Solving 3D Bin Packing Problem via Multimodal Deep Reinforcement Learning

Extended Abstract

Yuan Jiang
Nanyang Technological University
Singapore
jiang.yuan@ntu.edu.sg

Zhiguang Cao
National University of Singapore
Singapore
zhiguangcao@outlook.com

Jie Zhang
Nanyang Technological University
Singapore
zhangj@ntu.edu.sg

ABSTRACT

Recently, there is growing attention on applying deep reinforcement learning (DRL) to solve the 3D bin packing problem (3D BPP), given its favorable generalization and independence of ground-truth label. However, due to the relatively less informative yet computationally heavy encoder, and considerably large action space inherent to the 3D BPP, existing methods are only able to handle up to 50 boxes. In this paper, we propose to alleviate this issue via an end-to-end multimodal DRL agent, which sequentially addresses three sub-tasks of sequence, orientation and position, respectively. The resulting architecture enables the agent to solve large-scale instances of 100 boxes or more. Experiments show that the agent could learn highly efficient policies that deliver superior performance against all the baselines on instances of various scales.

KEYWORDS

Bin Packing Problem; Combinatorial Optimization Problem; Multimodal Learning; Deep Reinforcement Learning

ACM Reference Format:
Yuan Jiang, Zhiguang Cao, and Jie Zhang. 2021. Solving 3D Bin Packing Problem via Multimodal Deep Reinforcement Learning: Extended Abstract. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 3 pages.

1 INTRODUCTION

3D Bin packing problem (BPP) is a well-known class of combinatorial optimization problem in operations research and computer science [10, 17]. In a classic 3D BPP, a number of cuboid boxes with certain inherent constraints (e.g., the box cannot overlap with each other, or violate the bin dimensions) are packed into a bin, to maximizing the utilization rate. Due to the strong NP-hardness, it is always impractical to exactly solve the problem [14], while most of the attempts emphasize on heuristics to yield approximate solutions. However, conventional heuristics usually need hand-crafted decision rules to guide the solving process, which require substantial domain knowledge and engineering efforts to design and ignore the underlying pattern that those instances may share, thus likely lead to limited performance.

Recently, there is a growing trend towards applying deep reinforcement learning (DRL) to solve combinatorial optimization problems [1, 6, 9, 12, 16, 18–21]. Inspired by those seminal works, several DRL methods are also adapted to solve the 3D BPP [5, 7, 8], which decompose the problem into three sub-tasks, i.e., deciding, 1) the sequence to place boxes; 2) the orientation of the selected box; and 3) the position to place the selected box, respectively. Nevertheless, due to the harder nature of the 3D BPP compared with other general combinatorial optimization problems, the adapted encoders appear to be less informative while computationally heavy. The resulting decoders are unable to efficiently cope with the large action space, especially for the position sub-task. Consequently, those attempts are only able to handle up to 50 boxes, which significantly hinder the wide applications of DRL for solving 3D BPP.

To address these issues, we propose an end-to-end DRL agent to solve the 3D BPP. In specific, the agent exploits a multimodal encoder to produce more informative embedding while maintaining light computation for learning the packing policy, which enables solving large instances of 100 boxes or more.

2 METHOD

We solve same BPP problem as [17], in which 1) any two boxes do not overlap with each other; 2) all boxes do not violate the bin dimensions; 3) only one orientation is allowed for a box. The objective is to minimize the maximum stacked height (equal to maximizing the utilization rate of the bin).

Our agent adopts an encoder-decoder diagram to learn the packing policy and sequentially perform the sub-tasks of *sequence*, *orientation* and *position*. The multimodal encoder maps the states into feature embeddings, and a decoder is responsible for incrementally constructing solutions for the three sub-tasks. Particularly, in the multimodal encoder, a sparse attention sub-encoder is exploited to embed the box state (i.e., the basic information of the box including the indication of whether packed, orientation and position) while maintaining relatively light computation; and a CNN (convolutional neural network) sub-encoder is used to embed the view state for more informative auxiliary representation, which captures the top-down view of the current packing layout. In the decoder, an action representation learning is leveraged to deal with the large action space that mainly caused by the position sub-task.

The *input state* at each step includes *box state* and *view state* , respectively. The *box state* retrieves the status (rotations and coordinates) of all boxes at each step. The *view state* is described as the top-down view of the bin. Accordingly, the value of each grid cell in view state refers to the total height of stacked boxes in that cell. Intuitively, the objective value directly relates to the stacked height of each grid cell, which might not be well captured solely by the box state. Therefore, the view state, which represents the

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.

