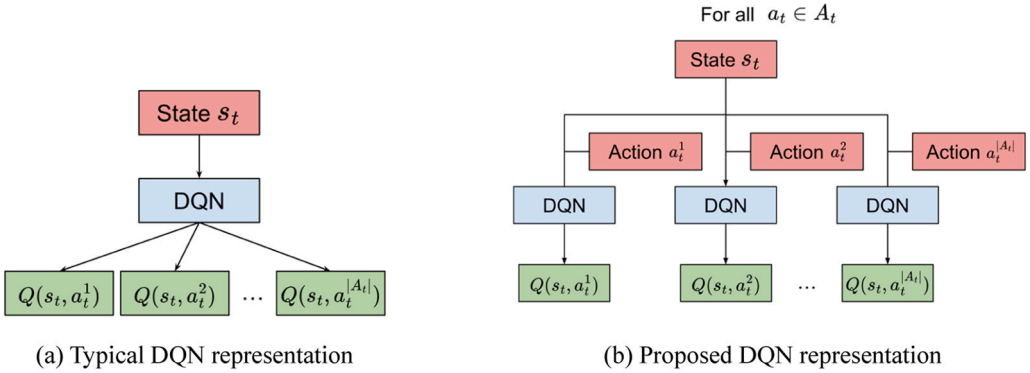


Fig. 6. Network architecture used in the proposed DQN per bin.



(a) Typical DQN representation (b) Proposed DQN representation

Fig. 5. Typical and proposed representation of the DQN.

time increases linearly with the number of feasible actions. Through taking the action with the maximal state-action value, the environment generates (1) the reward r_t, (2) the next state s_{t+1}, and (3) termination signal.

$$s_t = \{\mathscr{H}, A_0, I_0, A_1, I_1, \ldots, A_{|A|-1}, I_{|A|-1}\} \tag{17}$$

Additionally, this proposed method studies two bin replacement strategies to manage operations when bins reach capacity or cannot accommodate new items: (i) replaceAll and (ii) replaceMax. The replaceAll strategy entails replacing both bins with new ones if any of the lookahead k items cannot fit into the current bins. The replaceMax strategy, on the other hand, calls for replacing only the bin that has higher space utilization when faced with similar circumstances, preserving the bin with available space for continued packing.

To address the online 3D-DBPP effectively, a DRL-based optimiser is proposed. This optimiser is adept at accommodating both bin replacement strategies, learning from interactions with the environment to make informed decisions that enhance packing efficiency and space utilization. It dynamically updates its packing strategy based on real-time feedback, ensuring robust performance within the dynamic environment of a robotic warehouse.

4. Case study and analysis

Based on the design of the online 3D-DBPP optimiser, a set of simulation experiments are conducted in this section to verify its viability and performance, where the bin packing activities are referred to the real-life operations of a case company. By following the defined experimental protocol, the proposed optimiser is utilised to solve the 3D-DBPP, while the comparison with the existing heuristics approaches is included.

4.1. Case background

For the evaluation of proposed optimiser, a case study on the logistics operations of AOC Ltd. (a pseudonym) were conducted for developing the simulation experiment, which closely mimics real-world bin packing activities. Established in 1978, AOC Ltd. is a comprehensive logistics service provider, offering a wide range of services including international freight, warehousing, transportation, and e-fulfillment. The e-fulfillment process at AOC Ltd. is characterized by a standard workflow of order picking, packing, palletization, and last-mile delivery, with regular stock replenishment within the storage area.

To streamline the order fulfillment process and maintain a high level of service quality, AOC Ltd. has invested in advanced robotic systems, such as autonomous mobile robots and four-way shuttle systems, to support a goods-to-person picking methodology. Following the picking phase, carton boxes are selected for order packing, which in turn establishes the final dimensions of the parcels. Currently, parcels awaiting shipment are stored temporarily in a staging area before they are palletized for distribution via cargo vans. However, the e-fulfillment process often faces challenges related to staging space management and workforce allocation for palletization, especially when dealing with fluctuating order volumes. These challenges create a bottleneck in the e-fulfillment pipeline, limiting the capability to optimize throughput rates. Given these operational impediments, there is a pressing need for an online optimization solution tailored to the palletization stage of the e-fulfillment process, one that can integrate seamlessly with the existing conveyor belt system, as illustrated in Fig. 7. Two robotic arms can serve for one parcel conveyor belt in the bin packing process. Consequently, the operational parameters for the forthcoming experiments have been meticulously set to mirror the real-life conditions faced by the company.

4.2. Protocol of the system implementation

In the simulation experiments, the bin size of (32, 32, 32) is considered, while the list of 200 items are randomly generated within the range of (6, 6, 6) and (12, 12, 12). Also, the items are shuffled to avoid any unintended biases in the bin packing optimisation. During the bin packing process, the lookahead value k is controlled to reveal the set of item information to the optimiser, where the lookahead values are set at 5, 10 and 15 in the experiment. These settings are applied in two experimental scenarios, namely (i) online 3D single bin packing (S1) and (ii) online 3D dual bin packing (S2). In the former scenario, only one bin is considered at a time in the online 3D bin packing optimisation process, where a new bin is created when no feasible action is found. In the latter scenario, two bins are simultaneously considered for the online 3D bin packing optimisation process. As mentioned in Section 3.2.2, two bin replacement strategies are deployed, namely (i) replaceAll and (ii) replaceMax, to pack all the items in the bins. In brief, the strategy 'replaceAll' replaces all bins, while the strategy 'replaceMax' replaces the bin with the highest space utilisation, when there is no available item packed into the bin. Especially for the proposed DQN-based optimiser, the number of iterations and episodes for training the agent are set at 100 times and 1000 times, respectively. For both scenarios, the space utilisation of the completed bins, excluding those bins who are yet fully packed, are averaged for the performance comparison.

