FOUNDATIONS OF COMPUTING AND DECISION SCIENCES
Vol. 49 (2024) No. 1
ISSN 0867-6356
DOI: 10.2478/fcds-2024-0005 e-ISSN 2300-3405

Online Three-Dimensional Bin Packing: A DRL Algorithm with the Buffer Zone

Jiawei Zhang, Tianping Shuai *

Abstract. The online 3D bin packing problem(3D-BPP) is widely used in the logistics industry and is of great practical significance for promoting the intelligent transformation of the industry. The heuristic algorithm relies too much on manual experience to formulate more perfect packing rules. In recent years, many scholars solve 3D-BPP via deep reinforcement learning(DRL) algorithms. However, they ignore many skills used in manual packing, one of the most important skill is workers put the item aside if the item is packed improperly. Inspired by this skill, we propose a DRL algorithm with a buffer zone. Firstly, we define the wasted space and the buffer zone. And then, we integrate them into the DRL algorithm framework. Importantly, we compare the bin utilization with different thresholds of wasted space and different buffer zone sizes. Experimental results show that our algorithm outperforms existing heuristic algorithms and DRL algorithms.

Keywords: 3D bin packing, buffer zone, wasted space, actor-critic

1. Introduction

The rapid development of e-commerce and the logistics industry has greatly increased the burden on packaging workers. Industrial intelligence is the general trend for the development of the global manufacturing industry. Designing intelligent algorithms to replace human labor with machines is the key way to the high-quality development of the manufacturing industry. A practical packing scenario in the logistics industry is that items are delivered through a conveyor belt, and the packaging workers cannot know the information of all the items to be packed in advance. Only when the item arrives can they know the information of the item. The goal is to pack them into a bin to maximize the bin utilization. This problem can be expressed as an online

*School of Science, Beijing University of Posts and Telecommunications, Beijing, China, {jwzhang, tpshuai}@bupt.edu.cn

3D bin packing problem(3D-BPP). The existing algorithms can be divided into four categories: the approximation algorithm, the exact algorithm, the heuristic algorithm, and the deep reinforcement learning(DRL) algorithm.

In recent years, most scholars use heuristic algorithms to solve the bin packing problem(BPP)[12][24][11]. However, there is a problem with the cold start. The algorithm has to restart the slow climbing search to find the optimal solution, which cannot continue to learn from existing experience. Therefore, it spends a large time when solving large-scale problem instances. Moreover, due to the extra complexity, most scholars ignore some constraints such as the physical stability and the rotatability of items. Therefore, scholars try to use some other algorithms, such as DRL.

DRL algorithm combines deep learning and reinforcement learning algorithms, which integrates the strong understanding ability of deep learning on visual perception and the decision-making ability of reinforcement learning[18]. DRL algorithm realizes end-to-end learning and has been proved that it can be used to solve combinatorial optimization problems[4]. In recent years, some scholars have used DRL to solve the bin BPP [25][13][10]. However, they have ignored some human packing skills.

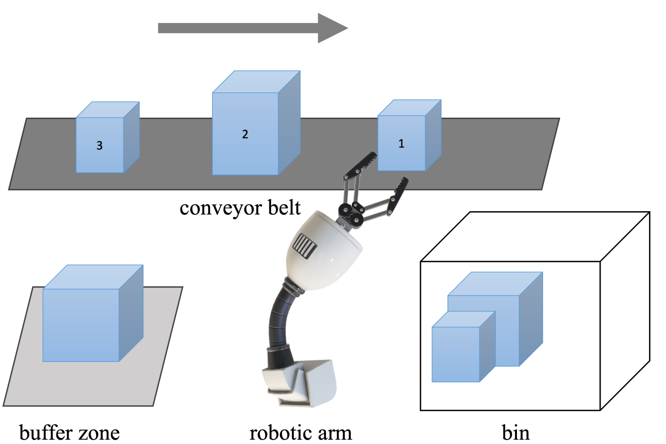

Figure 1. Online 3D-BPP with the buffer zone

In this paper, to improve bin utilization based on Hang Zhao et al.[25], we consider the human packing skill and propose a DRL algorithm with the wasted space and the buffer zone to solve the online 3D bin packing problem. As shown in Figure 1, items arrive sequentially through the conveyor belt. At this time, item 1 be packed, and we can only know the information of its size. We choose to pack the item into the bin or store it in the buffer zone. Specifically, we calculate the wasted space caused by packing the current item into the bin to judge whether put the item into the buffer zone. If there are items in the buffer zone, we randomly select the item on the conveyor belt and the item in the buffer zone to be packed.

