基于深度强化学习的作业车间调度问题优化*

乔东平[①②] 段绿旗[①②] 黎宏磊[①②] 肖艳秋[①②]
（①河南省机械装备智能制造重点实验室，河南 郑州 450002；
②郑州轻工业大学机械工程学院，河南 郑州 450002）

摘 要：针对作业车间调度问题求解的复杂性，以最小化最大完工时间为目标，提出基于深度强化学习优化算法求解作业车间调度问题。首先，基于析取图模型构建深度强化学习的调度环境，并建立三通道状态特征，设计 20 种复合启发式调度规则作为动作空间，将奖励函数等价为机器利用率；利用深度卷积神经网络搭建动作网络和目标网络，以状态作为输入，输出每个动作的 Q 值，进而使用行动有效性探索和利用策略选取动作；最后，计算即时奖励和更新调度环境。使用标准案例验证了算法可以平衡求解质量和时间，训练好的智能体对非零初始状态下调度问题具有很好的泛化性。

关键词：深度强化学习；作业车间调度；调度规则
中图分类号：TH1658 文献标识码：A
DOI：10.19287/j.mtmt.1005-2402.2023.04.023

Optimization of job shop scheduling problem based on deep reinforcement learning

QIAO Dongping[①②], DUAN Lvqi[①②], LI Honglei[①②], XIAO Yanqiu[①②]
(①Henan Key Laboratory of Intelligent Manufacturing of Mechanical Equipment, Zhengzhou 450002, CHN;
②College of Mechanical and Electrical Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, CHN)

Abstract: Aiming at the optimization problem of minimizing the maximum completion time in job shop scheduling, a deep reinforcement learning optimization algorithm is proposed. First, a deep reinforcement learning scheduling environment is built based on the disjunctive graph model, and three channels of state characteristics are established. The action space consists of 20 designed combination scheduling rules. The reward function is designed based on the proportional relationship between the total work of the scheduled operation and the current maximum completion time. The deep convolutional neural network is used to construct action network and target network, and the state features are used as inputs to output the Q value of each action. Then, the action is selected by using the action validity exploration and exploitation strategy. Finally, the immediate reward is calculated and the scheduling environment is updated. Experiments are carried out using benchmark instances to verify the algorithm. The results show that it can balance solution quality and computation time effectively, and the trained agent has good generalization ability to the scheduling problem in the non-zero initial state.

Keywords: deep reinforcement learning; job shop scheduling; scheduling rules

作业车间调度问题 (job shop scheduling problem, JSSP) 作为具有广泛工程背景的组合优化问题，其理论和求解方法的研究具有重要的理论和现实意义[1]。在过去的几十年中，众多的学者和工程人员对作业车间调度问题进行了大量的研究工作，提出的求解算法主要包括启发式规则调度算法和元启发式算法两类[2]。其中元启发式算法在诸多调度问题求解过程中取得了满意的性能，但其迭代搜索过程通常比较耗时，并且较少利用历史信息来调整搜索行为[3-4]。启发式调度规则算法在求解作业车间调

* 河南省高等学校重点科研项目计划支持（20A460029）

度问题时具有较低的时间复杂度，能够快速获得可行的调度方案，但研究表明调度规则具有短视的天性[5−8]：单一的规则在所有的算例中不具有最优性，混合组合多种规则比单一规则的求解效果要好。另外这两种调度方法都是针对特定的调度问题求解设计的，在问题发生变化时通常需要对相关参数和规则进行调整或重新设计，导致其扩展性和适应性不强[9]。因此，研究新的作业车间调度求解方法，充分利用历史信息快速获得高质量的解，提高算法的适应性和泛化性对提升企业生产管理水平具有重要的现实意义。

深度强化学习在游戏决策[10]、组合优化问题[11]和资源调度[12]等领域成功应用。深度强化学习结合调度规则可以弥补传统调度方法对历史数据应用不足的缺陷，获得满足生产实际需要的调度方案。Zhao Y 等人[13]提出了基于深度强化学习的动态车间生产调度，验证了深度强化学习和调度规则结合的动态车间生产控制算法的有效性。王凌等人[14]提出了一种基于深度强化学习与迭代贪婪算法的框架用于求解流水车间调度，充分利用了局部搜索来提高求解的质量。肖鹏飞等人[15]自定义了 15 个表达制造过程的状态作为特征数据输入，使用启发式算法作为动作空间，采用深度时序差分强化学习算法训练模型，证明了该算法具有较高的灵活性和适应性。Luo S[16]将深度 Q 网络 (deep Q-network ,DQN) 算法与调度规则结合，设计了 7 种调度规则，并将其应用到动态作业车间调度问题的求解过程中。现有的研究主要是基于深度强化学习方法针对特定的单一调度问题进行求解，求解模型对问题规模较敏感，较少考虑方法的泛化性，对于复杂的作业车间调度问题还需要进一步进行研究。