In addition, the proposed optimiser is compared with four commonly-used heuristics approaches as follows:

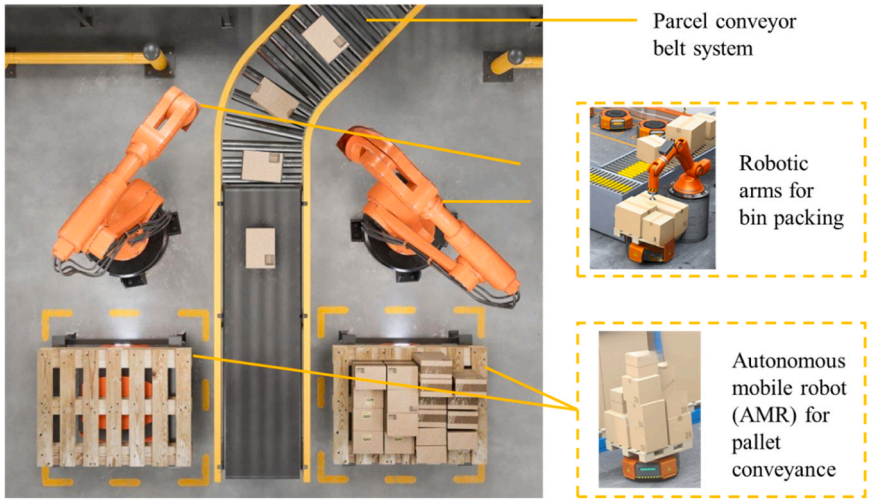
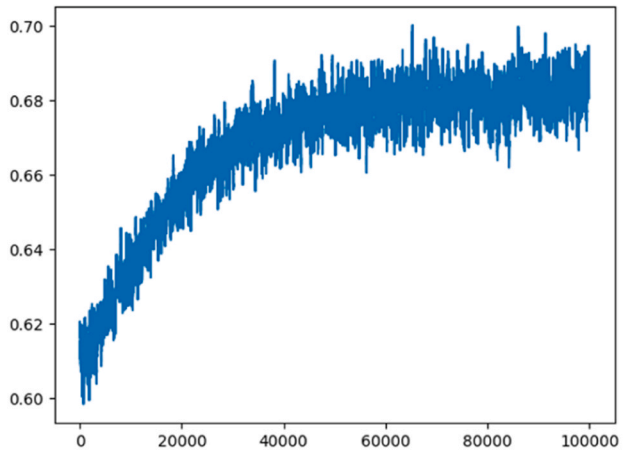
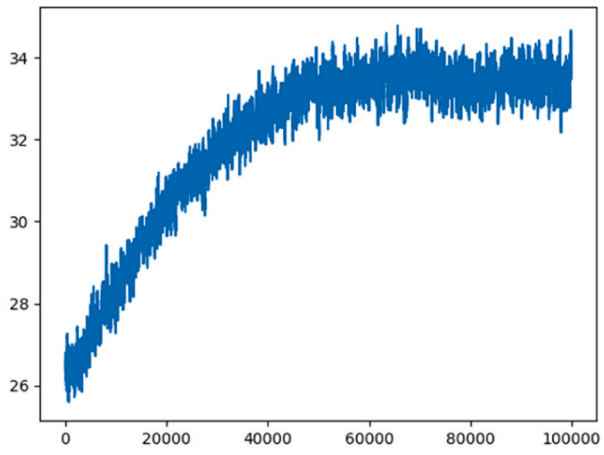


Fig. 7. Illustration of the solution implication in robotic warehouses.

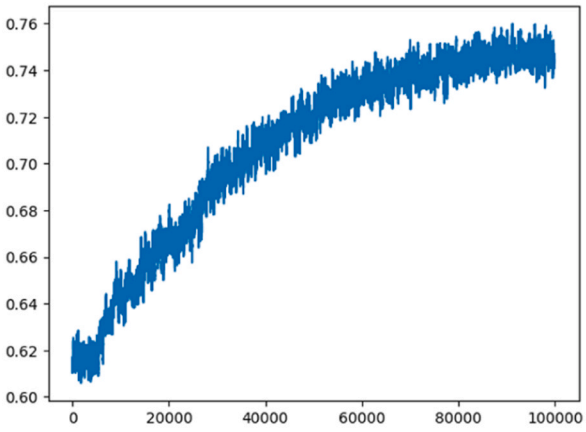
- *Bottom-left (H1-BL)*: H1-BL aims to position each rectangle such that the height is minimized, effectively placing the rectangle as low as possible within the space. In scenarios, where multiple valid positions exist that satisfy this condition, the position with the leftmost position is selected. This approach ensures that cuboids are oriented and placed in a manner that conservatively uses the bottom-left area, facilitating compact packing.
- *Best-volume-fit (H2-BVF)*: H2-BVF selects the smallest available 3D space for placing the next cuboid, thereby minimizing wasted space. In the event of equal areas—a tie—the Best Short Side Fit rule is



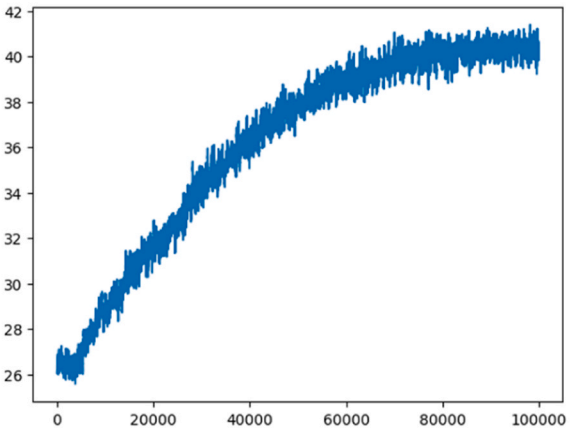
(a) Change of space utilisation (k=5)



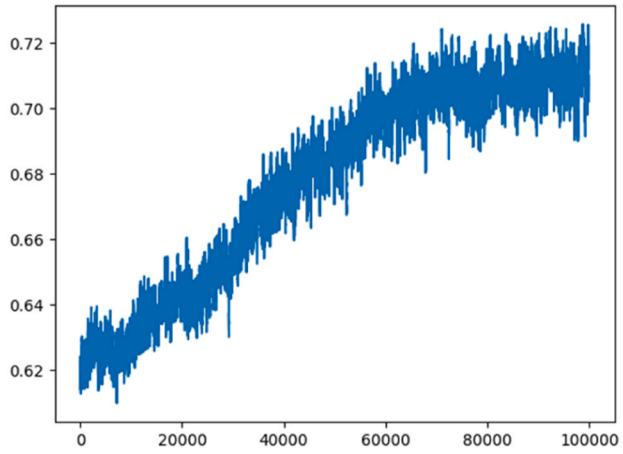
(a) Change of episode reward (k=5)



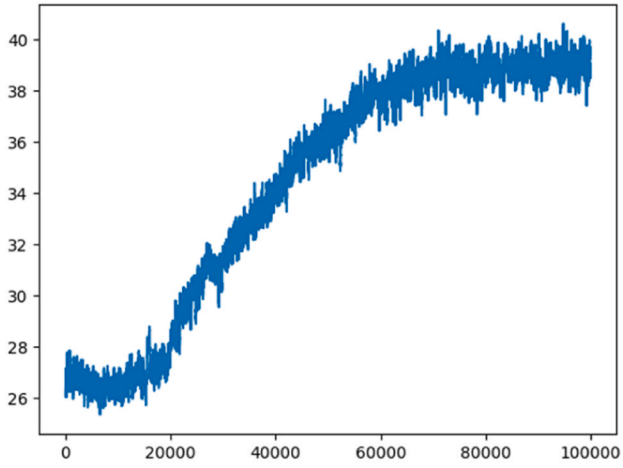
(c) Change of space utilisation (k=10)



(d) Change of episode reward (k=10)



(e) Change of space utilisation (k=15)



(f) Change of episode reward (k=15)

Fig. 8. Changes of space utilisation and episode reward over the training period.

applied as a tiebreaker. By prioritizing the placement into the smallest area, this heuristic contributes to a more efficient use of space with regard to the volume.

