Available online at www.sciencedirect.com

ScienceDirect

Advances in Space Research 74 (2024) 6366–6383

ADVANCES IN SPACE RESEARCH (a COSPAR publication)

www.elsevier.com/locate/asr

An autonomous ore packing system through deep reinforcement learning

He Ren, Rui Zhong *

School of Astronautics, BeiHang University, Beijing, China

Received 9 August 2023; received in revised form 28 December 2023; accepted 31 January 2024
Available online 3 February 2024

Abstract

In the contemporary era, the limited availability of terrestrial resources has prompted an increasing number of nations to turn their attention towards space, wherein extraterrestrial minerals hold considerable allure for both resource provisioning and scientific inquiry. Numerous nations have initiated the deployment of unmanned operational platforms towards extraterrestrial asteroids with the objective of accomplishing sampling return missions. Given the restricted storage capacity and energy limitations of these platforms, the optimization of mineral placement algorithms assumes paramount importance in enhancing the efficacy of these missions. In this paper, we propose an autonomous ore packing system capable of autonomously measuring ore characteristics and addressing the ore packing optimization problem in extraterrestrial unfamiliar environments. A deep reinforcement learning method that utilizes physical constraints to enhance overall performance was proposed to solve the ore packing problem with uncertainty, while meeting the high demands for real-time performance in the mission. To augment the autonomy and adaptability of our approach, we leverage advanced visual technology to transform the spatial distribution of bin utilization and the characteristics of the ore into a matrix representation. This empowers the robotic system to autonomously perceive bin information and capture essential ore features. The empirical findings substantiate that our algorithm attains a human-level performance in the majority of instances, rendering an approximate optimal solution within a concise timeframe. Additionally, we introduce a novel reinforcement learning training technique known as Maximum Worth Reinforcement Learning (MWRL) to address the optimization conundrum associated with incomplete Markov chains, which outperforms existing approaches in our comparative analysis. Lastly, we validate the efficacy of our algorithm in real-world scenarios by deploying it on a robotic manipulator.

Keywords: Extraterrestrial mining; Deep reinforcement learning; Ore placement optimization; Onboard autonomy

1. Introduction

It is well-understood that natural resources are scarce, and the mineral resources are becoming increasingly difficult to find and extract profitably. Outer space contains a vast amount of resources that offer virtually unlimited wealth to the human that can access and use them for commercial purposes (Howell and Spencer, 1986). The necessity to consider the feasibility of resource extraction in new location is paramount to our continued prosperity. As technology is rapidly expanding in the field of space exploration and the prospect of mining on the moon, asteroids, and other celestial bodies looms nearer. Numerous nations have launched unmanned operational platforms to extraterrestrial asteroids to accomplish ore sampling return missions (Qiao et al., 2012). Automatic mining technology in an unmanned environment has become a current research hotspot (Zhang et al., 2018).

* Corresponding author.

https://doi.org/10.1016/j.asr.2024.01.061
0273-1177/

Robotic platforms employed in extraterrestrial mining operations are typically non-manipulable by human operators. These robots autonomously procure minerals (assuming solid ores with unpredictable geometries in this investigation) and proficiently allocate them within storage bins. The primary objective of the robot is to maximize mineral acquisition in order to optimize storage bin occupancy and enhance overall operational efficiency (Zhang et al., 2022). A crucial hurdle in facilitating extraterrestrial mining lies in achieving robotic autonomy. Operating within an entirely unexplored environment, the robot is tasked with acquiring the ability to perceive its surroundings, evaluate storage bin occupancy, analyze ore properties, and make informed decisions regarding the optimal placement of each individual piece of ore. Additionally, given the constraints of limited storage capacity and energy resources available to the mining robot, the implementation of an effective ore packing optimization algorithm holds significant potential in enhancing the overall operational efficiency of the robot. However, in contrast to the conventional terrestrial packing optimization problem, the unique working conditions inherent to extraterrestrial ore packing present a formidable challenge. The inherent randomness and disorderliness of the shape and size of the future ore to be collected, coupled with the robot's limited knowledge confined to the specific details of the grabbed or target ore, engender a pervasive uncertainty throughout the entire process, thereby augmenting the intricacy of the packing problem and rendering its optimization more arduous.

Extraterrestrial exploration has significance for scientific research, human civilization, and space resource mining (Xu et al., 2007). These samples provide valuable insights into the composition of asteroids, serving as a foundation for future endeavors in space mining of rare metals. There are several asteroid sample return missions (SRMs). The Japan Aerospace Exploration Agency (JAXA) successfully launched Hayabusa 1 and Hayabusa 2 missions, targeting asteroid 25,143 Itokawa and asteroid 1999JU3 Ryugu, respectively (Yurimoto et al., 2011; Yuichi Tsuda and Yoshikawa, 2013; Yamaguchi et al., 2018). These missions were significant as Hayabusa 1 marked the first successful asteroid sample-return mission. The United States National Aeronautics and Space Administration (NASA) also launched the OSIRIS-Rex mission, aimed at collecting samples from asteroid 101,955 Bennu and returning them to Earth in September 2023 (Sankaran et al., 2013). Additionally, the European Space Agency (ESA) has planned an asteroid sample return mission to acquire samples from asteroid 2008 EV5 (Ren and Shan, 2014), while the China National Space Administration (CNSA) proposed the ZhengHe mission to explore near-Earth asteroid 469,219 Kamo'oalewa (Zhang et al., 2021). Space exploration can not only reveal the mysteries of the universe, but also promote technological innovation and economic prosperity.

Overall, the successful execution of extraterrestrial mining tasks hinges upon the development of an autonomous intelligent system capable of autonomously extracting ore characteristic information, analyzing storage bin occupancy, and making informed placement decisions based on these factors. The packing process can be likened to the intricate challenge of arranging irregular objects within a designated space, akin to the complex problem of irregular cutting and packing. The irregular cutting and packing problem is a widely recognized combinatorial optimization challenge with diverse applications across various industries, including automotive, apparel, furniture, metalworking, shipbuilding, and aerospace manufacturing (Tole et al., 2023). It falls into the category of NP-hard problems, combining the intricacies of cutting and packing with the complexities of accommodating irregular convex and non-convex objects (Rodrigues and Toledo, 2017). The objective of the packing problem is to arrange a collection of objects within a rectangular bin without any overlaps, with a primary focus on minimizing raw material wastage or maximizing space utilization. The conservation of raw materials holds significant value, contributing to both economic and environmental performance within society, and consequently leading to substantial profitability growth in the manufacturing sector.

Numerous scholars have conducted research on the irregular bin packing problem. Albano and Sapuppo (1980) proposed an novel approach that transforms the packing problem into a search process through candidate solutions. This classical irregular packing algorithm has inspired much subsequent research. Toledo et al. (2013) developed a dotted-board model that discretizes the bin with points and achieved good performance on many instances. Martinez-Sykora et al. (2017) used a mixed-integer model that allows items to be continuously rotated. Similarly, Chernov et al. (2010) also permit continuous rotation of the irregular pieces. Terashima-Marín et al. (2010) proposed a hyper-heuristic algorithm for the 2D-IBPP that combines several packing heuristics and adds the possibility of rotating pieces by a finite set of angles. Rodrigues and Toledo (2017) proposed a clique covering model to reduce the number of constraints and improve the linear relaxation. Their model has achieved optimal solutions for up to 25 pieces, subject to grid discretization.

One limitation in existing research on the irregular bin packing problem is the common assumption of deterministic data, which can lead to solutions that do not reflect real-world scenarios(Queiroz and Andretta, 2022). The algorithms based on such assumptions certainly cannot be deployed on extraterrestrial mining systems. At the same time, the methods used in the above generally take a long time to obtain a solution, which also does not meet the requirements of real-time and fast solution for extraterrestrial mining. Recent advances in AI, particularly Deep Reinforcement learning(DRL) shows great potential in combinatorial optimization problems (Sun et al., 2022; Li et al., 2022), in some cases such as Alpha Go (Silver et al., 2016), robotic control (Lan et al., 2023), achieving human-level performance. DRL has attracted a lot of

research interest and has excellent performance in many problems (Hildebrandt et al., 2022; Hou and Li, 2023). This encourages a few scholars to apply DRL to figure out bin packing problem, and they have achieved good results. Kundu et al. (2019) proposed a data-driven vision-based algorithm with DRL to solve the bin packing problem, which does not rely on prior information. Zhao et al. (2022) proposed a constrained deep reinforcement learning method to solve the online 3D bin packing problem. Their algorithm reduces invalid exploration in training by using a mask predictor to predict feasibility, but it does not consider all orientations of items to be packed. Jiang et al. (2023) composed reinforcement learning and constraint programming to solve the 3D bin packing problem. Their method uses a multimodal encoder to mitigate computation and achieves good performance in comparison studies. Tian et al. (2023) used deep reinforcement learning with a sequence-to-sequence policy network to solve the multi-vehicle cooperative 3D bin packing problem. They used a Bi-LSTM network to predict the packing positions, and the learned policy improved the average space utilization by 4% compared to traditional methods. However, most studies involving reinforcement learning in packing problems can be found in 2D or 3D regular bin packing problems, with few publications considering the online irregular bin packing problem.

Reinforcement learning is a powerful approach that utilizes Neural Networks (NN) to represent the control system, making it well-suited for tackling sequential decision-making problems such as extraterrestrial mining. However, the safety–critical nature of space missions introduces practical challenges when implementing onboard DRL systems. Issues related to the accuracy and explainability of the NN models can obscure potential risks to the overall success of the mission. (Golmisheh and Shamaghdari, 2023). To solve this issue, it is crucial to acknowledge that numerous physical and biological systems possess extensive pre-existing knowledge and physical constraints that remain untapped in contemporary machine learning methodologies (Raissi et al., 2018). By using prior information and physical constraints as a regularization machine, the solution space can be constrained to a smaller size, accelerating the convergence rate and making the algorithm more robust. However, to date, only a limited body of research has explored the integration of NN with physical constraints. There definitely exists a wealth of empirical information and physical constraints in the ore packing process that can be utilized to enhance the neural network training.

