A Multi-task Selected Learning Approach for Solving New Type 3D Bin Packing Problem

Haoyuan Hu1*, Lu Duan2*, Yu Gong3, Kenny Q. Zhu4†
Xiaodong Zhang5, Jiangwen Wei6, Yinghui Xu7

1 S 2 5 6 7 Artificial Intelligence Department, Zhejiang Cainiao Supply Chain Management Co., Ltd
2 Search Algorithm Team, Alibaba Group
3 Shanghai Jiao Tong University

*Emails: {haoyuan.huhy,duanlu.dl}@cainiao.com, gongyu.gy@alibaba-inc.com, kzhu@cs.sjtu.edu.cn
†Emails: yuhui.zxd@cainiao.com, jiangwen.wjw@alibaba-inc.com, renji.xyh@taobao.com

ABSTRACT

This paper studies a novel real-world application: a new type of 3D bin packing problem (BPP), in which a number of cuboid-shaped items must be put into a bin one by one orthogonally. The objective is to find a way to place these items such that the surface area of the bin is minimized. This problem is based on the fact that there is no fixed-sized bin in many real business scenarios and the cost of a bin is proportional to its surface area. Based on previous research on 3D BPP, the surface area is determined by the sequence, spatial locations and orientations of items. It is a new NP-hard combinatorial optimization problem on unfixed-sized bin packing, for which we propose a multi-task framework based on Selected Learning, generating the sequence and orientations of items packed into the bin simultaneously. During training steps, Selected Learning chooses one of loss functions derived from deep reinforcement learning and supervised learning corresponding to the training procedure. We produce large scale 3D Bin Packing order data set from Cainiao Supply Chain for studying bin packing problems. Comprehensive experiments on this data set are conducted to validate the effectiveness of our research in reducing packing materials by comparisons with other baseline methods. Numerical results also demonstrate that the method proposed significantly outperforms the well-designed heuristic baselines by a substantial gain of 7.52%.

ACM Reference Format:
Haoyuan Hu, Lu Duan, Yu Gong, Kenny Q. Zhu, Xiaodong Zhang, Jiangwen Wei, Yinghui Xu. 2018. A Multi-task Selected Learning Approach for Solving New Type 3D Bin Packing Problem. In Proceedings of The 2018 Conference on Information and Knowledge Management (CIKM ’18). ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnn.nnnnnn

1 INTRODUCTION

Bin packing problem (BPP) is a classical and important optimization problem in logistic and production systems. There are many variants of BPP and the most meaningful and challenging one is 3D BPP, in which a number of cuboid-shaped items with different sizes should be packed into bins orthogonally. For vanilla 3D BPP, the size and cost of bins are fixed and the objective is to minimize the number of bins used, i.e., to minimize the total cost. BPP is a typical and interesting combinatorial optimization problem and is strongly NP-hard [9], so it is a very popular research direction in optimization area. In addition, BPPs have numerous relevant industrial applications. An effective bin packing algorithm means the reduction of computation time, total packing cost and increase in utilization of resources.

The cost of packing materials is mainly determined by their surface area and this occupies the most part of packing cost. In many real business scenarios such as e-commerce, bins with variable sizes, e.g., PE bag which is made of flexible and soft packing materials, are used to pack items. In this case, our research is engaged in a new type of 3D BPP, of which the objective is to pack all items into a bin with least surface area.

Due to the difficulty of obtaining optimal solutions of BPPs, many researchers have proposed various approximation or heuristic algorithms. To achieve good results and guarantee performance, heuristic algorithms have to be designed specifically for different types of problems or situations, therefore still rely on human created heuristic rules. In recent years, artificial intelligence, especially deep reinforcement learning (DRL), has received intense research and achieved amazing results in many fields [24, 29]. In addition, DRL method has shown huge potential to solve combinatorial optimization problems [2, 32]. Previous DRL-based method that attacks this new type of 3D BPP [18] relies heavily on the output of the heuristic algorithm because the network just produces the sequence of packing items.

As the surface area is determined by three kind of factors: the sequence, spatial locations and orientations, each kind of decision can be treated as a task. Regardless of spatial locations, the sequence of putting the items into the bin will influence the orientations of each item and vice versa, so the two tasks are correlated. Meanwhile, with n items, the choice of orientations is 6^n and the choice of sequence is n!, so the two tasks are unbalanced. Inspired by multi-task learning, we overcome the limitations of simply outputting the placement sequence of items by adopting a new type of training mode named Selected Learning (SL) to solve the new type of 3D BPP. In Selected Learning, we prefer to select one kind of the loss function at each batch than to train the loss of all tasks at once.

Combination optimization problems are to minimize or maximize the objective under some constraints. The objective can be set as the reward of RL framework. In this new type of 3D BPP, the smaller the objective, the better. Through keeping track of the best...
solution sampled during the search, we find that combining Selected Learning with sampling at inference time works best in practice.

In this paper, we present a neural model that achieves state-of-the-art results on the proposed Large-scale 3D Bin Packing Problem Dataset (L3DBPD) and quantitative experiments are designed and conducted to demonstrate effectiveness of this method. The contributions of this paper are summarized below.

