Myopic Approaches for a Real World Palletizing Problem*

Florian Kagerer[1], Maximilian Beinhofer[1], Simon Hubmer[2], and Ronny Ramlau[2,3]

***Abstract*— In warehouse logistics, palletizing algorithms usually precalculate a pack pattern that determines the palletizing sequence of load carriers (LCs). Whenever this sequence is broken, troubles occur in robotic palletization. To overcome this issue, this paper presents three approaches that determine positions on a pallet for given LCs myopically, i.e., they avoid precalculating a load carrier sequence in advance. Based on simulations and experiments with a real robot, we show that even though our approaches do not predetermine an LC sequence, they produce reasonable palletizing results in a broad range of trials.**

I. INTRODUCTION

In warehouse logistics, pack pattern software is applied to build stable and densely packed pallets. This software precalculates a pallet with respect to assigned load carriers (LCs), e.g., cartons or plastic boxes. Hence, a sequence of the arrivals of the LCs is derived. Although such software provides excellent results for a wide range of applications, a violation of the predetermined sequence results in an enormous performance loss in the automated system supplying the robot with LCs. In addition, for many problems it might be unnecessary to precalculate the whole packing in advance.

To this end, we developed myopic approaches, i.e., methods that avoid to predetermine an LC sequence, one by one calculating a target position for the next (randomly) arriving LC. These approaches were tested on a real robot and benchmarked in simulations against a commercially used (non-myopic) pack pattern software.

There is a wide body of research in palletizing methods that can be used with robots [1], [2], [8]. Due to the complexity of such bin packing problems, not only heuristics, but also learning approaches were developed [3], [4]. In contrast to our approach, which works myopically, these methods generate predetermined LC sequences.

II. PROBLEM STATEMENT

The considered problem deals with the palletization of LCs, whose footprints – $0.6\,\mathrm{m} \times 0.4\,\mathrm{m}$ (big) and $0.4\,\mathrm{m} \times 0.3\,\mathrm{m}$ (small) – divide the base area of the palletizing target – $1.2\,\mathrm{m} \times 0.8\,\mathrm{m}$. Hence, an LC occupies either a single octant or a double octant (quadrant) of the pallet. Moreover, this paper incorporates the stackability of an object, which depends on its material (cartons, plastic boxes). On top of type 1

*This work is partly supported by the Austrian Research Promotion Agency (FFG) under grant 873923.

[1]F. Kagerer and M. Beinhofer are with TGW Logistics Group GmbH, Marchtrenk, Austria forename.surname@tgw-group.com

[2]S. Hubmer and R. Ramlau are with the Johann Radon Institute Linz, Linz, Austria forename.surname@ricam.oeaw.ac.at

[3]R. Ramlau is with the Institue of Industrial Mathematics, Johannes Kepler University, Linz, Austria ronny.ramlau@jku.at

Fig. 1: Experimental setup: robot with custom end-effector for palletizing cartons. See [7] for a video of the experiments.

(cartons) it is only allowed to place objects with the same footprint. Whenever two cartons with the small footprint are equally high, their stacking behavior is identical to type 2 (plastic boxes). On these, both footprints can be stacked onto.

The robot we consider (see Fig. 1) moves LCs from front right to back left to their target positions. Therefore, in order to reach all target positions, piles in the front and right octants need to be lower than those in the back and left.

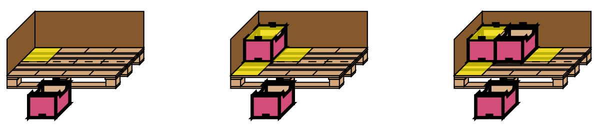
III. APPROACHES

We developed three approaches for finding a position on the pallet for an LC with respect to the mentioned restrictions. In general, these methods obtain (1) the current state of the pallet, and (2) the properties of the LC. This architecture was designed in consideration of reinforcement learning methods, which search for an action based on a given state of the environment. In order to make it unnecessary for an agent to learn the existing constraints, the action space is changeable, i.e., the agent only chooses its action out of the allowed ones. An example is depicted in Fig. 2.

In addition, all methods share a rearrangement property. Whenever no target position for the LC is found, the LC is rejected and considered again after three others.

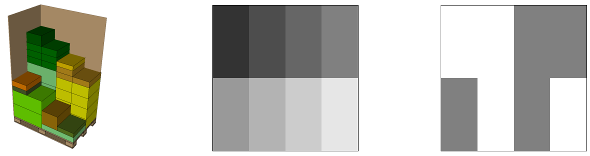
A. *Heuristic*

Firstly, we designed a heuristic that aims at creating pallet states such that a $0.6\,\mathrm{m} \times 0.4\,\mathrm{m}$ LC can be placed on top of as many quadrants of the pallet as possible at any given time.



(a) $A(t_1) = \{o_1\}$. (b) $A(t_2) = \{o_1, o_2, o_5\}$. (c) $A(t_3) = \{o_1, o_3, o_5\}$.

Fig. 2: The changeable action space $A(t)$ during the palletization of three identical load carriers. The yellow rectangles visualize the actions.



(a) Visualization (b) Heights (c) Stackabilities

Fig. 3: Example for the virtual image representation of a state of the pallet. LC colors: green – plastic box, yellow – carton.

Consequently, the heuristic tries to keep the heights of the piles in two neighboring octants at the same level.

This method uses the exact state of a pallet and selects one of the placements that satisfy the stackability and height requirements. In detail, for LCs with a small footprint, the algorithm chooses either the smallest pile or, if existing, the smaller pile in a quadrant with two different heights. On the other hand, for LCs with a big footprint, the algorithm chooses the highest possible pile. The only exception: if for big type 1 LCs there exist possible placements on top of other type 1 LCs, then the smallest of these piles is chosen.