The main contributions of this paper are three aspects: Firstly, we define the wasted space to guide the training of the DRL model. Secondly, we set up a buffer zone and apply it to the model framework of DRL. Finally, we compare the bin utilization with different thresholds of wasted space and different buffer zone sizes.

2. Related Work

BPP is one of the most classic discrete combinatorial optimization problems. It can be traced back to the 1960s. Many previous studies have focused on approximation algorithms and heuristic algorithms. The application of the next-fit(NF) algorithm laid the foundation for the development of subsequent research. Baker et al. sorted items in descending order first to propose the next fit descending(NFD), which worst-case performance ratio is 1.691[2]. Martello et al. defined the upper bounds of the maximum cardinality bin packing problem(CBP) to have an absolute worst-case performance ratio equal to $\frac{3}{2}$[14]. Andrea Lodi et al. discussed a simple deterministic approximation algorithm and discuss its worst-case performance. Then they use it as the initial solution of a tabu search approach to attain the same performance as the branch-and-bound algorithm[15]. Silvano Martello et al. discussed lower bounds and proved that the asymptotic worst-case performance ratio of the continuous lower bound is $\frac{1}{8}$. And they solved the BPP based on the branch-and-bound algorithm. However, due to the additional complexity, only small-scale BPP instances can be solved in a reasonable time[17]. Scholl et al. favorably combined the meta-strategy tabu search and a branch and bound procedure to propose an exact hybrid solution procedure[21].

Due to the additional complexity of 3D-BPP, many scholars use heuristic algorithms to solve this problem. Andrea Lodi et al. exploited a new constructive heuristic for the evaluation of the neighborhood to propose a tabu search framework[16]. Alvim et al. used lower bounding strategies to propose a hybrid improvement procedure for the BPP[1]. Kyungdaw Kang et al. proposed a hybrid genetic algorithm with a variant of the deepest bottom left packing strategy to solve the 3D-BPP and compared it with others[11]. Andreas Bortfeldt et al. proposed a hybrid genetic algorithm(GA) based on an integrated greedy heuristic and proved its good performance[3]. Hongfeng Wang et al. proposed a hybrid genetic algorithm that combines a special diploid representation scheme and heuristic packing methods to solve 3D-BPP[23]. Chuan He et al. proposed the Discrete Particle Swarm Optimization(DPSO) algorithm with redefined the velocity and position formula of basic Particle Swarm Optimization to solve 3D-BPP[8].

In recent years, with the development of the DRL, some scholars use DRL to solve the bin packing problem with some success. Hang Zhao et al. introduced a prediction-and-projection scheme to solve online 3D-BPP via the DRL method under the actor-critic framework. Their method might attain a human-level performance[25]. Shuai

Song et al. presented a DRL-based packing-and-unpacking network(PUN) to solve the online 3D-BPP, which can reliably complete the online 3D-BPP task in the real world[22]. Haoyuan Hu et al. used Pointer Net(Ptr-Net) to find a way to place these items that can minimize the surface area of the bin[9]. Quanqing Que et al. employed transformer architecture as the policy network and used proximal policy optimization(PPO) to train the network [20].

