Authors: Justine Evers & Michaël Schyns

HEC LIÈGE Management School - Liège Université

A Deep Reinforcement Learning approach for the 3D Multiple-Bin Packing Problem

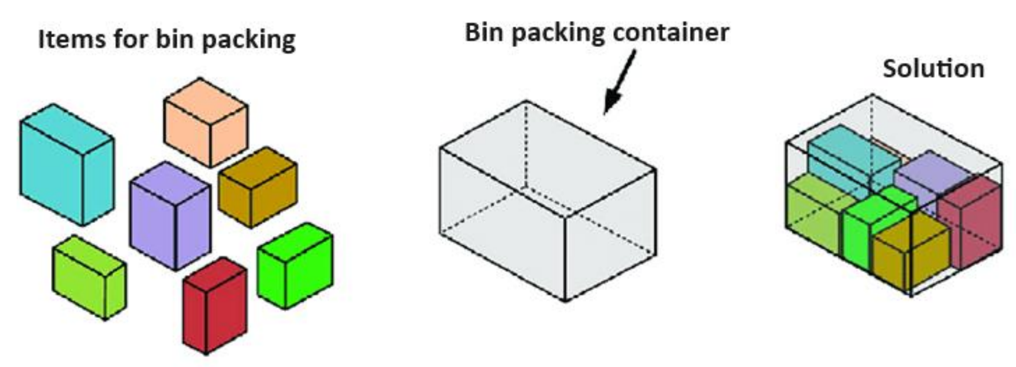
Introduction

This study investigates a variant of the 3D Bin Packing Problem, namely the 3D Multiple-Bin Packing Problem, with stability constraints. We build upon an existing Deep Reinforcement Learning model for 3DBPP (Que et al., 2023)[1] to consider several bins, limited bin volume and stability. We developed an implementation in Python to test our DRL approach on many artificial instances of different sizes with identical bins. Our system can quickly provide highly packed solutions for realistic large-scale instances.

Problem description

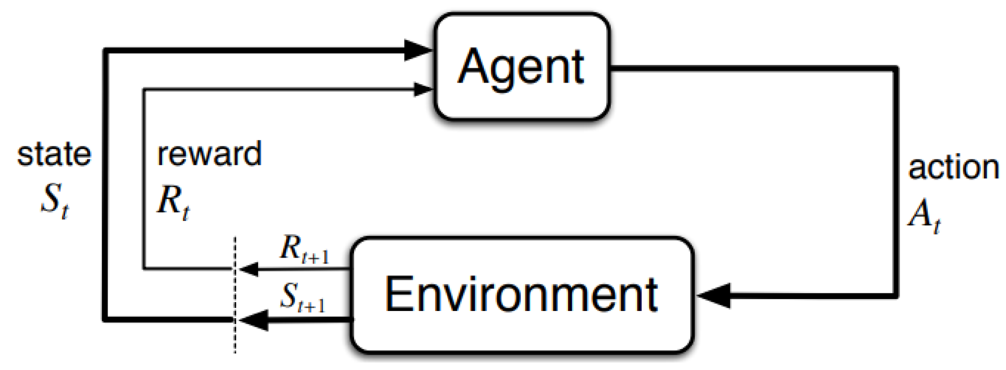
3D Bin Packing Problem:

- Set of weakly heterogeneous containers
- Set of highly heterogeneous cuboid boxes
- Place all boxes in a minimum number of containers
- First constraints: fit in the container, non-overlap of different objects and vertical stability



Reinforcement Learning

The **agent** performs **actions**, which results in a new **state** of the **environment** and triggers a **reward**



The agent learns (automatically) to recognize and take good actions over time.

Issue: RL is not designed for combinatorial problems with huge state spaces → Deep Learning is needed, and the approach becomes Deep Reinforcement Learning (neural net)

State Space

2 inputs for the system of neural networks:

- Box state (features of the boxes still to be placed)
- Container state (current filling level information)

Action Space

3 decisions to take before adding one box in the container:

1. Position selection in a bin
2. Box selection
3. Orientation selection

A complex set of neural networks determines the probability that each action is the best one.

[1] Que, Q., Yang, F., & Zhang, D. (2023). Solving 3D packing problem using Transformer network and reinforcement learning. *Expert Systems with Applications*, *214*, 119153.

