Article

A Hybrid Reinforcement Learning Algorithm for 2D Irregular Packing Problems

Jie Fang [1,2], Yunqing Rao [1,2,*], Xusheng Zhao [1,2] and Bing Du [1,2]

1 School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
2 State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
* Correspondence: ryq@hust.edu.cn

Abstract: Packing problems, also known as nesting problems or bin packing problems, are classic and popular NP-hard problems with high computational complexity. Inspired by classic reinforcement learning (RL), we established a mathematical model for two-dimensional (2D) irregular-piece packing combined with characteristics of 2D irregular pieces. An RL algorithm based on Monte Carlo learning (MC), Q-learning, and Sarsa-learning is proposed in this paper to solve a 2D irregular-piece packing problem. Additionally, mechanisms of reward–return and strategy-update based on piece packing were designed. Finally, the standard test case of irregular pieces was used for experimental testing to analyze the optimization effect of the algorithm. The experimental results show that the proposed algorithm can successfully realize packing of 2D irregular pieces. A similar or better optimization effect can be obtained compared to some classical heuristic algorithms. The proposed algorithm is an early attempt to use machine learning to solve 2D irregular packing problems. On the one hand, our hybrid RL algorithm can provide a basis for subsequent deep reinforcement learning (DRL) to solve packing problems, which has far-reaching theoretical significance. On the other hand, it has practical significance for improving the utilization rate of raw materials and broadening the application field of machine learning.

Keywords: irregular packing optimization; RL algorithm; machine learning; mathematical model; heuristic algorithm

MSC: 90-10; 90B35; 90C27

check for updates

Citation: Fang, J.; Rao, Y.; Zhao, X.; Du, B. A Hybrid Reinforcement Learning Algorithm for 2D Irregular Packing Problems. *Mathematics* **2023**, *11*, 327. https://doi.org/10.3390/math11020327

Academic Editor: Janez Žerovnik

Received: 29 November 2022
Revised: 27 December 2022
Accepted: 5 January 2023
Published: 8 January 2023

1. Introduction

With the rapid development of productivity brought about by technological innovation, reducing energy consumption has increasingly become a demand of various production industries [1]. At the same time, optimizing the raw materials used in manufacturing has become an important goal of the manufacturing system. In typical heavy industry, plates of metal or steel are the primary raw materials consumed in machinery manufacturing. The design of piece-cutting schemes is a necessary process and the first procedure in machinery manufacturing. Optimized packing schemes in this process can effectively improve the utilization rate of materials, thereby reducing manufacturing cost and improving economic benefits [2]. Packing-optimization problems deal with placing pieces to be arranged in a packing space in a certain way using the given space and certain constraints to achieve a specific optimization goal. Hereby pieces have different definitions on different occasions. In the metal processing industry, they refer to pieces to be processed; in the leather manufacturing industry, they refer to samples to be cut; and in the transportation industry, they refer to goods to be placed. According to the dimensions of pieces, packing problems can be classified into one-dimensional (1D) packing problems, two-dimensional (2D) packing problems, three-dimensional (3D) packing problems, and multidimensional packing problems. 2D irregular-piece packing [3,4], also known as special-shaped-piece packing, is a kind of 2D sheet-packing optimization. Compared to regular-piece packing, irregular-piece packing problems are very different in piece shape, solution strategy, and overlap-detection method. Therefore, they have a more extensive solution space and need more complex packing operations, which makes it challenging to obtain a satisfactory solution in polynomial time [5]. Many studies focus on obtaining an approximate optimal packing solution in an acceptable time.

Two-dimensional irregular packing is a classic mathematical and combinatorial optimization problem developed for decades. In the initial solution for 2D irregular packing, a single algorithm was applied, such as linear programming, a meta-heuristic algorithm, or a heuristic algorithm. These algorithms adjust the packing of items according to specific rules. Hopper et al. [6,7] studied the application of meta-heuristic and heuristic algorithms in 2D and 3D nesting. Bennell et al. [8] discussed a 2D irregular packing problem and related geometric problems. A solution to a packing problem of 2D irregular pieces developed from the previous rectangular-envelope algorithm [9] is adjacent packing based on real shape, which is an update and improvement of the packing method. Among the piece-packing methods based on real shapes, no-fit polygons (NFP) [10,11], raster methods (also called pixel methods) [12], linear programming (LP), and mixed integer linear programming (MIP) [13] are generally used to judge overlapping of pieces. Regarding placement rules, the bottom-eft (BL) [14] algorithm is the most commonly used method. Another problem worth studying is the sequence optimization of pieces, which is crucial to the final packing result [11]. Baosu Guo et al. [15] showed that the mathematical model for the 2D packing problem is mature, and there has been almost no disruptive technology in recent years. Another study [16] showed that most of the current packing algorithms improve the original method. However, with updates in packing technology, hybrid algorithms are increasingly used. Improvement of packing technology is not only the optimization of sequence or position but also the improvement of multiple packing-technology points simultaneously. Danyadi et al. [17] proposed a bacterial-evolution method aiming at the three-dimensional version of the bin packing problem in actual logistics, and fuzzy logic was utilized in the fitness calculation. Elkeran [18] adopted a method of combining irregular pieces in pairs and combining them with a rectangular-envelope algorithm to solve a packing problem of irregular pieces. This could effectively reduce the blank area, but the computational complexity was significantly increased. Sato et al. [19] not only adopted a heuristic paired-pack algorithm but also used a simulated annealing algorithm to guide the search process of the packing sequence and obtain specific packing effects. Beyaz et al. [20] proposed a hyperheuristic algorithm for solving a 2D irregular packing problem, which showed excellent robustness. In recent research on 2D irregular packing problems, Rao et al. [21] used a search algorithm hybridized with beam search and tabu search (BSTS), combined with a novel placement principle to complete packing of 2D irregular pieces. They obtained some results comparable to advanced algorithms in a short time. Souza Queiroz et al. [22] proposed a tailored branch-and-cut algorithm for a two-dimensional irregular-strip packing problem with an uncertain demand for the items to be cut, and developed a two-stage stochastic programming model considering discrete and finite scenarios, which achieved good nesting results. At present, the mainstream 2D irregular piece packing algorithms in the world are a hybrid algorithm based on metaheuristic sequencing algorithms (such as the particle swarm optimization algorithms [23], genetic algorithms [24], ant colony algorithms [25], and tabu search algorithms [21,26]); and a positioning algorithm based on NFP geometric operations (such as BL [27], bottom-left fill (BLF) [28], and maximal rectangles bottom-left (MAXRECTS-BL) [29,30]). Although the existing research on 2D irregular packing problems has made significant achievements and been applied to practical engineering problems, there are still some problems to be solved. First, solutions based on heuristic algorithms have poor universality, and the computing performance based on different data sets shows significant differences. Second, the intelligent optimization algorithm can fall into the local optimum on some problems, and the calculation cost is high.

The latest literature review on packing problems [15] shows that machine learning and deep learning algorithms may be helpful for sequential optimization of packing problems in the future. However, there is currently a lack of research in this area.

In recent years, artificial intelligence technology represented by RL has been widely studied and successfully applied in operational-research optimization, showing great potential to solve combinatorial optimization problems [31]. RL models the sequential-decision problem in operations research as an MDP and solves it. It improves the strategy by exploring and interacting with the environment. The characteristics of learning and online learning make it an important branch of machine learning research. Wang et al. [32] established a mathematical model and completed optimization of a centrifugal pump by using an artificial intelligence method. Kara et al. [33] solved a single-stage inventory decision-making problem considering product life using an RL method based on Sarsa-learning and Q-learning. Zhang et al. [34] used an improved Q-learning algorithm based on bounded table search to solve random customer demand and obtained good results. Kong et al. [35] tried to build a unified framework for solving linear combination optimization problems based on RL and used the knapsack problem as an example, whereby the gap between their results and optimal solutions was less than 5%. Chengbo Wang et al. [36] proposed using Q-learning to solve a problem of unmanned-ship-path planning. Laterre et al. [37] designed an algorithm based on ranked reward and applied it to a 2D bin packing problem; then the strategy was evaluated and improved using deep neural networks and Monte Carlo trees [38]. Wang Shijin et al. [39] studied a dynamic JSP problem of three scheduling rules by using Q-learning. The results showed that the Q-learning method can improve the agent's adaptability. Zhao et al. [40] solved a 2D rectangular packing problem by using a Q-learning search. As a progression of reinforcement learning, deep reinforcement learning (DRL) is also a mainstream machine learning method, which has led to some achievements in many fields, including combinatorial optimization. For example, Bello et al. [41] used a deep learning architecture and RL training to obtain the optimal solution for a large-scale TSP problem, and save computing costs. Hu et al. [42] and Duan et al. [43] used a pointer network, in combination with supervised learning or RL training, and combined it with certain heuristic algorithms to solve a 3D bin packing problem. Even so, the shape of 2D irregular pieces is special, resulting in high requirements for the input settings of neural networks. To the best of our knowledge, there is no research on deep reinforcement learning algorithms for 2D irregular packing problems, or on using reinforcement learning algorithms to solve 2D irregular-piece packing problems. The investigation of packing technology based on reinforcement learning can provide improved technical support for reinforcement training processes in deep reinforcement learning algorithms, explore new 2D irregular packing solutions, model designs, and expand the application field of machine learning. In addition, research on the solution method based on machine learning of 2D irregular-piece packing can reduce the design error, which leads to great practical application potential.