- This is the first attempt to define and solve the real-world problem of New Type 3D Bin Packing (Section 2). We collect and will open source a large scale 3D Bin Packing order dataset (Section 3).
- We use an intra-attention mechanism to solve the combinatorial optimization problem, which considers items that have already been generated by the decoder.
- We propose a multi-task framework based on Selected Learning, generating packing sequence and orientations at the same time.
- By sampling at inference time, we achieve 6.21%, 7.80%, 8.55% improvement than the new 3D bin packing heuristic algorithm (NBPH) (see Appendix B) for BIN8, BIN10 and BIN12.

## 2 PROBLEM

In a typical 3D BPP, a set of items must be packed into fixed-sized bins in the way that minimizes the number of bins used. Unlike typical BPP with fixed-sized bins, we focus on the problem of designing the bin with least surface area that can pack all the items. In real business scenarios, such as cross-board e-commerce, flexible and soft materials are used to fulfill the requirement of packing all items with no fixed-sized bin. After putting the items, the soft and flexible material such as PE bag is similar to rectangular-shaped bin. As the result, the cost of this material is directly proportional to the surface area of the rectangular-shaped bin. In this case, minimizing the surface area for the bin would bring huge economic benefits.

The exact formulation of our problem is shown below. Given a set of cuboid-shaped items and each item i is characterized by length (l_i), width (w_i) and height (h_i). Our target is to find the least-surface-area bin that can pack all items. We define (x_i, y_i, z_i) as the left-bottom-back (LBB) coordinate of item i and define (0, 0, 0) as the left-bottom-back coordinate of the bin. The details of decision variables are shown in Table 1.

As shown in the table, we set s_{ij} = 1 if item i is in the left side of item j, u_{ij} = 1 if item i is under item j and b_{ij} = 1 if item i is in the back of item j. We also set δ_{11} = 1 if the orientation of item i is front-up, δ_{12} = 1 if the orientation of item i is front-down, δ_{13} = 1 if the orientation of item i is side-up, δ_{14} = 1 if the orientation of item i is side-down, δ_{15} = 1 if orientations of item i is bottom-up and δ_{16} = 1 if orientation of item i is bottom-down. Consequently, our aim is to find a least surface area bin of which the size is (L, W, H), where L, W and H is the length, width and height of the bin, respectively.

Based on the descriptions of problem and notations, the mathematical formulation for the new type of 3D BPP is followed by

### Table 1: Decision Variables of New Type of 3D BPP.

| Variable | Type | Meaning |
|----------|------|---------|
| L        | Continuous | the length of the bin |
| W        | Continuous | the width of the bin |
| H        | Continuous | the height of the bin |
| x_i      | Continuous | LBB coordinate of item i in x axis |
| y_i      | Continuous | LBB coordinate of item i in y axis |
| z_i      | Continuous | LBB coordinate of item i in z axis |
| s_{ij}   | Binary   | item i is in the left side of item j or not |
| u_{ij}   | Binary   | item i is under item j or not |
| b_{ij}   | Binary   | item i is in the back of item j or not |
| δ_{11}   | Binary   | orientation of item i is front-up or not |
| δ_{12}   | Binary   | orientation of item i is front-down or not |
| δ_{13}   | Binary   | orientation of item i is side-up or not |
| δ_{14}   | Binary   | orientation of item i is side-down or not |
| δ_{15}   | Binary   | orientation of item i is bottom-up or not |
| δ_{16}   | Binary   | orientation of item i is bottom-down or not |

Constraints (9)(10)(11) denote the length, width, height of item i after orientating it. Constraints (1)(2)(3)(4)(5) are used to guarantee there is no overlap between two packed items while constraints (6)(7)(8) are used to guarantee the item will not be put outside the bin. Figure 1 explains the non-overlapping constraints in the problem definition.

We have tried to solve the problem by optimization solvers, such as IBM Cplex Optimizer1, but it is very difficult to solve in reasonable time limit and we will prove this problem is NP-hard in the appendix A.

## 3 DATA COLLECTION

To the best of our knowledge, available large scale 3D Bin Packing order dataset is very rare, especially with real-sized items. For that purpose, we create a dataset on 3D bin packing problem of real-world E-commerce platform. This L3DBPD consists of two parts: the customer order data is collected from Taobao2 E-commerce platform and item size data (with length, width and height) is collected from

1https://www.ibm.com/analytics/data-science/prescriptive-analytics/cplex-optimizer
2https://www.taobao.com/
We use one line to demonstrate one customer order. Therefore, we can TSP, where given a graph, the optimal solution is the permutation of nodes with minimal total edge weights (tour length). As mentioned above, the sequence generation task is part of our multi-task problem, and can be treated as searching the space of permutations to find the optimal one, which is similar to tackle TSP. For the new type 3D BPP, the original problem will rapidly degenerate into vanilla sequence to sequence problem when one can calculate the orientation and free space by a heuristic algorithm, so does in [18]. Nevertheless, [18] suffers from the limitedness of these heuristic-calculated values. In order to mitigate this issue, we would rather using the orientations generated by the network than by heuristic algorithm, as we do in our framework. Actually, the combination of results of our new type 3D BPP is $6^n n!$ in case of not taking free space into consideration. To search that huge space in limited time is impossible, so that we’d better employ neural network to make decisions straightforward. Consequently, for the sake of providing the sequence and orientation simultaneously, we adopt a multi-task encoder-decoder model based on Ptr as the main building block of the new type 3D BPP solver.