针对复杂的作业车间调度问题提出基于深度强化学习优化算法求解。首先，采用析取图模型构建调度环境，建立了工序工时层、工序需求层、完工时间层的三通道状态特征，使用复合启发式调度规则作为动作空间集，将已安排工序的总工时与当前最大完工时间之间的比例关系作为设计奖励函数的依据；利用深度卷积神经网络 (deep convolutional neural networks，DCNN) 搭建动作网络和目标网络，以三通道状态特征作为输入，输出每个动作的 Q 值；然后，采用行动有效性探索和利用策略 (action validity exploration and exploitation of strategy, AVEES) 选取动作空间中的复合启发式调度规则；最后，计算即时奖励和更新调度环境。通过标准案例验证了算法的有效性，使用训练好的智能体对非零的初始状态下调度问题进行优化求解，验证了该算法也具有很好的泛化性。

1 调度框架

1.1 问题描述

作业车间调度问题可以描述为：工件集 $\mathbf{J}=\{J_1, J_2, J_3, \cdots, J_n\}$由机器集$\mathbf{M}=\{M_1, M_2, M_3, \cdots, M_k\}$进行加工，每个工件按照规定的工艺路线，逐一安排到机器上进行加工。工件J_i具有n_i个工序，$O_{i,j}$表示工件J_i的第j道工序。最大完工时间 (makespan) 是所有工件最后一道工序完工时间记为$C_{\max}$，本文以最小化$C_{\max}$为目标。

为了讨论深度强化学习求解 JSSP 的可行性，使用析取图来表示调度智能体的学习环境。以 3 个工件和 3 台机器的作业车间调度问题为例，其加工信息如表 1 所示。JSSP 问题的学习环境可以由三元组表示$\mathbf{G}=(\mathbf{N};\mathbf{B},\mathbf{E})$，其中 **N** 为所有工序节点集合（包括了虚拟的起始节点 start 和终止节点 end）；**B** 为同一零件中由工艺决定的相邻工序之间的有向连接弧集，表达了工序之间的约束关系；**E** 为在同一机器上加工的工序间的析取弧集，表达了工序在设备上的加工顺序约束关系，如图 1 所示。

表 1　3 个工件、3 台机器加工信息

工件	加工信息（机器序列，加工时间）		
J_1	(1,3)	（2,2）	(3,5)
J_2	（1,3）	(3,5)	（2,1）
J_3	（2,2）	（1,5）	（3,3）

1.2 调度框架

本文构建的基于深度强化学习求解的作业车间调度问题框架如图 2 所示，框架由环境和网络训练组成。通过定义调度环境、状态、动作、奖励和调度策略，将作业车间调度过程映射为马尔科夫决策过程 (markov decision processes, MDP)。借助 DRL 对智能体的调度策略进行训练获取最优策略，确保智能体在每个决策点都能选到最合适的动作。

环境主要包括析取图、动作空间、状态和奖励函数。析取图作为深度强化学习的调度环境，从工序角度出发，分派工序到机器上进行加工。为调度智能体提供了调度决策点和执行动作的环境，决定