In this paper, research on 2D irregular-piece packing using RL methods is proposed for the first time to compensate for the weakness of traditional packing algorithms. We adopted a piece-sequence-generation method based on MC learning, Q-learning, and Sarsa-learning and designed a reward- and strategy-update mechanism based on piece packing. Combined with the classical BL positioning algorithm, packing of a 2D irregular piece can be realized. Finally, the piece packing test based on actual shapes was carried out with standard instances, proving the algorithm's effectiveness. In this paper, the packing problem is summarized first; then the 2D irregular packing problem is described and modeled. Secondly, a positioning strategy based on NFP and principles of sequence optimization based on an RL algorithm are introduced. Finally, the experimental settings and results are analyzed, and the significance and limitations of the algorithm are summarized and discussed.

2. Problem Statement

The principle of 2D packing problems is to place pieces of known quantity and size on a 2D plate to minimize the consumption of plates and achieve the highest utilization rate. Aline Leao et al. [44] summarized and explained the mathematical model for irregular packing problems, providing a reference for establishing the mathematical model of the packing problem in this paper. Typical heavy industry products are considered research objects in this research. The general requirement is to optimize the cutting of different shapes and sizes on a rectangular steel plate of a certain size according to the packing optimization. Either the lowest number of steel plates is utilized, or the usage rate of each steel plate must be the highest. The width of the plate is usually fixed, and the length of the plate occupied by packing pieces is reduced to improve the utilization rate of the plate. The mathematical model for 2D irregular-piece packing problems can be expressed as follows: Given a plate P of width W, a set group of pieces can be arranged with quantity n as $\{P_1, P_2, P_3 \ldots, P_n\}$. The piece number follows the ordered natural sequence; the objective function of piece packing and the constraints of packing optimization are shown in Formulas (1) and (2), respectively:

$$Max\rho = \frac{\sum\limits_{i=1}^{n} s_i}{WH} \tag{1}$$

$$\text{s.}t.\left\{ \begin{array}{l} P_i \in P \\ P_i \cap P_j = \varnothing \\ i \neq j, i, j \in \{1, 2, \ldots, n\} \end{array} \right\} \tag{2}$$

where S_i is the area of the *i-th* piece, and H is the plate height occupied by the pieces after packing. Here, the maximum utilization rate ρ of the optimization target is equivalent to the minimum total height H of the packing. The first constraint in Formula (2) states that all pieces must be entirely placed within the plate boundaries, while the second constraint states that pieces arranged into the plate boundaries must not overlap. Other constraints, such as plate defects, piece-rotation-angle restrictions, and piece-clearance requirements, may exist in some particular procedures.

3. BL Positioning Strategy Based on NFP

The geometric expression of irregular pieces involves a series of operations, such as saving, moving, rotating, and judging overlap, which are closely related to the efficiency and accuracy of the packing algorithm. Therefore, it is particularly important to choose the appropriate geometric expression method according to the needs of packing optimization. At present, there are various expressions for 2D irregular pieces, including polygon representation [2], envelope methods [45], and grid representation [46]. Polygon representation is widely used because of its relative simplicity, few control parameters, and low level of calculation necessary [21].

3.1. No-Fit Polygon

NFP is a real-shape method that can effectively judge the overlap between pieces, which was first proposed by Art [47] in 1966. There are generally three mainstream methods for generating NFPs: the orbiting algorithm [5,48], the decomposition algorithm [5,10], and the Minkowski sums algorithm [49,50]. The orbiting method is used to generate NFPs in this paper. The process of generating NFPs using the orbiting algorithm is as follows: Two polygons, A and B, are known. Assuming that A is fixed, any point on B is selected as the reference point *Ref*, and B slides tangentially along the outer edge of A until it returns to the original position. The trajectory polygon formed by the movement of the reference point in the sliding process is the no-fit polygon NFP_{AB}, as shown in Figure 1. Then, it can be determined whether polygons A and B overlap according to the positional relationship between the reference point *Ref* and NFP_{AB}. There are three positional relationships between polygons A and B, as illustrated in Figure 2. If the Ref on B is located in NFP_{AB}, A and B are overlapping; if the Ref on B is located on the boundary of NFP_{AB}, A and B are tangential; if the Ref on B is outside the NFP_{AB}, A and B are neither overlapping nor tangential. Therefore, the most reasonable state is that the reference point Ref on B is located on the boundary of NFP_{AB}.

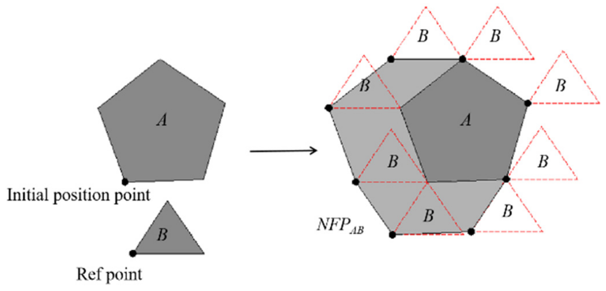

Figure 1. The no-fit polygon NFP_{AB} of two convex polygons, A and B.

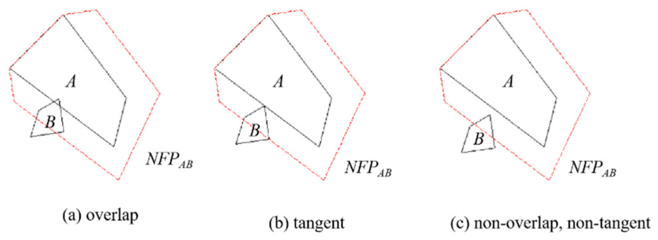

Figure 2. Relative positional relationship between polygons A and B.

3.2. BL Positioning Algorithm

A positioning algorithm determines the placement position and angle of pieces on the plate based on the sequencing optimization of pieces. Here, we calculate the placement of 2D irregular pieces using the classic BL positioning algorithm combined with NFP. The BL algorithm, a classic heuristic positioning algorithm, was proposed by Baker et al. [27]. The main principle of the algorithm is that under its constraints, pieces do not overlap and do not exceed the plate boundary, and pieces are packed into the plate from the upper right corner of the plate with the principle of going down and left as far as possible. When a piece touches other packed pieces, the angle needs to be changed, as given in Figure 3. A group of pieces with a quantity of seven (P_1, P_2, P_3, P_4, P_5, P_6, P_7) is calculated using the BL positioning algorithm and arranged into the plate with the piece number $P_1 \rightarrow P_6 \rightarrow P_4 \rightarrow P_2 \rightarrow P_5 \rightarrow P_3 \rightarrow P_7$.