- *Best-short-side-fit (H3-BSSF)*: H3-BSSF determines the placement of a cuboid by choosing the space where the difference between the space's width and the cuboid's width (W − w), the space's height and the cuboid 's height (H − h), or space's depth and the cuboid's depth (D −d) is minimized. The aim is to find the space where the space fits most snugly in terms of the shorter dimension, whether it be width, height or depth. Thus, H3-BSSF prioritizes placements that offer the least discrepancy on the shortest side, promoting a tight fit and potentially reducing gaps in packing.
- *Best-long-side-fit (H4-BLSF)*: Analogous to the H3-BSSF, H4-BLSF instead focuses on maximizing the fit along the longer side of the 3D space. By changing the evaluation from min() to max(), the chosen space will be one where the cuboid matches or exceeds one dimension of the space as closely as possible, thereby aligning with the longer dimension of the space. H4-BLSF is particularly useful for accommodating cuboids in a way that leverages the longer dimension for a snug fit, which can be beneficial in certain packing contexts.

Therefore, through the computation experiments, not only the viability of the proposed optimiser can be verified, and its advantages in different scenarios can be highlighted. All the simulation experiments are run in a computer with the specification of i7-10750H, 16 GB RAM and NVIDIA GeForce GTX 1660 Ti.

4.3. DQN training for the proposed optimiser

Through deploying the proposed lookahead configurations, the agent is trained in 100 iterations, where 1000 episodes are given to each iteration. The hyperparameters of the DQN are tested and tuned through a number of attempts so as to facilitate an effective learning and convergence over the DRL process. The set of tuned hyperparameters is: (i) discount factor of 0.95, (ii) maximum size of replay memory of 1000,000, (iii) initial and minimum exploration rates of 1 and 0.05 with the decay over epoch of 0.99 to govern the exploration-exploitation trade-off, (iv) warm-up period of 20 epochs with the learning rate of 1×10^{-5}, (v) learning rate of 1×10^{-3}, (vi) decease of the learning rate in every 10,000 epochs with the minimum learning rate of 1×10^{-5}, (vii) update of the target network at every 10 epochs. Fig. 8 shows the improvements of space utilisation and episode reward over the whole training process when the lookahead values are 5, 10 and 15.

It is found that the convergence is reached over time with a relatively stable episode reward. In other words, a stable policy is learnt which can maximise the expected cumulative rewards such that the space utilisation of bins can be maximised. Therefore, three DQN models are trained, which can be used for the inference of bin packing actions and the exploration rate is set at 0 to opt out the epsilon-greedy strategy. In other words, the DQNs can be regarded as the learnt policy based on the interactions with the environment. Dependent on the number of bins to be deployed, the same number of DQNs are independently implemented in the DRL process, while state-action values obtained from the DQNs are consolidated in a union set. Following the typical logic of DRL, the action with the highest state action value is selected. Based on the above deployment, the proposed optimiser is capable of handling multiple bins simultaneously, including single bin, dual bin, and more, to perform informed bin packing decisions.

4.4. Analysis of the single bin scenario (S1)

In the single bin scenario, only one bin's information, namely height map and action map, is processed by the DQN. Subsequently, the trained agents with the learnt policy are applied in the bin packing scenario for the performance evaluation, where the epsilon is set at 0 to avoid making any random actions. During the inference process, 1000 instances to pack 200 items in a bin with the control of lookahead values are considered. The average space utilisation by using the four aforementioned heuristics approaches and the proposed optimiser are summarised in Table 1. In brief, the proposed DRL-based optimiser outperforms the best heuristics approach with 5.73 %, 5.76 % and 3.15 % at $k = 5$, 10 and 15, respectively. Contrary to the heuristic approaches, the average space utilisation is relatively stable when the lookahead value is increased. Considering more items in the bin packing process does not greatly influence the space utilisation.

In addition, the analysis of variance (ANOVA) is conducted to evaluate the mean differences between the methods. It is found that at least one of the means is different from others with the 95 % confidence level for all three lookahead settings. Subsequently, the Tukey-Kramer test, the post-hoc test of ANOVA, is applied to examine the significance of mean differences between any two methods, where the test results are summarised in Table 2. The differences in means between the proposed optimiser and all other existing heuristic methods are statistically significant. When $k = 15$, the bin packing performance of methods H1, H2 and H3 is similar. Therefore, it can be summarised that the proposed optimiser significantly outperforms other heuristic methods in the online single bin packing scenario.

4.5. Analysis of the dual bin scenarios (S2)

In light of the viability of the single bin scenario, online bin packing for two bins simultaneously is considered in this scenario. Regarding the design of the DRL-based optimiser, two DQNs are deployed to handle the height and action maps for two independent bins, while a union set of Q values is constructed to perform informed bin packing actions. In addition, two proposed replacement strategies, namely replaceAll and replaceMax, are applied in the simulation experiments.

For using the replaceAll strategy, the average space utilisation and its variance obtained by the proposed optimiser and four existing heuristic algorithms are summarised in Table 3. When the size of lookahead items increases from 5 to 10, the average performance of the proposed optimiser slightly increases. The proposed optimiser outperforms the best heuristic approach in terms of average space utilisation ranged by 5.56 % and 4.62 %. However, when the lookahead value increases to 15, the average performance of the proposed optimiser drops to the utilization of 72.2 % This implies the limitation of proposed optimiser to deal with the large-scale combinational problem. In other words, the proposed optimiser does not always take the advantage of a larger space of combinatorial possibilities. To further validate the mean differences in 1000 testing instances, the ANOVA and Tukey-Kramer tests are applied, while the corresponding results are summarised in Table 4. It is found that only the performance between H2 and H3 are insignificant whatever the lookahead values are. Moreover, the performance of the proposed optimiser is significantly better than other heuristic approaches when $k = 5$ and 10.

For using the replaceMax strategy, the average space utilisation and its variance obtained by the proposed optimiser and four existing heuristic algorithms are summarised in Table 5. Differing to the use of replaceAll strategy, the proposed optimiser outperforms the best

Table 1
Performance summary among different approaches in the single bin scenario.