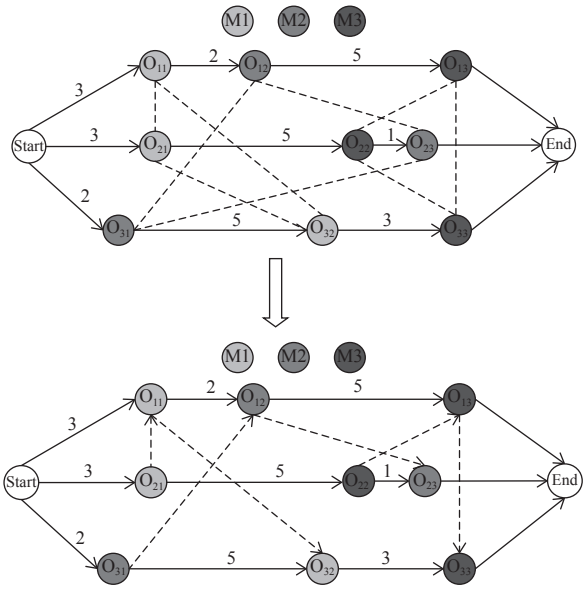

图 1 有向无环图

了动作对工件分派的影响。

在 JSSP 中状态需要反映工件和机器的整体和局部特征。在 JSSP 中动作是在启发式调度规则下选择优先级最高的工件，并将其分派到相应的机器中。在 JSSP 中，可以将不同状态下行为的评估函数或相反函数作为奖励，用于评价当前动作的好坏。调度策略作为深度强化学习的优化目标，决定了智能体在一定状态下选择的动作。智能体借助不同状态下采取动作得到的奖励信号对调度策略进行优化。

网络训练部分主要由记忆池、动作网络、目标网络和损失函数组成。记忆池用于存储环境提供的状态值、动作值和奖励值，为动作网络和目标网络的更新提供样本，通过随机抽取样本的方式打破数据之间的关联性；动作网络和目标网络通过环境和记忆池提供的训练样本分别计算出预测值和实际值，构建预测值和实际值的损失函数；采用梯度下降方式对动作网络参数进行更新，为了稳定目标网络，目标网络每隔固定步数 (C) 使用动作网络参数对目标网络参数进行更新。

2 基于 DRL 的作业车间调度问题实现

2.1 状态、动作和奖励

在 JSSP 的 DRL 框架中，状态必须正确地反映工件和机器的全局和局部信息。状态特征是对状态属性的数值表示，状态特征应易于计算，并进行归一化，保证尺度均匀性，对不同的调度问题状态特征需要具有一定的兼容性。本文设计了一个具有三层通道的状态特征s_t如下式：

$$s_t = \{\boldsymbol{PT}, \boldsymbol{PM}, \boldsymbol{PC}\} \tag{1}$$

其中：***PT*** 为工序工时层矩阵、***PM*** 为工序需求层矩阵、***PC*** 为完工时间层矩阵。矩阵 ***PT***、***PM***、***PC*** 的计算公式如下：

$$\boldsymbol{PT}_{i,j} = \begin{cases} P_{i,j} & \text{if } X_{i,j} = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\boldsymbol{PM}_{i,j} = \begin{cases} M_{i,j} & \text{if } X_{i,j} = 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$\boldsymbol{PC}_{i,j} = \begin{cases} T_{i,j} & \text{if } X_{i,j} = 1 \\ 0 & \text{otherwise} \end{cases}$$