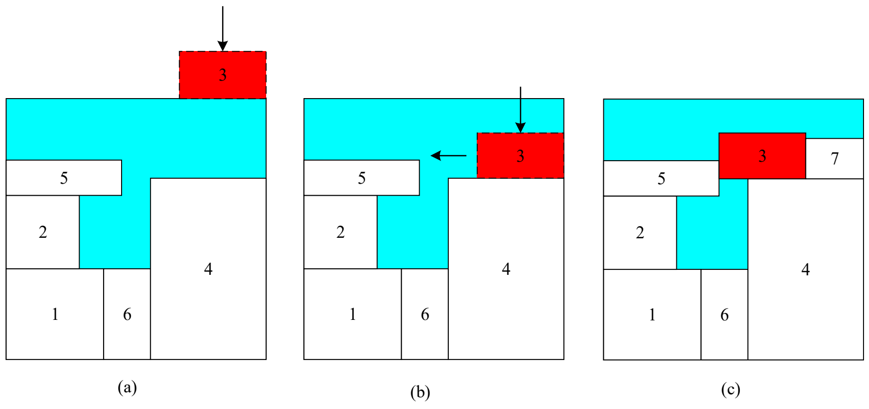

Figure 3. Piece-packing diagram of the BL algorithm. (**a**) Piece P_3 is at the upper right corner of the plate; (**b**) Piece P_3 moves downward; (**c**) Piece P_3 moves to the left.

4. Sequence Optimization Based on RL

RL regards many problems in the field of operational research optimization as sequential decision-making problems and models them as Markov decision-making processes [49]. Figure 4 depicts the fundamental organization of classic RL. In each time step, the agent perceives the environment state s_t and takes action a_t according to a certain strategy. Then, the immediate reward r_t can be obtained by executing a_t, and the environment is changed from state s_t to s_{t+1}. In this section, three hybrid RL methods are proposed for sequence optimization of 2D irregular-piece packing. Combined with the BL positioning strategy based on NFP, 2D irregular-piece packing can be achieved.

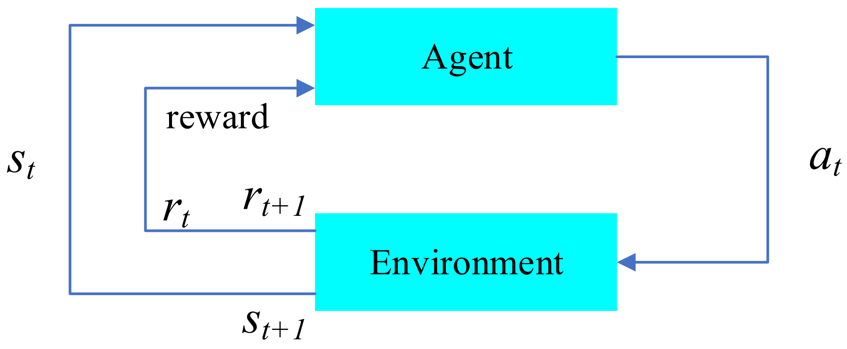

Figure 4. Basic structure of RL.

4.1. Reward-Model Based on Piece Packing

For a packing problem with n pieces, the reinforcement model is unknown, which means that the probability of state transition is uncertain. To acquire the optimal policy, the agent needs to interact with the environment to obtain some episodes, which are then used to evaluate and update the policy. The packing sequence optimization based on hybrid RL can be modeled as a multistage decision process, as follows:

$$s_0, a_1, s_1, r_1, a_2, s_2, r_2 \ldots, a_i, s_i, r_i, \ldots, a_n, s_n, r_n \qquad \pi(s \backslash a)$$

Figure 5 shows a model of the corresponding n-stage Markov decision-making process (MDP). Here, s_0 represents the state when pieces are not arranged, a_i and s_i are actions and states of the *i-th* stage, respectively, $i \in [1, n]$, and r_i is the immediate reward the piece obtains in state s_i at the *i-th* stage. π $(s \mid a)$ is a specific evaluation strategy, such as the greedy algorithm. The next piece selected by the agent in state $s_{i\text{-}1}$ is defined as a_i. After action a_i is completed, the environment state changes from $s_{i\text{-}1}$ to s_i, and $s_i = a_i$ is defined, representing the current piece number. In this paper, the reward r is given after completing a packing, $r_n = C/H \neq 0$ is set, C is a constant, H is the packing height obtained after each piece packing, and the immediate reward is $r_1 = r_2 = r_3 = \ldots = r_{n\text{-}1} = 0$. Therefore, the lower the packing height, the higher the utilization rate of the packing and the greater the reward value returned. In each episode, all pieces to be packed should be traversed once, and the packing should be completed once. That is, each episode contains n loops to select these n pieces. In the study of hybrid RL methods, Q (s, a) is used to express the expectation of the reward that the agent may obtain when taking action a in state s. Generally, Q (s, a) is expressed by the table value. Furthermore, it indicates the effect of selecting the next piece a on the packing height H in state s. In other words, the next piece that makes the packing height smaller can be selected according to the corresponding state information, and a current optimal packing sequence can be obtained according to the update of Q (s, a), which represents the optimal solution of the current piece packing problem.

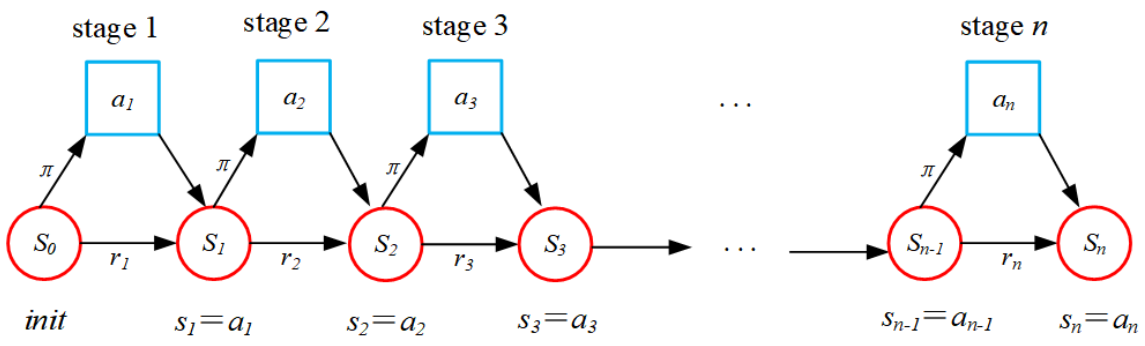

Figure 5. Markov decision-making process for the *n*-stage packing.

4.2. Hybrid RL Methods for Packing Problems

4.2.1. Monte Carlo Reinforcement Learning

The Monte Carlo reinforcement learning (MCRL) method refers to learning the state value directly from experiencing a complete episode (a complete traversal of pieces to be arranged) without knowing the state transition of the MDP, which is consistent with the reward–return strategy set in this paper. Generally, the value of a state is equal to the average of all rewards calculated using the state in multiple episodes. When the current strategy of the agent is to be evaluated, many episodes can be generated using the strategy $\pi(s\,|\,a)$, depicted in Figure 6. Then, the discount–reward–return value at state s in each episode can be calculated as shown in Formula (3). The average reward value can be calculated with two methods: the first visit or every visit. The first visit means that when calculating the value function at state s, only the value returned when state s is visited for the first time in each episode is used, as shown in Formula (4). While calculating the value function at state s, the return value of all visits at state s is utilized, called every visit, as shown in Formula (5). According to the characteristics of piece sequences, we used the first visit method to calculate the value function at state s and replaced the value function with the average reward value through different episodes:

$$R_i(s) = r_i + \gamma r_{i+1} + \ldots + \gamma^{n-1} r_n \tag{3}$$

$$Q(s) = \frac{R_{11}(s) + R_{21}(s) + \ldots}{N(s)} \tag{4}$$

$$Q(s) = \frac{R_{11}(s) + R_{12}(s) + \ldots + R_{21}(s) + \ldots}{N(s)} \tag{5}$$

where s is the state, r_i is the immediate reward of the *i-th* stage, γ is the discount factor, which represents how much the future reward can be observed in the current state, and $R_i(s)$ is the return value of the discounted reward at state s in the *i-th* episode. *Q(s)* is the average reward value at state s, which can help the agent select the next possible action a from the current state s. In other words, the next piece with a smaller packing height can be selected according to the corresponding state information. A current optimal sequence can be obtained according to the continuous update of $Q\,(s, a)$, represented by S_{opt}. The expression of $Q\,(s, a)$ is displayed in Formula (6), where *N(s)* represents the number of the same state–action pairs appearing in multiple episodes. The exploratory MC reinforcement learning (EMCRL) method (that is, the policy π is defined as each trial starting from a random initial state to the termination state) and the on-policy MC reinforcement learning (OMCRL) method (that is, the policy π is defined as using the ε-greedy algorithm for strategy improvement, as shown in Formula (7), where |A(s)| is the number of states, and ε is the probability of random exploration) are adopted to optimize the 2D irregular-piece packing sequence. The total number of episodes is set to m. After each episode, the state-sequence set and action-sequence set of pieces change with the change of the average return value of the reward accumulation, which promotes a change in the piece sequence, and further promotes a change in packing height and raw materials utilization. Therefore, the current optimal sequence after each episode update represents a solution to the 2D irregular-piece packing problem. The optimal packing sequence is changed and replaced towards a smaller packing height with the progress of multiple complete episodes, and the current optimal packing scheme is obtained. Two-dimensional irregular-piece packing based on MCRL is shown in Algorithm 1.