In response to these challenges, we have developed an autonomous ore packing system, which addresses the requirements for autonomy and stability in extraterrestrial planetary environments. Our system consists of two main components: environment perception and measurement, and an intelligent packing algorithm. In the first part, the robot uses hand-eye cameras to explore the environment and obtain the feature status of the storage bin and grabbed ore. The system autonomously create a No-fit matrix M_{No-fit} as the state representation. This approach effectively enhances the learning efficiency and generalization performance for the ore packing algorithm. Furthermore, we have devised a robust method called action mask that effectively identify and reject potentially hazardous actions. In the ore packing algorithm, we adopt an encoder-decoder diagram to learn the packing policy, where a encoder which consists of bidirectional LSTM layers to extracts the key feature, and a decoder which is equipped with an improved context-based attention mechanism is responsible for constructing solution. Incorporating insights from practical extraterrestrial mining processes, we have analyzed two distinct working modes, as illustrated in Fig. 1 **a**. The first mode, referred to as Ore Packing for 2 steps (OP-2s), involves the robot having access to information about both the next ore and the currently caught one. Since the Markov decision process (MDP) is complete in this mode, we employ the Proximal Policy Optimization (PPO) (Schulman et al., 2017) algorithm to train the system effectively. In the second mode, known as Ore Packing for 1 step (OP-1s), the agent only possesses information regarding the currently caught object, while the next ore to be collected remains uncertain. Consequently, the incompleteness of the MDP arises from the uncertainty of the next ore. To train the agent, we discretize the complete packing process into individual decision-making steps and propose a new training method called maximum worth reinforcement learning (MWRL) method. The MWRL leverages the new framework and achieves better solution in the experiments. Finally, we deploy our algorithm in the simulation environment and physical environment to verify its feasibility, as shown in Fig. 1 **b**. The contributions of this article are summarized as follows.

1. We propose an autonomous intelligent ore packing system capable of performing exploration, measurement, and autonomous optimization tasks for ore placement in complex environments. The system is deployed and validated on a physical platform, demonstrating its successful completion of all task requirements while ensuring safety performance.
2. We introduce a novel reinforcement learning method called MWRL to address optimization problems with incomplete Markov chains and uncertainties. Additionally, we incorporate physical constraints into the algorithm structure, effectively improving the generalization performance and accelerating the convergence speed.
3. We explore the use of a hand-eye camera for measuring irregular objects and storage bins. The measured information is fed into the network, and an improved context-based attention mechanism is employed for feature extraction. The effectiveness of this attention mechanism is verified through ablation experiments.

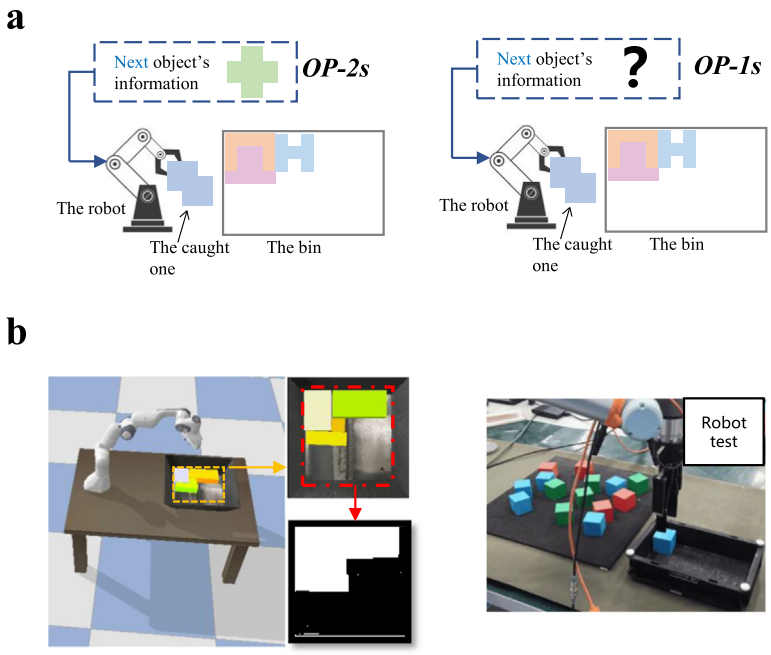

Fig. 1. The two working modes and experimental verification of the algorithm. **a**, The two working modes which commonly used in extraterrestrial mining processes: In the first mode, referred to as OP-2s, the agent has access to information about both the next object and the currently caught one; In the second mode, referred to as OP-1s, the agent only has access to information about the currently caught object. **b**, we verified the feasibility of the system in the simulation environment and the real platform. The robot uses the hand-eye camera to obtain the status of bins and objects, and inputs the image information into the network, and the algorithm gives a placement plan.

The organization of this paper is arranged as follows: The 1st section introduces the paper. The 2nd section and 3rd section gives the details of the method. The computational experiments and physical experiments are reported in Section 4. Finally, the conclusions of our paper are given in Section 5.

2. Problem definition

In this paper, we explore an autonomous ore packing system designed for extraterrestrial environments. The system entails the robot autonomously evaluating storage bin occupancy, detecting ore information, and selecting an optimal location for each collected ore. Taking into account various constraints such as volume, weight, energy limitations, and security, we assume the storage bin to have a cuboid shape. To prevent the instability of the platform caused by ore shaking, once the first layer of the storage bin is filled, the robot returns to the base to unload the ore. Let $O = \{o_1, \ldots, o_n\}$ be the set of n objects (for the convenience of description, use object to refer to ore) represented as various shapes to be packed into storage bins. The robot utilizes a camera to measure both the storage bin and the object to be packed, allowing it to obtain the feature status. These measurements are subsequently mapped to a 2-dimensional matrix space, resulting in the matrix representations M and m, as illustrated in Fig. 2. Hence, each object has limited rotations of $(0°, 90°, 180°, 270°)$. Since the robot's task involves filling

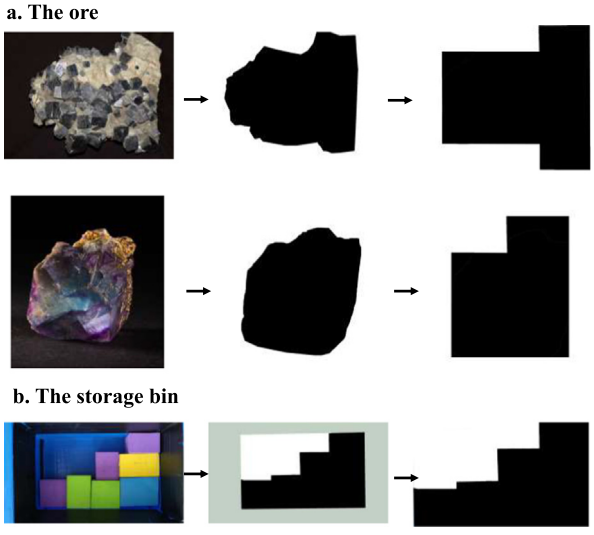

Fig. 2. Measurement method.

only one layer of the storage bin, the efficiency of the packing process can be evaluated by measuring the area utilization of the bin. We denote the area of object $o_i \in P$ by α_i, and the relationship between area α and its matrix representation m is $\alpha_i = \sum_i \sum_j m_i(i,j)$. In this paper, it is often necessary to sum all the elements of the matrix. In order to simplify the representation, we use $\sum$ to represent the sum of all the elements of the matrix, that is, $\alpha_i = \sum m_i$.

The objects need to be placed entirely within the bin and cannot overlap with other objects.

2.1. The Model

In ore packing problem, the goal is to pack as many objects as possible into a fixed-size bin with length L and width W (represented as M), where the maximum dimensions of an object are denoted as (l, w), and the number of unpacked objects is unlimited. A solution to the ore pecking problem is a packing sequence $P = \{p_1, \ldots, p_n\}$, where each $p_i(c_i, po_i, m_i)$ for $i = 1, \ldots, n$ consists of pack coordinates $c_i = [x_i, y_i]$ (representing the X and Y coordinates of the reference point of the object, with the origin at the upper left corner of the bin), an orientation angle po_i, and the matrix representation m_i which contains area and shape information. Here, the reference point of an object $o_i \in O$ is defined as the upper-left corner of the enclosing rectangle of the object. The objective is to maximize the packing density (PD), defined in Eq. 1, of the bin.

$$PD = \frac{\sum_{i=1}^{n} \alpha_i}{L \times W} \tag{1}$$

s.t.

$$\begin{aligned} &\forall p_i \in P, i = 1, \ldots, n; \\ &x_i + l_i \leqslant L \\ &y_i + w_i \leqslant W \\ &\max\left(M[x_i : x_i + l, y_i : y_i + w_i] + m_i\right) \leqslant 1 \end{aligned} \tag{2}$$

where the objective function Eq. 1 aims to maximum the material utilization of the bin, while constraints Eq. 2 guarantee that all the objects are packed into a bin without exceeding its capacity, and no object in the bin overlap with each other.

2.2. Notation

The MDP is a formal framework used to model decision-making problems in which an agent interacts with an environment. It is typically represented as a tuple (S, A, P, R), where S is a finite set of environment states, A represents the set of possible actions that the agent can take, $P: S \times A \rightarrow S\prime$ is the transition function that defines the probability of moving from state s to $s\prime$ given the action a taken by the agent, and $R: S \times A \rightarrow r$ is the reward function that specifies the reward received by the agent for taking an action a in state s.

2.3. Maximum worth reinforcement learning

To apply reinforcement learning methods to the ore packing problem, the Markov chain representing the packing process must be complete. In the bin packing problem, the environment state s can be expressed as $s = [M, m]$, consisting of two parts: the bin state and the object to be packed. The agent takes an action based on the state s, leading to the next state $s\prime = [M\prime, m\prime]$. The entire process described above constitutes a complete MDP. If any part is unclear or missing, the process becomes incomplete, making it difficult to train the algorithm effectively, since all reinforcement learning methods rely on the next state $s\prime$ to calculate the loss in the update step. For example, in the PPO algorithm (Schulman et al., 2017), the next state $s\prime$ is required to compute the advantage function $\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \cdots + (\gamma\lambda)^{T+1-t}\delta_{T-1}$, where $\delta t = r_t + \gamma V(s_{t+1}) - V(s_t)$.