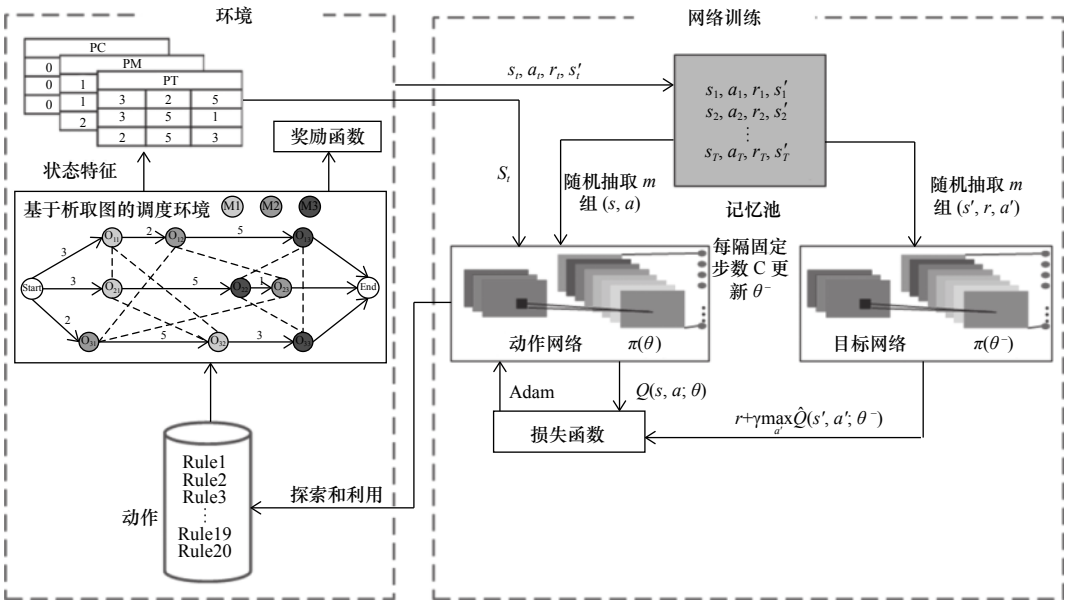

图 2 调度框架

其中：若工序$O_{i,j}$已完成加工则$X_{i,j}=1$，否则为0，$M_{i,j}$为分派的机器编号，$P_{i,j}$为加工时间，$T_{i,j}$为完工时间。以上***PT***、***PM***、***PC***矩阵的大小均为$n\times m$，n为工件数目，m为机器数目。

在JSSP的DRL框架中，动作是智能体可以执行的启发式调度规则。智能体通过选择的启发式调度规则对工件进行分派，确保调度结果接近预期的性能指标。以考虑工件全部工序加工信息的规则作为全局规则，只考虑工件部分工序加工信息的规则作为局部规则。选取多种具有全局和局部特性的启发式调度规则作为智能体的可选动作来克服单一规则短视性，也能够充分发挥深度强化学习的学习能力。本文通过20个启发式调度规则组成动作空间，如表2所示。

在基于析取图构建的调度环境中，智能体在依据当前状态按照选定的规则进行工序分派时，会出现多个工件符合规则的情况，此时智能体陷入“状态盲区”。此时，从时钟角度出发对工件进行二次选择，即使用先到先服务规则进行工件选择。因此，对选定20种启发式规则分别与先到先服务规则进行组合应用，形成复合启发式调度规则动作空间。

奖励函数优劣对深度强化学习的训练有重要的影响[17]，合理的奖励函数能够提高训练速度，使训练结果快速收敛。本文将JSSP中最大完工时间的最小化转化为最大化机器利用率U_k，如下式：

$$U_k=\frac{\sum_{i=1}^{n}\sum_{j=1}^{n_i}X_{i,j}\times P_{i,j}}{M\times C_{\max_k}} \tag{3}$$

其中：$k=1,2,\cdots,K$为已加工工序计数器，可以视为深度强化学习中的时间步长，$C_{\max_k}$为时间步长k处的最大完工时间。定义即时奖励：

$$r_k=U_k-U_{k-1} \tag{4}$$

则累计奖励如下式：

$$R=\sum_{k=1}^{K}r_k \tag{5}$$

从上式中可以推导出最小化最大完工时间可以等价为累计奖励最大化。

2.2 行动有效性探索和利用策略

在JSSP中探索和利用是调度智能体选择动作的两种相互冲突的重要策略。为了平衡智能体的探索和利用，使用epsilon-decreasing策略[18]：在开始时探索高于利用，随着智能体的学习进行探索被逐步转移到利用，当探索率$\varepsilon=0$时，选择最优动作执行。基于该策略的动作选择如下：

$$a=\begin{cases}\arg\max_a Q(s_t,a_t;\theta) & P\geqslant\varepsilon\\ random & P<\varepsilon\end{cases} \tag{6}$$

其中：a为调度规则的编号，P值为服从0到1均匀分布函数的随机数，ε用以下公式更新：

$$\varepsilon=\varepsilon_0\left(1-\frac{n_{iter}}{N}\right) \tag{7}$$

其中：ε_0为ε的初始值，n_{iter}为当前步长计数器，N为总的搜索步长。

当可供空闲机器选择的工件唯一时，状态转移唯一，智能体无法有效地利用历史经验，为了提高智能体的学习效率，直接将工件分派到空闲机器上，使智能体的行动更加有效。将行动有效性和epsilon-greedy策略相结合，提出了行动有效性探索和利用