Algorithm 1 MCRL for a 2D irregular packing problem

```
Initialize, for all $s \in n$, $a \in n$, Q table as a matrix of $\mathbf{0}$
Return $(s, a) \leftarrow$ empty list
for $t = 1$ to $m$ do:
    Choose $a_i$ at $s_{i-1}$ according policy $\pi$
    Update $s_i = a_i$, generate an episode using $\pi(a|s)$
    if $i = n$, $H \leftarrow$ piece positioning strategy, then $r_i = C/H$, else $r_i = 0$
    for each pair $(s, a)$:
        $R \leftarrow$ first visit $(s, a)$
        Append $R$ to Returns $(s, a)$
        $Q(s, a) \leftarrow$ average(Returns $(s, a)$)
    end for
    Update $S_{opt}$
end for
Output $S_{opt}$
```

$$Q(s,a) = Q(s,a) + \frac{1}{N(s,a)}(R - Q(s,a)) \tag{6}$$

$$\pi(a|s) \leftarrow \begin{cases} 1 - \varepsilon + \frac{\varepsilon}{|A(s)|}, a = \operatorname{argmax}_a Q(s,a) \\ \frac{\varepsilon}{|A(s)|}, a \neq \operatorname{argmax}_a Q(s,a) \end{cases} \tag{7}$$

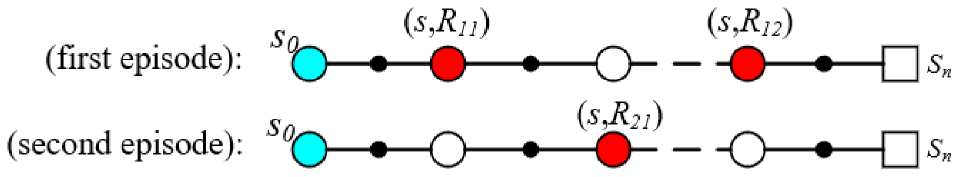

Figure 6. Episodes in Monte Carlo.

4.2.2. Q-Learning and Sarsa-Learning

Q-learning [51] and Sarsa-learning [52] are important components of classic RL, both of which belong to temporal-difference learning (TD learning). Similar to Monte Carlo learning, TD learning also learns from episodes without understanding the model itself, but it can learn incomplete episodes and realize single-step updates. To avoid the current best action falling into a local optimum, a certain probability of exploration in generating state–action pairs is used, and the ε-greedy algorithm is set as π strategy. In addition, Q-learning and Sarsa-learning are off-policy and on-policy algorithms, respectively. Off-policy means that the strategy for generating data is not the same strategy as for evaluating and improving, while the on-policy refers to the strategy for generating data being the same as the strategy for evaluating and improving. Q-learning and Sarsa-learning are updated through continuous interaction with the environment, and the agent automatically learns the action strategy of each step to accumulate the maximum reward. The long-term cumulative reward is represented by the $Q(s, a)$ value table, which guides the packing sequence of the next piece. The updates of Q (s, a) for Q-learning and Sarsa-learning are shown in Formulas (8) and (9), respectively:

$$Q(s,a) = Q(s,a) + \alpha[r(s,a) + \gamma * Max(Q(s\prime,a\prime)) - Q(s,a)] \tag{8}$$

$$Q(s,a) = Q(s,a) + \alpha[r(s,a) + \gamma * Q(s\prime,a\prime) - Q(s,a)] \tag{9}$$

where α is the learning rate, γ is the discounted factor, and s' and a' represent the state and action of the next stage, respectively.

The reciprocal of the packing height is returned as a reward after each complete traversal of the pieces in Q-learning and Sarsa-learning, to guide the new generation of the sequence of pieces. After each episode, the corresponding state sequence $\{s_1, s_2, s_3..., s_n\}$ or action sequence $\{a_1, a_2, a_3..., a_n\}$ represent the sequence solution of a 2D irregular packing problem, which is defined by S_{opt}. With the continuous updates of Q (s, a) and the episode, S_{opt} is continuously replaced by the sequence scheme towards a smaller packing height. Finally, the optimal packing result is obtained. Algorithms 2 and 3 illustrate 2D irregular-piece packing algorithms based on Q-learning and Sarsa-learning, respectively. After sequence optimization with the hybrid RL method, combined with the BL positioning strategy based on NFP, 2D irregular-piece packing can be realized. The process of the algorithm is displayed in Figure 7.

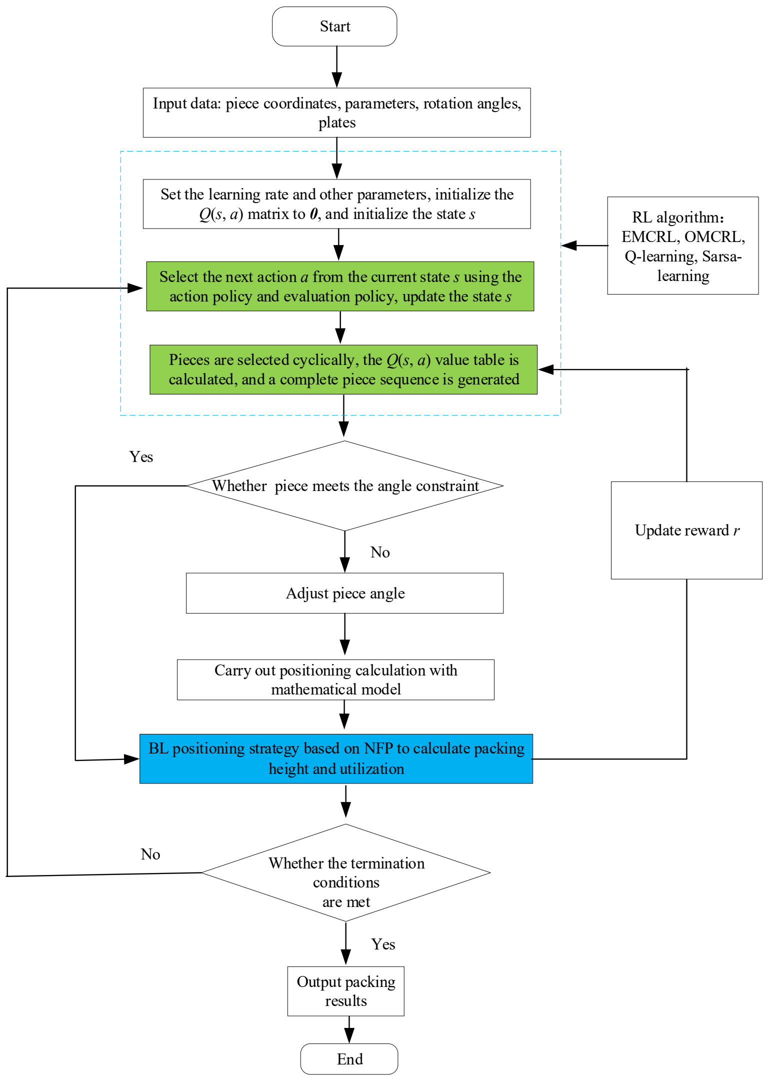
Start
Input data: piece coordinates, parameters, rotation angles, plates
Set the learning rate and other parameters, initialize the Q(s, a) matrix to 0, and initialize the state s
Select the next action a from the current state s using the action policy and evaluation policy, update the state s
Pieces are selected cyclically, the Q(s, a) value table is calculated, and a complete piece sequence is generated
RL algorithm: EMCRL, OMCRL, Q-learning, Sarsa-learning
Yes
Whether piece meets the angle constraint
No
Adjust piece angle
Carry out positioning calculation with mathematical model
BL positioning strategy based on NFP to calculate packing height and utilization
Update reward r
No
Whether the termination conditions are met
Yes
Output packing results
End

Figure 7. The process of the hybrid RL algorithm for 2D irregular-piece packing.