In our problem, two working modes which are most likely to occur in extraterrestrial mining process are considered. In OP-2s, after the agent takes an action to place the caught object into the bin, the occupancy of the bin will be updated, and the information of the next object is clear, so the next state $s\prime$ is clear. The MDP in OP-2s is complete. Therefore, we adopt The PPO method to train our agent, the objective of the stochastic policy is to maximize the expected sum of the discount reward.

$$J(\pi_{\mathrm{OP-2s}}) = \sum_{t=0}^{T} \mathbb{E}_{(s_t, a_t)\sim\tau}[\gamma^t r(s_t, a_t)] \tag{3}$$

where $\gamma \in [0, 1]$ is the discount factor and $\tau = (s_0, a_0, s_1, \cdots)$ is a trajectory samples based on the policy π .

In the case of OP-1s, where the uncertainty of the next object leads to an incomplete Markov chain, estimating the cumulative discounted reward becomes challenging. Traditional reinforcement learning methods are ill-suited for training in such an environment. To address this challenge, we have developed a novel approach called MWRL to assist the agent in learning the packing policy. In MWRL, the agent does not consider the effect of the current action on the future but strives to maximize the reward for each decision at every step. Maximizing the reward at each step implies that each decision of the agent should make the ores in the storage bin more compact and the overall layout of the storage more orderly. Therefore, we reformulate each single packing step as a MDP. Our approach aims to maximize the value of each action the agent takes in every interaction with the environment. The agent needs to learn how to achieve the best layout of the bin at various states. The objective function of MWRL is written as follows.

$$J(\pi_{\mathrm{OP-1s}}) = \mathrm{E}_{(s_t, a_t)\sim\rho_\pi}[r(s_t, a_t)] \tag{4}$$

3. The autonomous ore packing system

This section provides a comprehensive overview of the autonomous ore packing system. The system is comprised of two key components: environment perception and measurement, and an intelligent ore packing algorithm. In the first part, we delve into the utilization of a hand-eye camera

to capture and measure the distinctive features of both the storage bin and the object. This process involves extracting relevant information from the camera images to gain insights into the spatial characteristics and dimensions of the bin and object. In the second part, we present the network structure, algorithm composition, and training process of our intelligent ore packing system. We elucidate how these components work in synergy to facilitate effective packing decisions. This includes a detailed explanation of the neural network architecture and the step-by-step training process to optimize the system's performance.

3.1. Environment perception and measurement

Automation is the key to extraterrestrial mining missions. The efficient utilization of robots in extraterrestrial mining operations necessitates the integration of a dedicated measuring tool capable of assessing the ore and the storage bin to determine the optimal packing location. However, the conventional use of precision instruments for measurement significantly hampers packing efficiency due to time-consuming processes. In this study, we propose a novel approach that exclusively employs a hand-eye camera to detect crucial features, enabling the robot to swiftly react and work with heightened efficiency.

Considering the significant shaking of the ore during the robot's movement when placing multiple layers, there is a risk of platform overturning. To mitigate this risk, this study adopts the assumption that the robot returns to the base for unloading once the storage bin is fully packed on a single layer. This assumption also provides the necessary conditions for utilizing only a hand-eye camera for measuring the storage bin.

Both the bin and the ores are scanned using a hand-eye camera mounted on the robotic arm. The system need the measurement of ore shapes and sizes, as well as the detection of bin occupancy. For accurate size determination of the ores, we utilize a reference object of known dimensions within the image. Subsequently, the image is rasterized based on the reference object, creating a binary matrix where each element is either 0 or 1. The size of the ore is then determined by analyzing the position of the '1' element in this matrix. This process can be summarized into the following three distinct steps.

First, the system captures photographs of the ore and bin from a top-down perspective. Next, the captured images are transformed into binary images based on the contrast between foreground and background colors. In the final step, the system rasterizes the binary image with respect to a known-length reference object present in the image. This reference object, which can be the bin itself, serves as a known size reference. By establishing the unit standard length of this reference object, we can accurately determine the size of the ore by analyzing the ratio between the ore and the reference object. This process is visually represented in Fig. 2.

Moreover, the flexibility of this method is worth noting. By adjusting the standard size of the reference object, we can enhance the accuracy of the matrix representation, enabling more precise approximations of the ore's shape and size. The smaller the unit standard length we set, the more precise the system's measurements become, as depicted in Fig. 3. This concept is akin to adjusting the resolution of photographs, where a smaller unit standard length corresponds to a higher 'resolution'. However, it's essential to consider that higher 'resolution' in the matrix representation demands more computational resources. Given the constraints of computational resources on the spacecraft during asteroid exploration, selecting an appropriate 'resolution' becomes imperative.

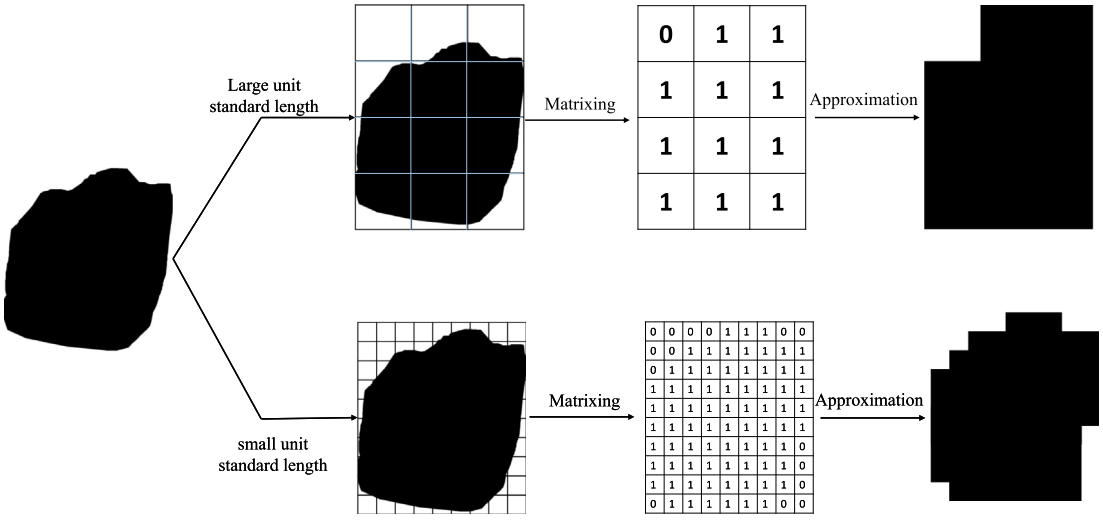

Fig. 3. Choosing different unit standard length of the reference object can bring different approximate results. Using small unit standard length can get a more accurate approximation.

After obtaining the binary matrix representation of the storage bin and the ore, there is no need for further conversion to determine the actual size of the ore. The system seamlessly inputs this matrix into the neural network, which autonomously extracts the necessary key features of both the bin and the ore. Subsequently, the algorithm generates the optimal ore placement based on these key features. This measurement process closely mirrors how humans interpret visual information through their eyes. Notably, our system operates at a significantly higher speed compared to conventional measurement methods, resulting in a substantial increase in packing efficiency. A flowchart of the complete process is shown in Fig. 4.

3.2. The ore packing algorithm

Most traditional algorithms for irregular bin packing problem assume complete knowledge of all object information and focus on mathematical modeling and numerical simulation. However, they are not well-suited for real-time optimization tasks with uncertainty, such as extraterrestrial mining. In this paper, we propose an online placement optimization algorithm that combines reinforcement learning, enabling the agent to make optimal decisions based on partial information.

3.2.1. State representation based on prior information

The environment state of the Ore Packing Problem consists of two parts: the current configuration of the bin and the information of the coming object. We denote the matrix representation of the storage bin as M_i, where the subscript denotes the next item to be packed, and the matrix representation of the objects as m_i. Together, the current environment state can be expressed as $s_i = \{M_i, m_i\}$, where both information matrices contain shape and size information.

However, it is challenging for the algorithm to establish the potential connection between the two parts of the current environment state $s_i = \{M_i, m_i\}$ if we simply input it into the neural network. This can lead to a decrease in the effectiveness of the training process. Through observation, we discovered that important prior information is concealed within the state s_i, which can expedite the training process. One geometric constraint of the packing problem is that each object must be fully placed inside the board and cannot overlap with any other objects. We found that the sum of the Hadamard product $\circ$ of the m_n and the block matrix of the placement location in the M_i can indicate whether the object can be placed at that location without overlapping. By computing the Hadamard product for each position in the bin, we obtain a matrix with the same size as the M_i, which we refer to as the no-fit matrix (NFM), also called prior information-based matrix. The equation for computing the NFM is shown in Eq. 5.

$$M_{No-fit}[x_i, y_i] = \frac{\sum (m_i \circ M_i[x_i : x_i + h_n, y_i : y_i + w_n])}{\sum m_i} \tag{5}$$

where the m_i is the matrix represent of the object to be packed, $M_i[x_i : x_i + h_n, y_i : y_i + w_n]$ is the block matrix of the placement location in the M_i, and $\circ$ is Hadamard Product. In order to simplify the formula, we use $\sum$ to represent the sum of the each element of the matrix in this paper. To reduce the variance of the input data, we normalize the NFM by dividing it by the sum of the m_i matrices. If $M_{No-fit}[x_i, y_i] = 0$, it indicates that the position $[x_i, y_i]$ in the bin is a feasible position for the object, while values other than zero denote overlapping or being out-of-bounds. In other words, the NFM matrix reveals which positions in the bin can accommodate the object without violating the constraint that it should be entirely placed inside the board and cannot overlap with other objects. Finally, the state is defined as $s_i = [M_{No-fit}, M_i]$

3.2.2. Action space and action mask

In the problem under investigation in this study, it is necessary to determine the coordinates of the objects in the storage bin in both the x-direction and y-direction in order to determine their placement position. Assuming

