A reinforcement learning algorithm for the 2D-rectangular strip packing problem

A reinforcement learning algorithm for the 2D-rectangular strip packing problem

Xusheng Zhao[a], Yunqing Rao[b*], Jie Fang[c]

School of Mechanical Science & Engineering, Huazhong University of Science and Technology, Wuhan, Hubei, 430074, China

Email: [a]m201671243@alumni.hust.edu.cn, [b*]ryq@hust.edu.cn, [c]fangjie@hust.edu.cn

Abstract. The 2D packing problem is categorized as one branch of the cutting and packing problems, which is widely spread in the manufacturing industries. Over the years many meta-heuristics have been proposed and applied on the packing problem. Recently, the approach combined with machine learning serves as a novel paradigm for solving the combinatorial optimization problem. However, the machine learning approaches have very limited literature reports on the appliance of the packing problem. We propose a reinforcement learning method for the 2D-rectangular strip packing problem. The solution is represented by the sequence of the items and the layout is constructed piece by piece. We use the lowest centroid placement rule for the piece placement, then a Q-learning based sequence optimization is applied. Three groups of conditions are set for the testing, the computational results show the Q-learning approach has good effect on the compaction of the layout.

1. Introduction

The 2D rectangular packing problem, also known as polygon placement, marker making, non-convex cutting stock, or some combination of any of these terms [1], is widely spread in the manufacturing industries such as steel, glass, paper, leather, and textile production. A higher utilization layout will reduce the material costs and significantly promotes the profitability [2].

The 2D packing problem lies in one branch of the C&P problem (cutting and packing problem), i.e. a type of arrangement that small items should be contained inside larger objects [3]. Wascher [4] gave the universally accepted definition of the C&P problem: a set of large objects and a set of small items, the small items of the subset do not overlap and lie entirely within the large object, with a given objective function to be optimized. The objective function could generally be classified into the fewest resources as possible shall be used to achieve a predefined goal, or the given resources shall be used to provide the best possible result [5].

As there are different parameter settings and specified characteristics in practice, they define a wide variety of particular C&P problems [6]. The C&P problems could be classified as regular [7] and irregular [8] subtypes according to the shape of items, or one-dimensional [9], two-dimensional [10] and three-dimensional [11] cases according to the dimensionality. The 2D rectangular packing problem is the two-dimensional regular type of the C&P problem, and it is one of the NP-hard combinatorial optimization problems. Although exact algorithms can provide optimal solutions, they usually take huge amount of computational effort. Since meta-heuristics are able to produce satisfactory solutions within reasonable time, over these years, many meta-heuristics have been applied to solve this problem [12], such as tabu search, simulated annealing, and genetic algorithms. Meta-heuristics have been taken as popular choices and proved to present good optimization abilities.

The approach combined with machine learning serves as a novel paradigm to solve the combinatorial optimization problem. Especially the deep neural networks offer satisfactory solutions to the originally problems which are deemed as complicated and difficult to address, such as face recognition, machine translation, and intelligent recommendation systems. Machine learning has achieved widespread success in both academia and industry. Seq2seq (Sequence to Sequence) belongs to RNN (Recurrent Neural Network) and is capable of fitting the mapping between two sets of sequences [13], the attention mechanism [14] has brought extra memory into the seq2seq and provided it with additional capacity to handle long sequences.

Vinyals [15] proposed the architecture based on seq2seq named Pointer Network, and used it as the approximate solver for optimization problems, e.g. convex hulls, Delaunay triangulations, and TSP (Travelling Sales Problem). Bello [16] combined this architecture with policy gradient as the DRL (deep reinforcement learning) approach to solve TSP and observed better results. As for the C&P problems, Hu [17] and Duan [18] also applied the DRL to solve a new 3D bin packing problem arising at the e-commerce platform. Other researches integrated with machine learning approaches include VRP (Vehicle Routing Problem) [19] and the binary quadratic programming [20].