Algorithm 2 Q-learning for a 2D irregular packing problem

```
Initialize Q table as a matrix of $\boldsymbol{0}$
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 $\varepsilon$-greedy policy
            Take $a_i$, enter stage $i$, $s_i = a_i$
            if $i = n$, $H \leftarrow$piece positioning strategy, then $r_i = C/H$, else $r_i = 0$
            Update Q ($s_{i-1}$, $a_i$)
            $s \leftarrow s'$, $a \leftarrow a'$
        end for
    Update $S_{opt}$
end for
Output $S_{opt}$
```

Algorithm 3 Sarsa-learning for a 2D irregular packing problem

```
Initialize Q table as a matrix of $\boldsymbol{0}$
Initialize $S_{opt}$
for $t$ = 1 to $m$ do:
    Initialize $s_0$
    Choose $a_i$ at $s_{i-1}$ according to $\varepsilon$-greedy policy
        for $i$ = 1 to $n$ do:
            Take $a_i$, enter stage $i$, $s_i = a_i$
            Choose $a'_i$ at $s'_{i-1}$ according to $\varepsilon$-greedy policy
            if $i = n$, $H \leftarrow$piece positioning strategy, then $r_i = C/H$, else $r_i = 0$
            Update Q ($s_{i-1}$, $a_i$)
            $s \leftarrow s'$, $a \leftarrow a'$
        end for
    Update $S_{opt}$
end for
Output $S_{opt}$
```

5. Computational Experiments and Discussion

The algorithm was written in Python. Computational tests of hybrid RL methods for 2D irregular-piece packing were performed on a computer with a 2.30 GHz AMD Ryzen 7 3750H CPU with 4 kernels and 8 GB of RAM. To test the performance of the proposed algorithmic model, the method was tested using packing problem instances, which were also used as benchmark problems in other studies. The data file for the test was obtained from the EURO Special Interest Group on Cutting and Packing (ESICUP, https://www.euro-online.org/websites/esicup/data-sets/ accessed on 8 September 2022). The sample information for these data is provided in Table 1. The total number of episodes m was set to 300, the discount factor $\gamma = 1$, and the constant $C = 100$, which is a reasonable parameter setting obtained through many experimental calculations. The random exploration rate ε was 0.1, and the learning rate α was 0.5 [30]. In order to avoid random deviation caused by convergence of the RL method, the packing height H returned after each episode was recorded. Meanwhile, the smaller H between the current minimum packing height and the packing height returned by the last episode was taken as the updated optimal packing height, and the sequence of corresponding pieces was the current optimal packing sequence.

Table 1. Details of the benchmark problems.

Problem Instance	Number of Piece Types	Total Number of Pieces	Rotational Constraints	Sheet Width
Dighe1	16	16	0 absolute	100
Dighe2	10	10	0 absolute	100
Fu	12	12	90 incremental	38
Jakobs1	25	25	90 incremental	40
Jakobs2	25	25	90 incremental	70
Marques	8	24	90 incremental	104
Shapes0	4	43	0 absolute	40
Shapes1	4	43	0, 180 absolute	40
Shapes2	7	28	0, 180 absolute	15
Shirts	8	99	0, 180 absolute	40
Swim	10	48	0, 180 absolute	5752
Trousers	17	64	0, 180 absolute	79

For each case, the packing algorithm based on hybrid RL was run 10 times and partially obtained better packing results compared to the corresponding literature (i.e., the genetic algorithm of bottom-left (GABL) [53], a hybrid nesting algorithm based on heuristic placement strategy and an adaptive genetic algorithm (AGAHA) [54], a simulated annealing hybrid algorithm (SAHA) [55], and the hybrid beam search with tabu search algorithm (BSTS) [21]). The differences in the computing environments of these algorithms are shown in Table 2. The relevant algorithm parameters are described in the corresponding literature. Tables 3 and 4 present the experimental results of these excellent algorithms and the packing results of our hybrid RL algorithm for a 2D irregular packing problem, respectively. Taking the instance as the horizontal coordinate, and the average utilization rate of the plate as the vertical coordinate, the results in Tables 3 and 4 were processed into a curve, as shown in Figure 8, representing the average utilization rate of the plate calculated based on different algorithms for each instance.

Table 2. Computational environments.

Algorithm	Language	CPU	RAM
Hybrid RL	Python	AMD Ryzen 7 3750H CPU at 2.30 GHz	8.0 GB
GABL AGAHA	JavaScript	Intel(R) Celeron(R) CPU at 2.60 GHz	4.0 GB
SAHA	C++	Pentium IV CPU at 2.13 GHz	0.5 GB
BSTS	VS C++	Core 2 D CPU at 2.0 GHz	1 G

Table 3. Packing results in the corresponding literature.

Instance	GABL		AGAHA		SAHA		BSTS	
	Best%	Avg%	Best%	Avg%	Length	Avg%	Length	Avg%
Dighe1	87.31	85.01	88.14	85.13	122	81.97	125.66	79.48
Dighe2	86.92	81.42	88.78	78.94	119.53	83.66	121.12	82.56
Fu	83.82	82.22	84.24	81.36	32.70	87.15	32.99	86.38
Jakobs1	80.46	76.50	81.67	80.94	12.93	75.80	11.98	81.87
Jakobs2	71.24	69.69	73.36	71.48	25.86	74.62	25.84	74.66
Marques	84.14	83.55	84.36	83.12	79.63	86.87	79.94	86.40
Shapes0	57.11	56.59	61.86	60.45	63.15	63.18	66.11	60.38
Shapes1	63.33	63.03	65.41	63.23	58.17	68.59	60.14	65.26
Shapes2	-	-	-	-	26.53	81.41	28.48	75.82
Shirts	71.04	70.65	71.53	70.85	-	-	-	-
Swim	67.61	66.81	68.35	67.76	6121.39	72.27	6369.1	69.46
Trousers	86.43	86.06	86.90	86.19	244.68	89.01	252.07	86.41

Table 4. Results for piece packing of hybrid RL.

Instance	EMCRL		OMCRL		Q-Learning		Sarsa-Learning	
	Best%	**Avg%**	**Best%**	**Avg%**	**Best%**	**Avg%**	**Best%**	**Avg%**
Dighe1	**77.40**	**77.28**	77.02	76.94	70.80	70.38	68.7	68.60
Dighe2	**100.0**	**99.74**	76.20	76.14	77.60	77.52	77.6	77.22
Fu	77.03	76.71	77.50	77.23	**82.06**	**81.97**	78.05	78.05
Jakobs1	**76.12**	75.81	74.56	74.52	75.38	75.21	75.38	75.29
Jakobs2	67.85	67.53	68.20	68.00	64.33	64.26	**68.93**	**<u>68.81</u>**
Marques	**80.43**	**79.48**	77.72	77.61	77.41	77.38	76.94	76.85
Shapes0	56.52	56.36	**57.35**	57.09	**57.35**	**57.19**	56.93	56.83
Shapes1	58.21	58.14	57.35	57.15	**58.65**	**55.36**	58.21	58.09
Shapes2	70.82	70.67	69.68	69.26	68.57	68.39	**71.05**	**70.78**
Shirts	**81.37**	**81.10**	78.71	78.56	73.67	73.54	72.70	72.54
Swim	64.22	64.12	64.41	64.23	64.92	64.53	**65.41**	**65.24**
Trousers	77.46	77.41	**78.09**	**77.84**	77.36	77.24	76.37	76.33

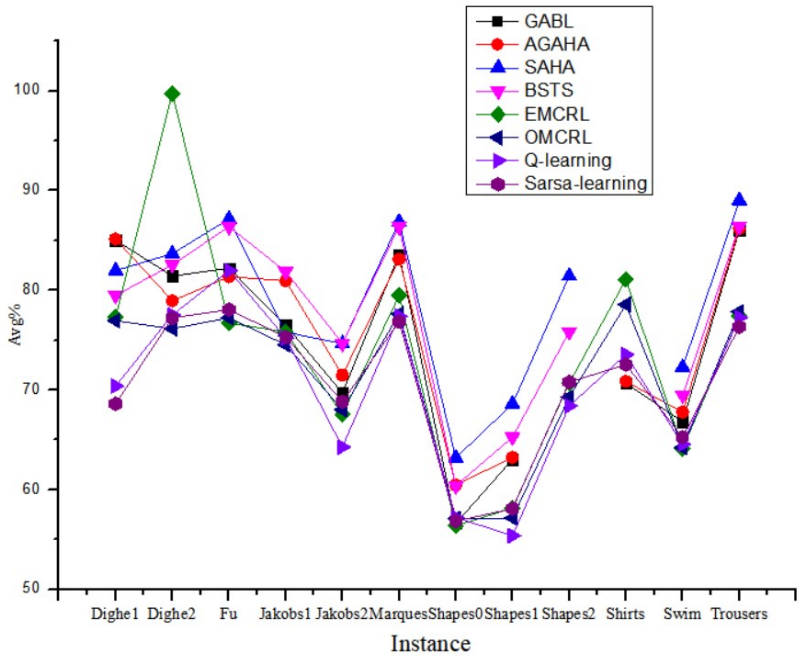

Figure 8. Average plate-utilization rate calculated using different algorithms for each instance.