Reward function

$$r_t = gap_{t-1} - gap_t$$

where $gap_t = (LWH \times bin_t + LW\widetilde{H_t}) - \sum_i w_i l_i h_i$

Managing constraints with a mask

Some constraints can be imposed via a mask. The mask checks whether an action recommended by the network is feasible. If not, the agent selects the next most probable action.

Constraints:

- Box fits within the container
- Limit on the height of the packing
- 3-point vertical stability

Dealing with large instances with downsampling

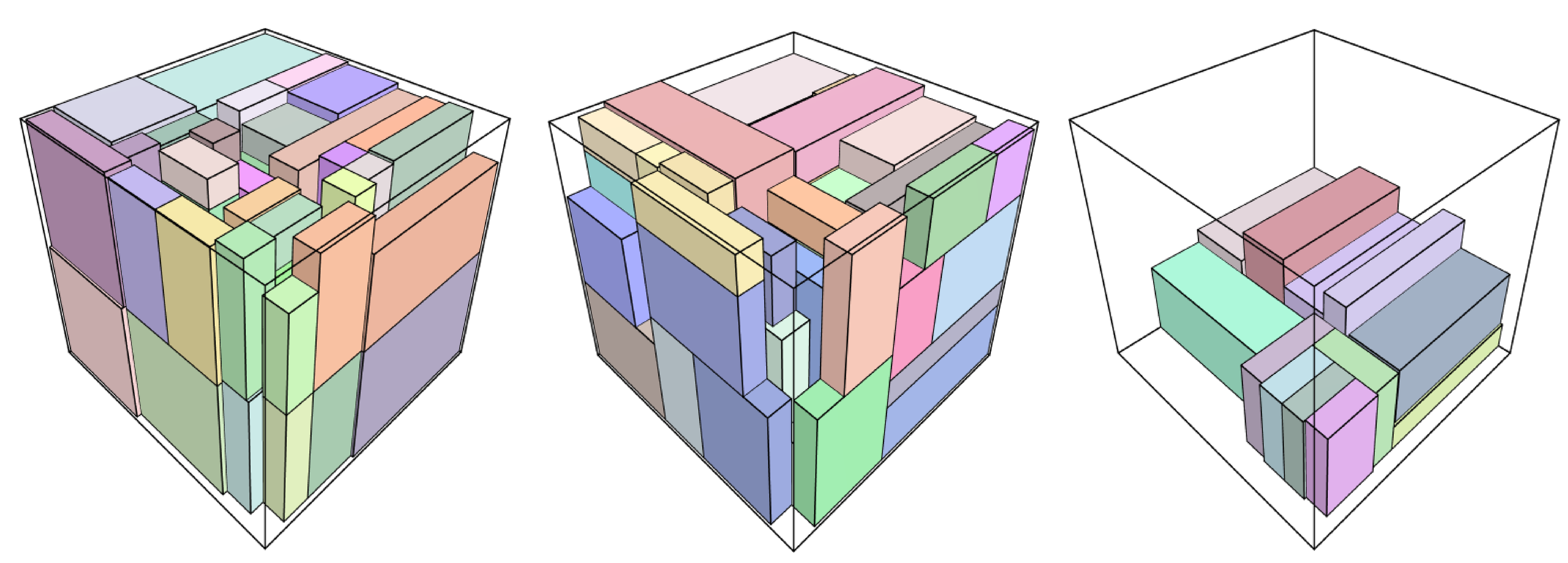
- Reduces the action space size
- Builds non-overlapping patches of arbitrary size

First computational results

Dataset of 1024 randomly generated instances

$120 \times 120 \times 100$ cubes and 96 boxes

- Best filling rate: 82,72%
- Worst filling rate: 60,45%



Conclusion & future directions

We have developed a new DRL model for the 3D-MBPP taking into account 3-point vertical stability, as it has not been addressed in the literature yet.

Future work:

- Further modify the mask to incorporate more constraints into our model
- Developing a full solution from theory to practice with an Augmented Reality tool

Justine Evers | jevers@uliege.be
Michaël Schyns | m.schyns@uliege.be
HEC Liège