Average (variance)	$k = 5$	$k = 10$	$k = 15$
H1-BL	67.68 % (0.0033)	70.21 % (0.0023)	72.17 % (0.0022)
H2-BVF	69.61 % (0.0024)	71.18 % (0.0025)	72.04 % (0.0023)
H3-BSSF	70.10 % (0.0020)	71.37 % (0.0020)	72.06 % (0.0019)
H4-BLSF	61.56 % (0.0019)	62.58 % (0.0021)	63.37 % (0.0021)
Proposed optimiser	74.11 % (0.0022)	75.48 % (0.0017)	74.45 % (0.0020)
F-value	898.44	1034.58	877.14
p-value	0	0	0

Table 2
Results of the Tukey-Kramer post-hoc tests in the single bin scenario.

Method		$k = 5$			$k = 10$			$k = 15$		
A	B	AMD	Critical Q	Sig.	AMD	Critical Q	Sig.	AMD	Critical Q	Sig.
H1	H2	0.0192	0.0059	Yes	0.0097	0.0056	Yes	0.0013	0.0056	No
H1	H3	0.0241	0.0059	Yes	0.0116	0.0056	Yes	0.0011	0.0056	No
H1	H4	0.0613	0.0059	Yes	0.0763	0.0056	Yes	0.0881	0.0056	Yes
H1	Proposed	0.0643	0.0059	Yes	0.0528	0.0056	Yes	0.0228	0.0056	Yes
H2	H3	0.0049	0.0059	No	0.0019	0.0056	No	0.0002	0.0056	No
H2	H4	0.0805	0.0059	Yes	0.0860	0.0056	Yes	0.0867	0.0056	Yes
H2	Proposed	0.0451	0.0059	Yes	0.0431	0.0056	Yes	0.0241	0.0056	Yes
H3	H4	0.0854	0.0059	Yes	0.0880	0.0056	Yes	0.0869	0.0056	Yes
H3	Proposed	0.0402	0.0059	Yes	0.0411	0.0056	Yes	0.0239	0.0056	Yes
H4	Proposed	0.1256	0.0059	Yes	0.1291	0.0056	Yes	0.1108	0.0056	Yes

Remarks: AMD refers to the absolute mean difference; Critical Q denotes the Q critical value used in the Tukey-Kramer post-hoc test; Sig. denotes the testing significance.

Table 3
Performance summary among different approaches in using the replaceAll strategy.

Average (variance)	$k = 5$	$k = 10$	$k = 15$
H1-BL	67.73 % (0.0009)	67.80 % (0.0013)	67.69 % (0.0014)
H2-BVF	71.24 % (0.0006)	72.55 % (0.0007)	73.03 % (0.006)
H3-BSSF	71.13 % (0.0006)	72.37 % (0.0006)	72.92 % (0.006)
H4-BLSF	62.04 % (0.0006)	62.63 % (0.0006)	62.92 % (0.006)
Proposed optimiser	75.19 % (0.0008)	75.89 % (0.0004)	72.18 % (0.005)
F-value	3443.94	3583.00	2607.06
p-value	0	0	0

heuristic approach in terms of average space utilisation by 6.01 %, 5.22 % and 0.16 % for $k = 5,\ \ 10$and15, respectively. To further validate the mean differences in 1000 testing instances, the ANOVA and Tukey-Kramer tests are applied, while the corresponding results are summarised in Table 6. Apart from the insignificant performance differences between H2 and H3, the performance differences among the proposed optimiser, H2 and H3 at $k = 15$ is also similar, while the significant mean differences cannot be found.

Across two replacement strategies deployed in the proposed optimiser, it is found that the replaceMax strategy is effective to improve the average space utilisation, while the corresponding variance can be slightly reduced. For $k = 5$, 10 and 15, the average space utilisation in using the replaceMax strategy are higher with the change of 1.53 %, 1.60 % and 2.35 %. When the replaceMax strategy is applied, only one bin is replaced by a new one, while there is a likelihood that suitable future items can be packed in the remaining bin. Therefore, the space utilisation by using the replaceMax strategy is relatively better. However, the performance of the proposed optimiser drops at $k = 15$ such that its performance is similar or even worse than BVF and BSSF. It shows the limits of the capability of the double DQN to consider 15 lookahead items in the online optimisation process. Overall, Fig. 9 illustrates the boxplots of all experiments to visualise the distribution of the space utilisation over 1000 instances per case.

5. Discussion and implications

Based on the simulation experiments in various scenarios of the case study, the superiority of the proposed optimiser is observed, compared with four well-known heuristic approaches. In this section, the experimental results are further synthesized and discussed, while the implications of this research are provided.

5.1. Discussion on the online 3D-DBPP optimiser

Based on the experimental findings, it is evident that the proposed optimiser is capable of effectively performing online 3D bin packing without complete prior knowledge of incoming items. The optimiser demonstrated superior average space utilization across all three simulation experiments, outpacing the best-performing heuristic by margins of 5.77 %, 5.20 %, and 0.72 % when the lookahead values were set to 5, 10, and 15, respectively. Fig. 10 illustrates the average space utilization and the performance differentials between the proposed optimiser and

Table 5
Performance summary among different approaches in using the replaceMax strategy.

Average (variance)	$k = 5$	$k = 10$	$k = 15$
H1-BL	68.91 % (0.0007)	69.42 % (0.0007)	69.96 % (0.0007)
H2-BVF	72.01 % (0.0005)	73.28 % (0.0006)	73.75 % (0.0005)
H3-BSSF	71.89 % (0.0005)	73.06 % (0.0005)	73.59 % (0.0005)
H4-BLSF	63.25 % (0.0005)	63.78 % (0.0005)	64.04 % (0.0005)
Proposed optimiser	76.35 % (0.0007)	77.11 % (0.0003)	73.87 % (0.0004)
F-value	4037.70	4877.51	3301.89
p-value	0	0	0

Table 4
Results of the Tukey-Kramer post-hoc tests in using the replaceAll strategy.

Method		$k = 5$			$k = 10$			$k = 15$		
A	B	AMD	Critical Q	Sig.	AMD	Critical Q	Sig.	AMD	Critical Q	Sig.
H1	H2	0.0351	0.0032	Yes	0.0475	0.0033	Yes	0.0533	0.0033	Yes
H1	H3	0.0340	0.0032	Yes	0.0457	0.0033	Yes	0.0523	0.0033	Yes
H1	H4	0.0570	0.0032	Yes	0.0517	0.0033	Yes	0.0477	0.0033	Yes
H1	Proposed	0.0746	0.0032	Yes	0.0809	0.0033	Yes	0.0448	0.0033	Yes
H2	H3	0.0010	0.0032	No	0.0017	0.0033	No	0.0011	0.0033	No
H2	H4	0.0920	0.0032	Yes	0.0991	0.0033	Yes	0.1011	0.0033	Yes
H2	Proposed	0.0396	0.0032	Yes	0.0335	0.0033	Yes	0.0085	0.0033	Yes
H3	H4	0.0910	0.0032	Yes	0.0974	0.0033	Yes	0.1000	0.0033	Yes
H3	Proposed	0.0406	0.0032	Yes	0.0352	0.0033	Yes	0.0074	0.0033	Yes
H4	Proposed	0.1316	0.0032	Yes	0.1326	0.0033	Yes	0.0926	0.0033	Yes

Remarks: AMD refers to the absolute mean difference; Critical Q denotes the Q critical value used in the Tukey-Kramer post-hoc test; Sig. denotes the testing significance.