The buffer zone was first used in the 1D-BPP. Galambos et al. presented a new next-fit type algorithm for the 1D-BPP with unit size buffer-bins in 1985 and a polynomial time approximation algorithm while using three active bins in 1993[5][6]. Feifeng Zheng et al. proposed an NF-based online algorithm with a capacitated buffer and proved the asymptotic competitive performance(ACP) is $\frac{13}{9}$, the theoretical results show that using a buffer helps in improving the ACP of NF-based algorithms[26]. In recent years, some scholars try to use the buffer zone to solve 3D-BPP. Puche et al. established a buffer zone to receive the information of multiple items for online 3D-BPP and used the Monte-Carlo Tree Search strategy to calculate the best position, which requires a large amount of calculation[19].

3. Algorithm

We define the wasted space and the buffer zone and then propose a DRL algorithm with the buffer zone to solve the online 3D bin packing problem.

3.1. Problem Statement

Considering the practical application scenario of the logistics industry as shown in Figure 1. We give a bin and can only know the current arrival item information. In the process of packing, we put items into the bin to maximize the bin utilization. The packing process is over when the currently arrived item cannot be loaded into the bin. We make the following assumptions:

- The sizes of the items and the bin are positive integers.
- Items can only be placed orthogonally.
- Items can only be placed vertically from the top of the bin.
- Items are allowed to rotate horizontally.
- Items cannot be moved after being packed into the bin.

In this paper, we make the following constraints:

- Items cannot be placed beyond the boundary of the bin.
- Items cannot be overlapped.

- The placement of items needs to satisfy the physical stability.

For definitions of the physical stability and the rotation horizontally, we refer to [25] and store the feasibility of all the packing positions for item i with a feasibility mask M_i, we are not going into details here. Under these assumptions, our objective is to maximize the bin utilization to satisfy the constraints as shown in equation 1.

$$\frac{\sum_{i=1}^{N} l_i w_i h_i}{LWH} \rightarrow max, \tag{1}$$

where l_i, w_i, h_i represent the length, width and height respectively of the item i, L, W, H is the length, width and height respectively of the bin.

3.2. DRL Algorithm

The DRL algorithm formulates the BPP as the Markov decision process(MDP). At step t, the state s_t represents the information about the bin and the item. Based on state s_t, the agent selects the packing position of the item as the action a_t. According to executing the action a_t, we determine the next state s_{t+1} and the reward $R_t(s_t, a_t)$. The algorithm's goal is to seek a policy $\pi(a_t|s_t, \theta)$ to maximize the discount reward as shown in equation 2 where θ is a parameter.

$$u_t = \max_{\theta} \sum_{t=1}^{T} \gamma^{t-1} R_t(s_t, a_t), \tag{2}$$

where γ is the discount factor. It is generally believed that the current reward is more important than the future reward. Figure 2 shows a detailed description of the DRL algorithm framework with the buffer zone(defined later) proposed in this paper. The state includes the state of the item and the state of the bin. Firstly, the feasibility mask shows the stability of the item at every position in the bin to guide the output of the actor-network. Secondly, the buffer module calculates whether to put it in the buffer zone. Finally, the critic-network is used to evaluate the status to update the parameters of the model.

State representation. At step t, the state s_t consists of two parts: the the bin state and the item state. The bin with size $[L, W, H]$ and the item k with size $[l, w, h]$, we separate it into a two-dimensional grid($L \times W$) according to the length(L) and width(H) of the bin as shown in Figure 3, the blue squares in the box represent the items that have been packed into the bin, and the red dotted line represents the wasted space by packing item $k-1$. The value on each grid represents the height that has been placed here in the bin. Therefore, the bin state is represented by a height graph H_t. Assuming that the sizes of the bin and items are positive integers, then $H_n \in [N]^{L \times W}$, where $N \in [0, H]$. The item state with size $[l_t, w_t, h_t]$ is represented by a 3-channel $L \times W$ tensor d_t. The first $L \times W$ tensor is assigned with l_t, the second

