Software Impacts 23 (2025) 100732

Contents lists available at ScienceDirect

Software Impacts

journal homepage: www.journals.elsevier.com/software-impacts

Original Software Publication

DeepPack3D: A Python package for online 3D bin packing optimization by deep reinforcement learning and constructive heuristics

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 Special Administrative Region
[b] Department of Supply Chain and Information Management, The Hang Seng University of Hong Kong, Shatin, Hong Kong Special Administrative Region

ARTICLE INFO

Keywords:
Online optimization
3D bin packing
Deep reinforcement learning
Constructive heuristics
Python

ABSTRACT

The rapid advancement of industrial robotic automation has increased the significance of online 3D bin packing optimization for applications, like palletization and container loading. Despite numerous learning-based methods emerging for informed decision-making in this process, the absence of a standardized benchmark makes it challenging to experience the process and validate new algorithms. To bridge this gap, we introduce DeepPack3D, a software package that integrates deep reinforcement learning and constructive heuristic approaches for online 3D bin packing optimization. DeepPack3D provides a foundation for benchmarking, allowing users to evaluate performance using customizable item lists and lookahead values, thereby facilitating consistent research advancements.

Code metadata

Current code version	v1.0
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2024-311
Permanent link to reproducible capsule	https://codeocean.com/capsule/2079012/tree
Legal code license	MIT License
Code versioning system used	Git
Software code languages, tools and services used	Python
Compilation requirements, operating environments and dependencies	Numpy (1.26.4), Matplotlib (3.9.0), tensorflow (2.10.0)
If available, link to developer documentation/manual	
Support email for questions	yungpo.tsang@polyu.edu.hk

1. Motivation and impact

The advent of industrial robotics has revolutionized supply chain operations, automating traditionally labor-intensive tasks such as order packing, palletization, and container loading [1,2]. This transformation not only enhances operational efficiency but also optimizes space utilization, driving a surge of interest in online 3D bin packing optimization. Effective bin packing algorithms facilitate informed decision-making processes, even with limited prior knowledge, thereby addressing critical challenges in dynamic and complex environments. Ali et al. [3] identified key obstacles in solving online 3D bin packing problems, primarily focusing on eliminating the improvement phase during the packing process. They highlighted two main streams of solution algorithms: constructive heuristics and machine learning-based methods. Given the high complexity and significant industrial value of online 3D bin packing optimization, recent research has increasingly explored these approaches to develop more efficient and scalable solutions. For instance, Zhao et al. [4] introduced a constrained deep reinforcement learning (DRL) method tailored for online 3D bin packing optimization. Their approach accounts for factors such as lookahead items, multi-bin packing, and item orientation, demonstrating that DRL can effectively learn optimal policies for sequential decision-making tasks in this domain. Similarly, other studies have pursued algorithmic advancements to enhance space utilization in online 3D bin packing [5,6], further validating the potential of machine learning techniques in overcoming traditional limitations. Moreover, Tsang et al. [7] extended the bin packing framework to accommodate the

DOI of original article: https://doi.org/10.1016/j.compind.2024.104202.
* Corresponding author.
E-mail address: yungpo.tsang@polyu.edu.hk (Y.P. Tsang).

https://doi.org/10.1016/j.simpa.2024.100732
Received 18 December 2024; Accepted 23 December 2024

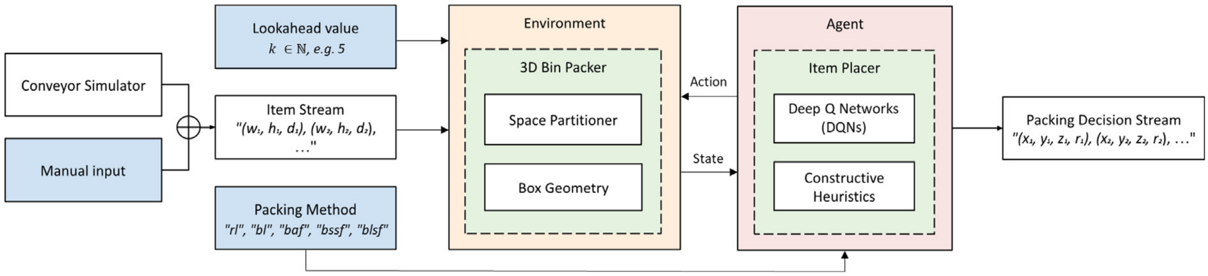

Fig. 1. Framework of DeepPack3D.

simultaneous packing of two bins, thereby supporting more flexible robotic palletization processes in warehouse settings. Their study not only provided a DRL-based solver but also incorporated solutions from various constructive heuristics, including Bottom Left (BL), Best Area Fit (BAF), Best Short Side Fit (BSSF), and Best Long Side Fit (BLSF). These heuristic approaches, originally developed for two-dimensional rectangle bin packing [8], were adapted to address the additional complexity introduced by three-dimensional spaces and concurrent bin packing requirements. The substantial industrial value derived from these research endeavors underscores the necessity for standardized and comparable baselines. Such benchmarks are crucial for evaluating the performance of emerging solution algorithms and fostering further innovation in the field.