Our model reads the sequence of input tokens (items) with Long Short-Term Memory (LSTM) [16] encoder and meanwhile decodes the sequence of output tokens (item and orientation). Intuitively, the inputs of our encoder model can be denoted as $x = \{(l_i, w_i, h_i)\}_{i=1}^n$, where $l_i, w_i$ and $h_i$ represents the length, width and height of the $i$th item respectively, $n$ is the total number of items. The outputs of our decoder network is the orientation $o$ and the permutation of the original input $x$ as sequence $s$, in which the items are packed into the bin. Thus the input of decoder cell for time-step $i$ contains two parts denoted as $y_i = (s_i, o_i)$ instead of just copying one of the original input $x$ as the current sequence input $s_i$.

4 APPROACH

In this section, we describe our multi-task selected learning for solving the new type 3D BPP. The overall architecture of our multi-task networks is shown in Figure 2. Basically, the procedure of solving a new type 3D BPP can be divided into three tasks: sequence generation, orientation generation and free space strategy (spatial locations). In order to leverage useful information contained in these multiple related tasks to help improve the generalization performance of all the tasks, we adopt a multi-task framework. However, among these tasks, the free space set is dynamic and hard to incorporate with the existing neural network. In this work, we settle the free space strategy and apply least surface area heuristic for picking free space (see Algorithm 5 in Appendix B) by default. Therefore, we can concentrate on the sequence and orientation generation problems.

We are inspired by previous work [2] that address the sequence to sequence problem like machine translation. In [2], they employ the pointer network (Ptr) [1] architecture as the policy model to parameterize to the probability of a tour in traveling salesman problem (TSP). However, this approach can only address the problem like TSP, where given a graph, the optimal solution is the permutation of nodes with minimal total edge weights (tour length). As mentioned above, the sequence generation task is part of our multi-task problem and can be treated as searching the space of permutations to find the optimal one, which is similar to tackle TSP. For the new type 3D BPP, the data will be published soon after accepted.
items into consideration means that the network has a priori knowledge to try to avoid repetition to some extend. To achieve this, we introduce an intra-attention mechanism [26], which is first proposed to solve the current combinatorial optimization problem.

For notation purposes, let us define encoder and decoder hidden states as $h^e_t$ and $h^d_t$ respectively. Here $h^e_t$ and $h^d_t$ are computing from the embedding vector of $x_i$ and $y_t$ respectively. We define $attn^d_{ij}$ as the intra-attention score of the previous hidden output state $h^d_j$ at decoding step $t$:

$$attn^d_{ij} = \text{softmax}(v^T \tanh(W_1 h^d_j + W_2 h^d_i)),$$

where $W_1$, $W_2$ and $v$ are trainable parameters for intra-attention. Especially for the first decoder cell, we set $f^{intra}_t$ to a vector of zeros since the already generated sequence is empty.

The previous intra-attention mechanism is either used to modify the underlying LSTM function [7] or to be applied to abstract summarization problems [26]. To our knowledge, this work is the first to introduce an intra-attention mechanism [26], which is first proposed to solve the current combinational optimization problem.

Especially for the first decoder cell, we set $f^{intra}_t$ to be the “attention” mask over the inputs.

$$f^{intra}_t = \sum_{j=1}^{t-1} attn^d_{ij} h^d_j,$$

where $W_1$, $W_2$ and $v$ are trainable parameters for intra-attention feature $f^{intra}_t$ of the current decoded item $s_t$ is calculated by using the computed intra-attention weights, for $i > 1$:

$$f^{intra}_t = \sum_{j=1}^{t-1} attn^d_{ij} h^d_j,$$

where $W_1$, $W_2$ and $v$ are trainable parameters for intra-attention.

4.2 Orientation Task

When placing a cuboid-shaped item, there are 6 orientations, for which we have 3 choices in the first dimension and 2 in the second. For new type 3D BPP, these 6 kinds of orientations are defined as: front-up, front-down, side-up, side-down, bottom-up and bottom-down. For orientation generation, the decoder computes the orientation based on a context vector $c$, which describes the whole problem and is based on the partial solution constructed so far. That is the context of the decoder at time step $t$ comes from the encoder outputs and the output up to time step $t$. For each time step $t$, we calculate the context vector:

$$c_t = [h^e; h^d_t, f^{intra}_t],$$

Here, $[...; ...]$ denotes vector concatenation, $h^e$ donates the representation of input sequence and is calculated by an attention-polling function which is defined as follows:

$$h^e = \sum_{j=1}^{n} attn^d_{ij} h^e_j.$$

$$attn^d_{ij} = \text{softmax}(v^T \tanh(W_6 h^e_j + W_7 h^d_i + f^{intra}_t)),$$

where $W_6$, $W_7$ and $v$ are trainable parameters for attention-polling function.

Thus, the probability distribution of orientations for the current decoding step $t$ is generated by as follows:

$$p(o_t) = \text{softmax}(W_{ori} c_t + b_{ori}),$$

where $W_{ori}$ and $b_{ori}$ are trainable parameters for orientations output task.