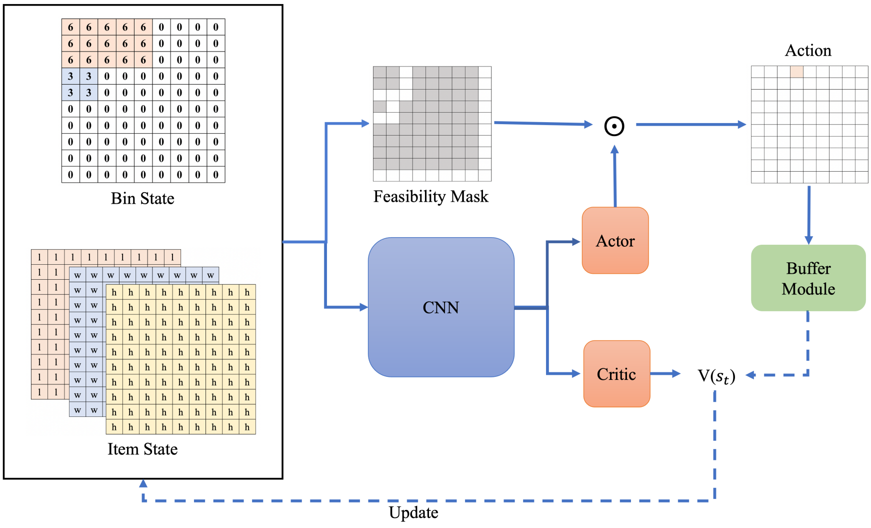

Figure 2. Network framework with the buffer zone

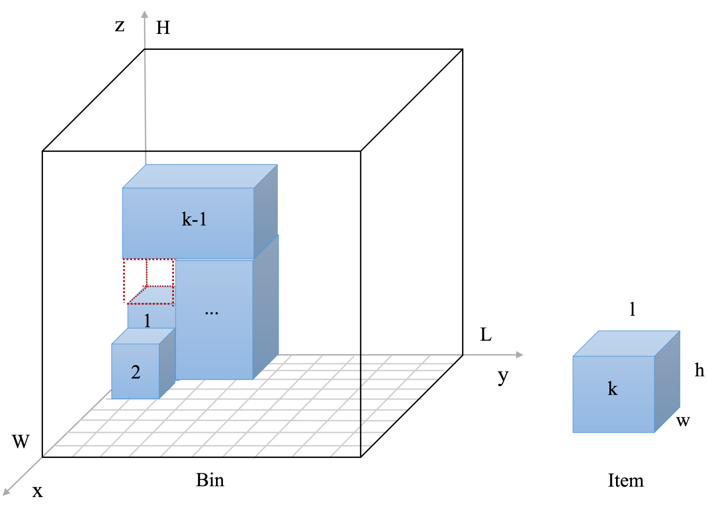

Figure 3. The packing status of the bin and the information about the next item to be packed

$L \times W$ tensor is assigned with w_t and the third $L \times W$ tensor is assigned with h_t as shown in Figure 2.

Action representation. We establish an axis with a corner of the bin as the origin as shown in Figure 3. The action is defined that packing the front-left-bottom of the item at the(x_n, y_n). Thus the action space size is $L \times W$ and the action space A is as follows:

$$\begin{bmatrix} (x_0, y_0) & (x_0, y_1) & \cdots & (x_0, y_{L-1}) \\ (x_1, y_0) & (x_1, y_1) & \cdots & (x_1, y_{L-1}) \\ \vdots & \vdots & \ddots & \vdots \\ (x_{W-1}, y_0) & (x_{W-1}, y_1) & \cdots & (x_{W-1}, y_{L-1}) \end{bmatrix}.$$

Reward function representation. The goal of our model is to maximize the bin utilization, which is to pack as many items as possible. Therefore, we define the free and unpacked space as wasted space, as shown in the red dotted line in Figure 3. We integrate the volume of the current item and the wasted space as our reward to guide the training of the DRL algorithm. The reward function of each step is as follows:

$$R_t(s_t, a_t) = \varphi \times (\frac{l_t w_t h_t}{LWH} - \frac{V_{waste}}{LWH}), \tag{3}$$

where φ is a hyperparameter and V_{waste} is the size of wasted space by executing action a_t.