Using machine learning approaches to solve the combinatorial optimization problem is a relatively novel direction, and there are very few literature reports on its appliance in the 2D packing problem. In this paper, we adopt the Q-learning method, i.e. a paradigm of reinforcement learning, to solve the 2D rectangular strip packing problem.

2. Problem description

In the 2D rectangular strip packing problem, the stock sheet has a fixed width W, and one open dimension, i.e. the length L, each item is moved into the stock sheet and positioned at the place where it can be accommodated. The distance between the right side of the last item being placed and the y axis is nominated as length L', and L' is to be minimized, as shown in figure 1.

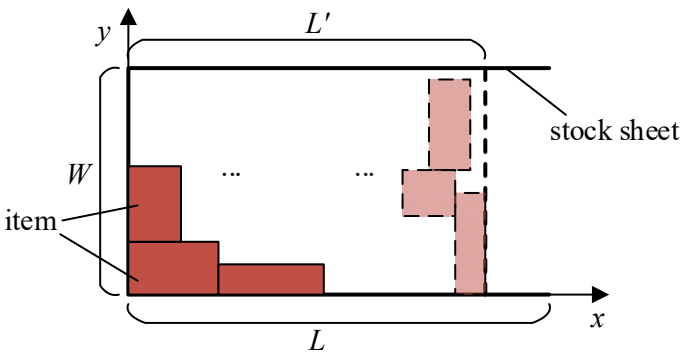


Figure 1. Rectangular strip packing problem

When the layout is constructed with items piece by piece [12], the solution is naturally represented as the sequence of the items, along with the placement rules, each solution is guaranteed with feasibility. We applied this layout construction method to solve the 2D rectangular strip packing problem.

3. Algorithm

In this section, the placement rule we choose is presented, then we describe the Q-learning approach for the sequence optimization of the items packing.

3.1. Lowest centroid placement rule

The lowest centroid placement rule is adopted, i.e. for the piece to be placed, each position with enough accommodation is searched over while the piece is allowed to possess a rotation of 90 degrees. The position with the smallest x-coordinate of the piece's centroid (i.e. the most left position) when the

piece being placed is selected as the location point, as well the bottom left corner of the piece should be adjacent to other parts or the boundary of the stock sheet. For example, suppose we are constructing a layout and the current piece to be packed is denoted by k. The left side of the boundary is filled with already packed items and not appropriated for new pieces placement, while the right side has enough accommodation. Suppose the position p is selected for the piece k, and the length and width of k is denoted by l_k and w_k respectively, as shown in figure 2 (a). The x-coordinate of p is denoted as x_p, and k is assumed to be located at p and the x-coordinate of its centroid is calculated as $cent_{x1} = x_p + l_k / 2$, as shown in figure 2 (b), and the same calculation is performed again while k is assumed to be located at p with a rotation of 90°, $cent_{x2} = x_p + w_k / 2$, as shown in figure 2 (c). In this example, $cent_{x2} < cent_{x1}$, then k with the rotation of 90° is confirmed, and the layout is the same as shown in figure 2 (c).

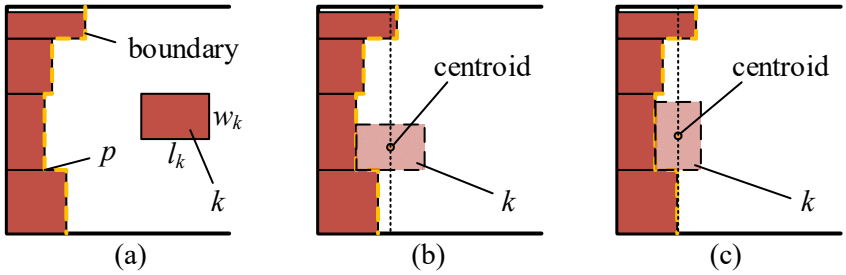


Figure 2. Placement decision of each piece during the packing. (a) The current packing state, the piece to be placed is denoted by k. (b) Piece k takes a rotation of 0°. (c) Piece k takes a rotation of 90°.