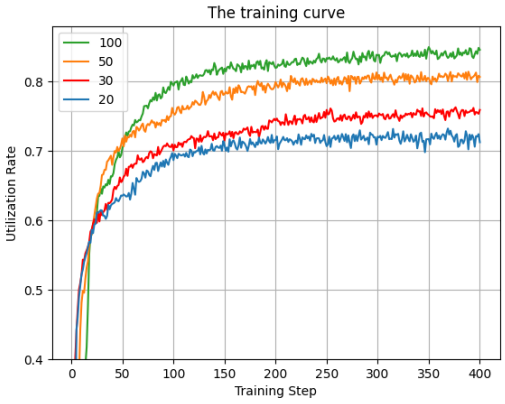

Figure 1: Learning curves of our model for different numbers of boxes in 100×100 bin.

top-down view of the bin at each step, could be exploited to offer more informative spatial features to facilitate the packing.

For the sparse attention sub-encoder, we adopt the Transformer architecture [15] as the backbone and discard the positional encoding. To alleviate the computation cost for instances with large number of boxes, we exploit a dilated sparse attention mechanism [3]. For the convolutional sub-encoder, we adopt a CNN architecture to encode the image-like view state. At each step, the decoder (consisting of 3 transformer sub-decoders) takes the output of the sparse attention sub-encoder and the output of the convolutional sub-encoder as input, and sequentially outputs the box index i, orientation (l_i^r, w_i^r, h_i^r) and position (x_i, y_i, z_i). However, even though we reduce the action space to 2D by straightly dropping the box from top, the number of possible actions for the position sub-task is quadratic to the size of bin, which is still considerably large (e.g., 40000 possible actions for a 200×200 bin). Previous DRL methods [5, 8] fail to take this into account, which limits them to instances of small-scale. To address this issue, we introduce the action representation learning [2] into the position sub-decoder to improve generalization over large finite action space, which allows the agent to infer the consequence of an action from the similar historical ones. To be precisely, we learn the embedding of position actions and use it to calculate a position for placing the box.

To better balance the variance and bias of the rewards, we adopt the A2C [11] with Generalized Advantage Estimation (GAE) [13] to train the agent. Meanwhile, we integrate supervised learning for action representation in the position sub-task.

3 EXPERIMENTS AND RESULTS

In the experiments, we follow a similar setting in [8] to randomly generate the data. With respect to the metric, utilization rate (UR) is adopted to measure the packing performance. Before comparing with others, we first present the learning curves of our method on 20, 30, 50 and 100 boxes with a 100×100 bin in Figure 1, respectively. We observe that, our method converges smoothly for all different scales of instances, demonstrating the favorable potential to handle further larger number of boxes (especially in light of that the following two DRL baselines can only handle up to 50 boxes). We then compare it with four different methods, including: 1) Genetic Algorithm (GA) [17]; 2) Extreme Point (EP) [4]; 3) Multi-Task Selected Learning (MTSL) [5], which is adapted to the 3D BPP in this paper; and 4) Conditional Query Learning (CQL) [8], the state-of-the-art DRL method for solving 3D BPP. All the DRL based methods including ours, sample 128 solutions according to the learnt policies and retrieve the best one as the final output as [5]. The superior performance reflected in Table 1 well justified the overall efficacy of the designed sparse attention sub-encoder, CNN sub-encoder and action representation learning in our method for boosting the solution quality and computation efficiency. We also notice that, as the number of boxes becomes larger, the average utilization rate of our method also increases. Larger number of boxes often means more information in the input sequence, which could be well exploited by the attention mechanism and DRL algorithm to learn the relationship among boxes and facilitate the long-term planning.

Table 1: Utilization rate (%). Genetic Algorithm (GA) [17]; Extreme Point (EP) [4]; Multi-Task Selected Learning (MTSL) [5]; Conditional Query Learning (CQL) [8]

Box Number	*GA*	*EP*	*MTSL*	*CQL*	*OUR*
20	68.3%	62.7%	62.4%	67.0%	**71.8%**
30	66.2%	63.8%	60.1%	69.3%	**75.5%**
50	65.9%	66.3%	55.3%	73.6%	**81.3%**
100	62.4%	67.5%	-	-	**84.4%**

We also conduct ablation studies by removing the respective components separately. The results (not displayed here due to space limitation) reveal that without the CNN sub-encoder or the action representation learning, the utilization rates for our method drop obviously, implying that both components could benefit the DRL algorithm for engendering high-quality solutions. Besides, the absence of CNN sub-encoder has more significant impacts on the performance compared with that of action representation. This finding justified that the view state empowered by the CNN sub-encoder could produce more informative auxiliary embedding for learning the packing policy. Meanwhile, with the absence of both CNN sub-encoder and action representation learning, the utilization rates further decreased.