3.3. Network Frame with the Buffer Zone

Buffer module. In this paper, we assume that the item cannot be adjusted after being packed. It may leads to a poor packing result if packing an item with a largely wasted space. Therefore, we define a buffer space to avoid this situation. Firstly, we set the maximum number of items that can be put into the buffer zone as B. When the number of items in the buffer zone is equal to B, we cannot store a new item in the buffer space and must pack it. Secondly, we set the threshold of wasted space as δ. we store the item in the buffer zone if the wasted space is greater than the threshold δ and the number of items in the buffer space is less than B. Thirdly, we adopt the first-come-first-packed placement rule for the items in the buffer zone. We randomly select the item that enters the buffer zone first and the item that arrives on the conveyor belt as the next item to be packed.

Network with the buffer zone. We adopt the Actor-Critic framework as shown in Figure 2. At learning step t, firstly, we extract features of bin and item states s_t by the convolutional neural network(CNN). Secondly, we select the action a_t by the actor-network $\pi(a_t|s_t, \theta)$ and judge whether store the item in the buffer zone. We store the item in the buffer zone if the wasted space exceeds the threshold δ and the number of items in the buffer space is less than B. Otherwise, we pack the item into the bin, update the bin height map H_t, and select the next item to pack. Thirdly, the critic-network $V(s_t, \omega)$ scores the current state's advantages and disadvantages to update the network parameter.

The process of packing an item is as follows:

- Step 1: If there are items in the buffer zone, randomly select the item that was first put into the buffer zone and the item that arrived on the conveyor belt as the current item d_t to be packed, otherwise select the item that arrived on the conveyor belt.
- Step 2: Compute the current state s_t according to the item information d_t and the bin height map H_t, then select the action a_t according to the strategy $\pi(a_t|s_t, \theta)$.
- Step 3: If the wasted space by executing a_t is greater than its threshold δ and the number of items in the current cache space is less than B, put the item d_t into the buffer space and back to step 1, otherwise, go to step 4.
- Step 4: Execute the action a_t, put the item d_t into the bin, get s_{t+1} and the reward $R_t(s_t, a_t)$, and update the bin height map H_t.

The pseudo-code of the model is presented in Algorithm 1.

Algorithm 1 DRL with the Buffer Zone

Input: $s_t = \{H_t, d_t\}$, δ, B, θ, ω, $t \leftarrow 0$
Output: π_θ

```
for each step t do
    if the number of items in buffer zone n ≠ 0 then
        randomly select d_t from the buffer zone and the conveyor belt
    else
        select d_t from the conveyor belt
    end if
    select a ~ π(·|s_t, θ) and calculate wasted space V_waste
    if V_waste > δ and the number of items in the buffer zone n < B then
        store d_t in the buffer zone, back to step 2
    else
        execute action a to pack d_t into the bin
        calculate reward R_t and update H_t
        calculate V(s_t, ω)
        update θ, ω
        t+ = 1
    end if
end for
```

4. Experiments

In this section, we conduct numerical experiments to demonstrate the effectiveness of the proposed algorithm. We use the CUT-2 dataset in [25].

To evaluate the effectiveness of the buffer zone, we conduct a comparative experiment with the buffer zone(red line) and without the buffer zone(dotted green line). The results are shown in Figure 4, where the horizontal axis is the number of learning steps, and the vertical axis is the bin utilization, it can be seen that the algorithm with the buffer zone has better performance in the same learning steps.