3.2. Q-learning for sequence optimization

Q-learning is a type of reinforcement learning method which is suitable for sequential decision. In each stage of the decision process, the agent takes an action, then the state of the environment changes and returns the agent a reward as the feedback. Through continuous interaction with the environment, the agent will automatically learn the strategy to act at each step to accumulate the largest amount of reward.

The construction of the layout of rectangular packing is modelled as MDP (Multistage Decision Process). Suppose there are n items to be placed, then the MDP has n stages. We denote the state at stage i by s_i, and action by a_i, reward by r_i, respectively, the MDP is as follows:

$$s_0, a_1, s_1, r_1, a_2, s_2, r_2 \ldots, a_n, s_n, r_n$$

The initial state s_0 represents the empty layout that none of these pieces have been placed. At the i-th stage, $i \in [1, n]$, we define the choice of the next piece to be placed at state s_{i-1} as a_i, $a_i \in [1, n]$. After the agent takes the action a_i, the environment state changes from s_{i-1} to s_i, we nominate that $s_i = a_i$, then s_i has the same range as a_i, $s_i \in [1, n]$. This is shown in figure 3.

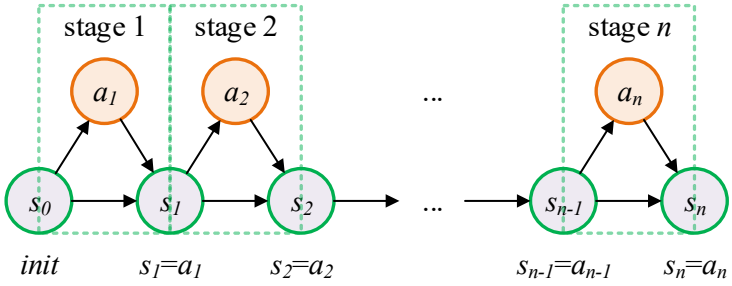


Figure 3. Markov Decision Process for the n stages packing

As for the action a_i, it represents the piece to be arranged, which is not supposed to repeat, then $a_0 \neq a_1 \neq \ldots \neq a_n$. For the reward r_i, since the length L' can be calculated only when all the pieces have been arranged, then we set the only non-zero reward at the n-th stage (i.e. the last stage of each

episode), $r_1 = r_2 = \ldots = r_{n-1} = 0$, $r_n \neq 0$. The target of reinforcement learning is normally designated as the maximum of the accumulative reward, while in this case we aim at the minimum of the length L', so we set $r_n = C/L'$, where C is a given constant.

Q(s, a) represents the expectation of the long term accumulative reward, which the agent may obtain at state s if it takes action a. In this case, it represents the impact that the choice of the next piece a may have on the length L' at state s. Q(s, a) is updated as in equation 1.

$$Q(s, a) = Q(s, a) + \alpha[R(s, a) + \gamma * \text{Max}(Q[s', a']) - Q(s, a)], \quad (1)$$

where α is the learning rate, it contains the updating efficiency of Q(s, a), and γ is the discounted factor, which represents how much the future reward can be observed in the current state.

We set m episodes for the Q-learning exploration. After these episodes, each sequence of the states $\{s_1, s_2, \ldots, s_n\}$ or the sequence of the actions $\{a_1, a_2, \ldots, a_n\}$ represents one solution to the 2D rectangular strip packing problem. We denote the optimal solution by S_{opt}, and S_{opt} is continually replaced in each episode by the better solution constructing the smaller length L'. In each episode, all the pieces should be selected once and placed on the stock sheet, then each episode contains n internal cycles for the choices of the n pieces. The Q-learning method is shown in algorithm 1.

Algorithm 1. Q-learning for rectangular packing

```
Initialize Q table as a matrix of 0
Initialize S_opt
for t = 1 to m do:
    Initialize s_0
        for i = 1 to n do:
            Choose a_i at s_{i-1} according to ε-greedy policy
            Take a_i, enter stage i, s_i = a_i
            if i = n then r_i = C/L'
            else r_i = 0
            Update Q(s_{i-1}, a_i)
        end for
    Update S_opt
end for
Output S_opt
```

4. Computational experiments

We test the Q-learning approach for the sequence optimization of the 2D rectangular strip packing problem. The stock sheet is set with a fixed width W, and we assign that the length l_i and width w_i of each piece i is randomly selected within a given range. Besides, we set W, l_i and w_i with integers. Three conditions are provided.