We define $o^* = \{o_1^*, o_2^*, ..., o_n^*\}$ as the ground-truth orientation sequence for a given input $x$ and generated packing sequence $s$. As a matter of fact, $o^*$ is calculated by the Algorithm 3 in Appendix B using the placement sequence generated by our sequence task model mentioned above. At each step, least surface area heuristic exhaustively search all possibilities of given free space set and orientation to find the one with least surface area for the current item.

As we apply supervised training in this part, the objective is to maximize the likelihood of all orientation labels $o^*$, given the generated placement sequence $s$ and model parameters $\theta_{ori}$ for orientation task:

$$\log p(o^*|s, \theta_{ori}) = \sum_{i=1}^{n} \log p(o_i|s_1, ..., s_{i-1}, o_{ori}, x).$$

Besides, we refer the probability distribution output of orientation task as $p_{\theta_{ori}}(o|\epsilon)$ in the following discussions.

4.3 Training and Testing

In this subsection, we will illustrate the training and testing procedure which is specially designed for our new type 3D BPP.

4.3.1 Training. To train a multi-task model, we use a hybrid loss function to combine supervised learning and reinforcement learning. For sequence task, we use REINFORCE algorithm [33] as before in [18]. However, for orientation task, the orientation is chosen stochastically from a distribution parameterized by supervised learning. Nevertheless, as the orientation task is much more complex than the sequence, the sequence task will suffer from the bad initializer of orientation if they are trained at the same time. To overcome this shortcoming, pre-training the orientation task first could be a reasonable idea. However, orientations of the items is tightly attached to the existence of packing sequence of items in new type 3D BPP. Consequently, we train different types of tasks separately for each random batch to keep them dynamic balance. We called this Multi-task Selected Learning (MTSL). Mathematically,
there are three kinds of basic loss function in our work:
\[ p(s_i) = p(s_i|s_1,o_1^*,...,s_{i-1},o_{i-1}^*,\theta_{seq},x), \]
\[ L_{seq} = (SA(s,o|x) - bl(x)) \sum_{i=1}^{n} \log p(s_i), \]
\[ L_{ori} = - \sum_{i=1}^{n} \log p(o_i^*|s_i,o_2^*,...,s_{i-1},o_{i-1}^*,s_i,\theta_{ori},x), \]
\[ L_{all} = L_{seq} + L_{ori}. \]

where \( SA(s,o) \) denotes the surface area of the bin in the case of placement sequence and its corresponding orientation of \( o \) and \( x \), and \( bl(x) \) denotes the baseline surface area by a well-designed algorithm NBPH. During training, our model will choose one of the three kinds of loss \( \{L_{seq},L_{ori},L_{all}\} \) to balance difficulty among them with a certain probability.

### 4.3.2 Baseline iteration: Exponential Moving Average.

For sequence task, as we use the REINFORCE estimator with baseline, a good baseline reduces the variance of the estimator and increases the convergence speed of learning. In our work, we resort a baseline iteration as there is no need to differentiate between inputs and simple to implement, rather than critic. For an sample \( x \), the baseline value \( bl(x) \) is initialized by calculating the surface area of a packing plan which is generated by the heuristic algorithm NBPH. In each step, the baseline value is updated as:
\[ bl'(x) = SA(s,o|x) + \alpha(b(x) - SA(s,o|x)), \]
where \( SA(s,o|x) \) is the surface area calculated at each training step and \( \alpha \) is the hyper-parameter for exponential moving average.

As a conclusion of the discussion above, the training procedure of our Multi-task Selected Learning method can be summarized in Algorithm 1.

### Algorithm 1 Multi-task Selected Learning.

1. Training set \( X \), number of training steps \( T \), batch size \( B \).
2. Initialize Sequence and Orientation parameters \( \theta_{seq}, \theta_{ori} \).
3. Initialize baseline value \( bl \) according to NBPH.
4. for \( t = 1 \) to \( T \) do
   5. Select a batch of samples \( x \) and calculate baseline \( bl(x) \).
   6. Sample sequence \( s \) by sequence task probability \( p_{\theta_{seq}}(\cdot|x) \).
   7. Obtain free space \( f \) and orientation \( o^* \) for \( s \) by Algorithm 3
   8. Calculate \( SA(s,o^*|x) \) by tuple \((s,o^*,f)\) by Algorithm 3
   9. \( g_{\theta_{seq}} = \mathbb{E}[(SA(s,o^*|x) - bl(x))\nabla_{\theta_{seq}} \log p(s|x)] \)
10. \( g_{\theta_{ori}} = \mathbb{E}[(\nabla_{\theta_{ori}} \log p(o^*|x)] \)
11. \( g_{\theta_{all}} = g_{\theta_{seq}} + g_{\theta_{ori}} \)
12. \( g = \text{choice}(g_{\theta_{seq}},g_{\theta_{ori}},g_{\theta_{all}}) \)
13. Update \( \theta = \text{ADAM}(\theta, g) \).
14. Update baseline \( bl(x) = SA(s,o^*|x) + \alpha(b(x) - SA(s,o^*|x)) \).
5. end for
6. return all parameters \( \theta_{seq}, \theta_{ori} \).