In response to this need, we present DeepPack3D, an open-source Python package designed to implement DRL approaches and four key constructive heuristics (BL, BAF, BSSF, and BLSF) for online 3D bin packing optimization. DeepPack3D is freely accessible to researchers and developers, providing a versatile tool for experimentation and development in this critical area of supply chain management. In a nutshell, DeepPack3D offers several key features, such as the following: Firstly, users can easily adjust the lookahead value to tailor the optimization process to specific online scenarios. Additionally, item lists can be generated randomly or input manually, allowing for flexibility in testing various packing conditions and constraints. Secondly, it includes two pretrained DRL models with lookahead values of 5 and 10 to demonstrate practical utility. Lastly, the package visualizes the bin-packing process through images, enabling users to comprehend items' spatial arrangement and orientation within the bins.

Consequently, the impact of DeepPack3D can be summarized into two primary facets. On the one hand, by offering an accessible platform, DeepPack3D enables students, junior researchers, and practitioners to intuitively grasp the application of DRL and constructive heuristics in solving online 3D bin packing problems. Researchers can train new DRL models using diverse item lists and varying lookahead values, facilitating the exploration of different optimization scenarios. On the other hand, it serves as a standardized baseline for evaluating the performance of advanced solution algorithms. Researchers and developers can leverage DeepPack3D to save time and resources otherwise spent on replicating existing research frameworks.

2. Design of DeepPack3D

The concepts and application scenarios of the proposed DeepPack3D optimizer in autonomous robotic warehouses can be referred to in the work [7]. This paper focuses on the implementation of the software modules within this optimizer, each aligning with the pipeline stages of a robotic palletization system. As shown in Fig. 1, the modules operate within a 3D coordinate system. Each cargo item is represented as a cuboid by its width, height, and depth dimensions. The container space where the cuboids are placed is another larger bounded cuboid. The cuboids are placed in the container space by x, y, and z coordinates, anchoring the position of the left-bottom-front corner of the cuboid relative to the container's origin.

2.1. 3D Bin Packer in the environment

The first module of the software is the 3D Bin Packer module, which is the environment in the context of the reinforcement learning framework. The module consists of three main components working collaboratively to achieve a goal. The primary function of this module is to simulate the placement of items as 3D cuboids within the container space with practical constraints, such as the non-overlapping constraint and the bottom support constraint, to prevent the items from floating in mid-air.

2.1.1. Conveyor simulator

The Conveyor Simulator component generates the dimensions of 3D cuboids given an underlying distribution of standard cargo sizes. This component mirrors the palletization process in warehouses, where the complete list of cargo sizes is not known in advance. Instead, the workers can only observe several incoming cargos and make placement decisions on site. In an automatic robotic warehouse, the cargo dimensions are measured at the beginning of the conveyor. It provides real-time data of limited lookahead information to the robotic system to support its placement strategy. The conveyor simulator allows the robotic arms to pick the best candidate cargo within their reach. This flexibility significantly improves compactness because robotic arms can only place cargo from the top rather than from the side. Therefore, the sequential placement of cargo is crucial, as poor choices can obstruct subsequent placements.

2.1.2. Box geometry

Before cargos approach the robotic arms, their dimensions as 3D cuboids are measured by a dimension measurement device. The Box Geometry component declares the position and dimension attributes of a 3D cuboid and implements the geometric set operations required for 3D bin packing. The geometric set operations, including intersection, difference, and complement in 3D space, are extensively utilized in the space partitioning algorithm. The Box Geometry component implements these set operations as a building block for the space partitioner. More importantly, it performs the non-overlapping and bottom support checks to support the practical constraints.

2.1.3. Space partitioner

The Space Partitioner component implements the maximal cuboids algorithm for non-guillotine space partitioning. This component facilitates the placement of cuboids into the container space. It maintains a data structure that represents the occupancy state of a cuboid space. The data structure supports non-discrete coordinates when the space is partitioned along the axes. These features perfectly align with the cargo loading operations in the robotic palletization process. In addition to the space partitioning algorithm, this component generates the height maps as the observation state of the reinforcement learning and renders the placed cuboids for visualization.