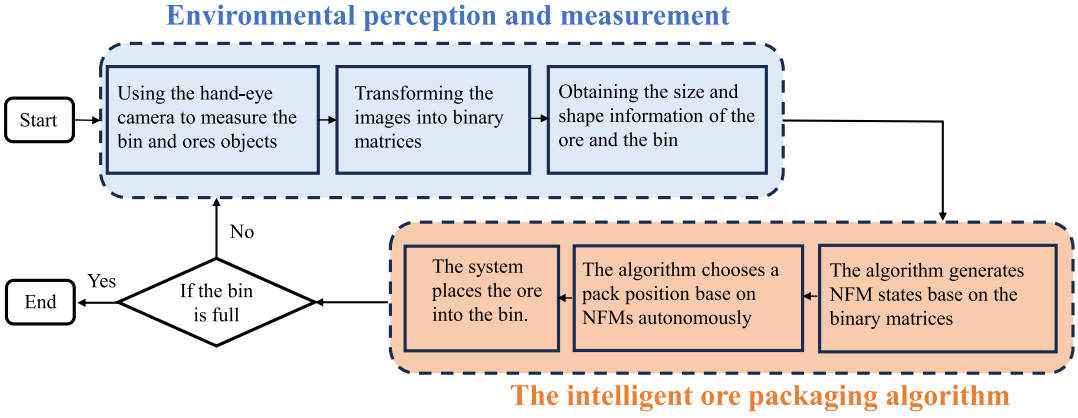

Fig. 4. The flowchart of the complete work process of the system.

the length of the storage bin is L and the width is M, with the x-axis along the length and the y-axis along the width, the dimension of the action space is $(L \times W)$. Considering the rotation, the total action space is $(4 \times L \times W)$ The large action space results in significant computational complexity, increased difficulty in finding the optimal solution, and a higher likelihood of aggressive actions, which is unacceptable for space missions. To reduce the action space and improve algorithm efficiency, we utilize the physical constraint information contained in the Non-Fit Matrix M_{No-fit}. By determining the y-coordinate, we can find feasible positions in the NFM along the x-axis. To maximize packing density, we select the solution closest to the bottom of the storage bin. As a result, the dimension of the action space decreases from $(4 \times L \times W)$ to $(4 \times W)$. This approach ensures solution accuracy while significantly accelerating algorithm execution.

In space missions, ensuring stability and safety is of paramount importance in designing control methods. Reinforcement learning, as an end-to-end control approach, poses challenges in terms of interpretability and stability analysis, which are difficult to be formally proven. In this study, to ensure that the robot avoids dangerous actions, we propose a method called Action Masking, which filters out unsafe actions from the action space. Specifically, we identify behaviors that lead to object overlap or exceeding the boundaries of the storage bin as hazardous, and the robot should avoid such behaviors. The Action Masking method generates a masking vector based on the NFM, which contains information about constraints related to object overlap and boundary violation. By examining the existence of safe positions for each action in all directions along the x-axis, the feasibility of the action can be determined. In our code, V_{mask} can be constructed by checking whether each element in the i^{th} column of the NFM, corresponding to action a_i, is zero or not. If the element is non-zero, action a_i is infeasible, and if it is zero, action a_i is worth exploring. This is expressed in Eq. 6. The Action Mask method also greatly improves the exploration efficiency and accelerates the train process.

$$V_{mask}[i] = \begin{cases} 0 & if\ any\ \left(M_{No-fit}[:\quad i]\right) \neq 0 \\ 1 & otherwise \end{cases} \tag{6}$$

3.2.3. reward function

The primary objective of our algorithm is to achieve maximum packing density. To this end, we have designed a reward function that incentivizes the agent to place objects in close proximity to each other. Moreover, the reward is further augmented if the agent can orient the objects in a way that they fit together snugly. We express the reward function in Eq. 7.

$$\begin{aligned} r_i = &-\eta \times r_{gap} - \gamma \times (l_i + w_i + 1 - \textstyle\sum m_{outer} + \sum m_i) \\ &-\delta \times r_{ori} - \varphi \times x_i + \kappa \times \left(\textstyle\sum m_{ref} - \sum m_i\right) \end{aligned} \tag{7}$$

where the term r_{gap} indicates whether the action generates a gap in the packing arrangement, where $r_{gap} = 1$ means the action generates a gap and $r_{gap} = 0$ means it does not. The variable $m_{outer} = M_i[x_i - 1 : x_i + 1 + h, y_i - 1 : y_i + 1 + w_i]$ represents a sub-matrix of the occupancy matrix M_i surrounding the position (x_i, y_i), and $l_i + w_i + 1 - \sum (m_{outer}) + \sum (m_i)$ is a measure of the proximity of the object to its neighboring objects. The term r_{ori} evaluates whether the packing orientation is suitable for embedding the object, taking the value of either 0 or 1, and x_i in the equation encourages the algorithm to choose a lower position for the object. The variable $m_{ref} = M_i[x_i : x_i + l_i, y_i : y_i + w_i]$ represents a sub-matrix of the occupancy matrix M_i that corresponds to the space that the object would occupy if it were placed in position (x_i, y_i), and $\sum (m_{ref}) - \sum (m_i)$ indicates whether the object fits in that space. The weight coefficients $\eta, \gamma, \delta, \varphi$, and κ are used to balance the contribution of these terms.

3.2.4. Neural network framework

The neural network framework is based on the actor-critic architecture, which is illustrated in Fig. 5. The actor learns the packing policy and outputs the probability of packing, while the critic estimates the expected value of the current state. The framework comprises two primary components: the encoder and the decoder. To reduce the network's size, the actor and critic share the encoder structure.

In the encoder, two embeddings are used to map the inputs $\left[M_{No-fit}, M_i\right]$ into a $\mathcal{D}$-dimensional space. For the bin matrix M_i, the overall layout of the bin needs to be learned, which is achieved by a two-dimensional convolution with m kernels, followed by the extraction of bin feature vector with a multi-layer perceptron. This process is expressed as $V_{BFV} = \tanh\left(W_{mlp} \cdot [F_{Conv2d}(M_i)] + b_{mlp}\right)$. For the M_{No-fit}, as the algorithm needs to weigh the pros and cons of different actions and the action space is evenly distributed on the y-axis, it is processed by a one-dimensional convolution along the x-axis direction, which is defined as follows:

$$V_{OFVs} = \tanh\left(W_O \cdot [u_1; \cdots; u_i; \cdots; \mathrm{u}_n] + b_O\right) \tag{8}$$

Here, $u_i = F_{Conv1d}\left(M_{No-fit}[:\quad i]\right)$, where ";" denotes the concatenation of two vectors.

In the decoder, the critic is responsible for estimating the expected value, therefore, we use bi-LSTM layers to capture the correlation between the objects and the bin, as shown in Eq. 9.

$$\begin{aligned} f_i &= \sigma\left(W_f \cdot \left[V_{BFV}; v_{OFVs}^{i-1}\right] + b_f\right) \\ i_i &= \sigma\left(W_i \cdot \left[V_{BFV}; v_{OFVs}^{i-1}\right] + b_i\right) \\ \widetilde{C}_i &= \tanh\left(W_c \cdot \left[V_{BFV}; v_{OFVs}^{t}\right] + b_C\right) \\ o_i &= \sigma\left(W_o \cdot \left[V_{BFV}; v_{OFVs}^{i-1}\right] + b_o\right) \\ h_i &= o_i \cdot \tanh\left(f_t \cdot V_{BFV} + i_t \cdot \widetilde{C}_t\right) \end{aligned} \tag{9}$$

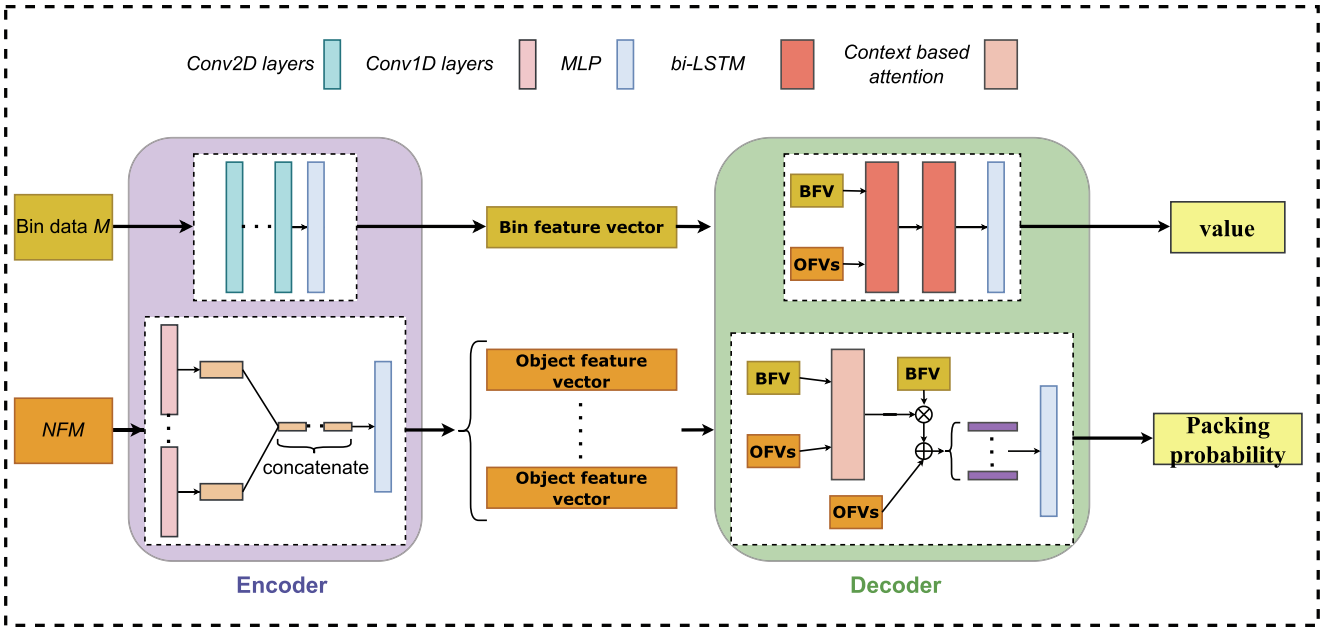

Fig. 5. The neural network. In this diagram, data vectors are depicted as horizontal rectangles, while neural network layers are illustrated as vertical rectangles. The key layers employed within our framework are highlighted and labeled above, with distinct colors indicating their respective types.

where $f_i, i_i, \widetilde{C}_i, o_i$, and h_i are intermediate variables produced by the bi-LSTM layers. The sigmoid function σ is used, and h_i is the output of the bi-LSTM layers, which is used to estimate the expected value of the current state by the critic.