Table 6
Results of the Tukey-Kramer post-hoc tests in using the replaceMax strategy.

Method		$k = 5$			$k = 10$			$k = 15$		
A	B	AMD	Critical Q	Sig.	AMD	Critical Q	Sig.	AMD	Critical Q	Sig.
H1	H2	0.0310	0.0029	Yes	0.0387	0.0028	Yes	0.0379	0.0029	Yes
H1	H3	0.0298	0.0029	Yes	0.0364	0.0028	Yes	0.0363	0.0029	Yes
H1	H4	0.0566	0.0029	Yes	0.0564	0.0028	Yes	0.0593	0.0029	Yes
H1	Proposed	0.0744	0.0029	Yes	0.0769	0.0028	Yes	0.0391	0.0029	Yes
H2	H3	0.0012	0.0029	No	0.0022	0.0028	No	0.0016	0.0029	No
H2	H4	0.0876	0.0029	Yes	0.0950	0.0028	Yes	0.0972	0.0029	Yes
H2	Proposed	0.0433	0.0029	Yes	0.0383	0.0028	Yes	0.0012	0.0029	No
H3	H4	0.0864	0.0029	Yes	0.0928	0.0028	Yes	0.0956	0.0029	Yes
H3	Proposed	0.0445	0.0029	Yes	0.0405	0.0028	Yes	0.0028	0.0029	No
H4	Proposed	0.1309	0.0029	Yes	0.1333	0.0028	Yes	0.0984	0.0029	Yes

Remarks: AMD refers to the absolute mean difference; Critical Q denotes the Q critical value used in the Tukey-Kramer post-hoc test; Sig. denotes the testing significance.

the best heuristic approach among H1, H2, H3, and H4.

For heuristic methods, the observed trend aligns with expectations that average space utilization increases as the lookahead window widens, allowing for more information about upcoming items. The use of the proposed DRL-based optimiser resulted in an initial performance gap of approximately 5 % when $k = 5$ compared to the best heuristics. An upward trend in bin packing performance is noted when the lookahead value is increased to 10, permitting consideration of a larger set of lookahead items. However, in the dual bin scenario using the replaceAll strategy, the performance increase is less pronounced. Surprisingly, further elevating the lookahead value to 15 leads to a significant performance decline. In the single bin setting, the performance differential between the proposed optimiser and the best heuristics narrows to approximately 2 %. In the dual bin scenario, the replaceAll strategy's performance falls behind the best heuristics, while the replaceMax strategy's performance is on par with the best heuristics. This outcome challenges the conventional expectation that a higher lookahead value correlates with improved bin packing performance. The observed decline at a lookahead value of 15 is attributed to the proposed DQN structure reaching its capacity, as depicted in Fig. 6, resulting in less informed decision-making.

The experimental results reveal a phenomenon when the lookahead value (k) is increased to 15. While the constructive heuristic methods consistently improve their performance as k increases, our proposed optimizer exhibits a different pattern. Specifically, the performance gap between the proposed optimizer and the best heuristic method narrows or even reverses at $k = 15$, contrasting with the clear advantages observed at $k = 5$ and $k = 10$. Based on the investigation, this phenomenon can be attributed to two main factors as follows. Firstly, as the lookahead value increases, the feasible action space expands notably. This expansion poses a challenge for the reinforcement learning agent, as it requires more extensive exploration to accurately estimate expected returns. The larger action space makes it more difficult for the agent to converge on an optimal policy within the given training time. This effect particularly happens at $k = 15$, where the action space complexity may outpace the agent ability to effectively explore and learn. Secondly, the performance decline at $k = 15$ may also be attributed to limitations in the model capacity to handle the increased complexity. In our study, we employ a convolutional neural network (CNN) to approximate the Q-value function of the policy (as illustrated in Fig. 6). While this architecture has proven effective for smaller lookahead values, it may struggle to memorize and generalize from all explored state-action pairs when k is increased to 15. This suggests that the current CNN architecture may be insufficient to capture the full complexity of the problem at higher lookahead values.

Furthermore, except for the use of the proposed optimiser at $k = 15$, the average space utilization in the dual bin scenario surpasses that of the single bin scenario. Packing in two bins concurrently provides added flexibility for item placement, which in turn can boost bin packing performance. Notably, in the dual bin scenario, the replaceMax strategy consistently outperforms the replaceAll strategy, irrespective of the algorithms used. In certain situations, the remaining space in completed bins, based on the current lookahead information, can be more effectively utilized. In summary, the replaceMax strategy emerges as the preferred approach in the dual bin scenario when compared to the replaceAll strategy.

5.2. Theoretical and practical implications

From the research perspectives, this study makes a significant contribution to the field of operations research by applying DRL to address the online 3D bin packing problem, specifically by managing multiple bins concurrently to increase space utilization. Utilizing a DDQN approach, this research delineates a near-optimal policy for 3D bin packing that requires minimal prior knowledge of incoming items. This study surpasses four established heuristic methods, demonstrating the superior capabilities of the proposed DRL approach in dealing with complex decision problems. Additionally, we propose and evaluate two innovative bin replacement strategies, namely replaceAll and replaceMax, within a dual bin context. The findings indicate a clear preference for the replaceMax strategy when managing dual bins, laying a robust theoretical groundwork for enhancing a wide array of online 3D bin packing applications.

From the practical perspectives, the optimiser devised in this study can revolutionize the order packing process by enabling automation that integrates seamlessly with existing order picking technologies, thereby moving towards full automation in robotic warehouses. The simulation experiments indicate that transitioning from a single bin to a dual bin scenario can substantially improve space utilization, thereby contributing to economic and environmental sustainability within the entire order fulfillment process. By optimizing the packing of ordered items into pallets for last-mile delivery, the proposed optimiser ensures that truck capacities are maximized. This optimization not only contributes to cost savings but also reduces the carbon footprint associated with transportation by decreasing the number of trips required. As such, the advancements presented in this research can be seen as a pivotal step in enhancing the sustainability and efficiency of robotic warehouse operations.