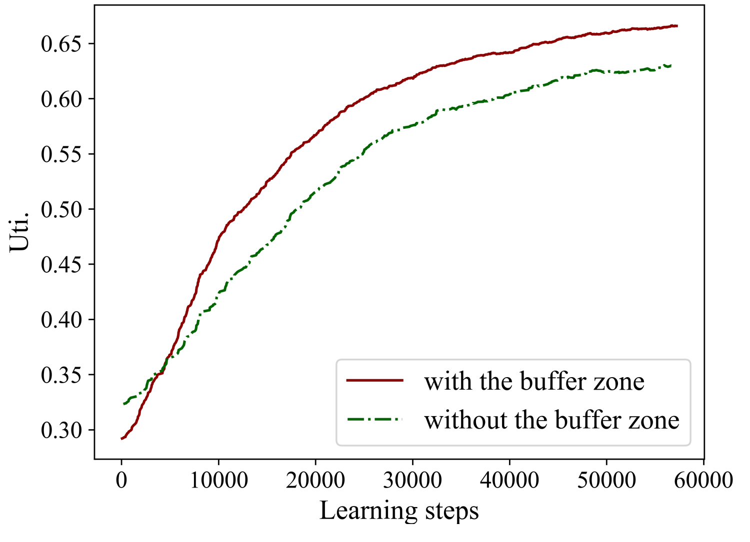

Figure 4. The learning process with the buffer zone and without the buffer zone

In this paper, to optimize the buffer zone and wasted space, the maximum number of items that can be put in the buffer zone is set as $B \in \{2, 5\}$, and the threshold for wasted space is set as $\delta \in \{0.2, 0.05, 0.02\}$. As shown in Table 1, we ran 100's experiments to calculate the average number of items packed in the bin and the average bin utilization. Here, Num. is the average number of items that can be packed in the bin, and Uti. is the average bin utilization. With the increase of the maximum number of items that can be put in the buffer zone B and the decrease of the threshold for wasted space $\delta(\mathrm{Ours}(B, \delta))$, the number of items packed in the bin and the average bin utilization are increased to 19.0 and 71.8% respectively.

To evaluate the proposed method, we compare our algorithm with two heuristic-based methods and a DRL method as shown in Table 2. BPH is a heuristic-based online algorithm, which assigns different priorities to the empty maximal spaces(EMSs)[7]. LBP is a heuristic-based offline algorithm, which defines two different heuristics to obtain a good upper bound[17]. BPP-1 is a DRL algorithm, which adopts the actor-critic

Table 1. Comparison with different thresholds of wasted space and different sizes of the buffer zone

Method	Num.	Uti.
Ours(2, 0.2)	18.2	69.0%
Ours(5, 0.2)	18.6	70.2%
Ours(5, 0.05)	18.7	71.1%
Ours(5, 0.02)	19.0	71.8%

framework[25]. As shown in Table 2, BPH averagely achieves 49.2% bin utilization and averagely packs 13.1 items. LBP averagely achieves 59.5% bin utilization and averagely packs 15.2 items. BPP-1 averagely achieves 66.9% bin utilization and averagely packs 17.5 items. Our method averagely achieves 71.8% bin utilization and averagely packs 19.0 items. From the comparison in Table 2, the method proposed in this paper(Ours(5,0.02)) achieves about 5% improvement than BPP-1.

Table 2. Comparison with others

Method	Num.	Uti.
BPH[7]	13.1	49.2%
LBP[17]	15.2	59.5%
BPP-1[25]	17.5	66.9%
Ours	19.0	71.8%

5. Conclusion

This paper proposes a DRL algorithm with a buffer zone to solve the online 3D bin packing problem. Then, to optimize the buffer zone and wasted space, we compare bin utilization with different thresholds of wasted space and different buffer zone sizes. The experimental results show the effectiveness of the proposed algorithm. In the future, we would like to incorporate heuristic algorithms into our framework and learn to pack items with irregular shape.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 12171051,12171052.

References

[1] Alvim A. C. F., Ribeiro C. C., Glover F., et al. A hybrid improvement heuristic for the one-dimensional bin packing problem *Journal of Heuristics*, **10**, 2004, 205-229.

[2] Baker B. S., Coffman E. G. A Tight Asymptotic Bound for Next-Fit-Decreasing Bin-Packing *SIAM Journal on Algebraic Discrete Methods*, **2**, 2, 1981, 147-152.

[3] Bortfeldt A., Gehring H. A hybrid genetic algorithm for the container loading problem *European Journal of Operational Research*, **131**, 1, 2001, 143-161.

[4] Cappart Q., Moisan T., Rousseau L. M., et al. Combining reinforcement learning and constraint programming for combinatorial optimization *Proceedings of the AAAI Conference on Artificial Intelligence*, **35**, 5, 2021, 3677-3687.