An attention mechanism is a differentiable structure used for learning the packing policy in the actor component. Specifically, we employ a improved context-based attention mechanism that extracts relevant information between the bin and the object.

$$a_i = \mathrm{softmax}(u_i) \quad \text{where} u_i = v_a^T \tanh\left(W_a\left[V_{BFV}; v_{OFVs}^i\right]\right) \tag{10}$$

We compute the conditional probabilities by combining the context vector c_i, computed as

$$c_i = a_i v_{BFV}^i \tag{11}$$

with the context vector c_i, and then we get the packing probability by normalizing the output with the softmax function, as follows:

$$p_i\left(v_{OFVs}^i | s_i\right) = \mathrm{softmax}(\tilde{u}_i), \quad \text{where}\, \tilde{u}_i = v_c^T\left(\left[V_{BFV}; v_{OFVs}^i\right]\right) \tag{12}$$

In Eq. 10-Eq. 12, v_a, W_a and v_c are trainable variables

3.2.5. Train process

Although both situations have different objective functions, we train the algorithms in the same framework. We use the PPO training method to train the algorithm: the robot interacts with the environment first to generate trajectories. After the interaction is over, we use the collected trajectories for training and employ the importance sampling method to train offline, which improves the utilization rate of experience. The training details are demonstrated below.

OP-2s

In OP-2s, the Markov chain is complete. We consider a parameterized state value function $V_\phi(s_t)$ and a state-action value function $Q_\theta(s_t, a_t)$, The parameters of these networks are $\phi_{OP-2s}, \theta_{OP-2s}$. The state value function $V_\phi(s_t)$ is trained to minimize the squared residual error

$$J_V(\phi_{OP-2s}) = \mathbb{E}_{s_t \sim \mathcal{T}}\left[\frac{1}{2}\left(V_{\phi OP-2s}(s_t) - \sum_{i=0}^{N} \gamma^i r_i\right)^2\right] \tag{13}$$

where $\mathcal{T}$ is a length-N trajectory segment of the agent, concluding at the final state of the interaction. The state value function $V_\phi(s_t)$ need to estimate the baseline of every state. Then the advantage estimation is generalized:

$$\begin{aligned}\widehat{A}_{OP-2s}(t) &= -V_{\phi_{OP-2s}}(s_t) + r_t + \gamma r_{t+1} + \cdots \\ &+\gamma^{N-t+1} r_{N-1} + \gamma^{N-t} V_{\phi_{OP-2s}}(s_N) \\ &= \delta_t + \gamma\delta_{t+1} + \cdots + \gamma^{N-t+1}\delta_{N-1}\end{aligned} \tag{14}$$

where $\delta_t = r_t + \gamma V_{\phi_{OP-2s}}(s_{t+1}) - V_{\phi_{OP-2s}}(s_t)$.

OP-1s

In OP-1s, The Markov chain is incomplete. We formulate the process as a single-step Markov decision process and use MWRL method to train the agent. The objective of the agent is to maximize the worth of each action. Let the $\phi_{OP-1s}, \theta_{OP-1s}$ are the parameters of the state value function and the state-value value function. The state value function is trained as below:

$$J_V(\phi_{OP-1s}) = \mathbb{E}_{s_t \sim \mathcal{D}}\left[\frac{1}{2}\left(V_{\phi_{OP-1s}}(s_t) - \mathbb{E}_{(s_t, a_t) \sim \rho_\pi}[r(s_t, a_t)]\right)^2\right] \tag{15}$$

Where $\mho$ is the experience pool used to store training data, and the advantage estimation is expressed as:

$$\hat{A}_{OP-1s}(t) = r_t - V_{OP-1s}(s_t) \tag{16}$$

In both situation, the parameters of the policy network are updated as follows.

$$L_t^{CLIP}(\theta) = \min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}\hat{A}(t), clip\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}, 1-\epsilon, 1+\epsilon\right)\hat{A}(t)\right) \tag{17}$$

where the $\epsilon = 0.2$ is a hyperparameter.

With the above symbols defined, we present the learning process of the OP-1s and OP-2s in Algorithm 1 and Algorithm 2.

In OP-2s, the agent interacts with the environment for a fixed number of time steps T and generates trajectory segments. Next, we calculate the advantage estimates $\widehat{A}$ based on these T timesteps of data and optimize the algorithm using minibatch SGD for K epochs.

The main difference between OP-1s and OP-2s is that in OP-1s, the advantage estimate can be calculated at each time step, allowing for more frequent training of the neural network. To facilitate this, we create an experience pool $\mho$ where training data is stored and can be randomly sampled to train the agent. This approach significantly reduces correlation between training data, enabling valuable data to be retained and used multiple times.

Algorithm 1. The train process of OP-2s

Algorithm 1: The train process of OP-2s

Input: The environment and the agent
Output: The parameters of the neural network
initialization $\phi_{OP-2s}, \theta_{OP-2s}$;
for *Each iteration* **do**
  for $t = 1, 2, \cdots, T$ **do**
    $a_t \sim \pi_{\theta_{OP-1s}}$;
    generate next state $s' = \left[N'_{No-fit}, M'\right]$;
  end
  Compute the advantage estimates $\hat{A}_1, \cdots, \hat{A}_t$;
  Update ϕ_{OP-2s} by the $\hat{\nabla}_{\phi_{OP-2s}} J_V(\phi_{OP-2s})$ with K epochs and minibatch size M;
  Update θ_{OP-2s} by the $\hat{\nabla}_{\theta_{OP-2s}} L^{CLIP}(\theta_{OP-1s})$ with K epochs and minibatch size M;
end

Algorithm 2. The train process of OP-1s

Algorithm 2: The train process of OP-1s

Input: The environment and the agent
Output: The parameters of the neural network
initialization $\phi_{OP-1s}, \theta_{OP-1s}$;
for *Each iteration* **do**
  for $t = 1, 2, \cdots$ **do**
    $a_t \sim \pi_{\theta_{OP-1s}}$;
    Compute advantage estimates $\hat{A}_t$;
    $\mho \leftarrow \mho \cup \left\{\left(s_t, a_t, r(s_t, a_t), \hat{A}_t\right)\right\}$;
  end
  for *each gradientstep* **do**
    $\phi_{OP-1s} \leftarrow \phi_{OP-1s} - \hat{\nabla}_{\phi_{OP-1s}} J_V(\phi_{OP-1s})$;
    $\theta_{OP-1s} \leftarrow \theta_{OP-1s} - \hat{\nabla}_{\theta_{OP-1s}} L^{CLIP}(\theta_{OP-1s})$;
  end
end

4. Experiment

In this section, we present a variety of experimental results to validate the feasibility of the system. Firstly, we demonstrate the superiority of the online packing optimization algorithm by showcasing training results, conducting ablation experiments, comparative experiments, and analyzing generalization performance. Secondly, we evaluate the overall performance and feasibility of the system through experiments conducted in both a simulation environment and a physical platform. All simulations are performed on an AMD Ryzen 7 CPU with a Nvidia 2060 GPU. The robotic arm employed in the physical experiments is the 2L6_4L3 experimental robotic arm, and the hand-eye camera used is the Realsense D435.

4.1. Hyperparameter initialization and training setup

We trained the algorithm on a rectangular bin with dimensions $L \times W$, where L and W are determined based on selected unit standard length. During the training process of OP-2s and OP-1s, the next object is randomly sampled from the library, and all hyperparameters in the neural network are set to the same value. At each iteration, the agent packs as many objects as possible into the bin until the maximum packing density is a maximum. For the training data, we designed the shapes and sizes of the objects to closely simulate the randomness encountered in extraterrestrial mining processes. To demonstrate the intelligence of the system, we create two sample libraries with different unit standard length. The sample library 1 with large unit standard length, containing 20 ore objects, is used to illustrate the training effect of the neural network and perform ablation experiments, as shown in Fig. 14 in the Appendix. We can observe that there are still regular rectangular objects present in this sample library. There are two reasons for this. First, the main contribution of the regular-shaped objects is helping train a better policy with good generalization. Although we train the agent to learn how to pack irregular objects, the more samples the agent encounters, the better the generalization performance the policy. Second, the presence of small rectangular objects can help the agent to fill the bins as completely as possible, making it easier to estimate the expected sum of the discounted reward used in OP-2s. It's essential to clarify that the regular-shaped objects are exclusively used for training the neural network and do not feature in the test experiments. In the sections dedicated to comparative studies and generalization tests, all ore samples used in experiments are of irregular shape. The sample library 2 has a small unit standard length and comprises 7 ore objects. This library serves to validate the system's performance in complex, real-world scenarios, as depicted in Fig. 15 in the Appendix. You can notice that the objects in this library exhibit even more irregular shapes than real ores. The tests conducted with this library demonstrate the system's capability to handle complex situations effectively.

4.2. The training process of the OP-2s and OP-1s

The learning curves for both situations are presented in Fig. 6. In each training step, we record the average reward of those interactions as the score of that train step. From the curves, we can see that both MWRL and PPO methods learn an effective policy and eventually converge. Interestingly, the MWRL algorithm used in OP-1s achieves better performance than the PPO method used in OP-2s. We speculate that there are two main reasons for this. First, the MWRL algorithm constructs a replay buffer similar to DQN to store experiences, which reduces the correlation between experiences and improves their utilization. Second, the drawback of random sampling is that the expected

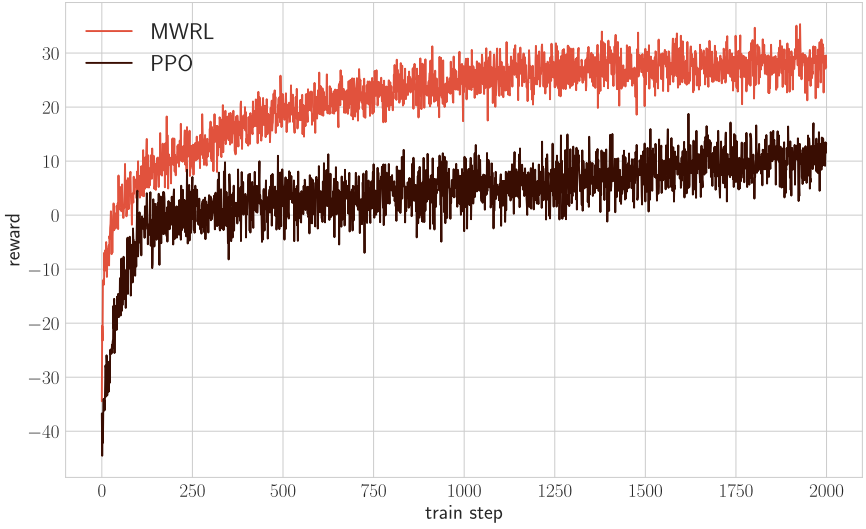

Fig. 6. The learning curves of the MWRL and PPO methods.

sum of discounted rewards is difficult to estimate, making it challenging to calculate the advantage function.