By analyzing Tables 3 and 4 and Figure 8, it is evident that our packing algorithm based on hybrid RL can achieve improved packing results compared to corresponding studies in some instances. The better results of our hybrid RL algorithm are highlighted in blue. Results below an average of 1% between algorithms are underlined. Moreover, the best results among the four RL algorithms analyzed are shown in bold. For the same instance, the solution results of different algorithms show the same changing trend, while in different instances, the packing results show great volatility, which shows the importance and complexity of algorithm design. In the four classic RL packing calculations based on 12 instances, the EMCRL algorithm has produced the optimal solution for five instances. The OMCRL algorithm and Q-learning algorithm have obtained the optimal solution for two and three instances, respectively, and the same optimal utilization rate of plates has been achieved on the instance based on Shapes0. The calculation based on the Sarsa-learning algorithm also obtained the optimal solution for three instances. In this view, the algorithm based on EMCRL has the strongest optimization among the four classic RL algorithms. Compared with the results of the classic heuristic algorithm in the relevant literature, the EMCRL algorithm has acquired a better solution for three instances

(Dighe2, Jakobs1, Shirts), while the calculations based on the OMCRL algorithm and Q-learning algorithm have obtained a better solution for two instances (Shapes0, Shirts) and three instances (Fu, Shapes0, Shirts), respectively. Although the Sarsa-learning algorithm produced a better solution for only one instance (Shirts) compared with the GABL and AGAHA algorithms, there is a less than 1% result gap for the instance Jakobs2 compared to the GABL method. In other words, compared with the GABL algorithm, AGAH algorithm, SAHA algorithm, and BSTS algorithm, better results for three (Dighe2, Shapes0, Shirts), three (Dighe2, Fu, Shirts), two (Dighe2, Jakobs1) and one (Dighe2) instance can be produced using the packing algorithm based on hybrid RL. Specifically, the EMCRL algorithm can achieve 100% utilization for instance Dighe2, which is approximately 10% higher than the result of the relevant heuristic algorithms (GABL, AGAHA, SAHA, BSTS). This is also the best calculation result among the four RL algorithms. For example, in Shirts the results based on the hybrid RL algorithm exceeds the results of 71.53% plate utilization in the classic heuristic algorithm (GABL, AGAHA). In addition, in most calculation results based on the RL algorithm, the gap between the optimal utilization rate and the average utilization rate is less than 3%, which indicates the algorithm's stability. From the above analysis, it can be seen that the hybrid RL algorithm can obtain comparable results or better results than some classic heuristic algorithms in certain instances. The EMCRL algorithm can produce better solutions, while the Sarsa-learning algorithm produces the least optimal solutions and has the most limited optimization effect. The layout of the best results for five instances of the packing algorithm based on RL can be found in Figure 9.

It can be seen from Table 4 and Figure 8 that the EMCRL algorithm can achieve better optimization results than the other three RL algorithms, which is related to the reward–return setting of the packing. When each piece to be packed is traversed, and the reciprocal of the packing height is used as a reward, it coincides with the reward–return strategy after each complete episode of the EMCRL. The data generation of EMCRL is more diverse than that of OMCRL, which is better for learning the update optimization of the sequence. Compared with Sarsa-learning, the Q-learning algorithm returns better packing optimization results, which is related to its sequence-update strategy. The difference between the two can be observed in Formulas (8) and (9). The decision-making piece is the same, but the global situation is considered by Sarsa-learning when performing actions (that is, the next action will be determined when updating the current Q value). Nevertheless, Q-learning chooses the action with the greatest current benefit every time, regardless of other states, meaning it is greedier. As a result, the Q-learning algorithm is similar to the EMCRL and has more diverse data, therefore it can achieve better learning efficiency and optimization effects. In addition, the hybrid RL algorithm can obtain comparable or better results than some classic heuristic algorithms in certain instances. The possible reasons are that, on the one hand, the reinforcement-learning search based on self-learning can gain a better packing sequence, and an appropriate positioning algorithm can obtain a better packing effect. On the other hand, pieces in different instances have various complexities, and the classic heuristic algorithm is inefficient in solving certain instances. Therefore, research of 2D irregular packing needs further exploration and improvement. There is a difference of more than a 5% result gap between the optimal utilization rate and the average utilization rate in the calculation based on the EMCRL algorithm, which is related to the agent exploration mechanism analyzed above. The use of this exploration mechanism can efficiently find the optimal solution and may lead to greater randomness.

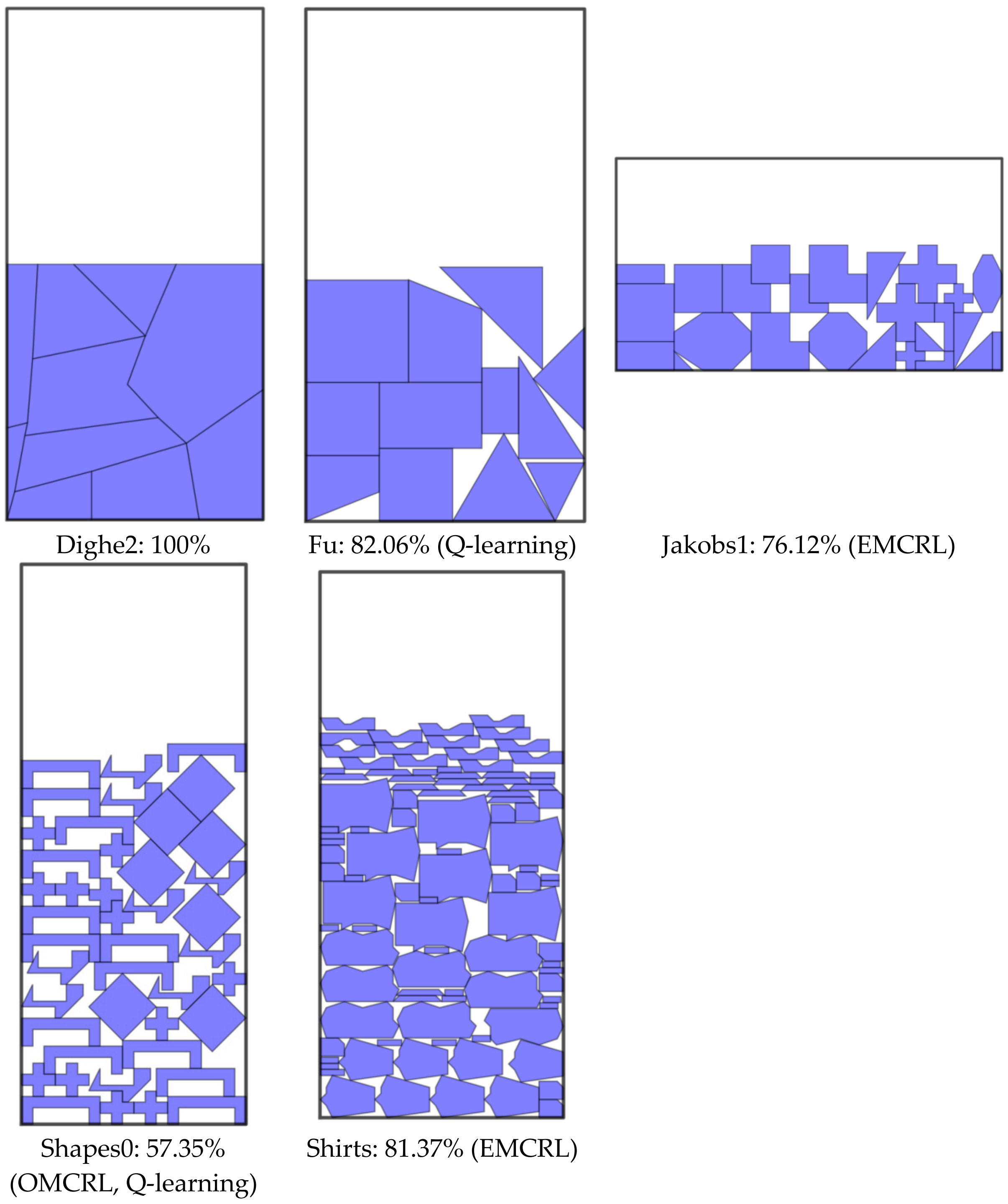

Figure 9. The layout of the best results for five instances of the packing algorithm based on RL.