### 4.3.3 Testing.

Moreover, different from traditional machine learning problems, combinational optimization problem are to minimize or maximize the objective under some constraints, which seems to have some of prior knowledge about how to evaluate the result. For our new type 3D BPP, the benchmark is the smaller the better.

As evaluating a surface area is inexpensive, we can simply generate multiple candidate solutions per order at inference time and select the best. The way generates candidates is based on the calculated probability \( p_{\theta_{seq}}(\cdot|x) \) and \( p_{\theta_{ori}}(\cdot|x) \). Meanwhile, as we can only obtain the packing sequence and orientation for each order, we still need Algorithm 5 in Appendix B to get the free space set for calculating the surface area.

### 5 EXPERIMENTS

We conduct experiments to investigate the behavior of the proposed multi-task selected learning BPP methods. As has been said above, our experiments are conducted on our 3D Bin Packing order data set L3DPBD.

#### 5.1 Implementation Details

Across all experiments, we use mini-batches of 128 sequences, LSTM cells with 128 hidden units, and embed the three coordinates of each item in a 128-dimensional space. We train our model with Adam optimizer [23] with initial learning rate of \( 10^{-3} \) and decay every 5000 steps by a factor of 0.96. All the parameters are initialized randomly in \([-0.08, 0.08]\) and clip \( L2 \) norm of our gradients to 1.0. We use the surface area calculated by heuristic algorithm NBPH as initial baseline input and apply \( \alpha = 0.7 \) during the baseline iteration. We use 100000 steps to train the model and it will take about 70 hours on Tesla M40 GPU machine. Model implementation with TensorFlow will be made available soon. Based on comprehensive consideration, the performance indicator is average surface area (ASA) which denotes the total cost of packing materials. The mathematical definition of ASA is as follows:
\[ ASA = \frac{\sum_{i=1}^{n} SA(i)}{n}, \]
where \( SA(i) \) is the surface area of \( i \)-th order and \( n \) is the number of order. The experimental procedure and results are as follows.

#### 5.1.1 Multi-task Selected Learning Training.

For MTSL experiments, at each step, we choose a loss function of \( \{L_{seq},L_{ori},L_{all}\} \) defined in section 4.3.1 at each batch adapting to the training process. Actually, we sample three loss with probabilities 0.3, 0.5, 0.2. The values of probabilities are annealed to 0.33, 0.33, 0.33 respectively over the first 10,000 training steps. Furthermore, we sample 800 times for each test instance and obtain the best as described in Section 4.3.3. For comparison, we computed two kinds of ASA: 1) the sequence is generated by the sequence task and orientation is generated by the Algorithm 3; 2) the sequence and orientation are both generated by our MTSL model. We refer to those results as MTSL single-sample and MTSL multiple-sample.

#### 5.1.2 Reinforcement Learning Training.

In addition to the described methods, we also implement and train a pointer network with reinforcement learning similarly to [18] as the strong baseline method. The stochastic policy of RL model is only related to the sequence of placing the items into bin, since other major factors are defined by the same heuristic algorithm NBPH mentioned above. Previous work has already proved the effectiveness of this method against the heuristic method, in which experiments shows that the reinforcement learning results with beam search at inference time are better than the NBPH baseline with an improvement of about
We regard it as strong baseline named RL vanilla-bs in our paper. On the contrary, the vanilla method never takes the packing sequence that has already been produced into consideration. We will empirically show the necessity of the introduced intra-attention mechanism through plenty of experiments and we refer this as RL intra-bs. Consequently, the remaining experiments about RL models have introduced the intra-attention mechanism by default. Furthermore, in order to carry out comparative test with the MTSL model, as our MTSL model adopts sampling strategy to obtain the best result at inference time, we also apply it in RL model. We call this approach RL single-sample. Meanwhile, to verify the effectiveness of probability distribution of orientations calculated by our MTSL model, we compare the results of the sequence generated by RL model following by random orientations, which is denoted as RL multiple-sample.

5.2 Results and Analysis

First of all, we evaluate different models including RL vanilla-bs and RL intra-bs on our proposed dataset L3DBPD. We report the ASA in Table 3. As the table shows, the RL vanilla-bs achieves 4.89%, 4.88%, 5.33% improvement than NBPH for BIN8, BIN10 and BIN12, whereas the improvement of the RL intra-bs is increased to 5.19%, 5.26%, 5.41% instead. Apparently, this demonstrates the usefulness of our intra-attention training method for New 3D BPP which can help reduce the surface area. Besides, we also conduct the approach BRKGA in [14] on L3DBPD to validate whether these method designed for fixed-sized bin are appropriate for our new 3D BPP. BRKGA is one of the state-of-the-art method to tackle 2D and 3D fixed-sized bin packing problems which adopts a heuristic method of Genetic Algorithm. According to Table 3, we find that BRKGA is dramatically limited by the non-fixed-sized bin. It even performs worse than the random method which randomly generates the packing sequence and adopts the orientation and free space calculated by Algorithm 3. All in all, the statistical analysis confirms that RL intra-bs is significantly better than all the other approaches outlined below.