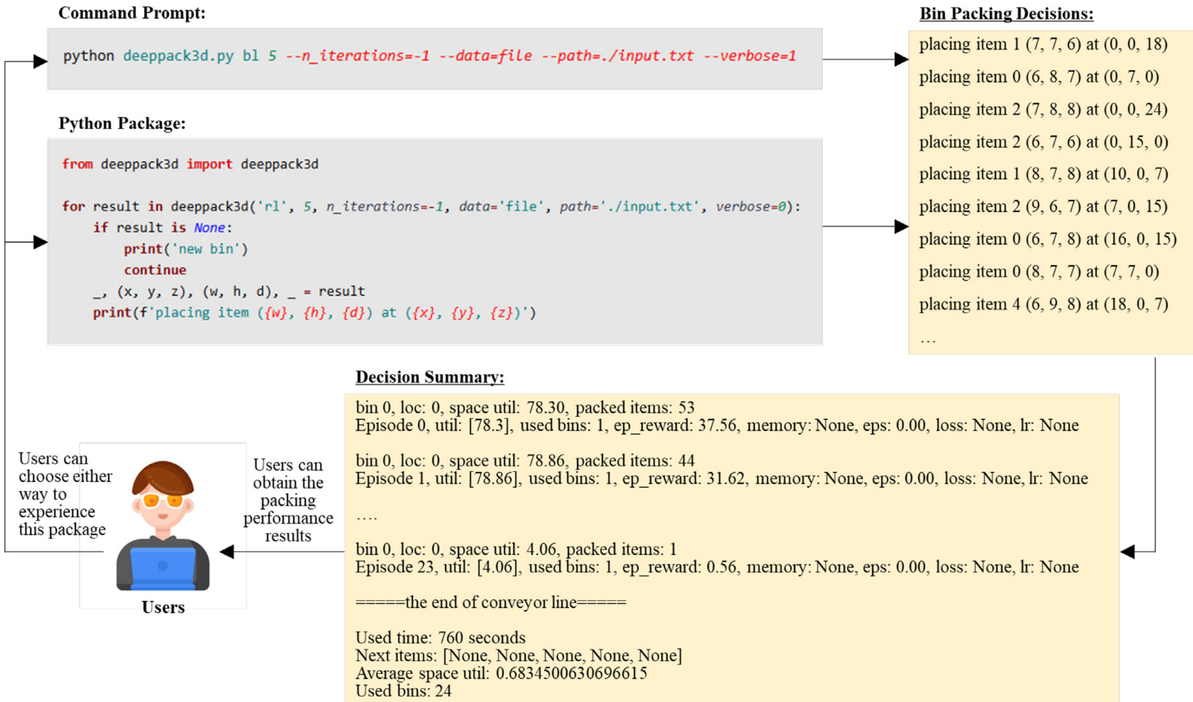

Fig. 2. Illustrative example of the package.

2.2. Item placer as the agent

The Item Placer module served as the agent to provide bin packing decisions, where DRL-based solver and constructive heuristics are provided [7]. Its primary function is determining the coordinates of item placement of the incoming cargo. The Item Placer can allow robotic arms to rotate the items by any combination of axes to fit the unoccupied space better. This strategy aligns with the capabilities of robotic arms.

2.2.1. DRL-based solver

The DRL-based solver is implemented using DQN models. The models take the dimensions of incoming items as the user's input and generate placement decisions that rotate the item cuboid and place it in an unoccupied space. The Bin Packer environment maintains an internal space representation for the state inputs. Before deployment, the DRL-based solver must be trained with a predetermined lookahead value. The DRL-based solver interacts with the 3D Bin Packer environment, which provides a reward for evaluating the quality of an action. The solver agent randomly explores possible actions that lead to higher rewards. Once the solver has been trained, the solver can perform inference without the reward feedback loop and is ready for deployment. The DeepPack3D package includes a pre-trained model for lookahead value of 5, i.e. $k = 5$, for test runs.

2.2.2. Constructive heuristics

In addition to the DRL-based solver, the DeepPack3D package includes a set of constructive heuristics to perform 3D bin packing decisions. These heuristics offer faster inference time but prone to sub-optimal solutions with lower space utilization. There are four heuristics available: Bottom Left (BL), Best Area Fit (BAF), Best Short Side Fit (BSSF), and Best Long Side Fit (BLSF). To ensure a fair comparison, the heuristics and the DRL method operate under the same problem constraints and simulation environment.

3. Illustrative example

DeepPack3D is developed in Python and optimized for GPU-enabled environments, particularly when utilizing reinforcement learning methods. This setup ensures faster training and inference times. The software offers flexibility by allowing execution via the command line for standalone operations or by being imported as a library into Python scripts, facilitating integration into customized workflows. Such versatility enables seamless incorporation with warehouse management systems, robotic controllers, and simulation environments. By default, DeepPack3D operates with a container size of (32, 32, 32) units and generates items with dimensions ranging from (6, 6, 6) to (12, 12, 12) units if the conveyor simulator is configured. Developers requiring different configurations can modify the parameters in the code as needed.