6. Conclusions

In this work, we establish a mathematical model for 2D irregular-piece packing and propose a novel solution scheme based on a hybrid reinforcement learning algorithm (Monte Carlo learning, Q-learning, and Sarsa-learning) to solve 2D irregular-piece packing problems. The algorithm is an early attempt to solve the 2D irregular packing problem by using the machine learning method. In the solution process, the piece packing sequence is self-learning optimized, and combined with the classic BL positioning algorithm based on NFP, to achieve 2D irregular-piece packing. During this process, mechanisms of reward–return and strategy-update based on piece packing are designed, and policy evaluation strategies of different RL methods are applied. Then, the packing results of different RL algorithms are compared.

The results show that the packing algorithm based on hybrid RL is an applicable and effective algorithm for the irregular packing problem, which can achieve 2D irregular-piece packing in an acceptable time. The proposed algorithm produces five better results and one comparable result within 1% of the average results for each of the 12 benchmark problems. That is, compared with some classic heuristic algorithms, the packing algorithm based on hybrid RL can achieve a partially similar or better optimization effect. At the same time, the EMCRL algorithm can achieve better results than the OMCRL algorithm, Q-learning algorithm, and Sarsa-learning algorithm, which provides a scheme for solving packing problems with different requirements. On the one hand, the packing algorithm based on RL can obtain a better packing utilization rate. On the other hand, the piece sequence to be arranged can be self-learning to reduce the probability of falling into local optimization and improve the reliability and intelligence of packing. These are of positive significance to practical packing applications. Furthermore, this algorithm is an early attempt at reinforcement learning to solve 2D packing problems, which provides a foundation for subsequent deep reinforcement learning to solve packing problems and has far-reaching theoretical significance. However, the update of the sequence of the algorithm depends on the update of the reward table, which needs more time to learn a better sequence. Moreover, there are many parameters in the proposed RL algorithm, which produce various results according to different settings, with certain complexity and uncertainty. Furthermore, for packing problems of a large scale, the search based on RL will cost a lot of time. For packing of pieces with certain defects, the performance of the proposed algorithm may be affected.

In future work, the packing performance based on the hybrid RL algorithm will be explored more deeply according to different packing characteristics, such as pieces with holes and defective plates. In addition, deep reinforcement learning based on neural networks will be investigated further to improve solutions to packing problems caused by variable variety batch production in actual heavy industry production.

Author Contributions: J.F.: conceptualization, methodology, validation, formal analysis, writing—original draft, writing—review and editing; Y.R.: supervision, project administration, writing—review and editing, funding. X.Z.: supervision, writing—review and editing; B.D.: writing—review and editing; All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (grant number. 51975231) and the Fundamental Research Funds for the Central Universities (grant number. 2019kfyXKJC043).

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the referees for their constructive comments that improved the presentation as well as the content of the paper.

Conflicts of Interest: We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, and that there are no professional or other personal interests of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of the manuscript entitled.

References

1. Wang, C.-N.; Yang, F.-C.; Nguyen, V.T.T.; Nguyen, Q.M.; Huynh, N.T.; Huynh, T.T. Optimal Design for Compliant Mechanism Flexure Hinges: Bridge-Type. *Micromachines* **2021**, *12*, 1304. [CrossRef]
2. Li., W. *Research and Application of polygon's Packing Optimization with Multiple Constraints*; Huazhong University of Science and Technology: Wuhan, China, 2016.
3. Liu, Y.; Chu, C.; Wang, K. A new heuristic algorithm for a class of two-dimensional bin-packing problems. *Int. J. Adv. Manuf. Technol.* **2011**, *57*, 1235–1244. [CrossRef]
4. Zhu, S.; Rao, Y. Heuristic Method for One-dimensional Cutting Stock. *Mach. Build. Autom.* **2014**, *43*, 52–55. [CrossRef]

5. Fang, J.; Rao, Y.; Liu, P.; Zhao, X. Sequence Transfer-Based Particle Swarm Optimization Algorithm for Irregular Packing Problems. *IEEE Access* **2021**, *9*, 131223–131235. [CrossRef]
6. Hopper, E.; Turton, B. A genetic algorithm for a 2D industrial packing problem. *Comput. Ind. Eng.* **1993**, *37*, 375–378. [CrossRef]
7. Hopper, E.; Turton, B. An Empirical Investigation of Meta-heuristic and Heuristic Algorithms for a 2D Packing Problem. *Eur. J. Oper. Res.* **2001**, *128*, 34–57. [CrossRef]
8. Bennell, J.A.; Oliveira, J.F. A Tutorial in Irregular Shape Packing Problems. *J. Oper. Res. Soc.* **2009**, *60*, S93–S105. [CrossRef]
9. Roger, B.G.; Tom, M.C. A new algorithm for the minimal-area convex enclosure problem. *Eur. J. Oper. Res.* **1995**, *84*, 522–538.
10. Burke, E.K.; Hellier, R.S.; Kendall, G.; Whitwell, G. Complete and robust no-fit polygon generation for the irregular stock cutting problem. *Eur. J. Oper. Res.* **2007**, *179*, 27–49. [CrossRef]
11. Ranga, P.A.; Julia, A.B.; Antonio, M. Jostle heuristics for the 2D-irregular shapes bin packing problems with free rotation. *Int. J. Prod. Econ.* **2018**, *195*, 12–26.
12. Mundim, L.R.; Andretta, M.; de Queiroz, T.A. A biased random key genetic algorithm for open dimension nesting problems using no-fit raster. *Expert Syst. Appl.* **2017**, *81*, 358–371. [CrossRef]
13. Cherri, L.H.; Mundim, L.R.; Andretta, M.; Toledo, F.M.; Oliveira, J.F.; Carravilla, M.A. Robust mixed-integer linear programming models for the irregular strip packing problem. *Eur. J. Oper. Res.* **2016**, *253*, 570–583. [CrossRef]
14. Dequan, L.; Hongfei, T. An improved BL-algorithm for genetic algorithm of the orthogonal packing of rectangles. *Eur. J. Oper. Res.* **1999**, *112*, 413–420.
15. Guo, B.; Zhang, Y.; Hu, J.; Li, J.; Wu, F.; Peng, Q.; Zhang, Q. Two-dimensional irregular packing problems: A review. *Front. Mech. Eng.* **2022**, *79*, 1–15. [CrossRef]
16. Liu, Q.; Cheng, H.; Tian, T.; Wang, Y.; Leng, J.; Zhao, R.; Zhang, H.; Wei, L. Algorithms for the variable-sized bin packing problem with time windows. *Comput. Ind. Eng.* **2021**, *155*, 107175. [CrossRef]
17. Danyadi, Z.; Foedesi, P.; Koczy, L.T. A bacterial evolutionary solution for three-dimensional bin packing problems using fuzzy fitness evaluation. *Aust. J. Intell. Inf. Process. Syst.* **2011**, *13*.
18. Elkeran, A. A new approach for sheet nesting problem using guided cuckoo search and pairwise clustering. *Eur. J. Oper. Res.* **2013**, *231*, 757–769. [CrossRef]
19. Sato, A.K.; Martins, T.D.C.; Tsuzuki, M.D.S.G. Placement Heuristics for Irregular Packing to Create Layouts With Exact Placements for Two Moveable Items. *IFAC Proc. Vol.* **2013**, *46*, 384–389. [CrossRef]
20. Muhammed, B.; Tansel, D.; Ahmet, C. Robust hyper-heuristic algorithms for the offline oriented/non-oriented 2D bin packing problems. *Appl. Soft Comput.* **2015**, *36*, 236–245.
21. Rao, Y.; Wang, P.; Luo, Q. Hybridizing Beam Search with Tabu Search for the Irregular Packing Problem. *Math. Probl. Eng.* **2021**, *2021*, 1–14. [CrossRef]
22. Queiroz, L.; Andretta, M. A branch-and-cut algorithm for the irregular strip packing problem with uncertain demands. *Int. Trans. Oper. Res.* **2022**, *29*, 3486–3513. [CrossRef]
23. Liu, D.; Tan, K.; Huang, S.; Goh, C.; Ho, W. On solving multiobjective bin packing problems using evolutionary particle swarm optimization. *Eur. J. Oper. Res.* **2008**, *190*, 357–382. [CrossRef]
24. Pinyapod, K. *An Efficient Genetic Algorithm for Rectangular Packing Problem*; National Taipei University of Technology: Taipei, Taiwan, 2014; pp. 1–53. [CrossRef]
25. Lin, T.D.; Hsu, C.C.; Hsu, L.F. Optimization by Ant Colony Hybrid Local Search for Online Class Constrained Bin Packing Problem. *Appl. Mech. Mater.* **2013**, *311*, 123–128. [CrossRef]
26. Zhang, C.; Li, P.; Guan, Z.; Rao, Y. A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem. *Comput. Oper. Res.* **2007**, *34*, 3229–3242. [CrossRef]
27. Baker, B.S.; Coffman, J.E.G.; Rivest, R.L. Orthogonal Packings in Two Dimensions. *SIAM J. Comput.* **1980**, *9*, 846–855. [CrossRef]
28. Chazelle. The Bottomn-Left Bin-Packing Heuristic: An Efficient Implementation. *IEEE Trans. Comput.* **1983**, *C-32*, 697–707. [CrossRef]
29. Jukka, J. A thousand ways to pack the bin-a practical approach to two-dimensional rectangle bin packing. 2010. Available online: http://pds25.egloos.com/pds/201504/21/98/RectangleBinPack.pdf (accessed on 28 November 2022).
30. Fang, J.; Rao, Y.; Guo, X.; Zhao, X. A reinforcement learning algorithm for two-dimensional irregular packing problems. In Proceedings of the 2021 4th International Conference on Algorithms, Computing and Artificial Intelligence, Sanya, China, 22–24 December 2021.
31. Weichert, D.; Link, P.; Stoll, A.; Rüping, S.; Ihlenfeldt, S.; Wrobel, S. A review of machine learning for the optimization of production processes. *Int. J. Adv. Manuf. Technol.* **2019**, *104*, 1889–1902. [CrossRef]
32. Wang, C.-N.; Yang, F.-C.; Nguyen, V.T.T.; Vo, N.T.M. CFD Analysis and Optimum Design for a Centrifugal Pump Using an Effectively Artificial Intelligent Algorithm. *Micromachines* **2022**, *13*, 1208. [CrossRef]
33. Kara, A.; Dogan, I. Reinforcement learning approaches for specifying ordering policies of perishable inventory systems. *Expert Syst. Appl.* **2018**, *91*, 150–158. [CrossRef]
34. Zhang, C.; Dellaert, N.P.; Zhao, L.; Van Woensel, T.; Sever, D. *Single Vehicle Routing with Stochastic Demands: Approximate Dynamic Programming*; Tsinghua Univ. Beijing, China, Tech. Rep, 2013, 425. Available online: https://pure.tue.nl/ws/portalfiles/portal/3567910/391671830622358.pdf (accessed on 19 September 2022).