Table 3: Comparison with RL vanilla-bs and RL intra-bs on the L3DBPD Dataset.

| model              | BIN8   | BIN10  | BIN12  |
|--------------------|--------|--------|--------|
| Random             | 44.70  | 48.38  | 50.78  |
| NBPH method        | 43.97  | 47.33  | 49.34  |
| BRKGA              | 59.44  | 65.73  | 68.47  |
| RL vanilla-bs      | 41.82  | 45.02  | 46.70  |
| RL intra-bs        | **41.69** | **44.84** | **46.67** |

Thus, we report the ASA of our rest approaches on BIN8, BIN10, and BIN12 in Table 4. Notably, results demonstrate that training with RL and MTSL significantly comfortably surpass the NBPH method. By calculation, the proposed MTSL multiple-sample achieves 6.21%, 7.80%, 8.55% improvement than NBPH method for BIN8, BIN10, and BIN12. In our experiments, MTSL multiple-sample proves superior than other methods in most cases but is slightly less competitive than RL single-sample in BIN8.

We present a more detailed comparison of our methods in Figure 3, where we show the relationship between their performances and corresponding sampling times. The curve reveals a fact that MTSL multiple-sample approach which produces the sequence and orientations simultaneously outperforms other learning configurations in most cases. The contrast between RL multiple-sample and MTSL multiple-sample demonstrably illustrates that the probability distribution of orientations calculated by our MTSL model is indeed effective, as the ASA with MTSL significantly improves over RL. Furthermore, we find that the orientation task can definitely promote better results, even though the MTSL multiple-sample is just a few percents worse than NBPH baseline and MTSL single-sample at the very beginning. Moreover, MTSL multiple-sample approach benefits from much less orientation search space and run efficient than the methods that adopt the heuristic orientation calculation which will search from 6th space at most.

Finally, we randomly pick an example order and produce results by different methods in Figure 4 for case study. The example order is from BIN8 and the results are obtained by sampling 500 times. As shown, the surface area computed by each method is listed on the top of each results. Obviously, the computational experiment results demonstrate that MTSL multi-sample can produce a more reasonable packing policy than other methods.

6 RELATED WORK

Our work applies the latest Deep Learning technology to solve a new type of 3D BPP. In this section, we will go through the main research result of 3D BPP and machine learning method in combinatorial optimization.

6.1 3D bin packing problem

Bin packing problem is a classical and popular optimization problem. Since 1970s, it has attracted great interest of many researchers and some valuable achievements have been obtained. As a strongly NP-hard problem, a lot of research focuses on approximation and heuristic algorithms for 3D BPP. [28] proposes the first approximation algorithm for 3D BPP and investigates the performance bound of the algorithm. And many effective heuristic algorithms, such as genetic algorithm [14], guided local search [12], extreme point-based heuristics [11], hybrid genetic algorithm [20], have been proposed. There are also some research about exact solution method for 3D BPP. [6] considers a problem of loading containers with cartons of non-uniform size, which is a generalization of 3D BPP where bins may have different sizes, and a mixed integer programming model is developed to obtain optimal solutions. Some variants of BPP from real world are also studied, such as variable size bin packing problem.
Figure 3: Results for multi-task binpacking network. (a), (b), (c) shows the result of BIN8, BIN10, BIN12, respectively.

Figure 4: Results for MTSL single-sample, MTSL multiple-sample, RL single-sample, RL multiple-sample and NBPH method. The surface area of these method are 27.24, 26.66, 30.75, 27.24 and 28.62 respectively.

Another class of geometrical packing problem, named strip packing problem (SPP), is also worth mentioning here, because it is very similar to the problem we studied. In SPP, a given set of cuboid-shaped items should be packed into a given strip orthogonally by minimizing the height of packing. The length and width of the strip is fixed and limited, and the height is infinite (for 2D SPP, the width of strip is fixed and the length is infinite). Different types of algorithms have been proposed to solve the problem, such as exact algorithms in [21], approximation algorithm in [30], heuristic algorithm in [4] and meta-heuristic algorithms in [3] and [17].

Our work focus on a new type of 3D BPP and its objective is to minimize the surface area of the bin that can place all the items. As shown in the Appendix A, this is a NP-hard problem. Previous methods of solving typical 3D BPP could not achieve persuasive performance (BRKGA) and we propose a new well-designed heuristic algorithm (NBPH). The heuristic algorithm can be seen as a series of decisions. We believe that if we use DNN to parameterize these decision processes, we can obtain a more powerful algorithm, similar to the performance boosted in [18].

6.2 Machine Learning in combinatorial optimization

Even though machine learning and combinatorial optimization have been studied for decades respectively, there are few investigations about application of machine learning method in combinatorial optimization problems. One research direction is designing hyper-heuristics based on reinforcement learning ideas. An overview of hyper-heuristics is presented in [5], in which some hyper-heuristics based on learning mechanism are discussed.

Recent advances in sequence-to-sequence model [31] have motivated the research about neural combinatorial optimization. Attention mechanism, which is used to augment neural networks, contributes a lot in areas such as machine translation ([1]) and abstractive summarization [26]. In [26], intra attention mechanism is proposed to address the repeating phrase problem in abstractive summarization. In [32], a neural network with a specific attention mechanism named Pointer Net was proposed and a supervised learning method is applied to solve the TSP. [2] develops a neural combinatorial optimization framework with RL, and some classical problems, such as TSP and Knapsack Problem are solved in this framework.