Condition 1: $W = 20$, $w_i \in [5, 7]$, $l_i \in [8, 10]$, $n = 10$

Condition 2: $W = 40$, $w_i \in [6, 10]$, $l_i \in [11, 15]$, $n = 20$

Condition 3: $W = 60$, $w_i \in [11, 15]$, $l_i \in [16, 20]$, $n = 30$

Under each condition, we generated a set of pieces, used Q-learning to search for the optimal sequence and constructed the layout, the length L' of which is denoted by $L'_{opt(t)}$, where t represents the number of the test. Then, with this set of pieces, we constructed the layout with a stochastic sequence, and the length L' of which is denoted by $L'_{(t)}$. The decline on length L' is denoted by $D_{L'}$, $D_{L'}$ is calculated and averaged on 100 times of performance as in equation 2.

$$D_{L'} = \sum_{t=1}^{100} \frac{L'_{(t)} - L'_{opt(t)}}{L'_{(t)}} \quad (2)$$

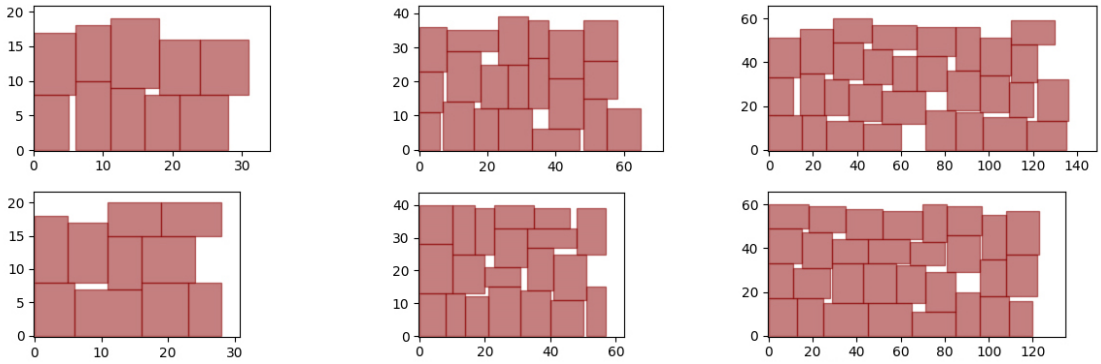
The results are shown in table 1.

Table 1. Average decline on length *L'* under the given conditions

	Condition 1	Condition 2	Condition 3
$D_{L'}$ (%)	7.254%	10.771%	8.217%

It can be observed from the table that the length *L'* constructed by the optimized sequence is smaller than the one constructed by a stochastic sequence, which means the sequence optimized by Q-learning constructs a tighter layout, this is beneficial for the higher material utilization.

We pick one layout comparison under each condition and present them in figure 4.



(a) (b) (c)

Figure 4. Layout comparison between the stochastic sequence and the optimized sequence under the three conditions. (a) Condition 1. (b) Condition 2. (c) Condition 3

5. Conclusion

A reinforcement learning method is proposed for solving the 2D rectangular strip packing problem. The solution is represented with the sequence of the items. For the placement strategy, we adopt the lowest centroid placement rule and the layout is constructed with items piece by piece. The construction of the layout is modelled as the multistage decision process. With the reciprocal of the layout length as the reward, the Q-learning approach is applied for the decision of the sequence of the items. Setting three groups of conditions about the width of the stock sheet, the item sizes and the number of items, we testify the effectiveness of the proposed method. Compared with random sequences, the computational results show the items placed with the sequences optimized by the Q-learning approach provide the tighter layout.