DeepPack3D can be used either via the command prompt or by importing the package in Python as shown in Fig. 2. The command prompt option provides a convenient way to obtain results without modifying the code, while the imported package offers greater flexibility and an interface for integration into existing systems.

During inference, DeepPack3D determines the placement decision, specifying the selected item along with its rotation and position. If the visualization flag is enabled, the software generates step-by-step snapshots of each item's placement into the container, as shown in Fig. 3.

4. Conclusion

This work has made strides in advancing online 3D bin packing optimization by providing a freely-accessible Python-based tool, i.e. DeepPack3D, that leverages deep reinforcement learning and constructive heuristics. It is convenient for users to learn the online 3D bin packing optimization process, and compare the space utilization performance with newly designed solution algorithms. Moreover, its potential in commercial settings underscores its practical value in logistics and warehouse management companies integrating DeepPack3D into automated systems for tasks, such as palletization and container

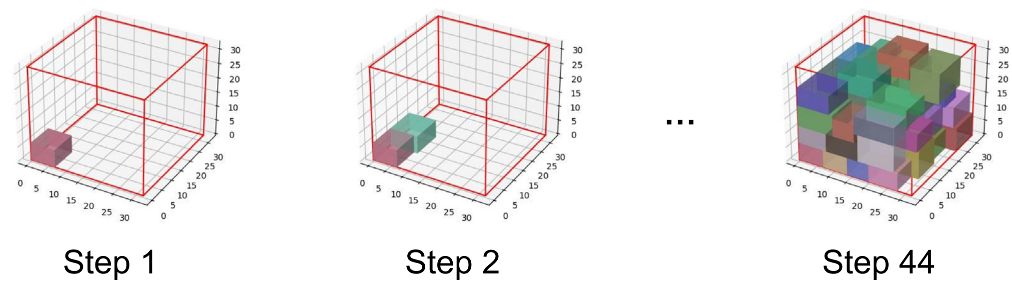

Fig. 3. Visualizations of the step-by-step placement of items into the container.

loading. Despite its potential, DeepPack3D faces certain limitations that present opportunities for future development. Currently, the software requires GPU-enabled environments to fully harness the capabilities of reinforcement learning, which may present accessibility challenges for smaller enterprises or educational institutions with limited resources. Future improvements could include optimizing the software for CPU-based environments to broaden its accessibility, integrating real-time data feeds to enable dynamic and adaptive packing strategies, and expanding the range of customizable parameters to accommodate a wider array of industrial scenarios.

CRediT authorship contribution statement

Y.P. Tsang: Formal analysis, Methodology, Writing – original draft. **D.Y. Mo:** Investigation, Validation, Writing – review & editing. **K.T. Chung:** Investigation, Methodology. **C.K.M. Lee:** Resources, Supervision, Writing – review & editing.

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.

Acknowledgments

This work is supported by a Grant of The Hong Kong Polytechnic University (project code: G-UARK).

References

[1] W. Zhu, Y. Fu, Y. Zhou, 3D dynamic heterogeneous robotic palletization problem, European J. Oper. Res. 316 (2) (2024) 584–596.
[2] G. Jiao, M. Huang, Y. Song, H. Li, X. Wang, Container loading problem based on robotic loader system: An optimization approach, Expert Syst. Appl. 236 (2024) 121222.
[3] S. Ali, A.G. Ramos, M.A. Carravilla, J.F. Oliveira, Online three-dimensional packing problems: A review of off-line and online solution approaches, Comput. Ind. Eng. 168 (2022) 108122.
[4] H. Zhao, Q. She, C. Zhu, Y. Yang, K. Xu, Online 3D bin packing with constrained deep reinforcement learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, (no. 1) 2021, pp. 741–749.
[5] S. Yang, S. Song, S. Chu, R. Song, J. Cheng, Y. Li, W. Zhang, Heuristics integrated deep reinforcement learning for online 3d bin packing, IEEE Trans. Autom. Sci. Eng. 21 (1) (2023) 939–950.
[6] H. Xiong, C. Guo, J. Peng, K. Ding, W. Chen, X. Qiu, et al., GOPT: Generalizable online 3D bin packing via transformer-based deep reinforcement learning, IEEE Robot. Autom. Lett. (2024).
[7] Y.P. Tsang, D.Y. Mo, K.T. Chung, C.K.M. Lee, A deep reinforcement learning approach for online and concurrent 3D bin packing optimisation with bin replacement strategies, Comput. Ind. 164 (2025) 104202.
[8] J. Jylänki, A thousand ways to pack the bin-a practical approach to two-dimensional rectangle bin packing, 2010, Retrived from http://clb.demon.fi/files/RectangleBinPack.pdf.