35. Kong, W.; Liaw, C.; Mehta, A.; Sivakumar, D. A new dog learns old tricks: RL finds classic optimization algorithms. In Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
36. Wang, C.B.; Zhang, X.Y.; Zou, Z.Q.; Wang, S.B. On Path Planning of Unmanned Ship Based on Q-Learning. *Ship Ocean Eng.* **2018**, *47*, 168–171.
37. Laterre, A.; Fu, Y.; Jabri, M.K.; Cohen, A.S.; Kas, D.; Hajjar, K.; Dahl, T.S.; Kerkeni, A.; Beguir, K. Ranked Reward: Enabling Self-Play Reinforcement Learning for Combinatorial Optimization. *arXiv* **2018**, arXiv:1807.01672. [CrossRef]
38. Browne, C.B.; Powley, E.; Whitehouse, D.; Lucas, S.M.; Cowling, P.I.; Rohlfshagen, P.; Tavener, S.; Perez, D.; Samothrakis, S.; Colton, S. A Survey of Monte Carlo Tree Search Methods. In *IEEE Transactions on Computational Intelligence and AI in Games*; IEEE: Piscataway, NJ, USA, 2012; Volume 4, pp. 1–43.
39. Wang, S.; Sun, C.; Zhou, B. Q-Learning Based Dynamic Single Machine Scheduling. *J. Shanghai Jiaotong Univ.* **2007**, *41*, 1227–1232. [CrossRef]
40. Zhao, X.; Rao, Y.; Fang, J. A reinforcement learning algorithm for the 2D-rectangular strip packing problem. *J. Phys. Conf. Ser.* **2022**, *2181*, 012002. [CrossRef]
41. Bello, I.; Pham, H.; Le, Q.V.; Norouzi, M.; Bengio, S. Neural Combinatorial Optimization with Reinforcement Learning. *arXiv* **2016**, arXiv:1611.09940. Available online: https://arxiv.org/abs/1611.09940 (accessed on 25 April 2021).
42. Hu, H.; Zhang, X.; Yan, X.; Wang, L.; Xu, Y. Solving a New 3D Bin Packing Problem with Deep Reinforcement Learning Method. *arXiv* **2017**, arXiv:1708.05930. [CrossRef]
43. Duan, L.; Hu, H.; Qian, Y.; Gong, Y.; Zhang, X.; Xu, Y.; Wei, J. 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, Montreal, QC, Canada, 13–17 May 2019.
44. Leao, A.A.; Toledo, F.M.; Oliveira, J.F.; Carravilla, M.A.; Alvarez-Valdés, R. Irregular packing problems: A review of mathematical models. *Eur. J. Oper. Res.* **2020**, *282*, 803–822. [CrossRef]
45. Błażewicz, J.; Hawryluk, P.; Walkowiak, R. Using a tabu search approach for solving the two-dimensional irregular cutting problem. *Ann. Oper. Res.* **1993**, *41*, 313–325. [CrossRef]
46. Baldacci, R.; Boschetti, M.A.; Ganovelli, M.; Maniezzo, V. Algorithms for nesting with defects. *Discret. Appl. Math.* **2014**, *163*, 17–33. [CrossRef]
47. Art, R. An Approach to the Two -Dimensional Irregular Cutting Stock Problem. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA; p. 1966.
48. Bennell, J.A.; Xiang, S. A comprehensive and robust procedure for obtaining the nofit polygon using Minkowski sums. *Comput. Oper. Res.* **2008**, *35*, 267–281. [CrossRef]
49. Li, Z.; Milenkovic, V. Compaction and separation algorithms for non-convex polygons and their applications. *Eur. J. Oper. Res.* **1995**, *84*, 539–561. [CrossRef]
50. Pankaj, K.A.; Eyal, F.; Dan, H. Polygon decomposition for efficient construction of Minkowski sums. *Comput. Geom. Theory Appl.* **2000**, *21*, 39–61. [CrossRef]
51. Dayan, P.; Watkins, C.J.C.H. Q-learning. *Mach. Learn.* **1992**, *8*, 279–292. [CrossRef]
52. Sprague, N.; Ballard, D.H. Multiple-Goal Reinforcement Learning with Modular Sarsa(0). In Proceedings of the 18th international joint conference on Artificial intelligence, Acapulco, Mexico, 9–15 August 2003.
53. Du, B.; Guo, X.; Fang, J.; Wang, P.; Rao, Y. A hybrid solving algorithm on two-dimensional irregular graphics nesting problem. *Forg. Stamp. Technol.* **2022**, *47*, 39–45. [CrossRef]
54. Xu, X.B.; Li, Z.P. Research Progress and Prospects for Application of Reinforcement Learning in Operations Research. *Oper. Res. Manag. Sci.* **2020**, *29*, 227–239.
55. Gomes, A.M.; Oliveira, J.F. Solving Irregular Strip Packing problems by hybridising simulated annealing and linear programming. *Eur. J. Oper. Res.* **2006**, *171*, 811–829. [CrossRef]