4.3. Ablation experiment

To validate the importance of the no-fit matrix M_{No-fit} and our improved context-based attention mechanism, we conducted a set of ablation experiments to test the impact of these structures on the algorithm. The four experiments consisted of the complete MWRL neural network, the neural network with the no-fit matrix M_{No-fit} removed, the neural network with the improved context-based attention mechanism removed, and the neural network with both structures removed. The results are demonstrated in Fig. 7.

From the results, we can clearly see that both the no-fit matrix M_{No-fit} and the improved context attention mechanism have a great impact on the algorithm, especially the M_{No-fit}. We can observe that the absence of M_{No-fit} makes the agent learn a confused policy, regardless of the presence of an attention mechanism. This indicates that the no-fit matrix M_{No-fit} can help the neural network extract the key features precisely, improving the training speed greatly. Additionally, we can see that the improved context-based attention mechanism can significantly enhance the policy. The ablation experiment confirms the importance of these structures.

4.4. Comparison study

In the comparison study, we compared our reinforcement learning-based online irregular ore packing algorithm with existing irregular bin packing algorithms. To our knowledge, few online irregular bin packing methods that obey uncertain demand constraints exist, we selected two algorithms for comparison. The first is the dotted-board heuristic algorithm proposed by Rodrigues and Toledo (2017), which discretizes the bin and represents positions in the bin with dots, similar to our method. The dotted-board heuristic algorithm has been proven to provide optimal layout solutions for commonly used irregular bin packing instances. The second algorithm is a heuristic search method proposed by Albano and Sapuppo (1980), which is a classic irregular bin packing algorithm often used for comparison studies. The two method are exact algorithm which require full information about all objects to be packed. We also compared our method with several commonly used reinforcement learning algorithms, namely Double DQN(van Hasselt et al., 2015), Dueling DQN (Wang et al., 2016), and Q-learning. All the reinforcement learning algorithms employed the same neural network architecture and hyperparameters. However, except for the PPO and MWRL methods, the other reinforcement learning algorithms did not utilize the Action Mask method, and all reinforcement learning method only know local information about the packing sequence. We trained our agent using the two sample libraries separately because altering the unit standard length leads to changes in the size of the state matrix representation. We observed the calculation speed, packing density, and overall system performance in libraries with varying unit lengths.

For the sample library 1 with large unit standard length, We conducted a total of eight sets of comparative experiments using different ore samples. For each category of experiments, we use the given objects to randomly generate assembly sequences, conduct 100 simulations, and calculate the average packing density in the 100 simulations. The results are shown in Table 1.

In Table 1, the instance names briefly describe the conditions of the objects used in each experiment. For example, in S-5–1, 'S-5' indicates that five types of objects with different shapes are used in the experiment,'1' is the experiment label.The DDQN1 is the double DQN method and DDQN2 is the dueling DQN method. From the results, it is evident that our method (MWRL) consistently outper-

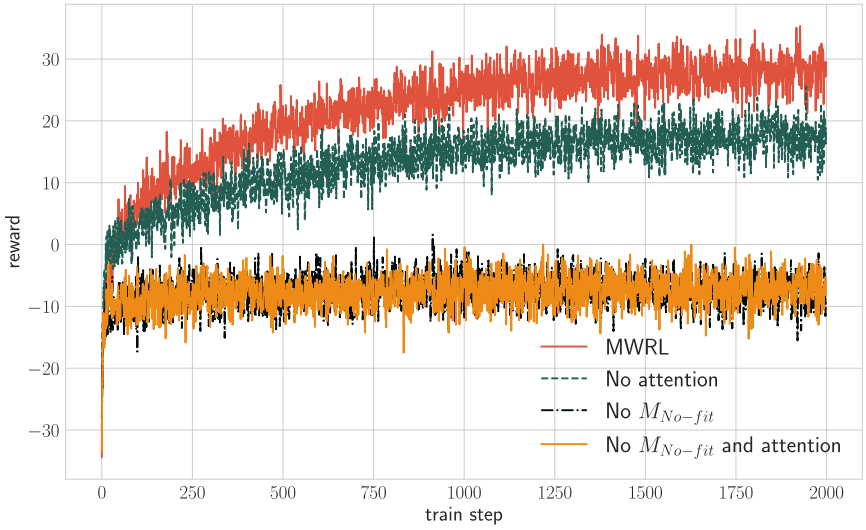

Fig. 7. The ablation experiments.

Table 1
The compare result with library 1.

Instance	Toledo	Albano	Q learning	DDQN1	DDQN2	PPO	MWRL
S-5–1	71.22%	66.19%	48.73%	59.27%	61.38%	72.89%	**74.93%**
S-5–2	77.37%	66.75%	49.24%	62.31%	65.36%	72.54%	**78.69%**
S-5–3	83.41%	79.15%	54.49%	74.77%	72.35%	79.36%	**85.23%**
S-5–4	73.93%	64.95%	49.36%	63.87%	62.16%	70.33%	**74.94%**
S-10–1	73.35%	67.33%	46.45%	66.92%	68.74%	74.31%	**75.36%**
S-10–2	77.50%	69.29%	50.82%	68.46%	69.97%	77.25%	**78.55%**
S-15–1	75.83%	68.48%	45.56%	69.02%	67.29%	72.31%	**78.65%**
S-20–1	77.73%	70.61%	46.11 %	66.90%	72.39%	73.77%	**79.20%**

Table 2
The compare result with library 2.

Instance	Toledo	Albano	Q learning	DDQN1	DDQN2	PPO	MWRL
E1	51.86%	50.97%	39.77%	45.89%	43.79%	53.17%	**60.74%**
E2	47.94%	46.46%	39.41%	50.49%	48.89%	51.35%	**61.41%**
E3	42.76%	40.25%	43.15%	48.16%	49.43%	56.71%	**64.96%**

forms other algorithms in terms of average packing density. In comparison to Q-learning, DDQN1, DDQN2, and PPO, MWRL increases the average packing density by 60.06%, 17.70%, 15.91%, and 5.53%, respectively. While there is a modest 2.19% increase in average packing density compared to the algorithm proposed by Toledo, it's important to note that the average decision time of the MWRL method is only 0.53 s, whereas Toledo's method has a decision time of 16.27 s.

For the second sample library with a small unit standard length, we conducted three sets of comparative experiments using storage bins of varying sizes: $40 \times 60, 60 \times 60$, and 80×60. In each category of experiments, ore objects were randomly selected from library 2, and the system's goal was to maximize the filling of the bin. We conducted 100 simulations for each category and calculated the average packing density across these simulations, as presented in Table 2. The results demonstrate that the MWRL method consistently outperforms the other six methods in terms of packing density. Notably, we observed that the packing density of the MWRL method increased with the size of the storage bin. It achieved a 31.25% higher packing density compared to Toledo's method. These comparative experiments in sample library 2 underscore the system's capability to tackle more challenging tasks. For a more visually intuitive presentation, we computed the overall average packing density based on all the experiments conducted with the two sample libraries. This data is depicted in Fig. 8.

Safety and stability are fundamental requirements for space missions. In this study, we have devised an Action Mask approach to eliminate hazardous actions within the action space, ensuring the reliability of the algorithm. To validate the effectiveness of the Action Mask method, we have conducted the aforementioned comparative experiments and recorded the occurrence of dangerous actions generated by each algorithm during the experimental process. We consider actions resulting in overlap or exceeding the boundaries of the storage bin as hazardous. The statistical results are illustrated in Fig. 9. From the results, it is evident that due to the utilization of Action Mask method in the PPO and MWRL methods, the experiment did not encounter any dangerous actions, similar to the two exact algorithm. However, the other three reinforcement learning methods, which did not impose constraints on the action space, were susceptible to taking actions that posed risks to the platform.

Our algorithm is based on an intelligent approach and is focused on solving online irregular bin packing problems with uncertain demands. The other two exact algorithms we selected for comparison have been shown to achieve optimal solutions in many irregular packing instances. Therefore, the results compared with these algorithms are convincing.

4.5. Generalization test

The key to determining whether an original algorithm can be widely used lies in its generalization performance. Good generalization performance allows the algorithm to perform well even without additional training. In extraterrestrial mining mission,the shape of ores varies significantly, and algorithms can only handle the blocks encountered during training, which may not fulfill the task requirements. Good generalization performance is essential to ensure successful completion of the mission. Therefore, we will test the generalization of our algorithm. We will make the agent pack objects that have never appeared in the training data. According to the possible ore shapes, six new ore samples were redesigned, as shown in Fig. 10.

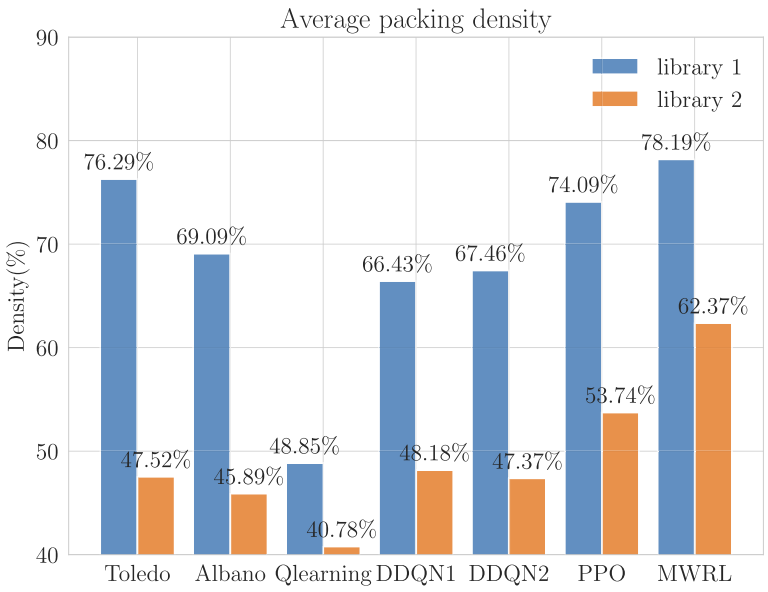

Fig. 8. The overall average packing density of all the experiments conducted within the two sample libraries.