4 CONCLUSIONS

In this paper, we present a multimodal DRL agent for solving 3D BPP. Specifically, we exploit a multimodal encoder with a sparse attention sub-encoder and a CNN sub-encoder to exploit multimodal information. Meanwhile, the action representation learning is adopted to cope with large action space. The resulting policy enables the agent to solve large instances of 100 boxes or more, while the state-of-the-art DRL method is only able to handle up to 50 boxes. Moreover, our method also delivers superior performance in terms of utilization rate against all the baselines.

5 ACKNOWLEDGEMENT

Zhiguang Cao is the corresponding author.

REFERENCES

[1] Irwan Bello, Hieu Pham, Quoc V Le, Mohammad Norouzi, and Samy Bengio. 2017. Neural combinatorial optimization with reinforcement learning. In *International Conference on Learning Representations*.

[2] Yash Chandak, Georgios Theocharous, James Kostas, Scott Jordan, and Philip S Thomas. 2019. Learning action representations for reinforcement learning. *arXiv preprint arXiv:1902.00183* (2019).

[3] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509* (2019).

[4] Teodor Gabriel Crainic, Guido Perboli, and Roberto Tadei. 2008. Extreme point-based heuristics for three-dimensional bin packing. *Informs Journal on computing* 20, 3 (2008), 368–384.

[5] Lu Duan, Haoyuan Hu, Yu Qian, Yu Gong, Xiaodong Zhang, Jiangwen Wei, and Yinghui Xu. 2019. A Multi-task Selected Learning Approach for Solving 3D Flexible Bin Packing Problem. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. 1386–1394.

[6] Wouter Kool, Herke van Hoof, and Max Welling. 2019. Attention, Learn to Solve Routing Problems!. In *International Conference on Learning Representations*.

[7] Alexandre Laterre, Yunguan Fu, Mohamed Khalil Jabri, Alain-Sam Cohen, David Kas, Karl Hajjar, Torbjorn S Dahl, Amine Kerkeni, and Karim Beguir. 2018. Ranked reward: Enabling self-play reinforcement learning for combinatorial optimization. *arXiv preprint arXiv:1807.01672* (2018).

[8] Dongda Li, Changwei Ren, Zhaoquan Gu, Yuexuan Wang, and Francis Lau. 2019. Solving Packing Problems by Conditional Query Learning. (2019). https://openreview.net/forum?id=BkgTwRNtPB

[9] Jingwen Li, Liang Xin, Zhiguang Cao, Andrew Lim, Wen Song, and Jie Zhang. 2021. Heterogeneous Attentions for Solving Pickup and Delivery Problem via Deep Reinforcement Learning. *IEEE Transactions on Intelligent Transportation Systems* (2021).

[10] Silvano Martello, David Pisinger, and Daniele Vigo. 2000. The three-dimensional bin packing problem. *Operations research* 48, 2 (2000), 256–267.

[11] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*. 1928–1937.

[12] Mohammadreza Nazari, Afshin Oroojlooy, Lawrence Snyder, and Martin Takác. 2018. Reinforcement learning for solving the vehicle routing problem. In *Advances in Neural Information Processing Systems*. 9839–9849.

[13] John Schulman, P. Moritz, S. Levine, Michael I. Jordan, and P. Abbeel. 2016. High-Dimensional Continuous Control Using Generalized Advantage Estimation. *CoRR* abs/1506.02438 (2016).

[14] Everton Fernandes Silva, Tony Wauters, et al. 2019. Exact methods for three-dimensional cutting and packing: A comparative study concerning single container problems. *Computers & Operations Research* 109 (2019), 12–27.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[16] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in neural information processing systems*. 2692–2700.

[17] Yong Wu, Wenkai Li, Mark Goh, and Robert de Souza. 2010. Three-dimensional bin packing problem with variable bin height. *European journal of operational research* 202, 2 (2010), 347–355.

[18] Yaoxin Wu, Wen Song, Zhiguang Cao, Jie Zhang, and Andrew Lim. 2020. Learning improvement heuristics for solving routing problems. *arXiv preprint arXiv:1912.05784* (2020).

[19] Liang Xin, Wen Song, Zhiguang Cao, and Jie Zhang. 2020. Step-wise Deep Learning Models for Solving Routing Problems. *IEEE Transactions on Industrial Informatics* (2020).

[20] Liang Xin, Wen Song, Zhiguang Cao, and Jie Zhang. 2021. Multi-Decoder Attention Model with Embedding Glimpse for Solving Vehicle Routing Problems. In *Proceedings of 35th AAAI Conference on Artificial Intelligence*.

[21] Cong Zhang, Wen Song, Zhiguang Cao, Jie Zhang, Puay Siew Tan, and Chi Xu. 2020. Learning to Dispatch for Job Shop Scheduling via Deep Reinforcement Learning. In *Advances in Neural Information Processing Systems*.