7 CONCLUSION

In this paper, we focus on a new type of 3D bin packing problem, in which the objective is to minimize the surface area of the bin that can pack all items. Due to the complexity of the problem, it is very difficult to obtain optimal solution. If we use DNN to parameterize these decision processes similar to what does in heuristic, we can obtain a more powerful algorithm. Therefore, we apply a multi-task selected learning based method to optimize the sequence and orientation of items to be packed. The model is trained and tested with a large number of real data. Experimental results show that the multi-task selected learning method outperforms both a well-designed, effective heuristic algorithm and the previous DRL-based method significantly. We also release a large-scale 3D bin packing order dataset. In the future research, we will focus on investigation of more effective network architecture and training algorithm. In
addition, integrating the selection of empty maximal space into the architecture of neural network is also worthy of study.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014).

[2] Irwan Bello, Hiru Pham, Quoc V Le, Mohammad Norouzi, and Samy Bengio. 2016. Neural Combinatorial Optimization with Reinforcement Learning. arXiv preprint arXiv:1611.09940 (2016).

[3] Andreas Bortfeldt. 2006. A genetic algorithm for the two-dimensional strip packing problem with rectangular pieces. EJOR 172, 3 (2006), 814–837.

[4] Andreas Bortfeldt and Daniel Mack. 2007. A heuristic for the three-dimensional strip packing problem. EJOR 183, 3 (2007), 1267–1279.

[5] Edmund K Burke, Michel Gendreau, Graham Kendall, Gabriela Ochoa, Ender Ozcan, and Rong Qu. 2013. Hyper-heuristics: A survey of the state of the art. Journal of the Operational Research Society 64, 12 (2013), 1695–1724.

[6] CS Chen, Shen-Ming Lee, and QS Shen. 1995. An analytical model for the container loading problem. EJOR 80, 1 (1995), 65–78.

[7] Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long Short-Term Memory-Addition, integrating the selection of empty maximal space into the architecture of neural network is also worthy of study. in CIKM ’18, October 2018, Lingotto, Turin, Italy H. Hu et al.

[28] Guntram Schröthauer. 1991. A three-dimensional bin packing algorithm. Elektronische Informationsverarbeitung und Kybernetik 27, 5/6 (1991), 263–271.

[29] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of Go with deep neural networks and tree search. nature 529, 7587 (2016), 484–489.

[30] A Steinberg. 1997. A strip-packing algorithm with absolute performance bound 2. SIAM J. Comput. 26, 2 (1997), 401–409.

[31] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In NIPS. 3104–3112.

[32] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In NIPS. 2692–2700.

[33] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning 8, 3–4 (1992), 229–256.

A NP-HARDNESS OF NEW TYPE OF 3D BPP

LEMMA A.1. The new type of 3D BPP proposed in this paper is NP-hard.

Proof: First of all, we will prove the new type of 2D BPP is NP-hard. To show it is NP-hard, we will give a reduction of 1D Bin Packing Problem.

Given a one-dimensional bin packing problem, it consists of \( n \) items with integer size \( w_1, \ldots, w_n \) and bins with integer capacity \( W \). The objective is to minimize the number of bins used to pack all items.

To convert it into new type of 2D BPP, we assume that there are \( n \) items with length \( w \) and width \( 1/(n \cdot \max(w_i)) \). Besides there is another item with length \( W \) and width \( W \cdot n \cdot \max(w_i) \), which is denoted as Base Item. The new type of 2D BPP problem is to find the bin with least surface area to pack all the generated \( n + 1 \) items.

Without loss of generality, we assume the Base Item is on the left-bottom of the bin. Adding one item on the right side of the Base Item yields the total surface area is increased at least \((W \cdot n \cdot \max(w_i))/(n \cdot \max(w_i))\) \( = W \). At the same time, even if all the items are added on the upper side of the Base Item, the total increased area is at most \( W \). Thus, all the items will be put on the upper side of the Base Item.

Next, we will prove the orientation of the item will not be changed. If reversing one item with width and length, the increased area is at least \( W \cdot \min(w_i) \) for this item. However, the increased area is at most \( W \) for all items if no item is reversed.

If we can find out a bin with least surface area to pack this \( n + 1 \) items, we find out the least number of bins of capacity \( W \) that can contain \( n \) items of size \( w_1, \ldots, w_n \). Therefore, if we can solve the new type of 2D BPP in polynomial time, the one-dimensional bin packing problem can be solved in polynomial time, which completes the proof that this new type of 2D BPP is NP-hard unless \( P = NP \).

For the new type of 3D bin packing problem, we will add height \( 1/(n \cdot \max(w_i))^2 \) for each item in the 2D case, which will ensure no item will be added on the height side. Proof is the same as the 2D case.

B NEW 3D BIN PACKING HEURISTIC ALGORITHM

New 3D bin packing heuristic algorithm (NBPH) is based on idea of choosing the item, orientation and free space according to least waste space criteria, which aims to obtain least surface area at each step. The detailed algorithm NBPH is as Algorithm 2:
Algorithm 2 New 3D Bin Packing Heuristic Algorithm.