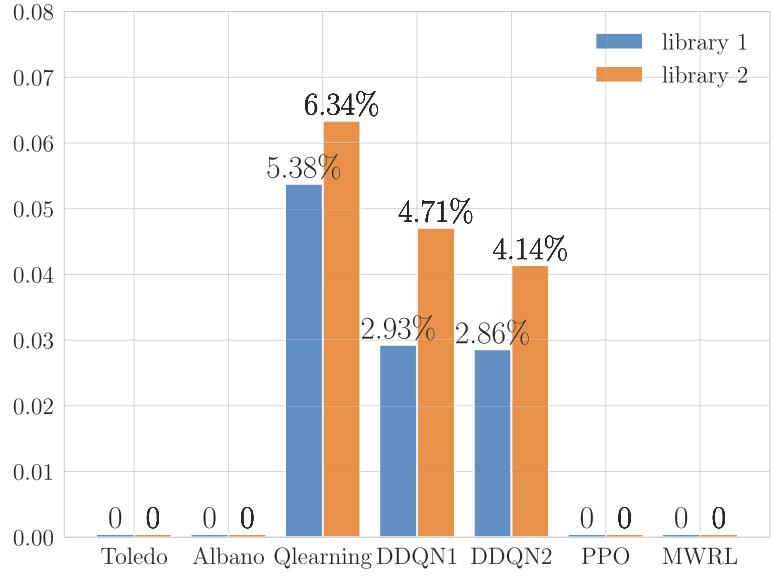

Fig. 9. The proportion of dangerous actions.

We conducted two comparative experiments with the six new ore samples. In each comparative experiment, we conducted 100 simulations, and calculated the average packing density in the 100 simulations. The packing results are illustrated in Table 2. Our method MWRL still achieves a better performance than the heuristic algorithm. In experiments, our method does not need to be retrained on the new six ore samples to give good decisions. But the heuristic algorithm needs to spend some time to find a solution in the solution space.

The above experiments have verified that our algorithm still achieves good performance when handling totally unfamiliar objects. We analyzed the main reasons for this success. Firstly, the pixelated objects and bins provide clear and concise information for the algorithm to understand the current state. Secondly, the prior information-based matrix M_{No-fit} projects the state data to a new state space, thus playing a role in normalizing the state data. This helps the algorithm to better handle various objects and generalize well.

Table 3
The experiment about generalization performance.

Instance	Toledo	MWRL
E-1	61.22%	**62.19%**
E-2	59.37%	**61.75%**

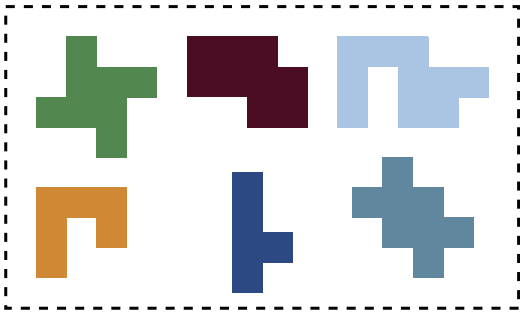

Fig. 10. The redesigned objects pool.

4.6. simulation experiments and physical experiments

The series of experiments mentioned above were conducted to assess the performance of the algorithm. Next, we validate the feasibility of the overall system. We need to test the two main functionalities of the system: first, whether the robot can use the hand-eye camera to measure the storage bin and objects, and generate No-fit matrix M_{No-fit}; and second, whether the proposed algorithm MWRL can be successfully applied to the robot system to complete the entire decision-making process. The experiments were conducted in both the simulation environment (PyBullet) and a physical platform.

Fig. 11 shows the complete packing process in the Pybullet, where the robot first uses the hand-eye camera to obtain a top-down view of the bin, then transforms it into a binary graph and generates the no-fit matrix M_{No-fit} to determine the packing position (represented by a circle point in the figure). After the packing action finishes, the bin state is updated and the resulting bin state matrix M_i is displayed as a sketch map to show the packing result. The successful experiments demonstrate the feasibility of using No-fit matrix matrices M_{No-fit} to describe the real-time status of ore assembly. We were able to complete all the measurements using a single hand-eye camera, significantly reducing the complexity of extraterrestrial mining tasks. Moreover, it is evident that the packing algorithm can rapidly provide reasonable assembly decisions that meet the requirements of the task. Throughout the experiments, we used rectangular blocks as our test objects, as non-convex blocks would cause the collision detection function in PyBullet to fail. However, this does not affect our ability to validate the effectiveness of the algorithm.

Finally, we conducted physical experiments to observe the system's smoothness and response time, which effectively reflects the operational efficiency of the system.We chose regular cubes for the physical experiments to demonstrate our system's compatibility with a robotic arm. Our goal was to verify the system's capability to extract key features from the hand-eye camera and make appropriate decisions based on the no-fit matrix M_{No-fit}. The entire packing process is depicted in Fig. 12. The robotic arm captures a top-down view image, selects an object, decides where to place it, and completes the packing action. From the start to filling the entire storage bin, it took a total of 567.75 s, with an average time of 37.85 s per step. The full bin confirms the feasibility of our algorithm.

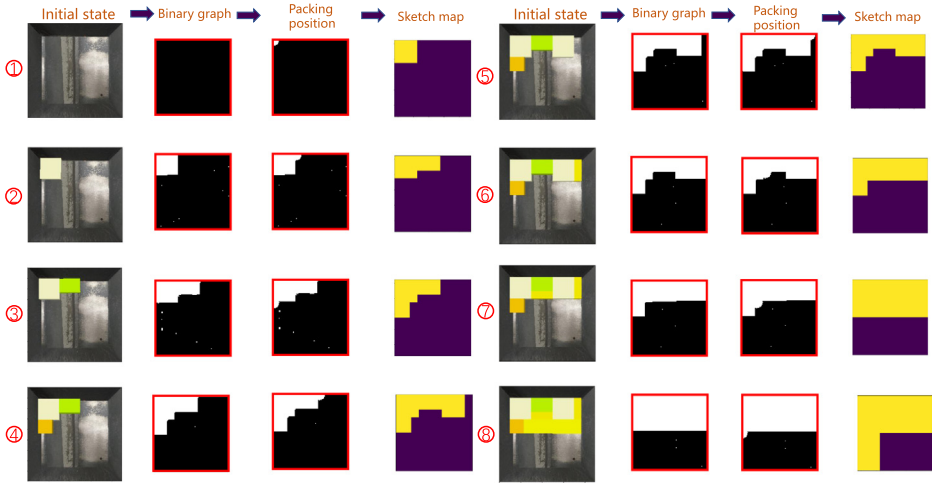

Fig. 11. The simulation experiment. The robot first uses the hand-eye camera to obtain the top view of the box, converts it into a binary image, generates prior information matrix M_{No-fit}, and finally makes a decision to complete an assembly.

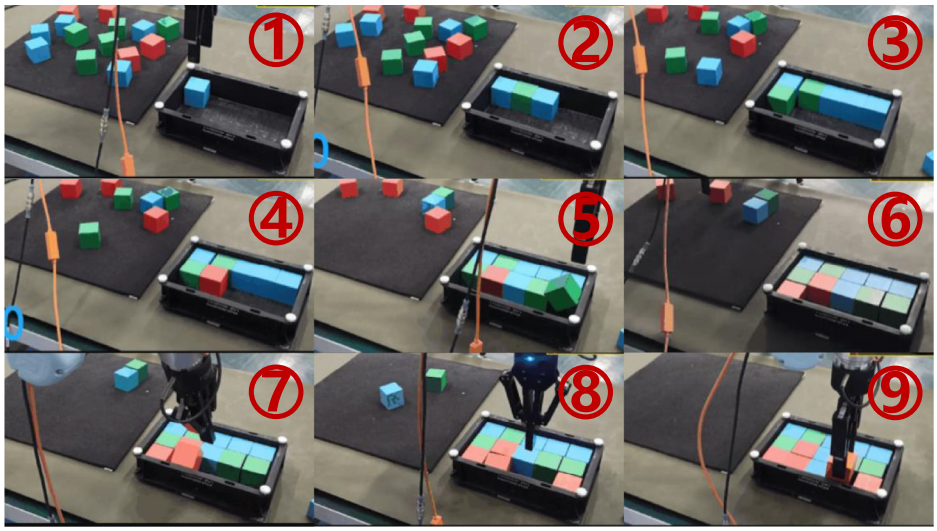

Fig. 12. The physical experiment.

Fig. 13. Some results of the experiment with rectangular objects.

To further test the repeatability of the system, we conducted 50 physical experiments with rectangular objects and calculated the average process time. The average processing time per step was 54.26 s, slightly longer than the time required with cube objects. Some results of the experiment are shown in Fig. 13. Throughout the process, the agent's decisions were similar to those made by a human using their naked eye to observe and judge, demonstrating the human-like intelligence of our algorithm.

5. Conclusion

In this paper, we propose an autonomous ore packing system capable of autonomously measuring ore characteristics and addressing the ore packing optimization problem in extraterrestrial unfamiliar environments. The system is divided into two parts: environment perception and measurement, and intelligent ore placement optimization algorithm. In the first part, we rely solely on a hand-eye camera to extract features of the ore and storage bin, introducing a novel state representation called the Non-Fit Matrix (NFM) that incorporates physical constraints and significantly enhances the generalization performance of the placement algorithm. In the second part, we employ reinforcement learning algorithms to meet the real-time intelligent requirements of mining tasks and introduce a novel training method called Maximum Worth Reinforcement Learning (MWRL), which we validate through experiments. Additionally, we propose an action masking method to ensure system safety and stability.

Our system can be further extended to more complex unmanned autonomous missions with simple modifications. In the future, we will continue to enhance the system's performance to tackle increasingly challenging tasks.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank the Editor-in-Chief, Associate Editor and anonymous referees for their invaluable comments and suggestions

Appendix A. Appendix

There are two sample libraries which are used in the experiments. (see Figs. 14,15).