Furthermore, the success of online and concurrent 3D bin packing optimization for robotic palletization contributes to the design of facility layout in robotic warehouses. Traditionally, only one robotic arm is deployed per conveyor belt for robotic palletization, operating independently from other robotic arms in the process. As illustrated in Fig. 11, the conventional single-arm palletization can be transformed into dual-arm and hybrid palletization configurations through the deployment of the proposed online 3D-DBPP optimizer. In the dual-arm palletization setup, although additional space is required to accommodate two robotic arms alongside the conveyor belt, the overall robotic palletization process benefits from increased productivity and bin packing performance across all conveyor belts. Conversely, in the hybrid

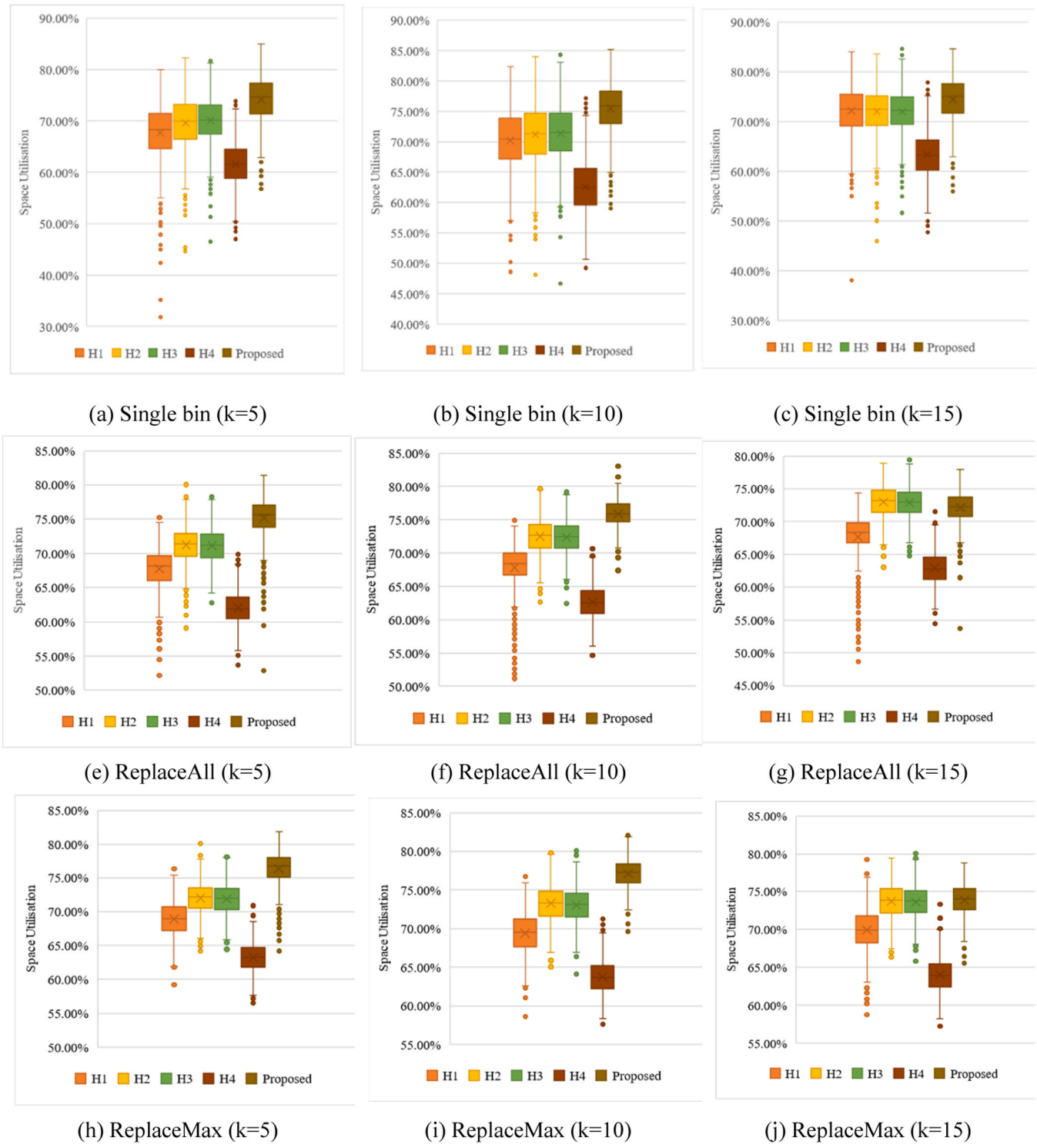


Fig. 9. Boxplots of the space utilisation among different methods.

palletization setup, only an additional robotic arm is integrated into the single-arm palletization system, enabling one of the conveyor belts to leverage the advantages of enhanced productivity and bin packing performance. This configuration not only preserves a higher degree of flexibility and responsiveness in packing items but also allows the proposed optimizer to be deployed selectively, further enhancing overall productivity and bin packing efficiency due to improved utilization by dual-arm palletization. The dual-arm palletization layout, which results in higher utilization from our proposed approach, enables companies to achieve greater efficiency, lower operational costs, and increased flexibility for lean operations in smart logistics.

6. Conclusion

This study addresses the online 3D bin packing problem through the lens of deep reinforcement learning (DRL), innovating the way that multiple bins are concurrently managed to heighten space utilization. Leveraging a double deep Q-network (DDQN) methodology, the