Acknowledgments

This work has been supported by the National Nature Science Foundation of China (No. 51975231) and the Foundational Research Funds for the Central Universities (No. 2019kfyXKJC043)

References

[1] Bennell J A, Oliveira J F. (2008) The geometry of nesting problems: A tutorial. European Journal of Operational Research, 184(2): 397-415.
[2] Elkeran A. (2013) A new approach for sheet nesting problem using guided cuckoo search and pairwise clustering. European Journal of Operational Research, 231(3): 757-769.
[3] Camacho E L, Marin H T, Ross P, Ochoa G. (2014) A unified hyper-heuristic framework for solving bin packing problems. Expert Systems with Applications, 41(15): 6876-6889.
[4] Wäscher G, Haußner H, Schumann H. (2007) An improved typology of cutting and packing problems. European Journal of Operational Research, 183(3): 1109-1130.
[5] Martinovic J, Scheithauer G, Carvalho J M V. (2018) A comparative study of the arcflow model and the one-cut model for one-dimensional cutting stock problems. European Journal of Operational Research, 266(2): 458-471.

[6] Leao A A S, Toledo F M B, Oliveira J F, Carravilla M A, Valdés R A. (2020) Irregular packing problems: A review of mathematical models. European Journal of Operational Research, 282(3): 803-822.
[7] Wu L, Tian X, Zhang J, Liu Q, Xiao W, Yang Y. (2017) An improved heuristic algorithm for 2D rectangle packing area minimization problems with central rectangles. Engineering Applications of Artificial Intelligence, 66: 1-16.
[8] Martins T C, Tsuzuki M S G. (2010) Simulated annealing applied to the irregular rotational placement of shapes over containers with fixed dimensions. Expert Systems with Applications, 37(3): 1955-1972.
[9] Castellanos M Q, Reyes L C, Jimenez J T, S C G, Huacuja H J F, Alvim A C F. (2015) A grouping genetic algorithm with controlled gene transmission for the bin packing problem. Computers & Operations Research, 55: 52-64.
[10] Mundim L R, Andretta M, Queiroz T A. (2017) A biased random key genetic algorithm for open dimension nesting problems using no-fit raster. Expert Systems with Applications, 81: 358-371.
[11] Wei L, Oon W C, Zhu W, Lim A. (2012) A reference length approach for the 3D strip packing problem. European Journal of Operational Research, 220(1): 37-47.
[12] Bennell J A, Oliveira J F. (2009) A tutorial in irregular shape packing problems. Journal of the Operational Research Society, 60: S93-S105.
[13] Sutskever I, Vinyals O, Le Q V. (2014) Sequence to Sequence Learning with Neural Networks. In: Advances in neural information processing systems. Montreal Canada. pp. 3104–3112.
[14] Bahdanau D, Cho K, Bengio Y. (2015) Neural machine translation by jointly learning to align and translate. https://arxiv.org/abs/1409.0473.
[15] Vinyals O, Fortunato M, Jaitly N. (2015) Pointer networks. In: International Conference on Neural Information Processing Systems. Montreal Canada. pp. 2692–2700.
[16] Bello I, Pham H, Le Q V, Norouzi M, Bengio S. (2016) Neural Combinatorial Optimization with Reinforcement Learning. https://arxiv.org/abs/1611.09940.
[17] Hu H, Zhang X, Yan X, Wang L, Xu Y. (2017) Solving a New 3D Bin Packing Problem with Deep Reinforcement Learning Method. https://arxiv.org/abs/1708.05930.
[18] Duan L, Hu H, Qian Y, Gong Y, Zhang X, Xu Y, Wei J. (2019) A Multi-task Selected Learning Approach for Solving 3D Flexible Bin Packing Problem. In: International Conference on Autonomous Agents and MultiAgent Systems. Montreal QC Canada. pp. 1386–1394.
[19] Yu J J Q, Yu W, Gu J. (2019) Online Vehicle Routing With Neural Combinatorial Optimization and Deep Reinforcement Learning. IEEE Transactions on Intelligent Transportation Systems, 20(10): 3806-3817.
[20] Gu S, Hao T, Yao H. (2020) A Pointer Network Based Deep Learning Algorithm for Unconstrained Binary Quadratic Programming Problem. Neurocomputing, 390: 1-11.