B. *Q-learning*

Secondly, we developed an approach that is based on Q-learning. For details on Q-learning, see [6]. For this approach, we defined the state of the pallet as a tuple that contains the discretized height of the pile in each octant. The rating of every state-action pair (deciding which action the agent chooses at a given state) is stored in the Q-learning table. Since a finer height discretization increases the size of the Q-learning table, this is the restrictive factor in this approach. During training, the ratings were updated at the end of each episode, i.e., after palletizing of the whole pallet was finished. The update of the rating depends on the filling degree of the pallet, i.e., the palletized volume divided by the maximum admissible one. Similar to neural networks, not only the last, but also all visited state-action pairs are updated, but we defined that the update rate decreases by 10% with each step back.

C. *Deep Q-network*

Thirdly, we created an approach that uses deep Q-networks (DQNs) on image data to solve the palletizing problem. DQNs receive an image and create different abstract representations to obtain a single output [5]. Consequently, we represented the state of the pallet as an image. See Fig. 3 for an example. The height of each pile on the pallet, which is stored in our software, is converted to the value of a pixel in a virtual gray-scale image. Similarly, the stackabilities of the uppermost load carriers defined a second image which extends the information about the pallet state. This conversion makes it possible to apply image based approaches to solve the palletizing problem.

The input layer of our approach can either be a single channel image, i.e., only the heights, or a two channel image, i.e., heights and stackabilities on the pallet. Before the agent takes an action depending on the approximated ratings, all prohibited actions are removed. As before, the filling degree of the pallet at the end of each episode defines the reward.

IV. EXPERIMENTS

We evaluated our three approaches both on simulation and in experiments with a real robot. Up until now, the real robot setup shown in Fig. 1 does not include a sensor for detecting differences between planned and real LC heights. The robot moves only based on planned positions. Qualitative tests with the real robot have shown that two factors strongly influence the quality of a built pallet: (1) the rate of small load carriers, and (2) the rate of load carriers with type 1 stackability.

A. *Simulation*

For the simulations, we randomly created sequences of LCs of mixed footprints and stackability types. We divided the created dataset into training data for the machine learning approaches and validation data. Rearrangement of LCs was only allowed three times per episode. Whenever the approach was unable to find a position for the next LC, palletization was aborted and the episode was finished.

B. *Results*

At the end of each episode, the filling degree η was collected and categorized. For evaluation, we created the categories *Successful* ($\eta \geq 85\%$), *Tolerable* ($\eta \in [70\%, 85\%)$), and *Failed* ($\eta < 70\%$). Fig. 4 compares the performances of the myopic approaches and a non-myopic benchmark.

The myopic approaches achieved a *Successful* or *Tolerable* result in a reasonable amount of sequences, even though not a single restriction was demanded on the sequence of load carriers. Up until now, our implementation of the three myopic approaches were not yet able to avoid *Failed* orders.

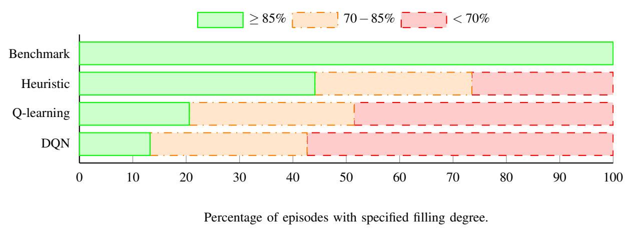


Fig. 4: Comparison of the results from all approaches.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we presented three myopic approaches for robotic palletizing, which avoid to predetermine a sequence of load carriers: one heuristic and two reinforcement learning approaches. Compared to conventional, non-myopic approaches, ours provide more flexibility in the delivery of load carriers to the palletizing robot. The filling degrees of our produced pallets were in an acceptable range for a high amount of scenarios, even though our approaches did not place a single restriction on the sequence of the load carriers.

In future work, we plan to optimize our implementations to further increase the achievable filling degrees. In our robot setup, we plan to integrate a sensor detecting differences between planned and real heights of the positioned LCs.

References

[1] M. Baldi, G. Perboli, and R. Tadei, "The three-dimensional knapsack problem with balancing constraints," *Applied Mathematics and Computation*, vol. 218, 2012.

[2] G. Demisse, R. Mihalyi, B. Okal, D. Poudel, J. Schauer, and A. Nüchter, "Mixed palletizing and task completion for virtual warehouses," *Virtual Manufacturing and Automation Competition (VMAC) Workshop at the Int. Conference of Robotics and Automation (ICRA)*, vol. 16, 2012.

[3] L. Duan, H. Hu, Y. Qian, Y. Gong, X. Zhang, Y. Xu, and J. Wei, "A multi-task selected learning approach for solving 3d flexible bin packing problem," *CoRR*, vol. abs/1804.06896, 2018.

[4] H. Hu, X. Zhang, X. Yan, L. Wang, and Y. Xu, "Solving a new 3d bin packing problem with deep reinforcement learning method," *CoRR*, vol. abs/1708.05930, 2017.

[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, 2015.

[6] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA: The MIT Press, 2018.

[7] TGW Logistics Group. Austrian Robotics Workshop 2021 - Myopic Approaches for a Real World Palletizing Problem. [Online]. Available: https://tinyurl.com/arw21-tgw

[8] F. Wang and K. Hauser, "Stable bin packing of non-convex 3d objects with a robot manipulator," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019.