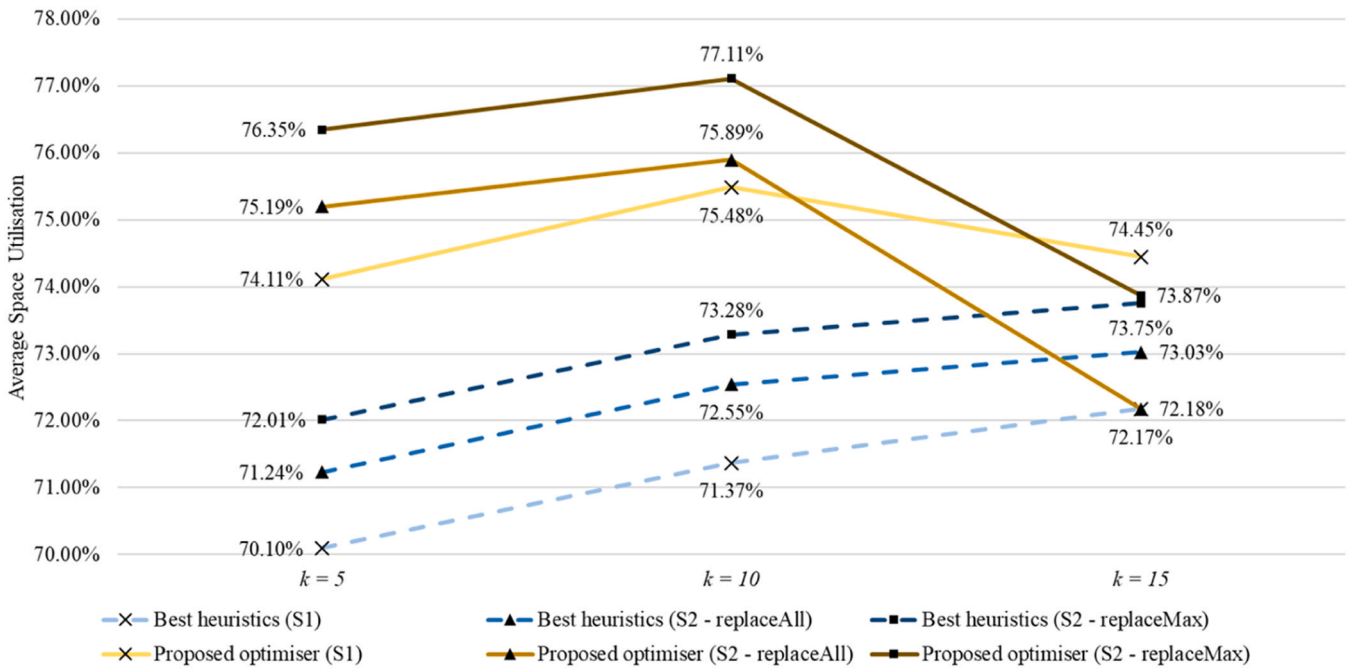


Fig. 10. Average bin packing performance over the lookahead values.

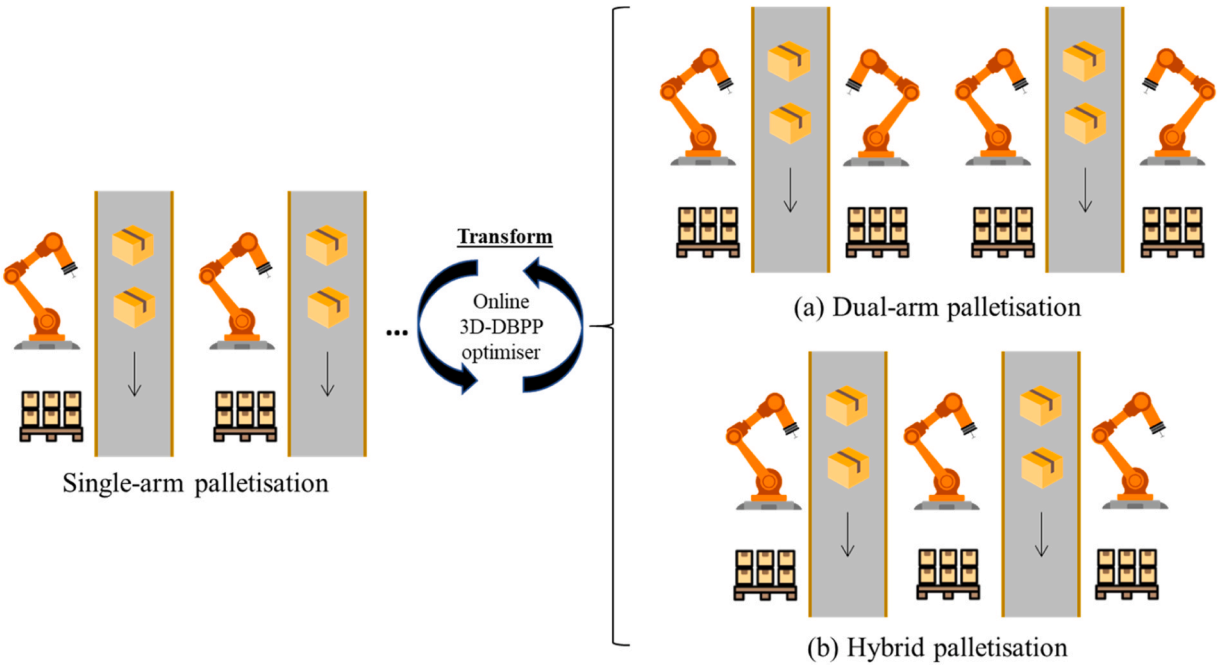


Fig. 11. Configuration changes in the robotic palletisation layout.

research identifies an optimal packing policy with limited prior item information, showcasing methodological advancements over traditional heuristic approaches. Two bin replacement strategies, replaceAll and replaceMax, are introduced and scrutinized within a dual bin environment, with replaceMax emerging as the preferred strategy. Simulation experiments reveal that the proposed optimiser significantly improves space utilization, advancing economic and environmental sustainability in robotic warehouse operations. This optimiser also paves the way for the full automation of order packing processes when integrated with existing order picking technologies. The outcomes of this study illustrate a potent solution that not only boosts operational efficiency but also fortifies the sustainability of modern warehousing systems.

The limitation of this study is its focus on a constrained set of bin and item sizes, which may consider different operational considerations such as loading balance, multiple objective measurements, irregular shape of items in diverse warehousing scenarios. While the proposed DRL model shows promise, its scalability and performance in handling a greater array of bin quantities and a wider variety of item dimensions is yet to be determined. Future research should aim to enhance the model robustness for dealing with different types of operational considerations to larger and more complex problem instances, thereby ensuring its efficacy and applicability in a broader range of real-world settings.

Moreover, alternative neural network architectures should be investigated, such as transformer-based models or graph neural networks, which may be better suited to capture the increased complexity at higher lookahead values. Advanced exploration techniques, such as intrinsic motivation or curiosity-driven exploration, can be deployed to help the agent navigate the larger action space more effectively.

CRediT authorship contribution statement

D. Y. Mo: Writing – review & editing, Methodology, Formal analysis. **K. T. Chung:** Methodology, Formal analysis, Data curation. **C. K. M. Lee:** Writing – review & editing, Supervision. **Y.P. Tsang:** Writing – original draft, Visualization, Methodology, Formal analysis, Conceptualization.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is supported by a Grant of Hong Kong Polytechnic University (project code: G-UARK), and in part by the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. UGC/FDS14/E05/21).