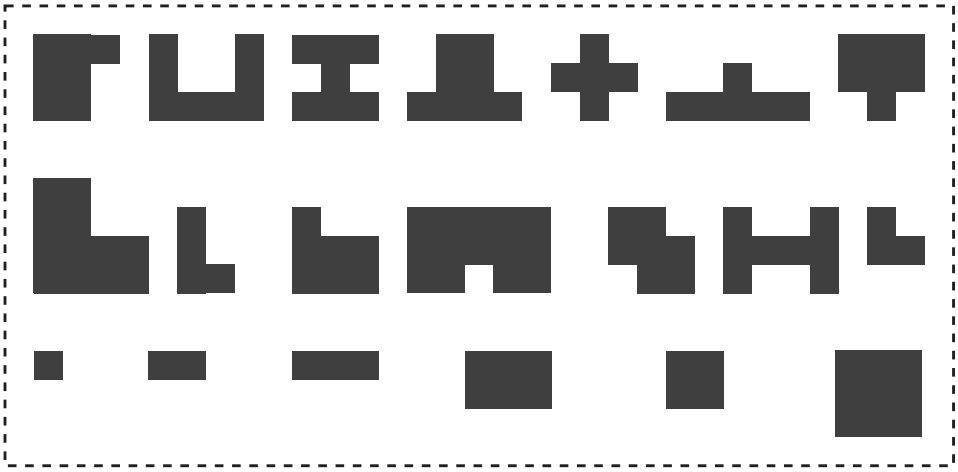

Fig. 14. The sample library 1 with large unit standard length.

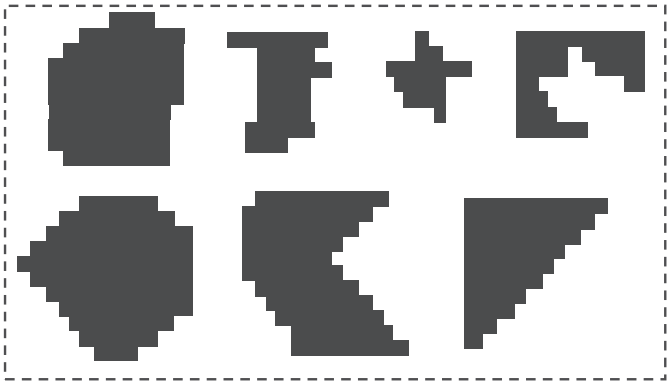

Fig. 15. The sample library 2 with small unit standard length.

References

Albano, A., Sapuppo, G., 1980. Optimal allocation of two-dimensional irregular shapes using heuristic search methods. IEEE Trans. Syst., Man, Cybernet. 10 (5), 242–248. https://doi.org/10.1109/TSMC.1980.4308483.

Chernov, N., Stoyan, Y., Romanova, T., 2010. Mathematical model and efficient algorithms for object packing problem. Comput. Geometry 43 (5), 535–553. https://doi.org/10.1016/j.comgeo.2009.12.003.

Golmisheh, F.M., Shamaghdari, S., 2023. Distributed safe formation maneuver control of euler–lagrange multi-agent systems in a partially unknown environment by safe reinforcement learning. Robot. Auton. Syst. 167, 104486. https://doi.org/10.1016/j.robot.2023.104486.

van Hasselt, H., Guez, A., Silver, D., 2015. Deep reinforcement learning with double q-learning, arXiv:1509.06461.

Hildebrandt, F., Thomas, B., Ulmer, M., 2022. Opportunities for reinforcement learning in stochastic dynamic vehicle routing. Comput. Oper. Res. 150, 106071. https://doi.org/10.1016/j.cor.2022.106071.

Hou, Y., Li, J., 2023. Learning 6-dof grasping with dual-agent deep reinforcement learning. Robot. Auton. Syst. 166, 104451. https://doi.org/10.1016/j.robot.2023.104451.

Howell, K., Spencer, D., 1986. Periodic orbits in the restricted four-body problem. Acta Astronaut. 13 (8), 473–479. https://doi.org/10.1016/0094-5765(86)90026-3.

Jiang, Y., Cao, Z., Zhang, J., 2023. Learning to solve 3-d bin packing problem via deep reinforcement learning and constraint programming. IEEE Trans. Cybernet. 53 (5), 2864–2875. https://doi.org/10.1109/TCYB.2021.3121542.

Kundu, O., Dutta, S., Kumar, S., 2019. Deep-pack: A vision-based 2d online bin packing algorithm with deep reinforcement learning. In: In 2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), pp. 1–7. https://doi.org/10.1109/RO-MAN46459.2019.8956393.

Lan, Y., Ren, J., Tang, T., et al., 2023. Efficient reinforcement learning with least-squares soft bellman residual for robotic grasping. Robot. Auton. Syst. 164, 104385. https://doi.org/10.1016/j.robot.2023.104385.

Li, K., Zhang, T., Wang, R., et al., 2022. Deep reinforcement learning for combinatorial optimization: Covering salesman problems. IEEE Trans. Cybernet. 52 (12), 13142–13155. https://doi.org/10.1109/TCYB.2021.3103811.

Martinez-Sykora, A., Alvarez-Valdes, R., Bennell, J., et al., 2017. Matheuristics for the irregular bin packing problem with free rotations. Eur. J. Oper. Res. 258 (2), 440–455. https://doi.org/10.1016/j.ejor.2016.09.043.

Qiao, D., Cui, P., Cui, H., 2012. Proposal for a multiple-asteroid-flyby mission with sample return. Adv. Space Res. 50 (3), 327–333. https://doi.org/10.1016/j.asr.2012.04.014.

Queiroz, L., Andretta, M., 2022. A branch-and-cut algorithm for the irregular strip packing problem with uncertain demands. Int. Trans. Oper. Res. 29. https://doi.org/10.1111/itor.13122.

Raissi, M., Perdikaris, P., Karniadakis, G., 2018. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. 378. https://doi.org/10.1016/j.jcp.2018.10.045.

Ren, Y., Shan, J., 2014. On tethered sample and mooring systems near irregular asteroids. Adv. Space Res. 54 (8), 1608–1618. https://doi.org/10.1016/j.asr.2014.06.042.

Rodrigues, M.O., Toledo, F.M., 2017. A clique covering mip model for the irregular strip packing problem. Comput. Oper. Res. 87, 221–234. https://doi.org/10.1016/j.cor.2016.11.006.

Sankaran, K., Hamming, B., Grochowski, C., et al., 2013. Evaluation of existing electric propulsion systems for the osiris-rex mission. J. Spacecr. Rock. 50, 1292–1295. https://doi.org/10.2514/1.A32505.

Schulman, J., Wolski, F., Dhariwal, P. et al., 2017. Proximal policy optimization algorithms. arXiv:1707.06347.

Silver, D., Huang, A., Maddison, C., et al., 2016. Mastering the game of go with deep neural networks and tree search. Nature 529, 484–489. https://doi.org/10.1038/nature16961.

Sun, Z., Wang, N., Lin, H., et al., 2022. Persistent coverage of uavs based on deep reinforcement learning with wonderful life utility. Neurocomputing 521. https://doi.org/10.1016/j.neucom.2022.11.091.

Terashima-Marín, H., Ross, P., Zárate, C., et al., 2010. Generalized hyper-heuristics for solving 2d regular and irregular packing problems. Annals OR 179, 369–392. https://doi.org/10.1007/s10479-008-0475-2.

Tian, R., Kang, C., Bi, J., et al., 2023. Learning to multi-vehicle cooperative bin packing problem via sequence-to-sequence policy network with deep reinforcement learning model. Comput. Industr. Eng. 177, 108998. https://doi.org/10.1016/j.cie.2023.108998.

Tole, K., Moqa, R., Zheng, J., et al., 2023. A simulated annealing approach for the circle bin packing problem with rectangular items. Comput. Industr. Eng. 176, 109004. https://doi.org/10.1016/j.cie.2023.109004.

Toledo, F.M., Carravilla, M.A., Ribeiro, C., et al., 2013. The dotted-board model: A new mip model for nesting irregular shapes. Int. J. Prod. Econ. 145 (2), 478–487. https://doi.org/10.1016/j.ijpe.2013.04.009.

Wang, Z., Schaul, T., Hessel, M., et al., 2016. Dueling network architectures for deep reinforcement learning, arXiv:1511.06581.

Xu, R., Cui, P., Qiao, D., et al., 2007. Design and optimization of trajectory to near-earth asteroid for sample return mission using gravity assists. Adv. Space Res. 40 (2), 220–225. https://doi.org/10.1016/j.asr.2007.03.025.

Yamaguchi, T., Saiki, T., Tanaka, S., et al., 2018. Hayabusa2-ryugu proximity operation planning and landing site selection. Acta Astronaut. 151, 217–227. https://doi.org/10.1016/j.actaastro.2018.05.032.

Yuichi Tsuda, M.A.H.M.S.N., Yoshikawa, Makoto, 2013. System design of the hayabusa 2—asteroid sample return mission to 1999 ju3. Acta Astronaut. 91 (2), 356–362. https://doi.org/10.1016/j.actaastro.2013.06.028.

Yurimoto, H., ichi Abe, K., Abe, M., et al., 2011. Oxygen isotopic compositions of asteroidal materials returned from itokawa by the hayabusa mission. Science 333 (6046), 1116–1119. https://doi.org/10.1126/science.1207776.

Zhang, J., Ding, Y., Chen, L., et al., 2022. A sweeping and grinding combined hybrid sampler for asteroid sample return mission. Acta Astronaut. 198, 329–346. https://doi.org/10.1016/j.actaastro.2022.06.019.

Zhang, J., Dong, C., Zhang, H. et al., 2018. Modeling and experimental validation of sawing based lander anchoring and sampling methods for asteroid exploration. Adv. Space Res., 61(9), 2426–2443. URL: https://www.sciencedirect.com/science/article/pii/S0273117718301169. doi: 10.1016/j.asr.2018.02.003.

Zhang, T., Xu, K., Ding, X., 2021. China's ambitions and challenges for asteroid–comet exploration. Nature Astronomy 5. https://doi.org/10.1038/s41550-021-01418-9.

Zhao, H., She, Q., Zhu, C., et al., 2022. Online 3d bin packing with constrained deep reinforcement learning, arXiv:2006.14978.