1. Denote the set of $n$ items as $I$. Each item is of length $l_i$, height $h_i$ and width $w_i$.
2. Initialize a sufficiently large bin $B$ with length $L$, width $W$, height $H$ (We could set $L = W = H = \sum_{i=1}^{n} \max(l_i, h_i, w_i)$).
3. Initialize the set of remaining items $\hat{I} = I$ and the set of free spaces as $ES = \emptyset$.
4. for $i = 1$ to $n$ do
5. if $i = 1$ then
6. Select an item $i$ with largest surface area.
7. Put the item into the bin and generate 3 free spaces $ES1$.
8. Update $ES = ES1$.
9. Update $\hat{I} \leftarrow \hat{I} \setminus i$.
10. else
11. Select a item $i$ from $\hat{I}$ according to (Algorithm 4).
12. Update $\hat{I} \leftarrow \hat{I} \setminus i$.
13. Select an free space $f_i$ from $ES$ and decide the orientation $\hat{o}_i$ according to Algorithm 3.
14. Update $ES \leftarrow ES \cup ES1 \setminus ES2$.
15. end if
16. end for
17. Return the surface area of the bin that could pack all items.

Algorithm 3 Least Surface Area Heuristic for Picking Free Space and Orientation.

1. Denote the set of free space as $ES$, the set of orientations as $o$.
2. Initialize the least surface area for item $i$ as $LSA_i$. $LSA_i$ is large enough.
3. Initialize best free space $\hat{s}_i = null$, best orientation as $\hat{o}_i = null$.
4. for each free space $f \in ES$ do
5. for each orientation $o \in o$ do
6. Calculate the surface area $LSA_{i,f,o}$ after putting item $i$ in free space $f$ with orientation $o$.
7. if $LSA_{i,f,o} < LSA_i$ then
8. Update $\hat{f}_i = f$.
9. Update $\hat{o}_i = o$.
10. Update $LSA_i \leftarrow LSA_{i,f,o}$.
11. else if $LSA_{i,f,o} = LSA_i$ (Apply tie-breaking rule) then
12. Set $l_f, w_f, h_f$ the length, width, height of free space $f$
13. Set $l_i, w_i, h_i$ the length, width, height of item $i$ with orientation $o$
14. Set $rest1 = min(l_f - l_i, w_f - w_i, h_f - h_i)$.
15. Set $rest2 = min(l_f - l_i, w_f - w_i, h_f - h_i)$
16. if $rest1 < rest2$ then
17. Update $\hat{f}_i = f$.
18. Update $\hat{o}_i = o$.
19. end if
20. end if
21. end for
22. end for
23. Update $ES \leftarrow ES \cup ES1 \setminus ES2$.
24. Return $\hat{f}_i, \hat{o}_i$ for item $i$ and $ES$.

Algorithm 4 Least Surface Area Heuristic for Picking Item.

1. Denote the set of remaining items as $\hat{I}$.
2. Denote the surface area of item $i$ as $SA_i$.
3. Initialize the best item $\hat{i} = null$.
4. Initialize the Least surface area as $LSA = (\max(l_i, w_i, h_i) \cdot n)^3$.
5. for each item $i \in \hat{I}$ do
6. Calculate the surface area $SA$ after packing item $i$ according to $\hat{f}_i, \hat{o}_i$ (which is determined by Algorithm 3).
7. Denote the least waste space of item $LSA_i = SA - SA_i$.
8. if $LSA_i < LSA$ then
9. Update $LSA = LSA_i$.
10. Update $\hat{i} = i$.
11. end if
12. end for
13. Return item $\hat{i}$.

Algorithm 5 Least Surface Area Heuristic for Picking Free Space.

1. Input: Item $t$ and Orientation $o_t$.
2. if $t = \emptyset$ then
3. Initialize the set of free space $ES = (L, H, W)$, $(L, H, W)$ represents a sufficiently large bin.
4. else
5. $ES$ is retrieved from the latest step
6. end if
7. Initialize the least surface area for item $t$ as $LSA_t$. $LSA_t$ is a large enough.
8. Initialize best free space $\hat{f}_t = null$.
9. for each free space $f \in ES$ do
10. Calculate the surface area $LSA_{t,f,o_t}$ after putting item $t$ in free space $f$ with orientation $o_t$.
11. if $LSA_{t,f,o_t} < LSA_t$ then
12. Update $\hat{f}_t = f$ and $LSA_t \leftarrow LSA_{t,f,o_t}$.
13. else if $LSA_{t,f,o_t} = LSA_t$ (Apply tie-breaking rule) then
14. Set $l_f, w_f, h_f$ the length, width, height of free space $f$
15. Set $l_t, w_t, h_t$ the length, width, height of item $t$ with orientation $o_t$
16. Set rest1 = min($l_f - l_t, w_f - w_t, h_f - h_t$).
17. Set rest2 = min($l_f - l_t, w_f - w_t, h_f - h_t$)
18. if $rest1 < rest2$ then
19. Update $\hat{f}_t = f$.
20. end if
21. end if
22. end for
23. Generate new free spaces($ES1$) and delete those that are intersected and overlapped($ES2$).
24. Update $ES \leftarrow ES \cup ES1 \setminus ES2$.
25. Return $\hat{f}_t$ for item $t$ and its orientation $o_t$ and free space set $ES$. 