Appendix A. Notation of 3D-bin packing problem

Table A
Notation of 3D-bin packing problem

Indices:	
i	Indices for items (i =1,…, n)
d	Indices for dimensions (d = 1, …, 3)
r	Indices for rotations (r = 1,…, 6)
Parameters:	
$z_{i,r,d}$	Size of item i rotated with rotation r along dimension d
Z_d	Size of bin along dimension d
m_i	Weight of item i
v_i	Volume of item i
V	Volume of bin
Decision variables:	
x_i	If item i is packed into the bin, $x_i = 1$; otherwise, $x_i = 0$
$R_{i,r}$	If item i is rotated with rotation r, $R_{i,r} = 1$; otherwise, $R_{i,r} = 0$
$C_{i,d}$	Coordinate of the packed position of the item i along dimension d
$y_{i,j,d}$	If item i is packed and item j is not packed into the bin along dimension d, $y_{i,j,d} = 1$; otherwise,$y_{i,j,d} = 0$

Appendix B. Notation of the variables used in the 3D-DBPP optimiser

Table B
Notation of the variables used in the 3D-DBPP optimiser

Variable/Set	Descriptions
k	Lookahead value
W, H, D	Dimensions of the bin
w, h, d	Dimensions of the item
$x_{\min}, y_{\min}, z_{\min}, \quad x_{\max}, y_{\max}, z_{\max}$	Bounding box of the cuboid (i.e., the placement and size of an item)
$\mathscr{H}$	Height map
$\mathscr{A}$	Action map
I	Item list
$\mathscr{M}$	Maximal partitioning space
s_t, a_t, r_t	State, action, and reward at timestep t
S, A, R, P	State space, action space, reward function, and state transition function in a Markov Decision Process
π	Policy of the agent

Data availability

Data will be made available on request.

References

Abdou, G., Yang, M., 1994. A systematic approach for the three-dimensional palletization problem. Int. J. Prod. Res. 32 (10), 2381–2394.

Ali, S., Ramos, A.G., Carravilla, M.A., Oliveira, J.F., 2022. On-line three-dimensional packing problems: A review of off-line and on-line solution approaches. Comput. Ind. Eng. 168, 108122.

Baldi, M.M., Perboli, G., Tadei, R., 2012. The three-dimensional knapsack problem with balancing constraints. Appl. Math. Comput. 218 (19), 9802–9818.

Chazelle, 1983. The bottomn-left bin-packing heuristic: An efficient implementation. IEEE Trans. Comput. 100 (8), 697–707.

Elhedhli, S., Gzara, F., Yildiz, B., 2019. Three-dimensional bin packing and mixed-case palletization. INFORMS J. Optim. 1 (4), 323–352.

Erbayrak, S., Özkır, V., Mahir Yıldırım, U., 2021. Multi-objective 3D bin packing problem with load balance and product family concerns. Comput. Ind. Eng. 159.
Gajda, M., Trivella, A., Mansini, R., Pisinger, D., 2022. An optimization approach for a complex real-life container loading problem. Omega 107, 102559.
Gonçalves, J.F., Resende, M.G., 2013. A biased random key genetic algorithm for 2D and 3D bin packing problems. Int. J. Prod. Econ. 145 (2), 500–510.
Ha, C.T., Nguyen, T.T., Bui, L.T., Wang, R., 2017. An online packing heuristic for the three-dimensional container loading problem in dynamic environments and the Physical Internet. In: In *Applications of Evolutionary Computation: 20th European Conference, EvoApplications 2017, Amsterdam, The Netherlands, April 19-21, 2017, Proceedings, Part II*, 20. Springer International Publishing, pp. 140–155.
Hassan, A., Pillay, N., 2019. Hybrid metaheuristics: An automated approach. Expert Syst. Appl. 130, 132–144.
Jylänki, J., 2010. A Thousand Ways Pack. Bin. -A Pract. Approach two-Dimens. rectangle Bin. Pack. (retrived from http://clb. demon. fi/files/RectangleBinPack. pdf).
Kurpel, D.V., Scarpin, C.T., Junior, J.E.P., Schenekemberg, C.M., Coelho, L.C., 2020. The exact solutions of several types of container loading problems. Eur. J. Oper. Res. 284 (1), 87–107.
Liu, H., Zhou, L., Yang, J., Zhao, J., 2023. The 3D bin packing problem for multiple boxes and irregular items based on deep Q-network. Appl. Intell. 53 (20), 23398–23425.
Martin, M., Oliveira, J.F., Silva, E., Morabito, R., Munari, P., 2021. Three-dimensional guillotine cutting problems with constrained patterns: MILP formulations and a bottom-up algorithm. Expert Syst. Appl. 168, 114257.
Martinez, S., Mariño, A., Sanchez, S., Montes, A.M., Triana, J.M., Barbieri, G., Abolghasem, S., Vera, J., Guevara, M., 2021. A digital twin demonstrator to enable flexible manufacturing with robotics: A process supervision case study. Prod. Manuf. Res. 9 (1), 140–156.
Moon, I., Nguyen, T.V.L., 2014. Container packing problem with balance constraints. OR Spectr. 36, 837–878.
Panzer, M., Bender, B., 2022. Deep reinforcement learning in production systems: a systematic literature review. Int. J. Prod. Res. 60 (13), 4316–4341.
Paquay, C., Limbourg, S., Schyns, M., Oliveira, J.F., 2018. MIP-based constructive heuristics for the three-dimensional Bin Packing Problem with transportation constraints. Int. J. Prod. Res. 56 (4), 1581–1592.
Puche, A.V., Lee, S., 2022. Online 3d bin packing reinforcement learning solution with buffer (pp.). In *2022 ieee/rsj international conference on intelligent robots and systems (iros)*. IEEE, pp. 8902–8909 (pp.).
Tian, R., Kang, C., Bi, J., Ma, Z., Liu, Y., Yang, S., Li, F., 2023. Learning to multi-vehicle cooperative bin packing problem via sequence-to-sequence policy network with deep reinforcement learning model. Comput. Ind. Eng. 177, 108998.
Xavier, E.C., Miyazawa, F.K., 2008. A one-dimensional bin packing problem with shelf divisions. Discret. Appl. Math. 156 (7), 1083–1096.
Yang, Y., Liu, B., Li, H., Li, X., Wang, G., Li, S., 2023. A nesting optimization method based on digital contour similarity matching for additive manufacturing. J. Intell. Manuf. 34 (6), 2825–2847.
Yang, S., Song, S., Chu, S., Song, R., Cheng, J., Li, Y., Zhang, W., 2023. Heuristics Integrated Deep Reinforcement Learning for Online 3D Bin Packing. IEEE Trans. Autom. Sci. Eng.
Yang, T.T., Tsang, Y.P., Wu, C.H., Chung, K.T., Lee, C.K.M., Yuen, S.S.M., 2023. Mixed reality-based online 3D pallet loading problem to achieve augmented intelligence in e-fulfilment processes. Oper. Manag. Res. 1–16.
Zhao, H., She, Q., Zhu, C., Yang, Y., Xu, K., 2021. Online 3D bin packing with constrained deep reinforcement learning. *Proc. AAAI Conf. Artif. Intell.* 35 (1), 741–749.
Zhao, H., Zhu, C., Xu, X., Huang, H., Xu, K., 2022. Learning practically feasible policies for online 3D bin packing. Sci. China Inf. Sci. 65 (1), 112105.
Zhu, W., Fu, Y., Zhou, Y., 2024. 3D dynamic heterogeneous robotic palletization problem. Eur. J. Oper. Res. https://doi.org/10.1016/j.ejor.2024.02.007.