[5] Galambos G. A new heuristic for the classical bin-packing problem *Univ. Augsburg, Inst. für Mathematik*, 1985.

[6] Galambos G., Woeginger G. J. *Repacking helps in bounded space on-line bind-packing*, **49**, 4, 1993, 329-338.

[7] Ha C. T., Nguyen T. T., Bui L. T., et al. An online packing heuristic for the three-dimensional container loading problem in dynamic environments and the Physical Internet *In European Conference on the Applications of Evolutionary Computation*, 2017, 140-155.

[8] He C., Zhang Y. B., Wu J. W., et al. Research of three-dimensional container-packing problems based on discrete particle swarm optimization algorithm *2009 International Conference on Test and Measurement*, **2**, 2009, 425-428.

[9] Hu H., Zhang X., Yan X., et al. Solving a new 3d bin packing problem with deep reinforcement learning method *arXiv preprint arXiv:1708.05930*, 2017.

[10] Jiang Y., Cao Z., Zhang J. Solving 3D bin packing problem via multimodal deep reinforcement learning *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, 2021, 1548-1550.

[11] Kang K., Moon I., Wang H. A hybrid genetic algorithm with a new packing strategy for the three-dimensional bin packing problem *Applied Mathematics and Computation*, **219**, 3, 2012, 1287-1299.

[12] Karabulut K., İnceoğlu M. M. A hybrid genetic algorithm for packing in 3D with deepest bottom left with fill method *International Conference on Advances in Information Systems*, 2005, 441-450.

[13] Kundu O., Dutta S., Kumar S. Deep-pack: A vision-based 2d online bin packing algorithm with deep reinforcement learning *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2019, 1-7.

[14] Labbé M., Laporte G., Martello S. Upper bounds and algorithms for the maximum cardinality bin packing problem *European Journal of Operational Research*, **149**, 3, 2003, 490-498.

[15] Lodi A., Martello S., Vigo D. Approximation algorithms for the oriented two-dimensional bin packing problem *European Journal of Operational Research*, **112**, 1, 1999, 158-166.

[16] Lodi A., Martello S., Vigo D. Heuristic algorithms for the three-dimensional bin packing problem *European Journal of Operational Research*, **141**, 2, 2002, 410-420.

[17] Martello S., Pisinger D., Vigo D. The three-dimensional bin packing problem *Operations research*, **48**, 2, 2000, 256-267.

[18] Mnih V., Kavukcuoglu K., Silver D., et al. Human-level control through deep reinforcement learning *Nature*, **518**, 7540, 2015, 529-533.

[19] Puche A. V., Lee S. Online 3D Bin Packing Reinforcement Learning Solution with Buffer *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, 8902-8909.

[20] Que Q., Yang F., Zhang D. Solving 3D packing problem using Transformer network and reinforcement learning *Expert Systems with Applications*, **214**, 2023, 119-153.

[21] Scholl A., Klein R., Jürgens C. Bison: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem *Computers Operations Research*, **24**, 7, 1997, 627-645.

[22] Song S., Yang S., Song R., et al. Towards Online 3D Bin Packing: Learning Synergies between Packing and Unpacking via DRL *Conference on Robot Learning*, 2023, 1136-1145.

[23] Wang H., Chen Y. A hybrid genetic algorithm for 3D bin packing problems *2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA)*, 2010, 703-707.

[24] Wei L., Oon W. C., Zhu W., et al. Andrew Lim, A skyline heuristic for the 2D rectangular packing and strip packing problems *European Journal of Operational Research*, **215**, 2, 2011, 337-346.

[25] Zhao H., She Q., Zhu C., et al. Online 3D BinPacking with Constrained Deep Reinforcement Learning *Proceedings of the AAAI Conference on Artificial Intelligence*, **35**, 1, 2021, 741-749.

[26] Zheng F. F., Luo L., Zhang E. NF-based algorithms for online bin packing with buffer and bounded item size *Journal of Combinatorial Optimization*, **30**, 2015, 30: 360-369.

Received 04.05.2023, Accepted 16.11.2023