表2 启发式调度规则

No.	特性	Rule	描述
1	全局	SPT	选择工序加工时间最短的工件
2		LPT	选择工序加工时间最长的工件
3		SPT*TWK	选择工序加工时间与总加工时间乘积最小的工件
4		LPT*TWK	选择工序加工时间与总加工时间乘积最大的工件
5		SPT/TWK	选择工序加工时间与总加工时间之比最小的工件
6		LPT/TWK	选择工序加工时间与总加工时间之比最大的工件
7		SPT*TWKR	选择工序加工时间与剩余加工时间乘积最小的工件
8		LPT*TWKR	选择工序加工时间与剩余加工时间乘积最大的工件
9		SPT/TWKR	选择工序加工时间与剩余加工时间之比最小的工件
10		LPT/TWKR	选择工序加工时间与剩余加工时间之比最大的工件
11	局部	SRM	选择除当前考虑工序外剩余加工时间最短的工件
12		LRM	选择除当前考虑工序外剩余加工时间最长的工件
13		SRPT	选择剩余加工时间最短的工件
14		LRPT	选择剩余加工时间最长的工件
15		SSO	选择下一工序加工时间最短的工件
16		LSO	选择下一工序加工时间最长的工件
17		SPT+SSO	选择当前工序加工时间与后续工序加工时间最短的工件
18		LPT+LSO	选择当前工序加工时间与后续工序加工时间最长的工件
19		SQN	选择等待时间最短的工件
20		LQN	选择等待时间最长的工件

策略。表 3 给出了行动有效性探索和利用策略的实施流程。

表 3 行动有效性探索和利用策略的动作选择流程

Input: 环境E_t	
Output: 状态s_t, s_{t+1}，动作a_t，奖励r_t，环境E_t	
1:	从E_t随机选择一台空闲机器M_k，建立允许在机器M_k上加工的工件集合O_k
2:	$s_t \leftarrow E_t$
3:	取随机数P服从[0,1]的均匀分布
4:	if $\|O_k\|$=1
5:	直接安排工序在机器M_k上加工，$a_t = -1$
6:	else if $P < \varepsilon$
7:	从动作集合**A**随机选择一个动作a_t
8:	Else
9:	$a_t = \underset{a'}{\text{argmax}} Q(s_t, a; \theta)$
10:	end if
11:	执行动作a_t
12:	end if
13:	更新环境E_t，$s_{t+1}, r_t \leftarrow E_t$
14:	return s_t, s_{t+1}, r_t, a_t, E_t

2.3 基于 DRL 的 JSS 实施

智能体的目标是找到最优调度策略$\pi^* \in \mathbf{\Pi}$，在车间状态s下遵循特定的调度策略π采取动作a所代表的复合启发式调度规则，获得最大化累计奖励和最小完工时间，定义式如下：

$$Q^{\pi^*}(s,a) = \max_{\pi \in \mathbf{\Pi}} Q^{\pi}(s,a) = E\left[\sum_{h=0}^{\infty} \gamma^h r_{t+h} \mid s_t = s, a_t = a\right] \quad (8)$$

其中：$\gamma \in (0,1]$，$Q^{\pi}(s,a)$为值函数。为了优化智能体的调度策略，一个合适的状态和动作之间的值函数是非常重要的。本文中采用动作网络对值函数进行拟合，如下式：

$$Q^{\pi}(s,a) \approx Q(s,a;\theta) \quad (9)$$

为了打破了数据之间的关联性引入经验回放机制，建立一个容量大小为N的记忆池D，其中存储智能体的每个调度决策点t的经验即(s_t, a_t, r_t, s_{t+1})。从D中随机抽取一批样本对动作网络的权重进行优化。当容量超出N后使用新的调度经验替换旧的调度经验，实现更好的数据使用效率。在网络权重的优化过程中为了确保动作网络的稳定性，建立一个与动作网络有相同结构的目标网络$\widehat{Q}(s,a;\theta^-)$。以动作网络的预测值和目标网络的实际值构建损失函数$L(\theta)$，如下式：

$$L(\theta) = \left(r + \gamma \max_{a' \in \boldsymbol{A}} \widehat{Q}(s', a'; \theta^-) - Q(s,a;\theta)\right)^2 \quad (10)$$

其中：$r + \gamma \max_{a' \in \boldsymbol{A}} \widehat{Q}(s', a'; \theta^-)$为实际值，$Q(s,a;\theta)$为预测值。动作网络的权重参数采用梯度下降的方式进行更新，如下式：

$$\theta = \theta + \alpha \nabla_{\theta} L(\theta) \quad (11)$$

其中：$\alpha \in [0,1]$为学习率，$\nabla_{\theta} L(\theta)$为损失函数有关权重$\theta$的梯度。为了稳定目标网络，目标网络中的权重$\theta_k^-$每隔固定步数 (C) 进行一次更新，更新公式如下：

$$\theta_k^- = \theta_k \quad (12)$$

基于深度强化学习的作业车间调度优化算法具体实施如表 4 所示。在表 4 中，对算法进行初始化设置 (第 1~3 行)。初始化调度环境和状态 (第 5 行)，时间 T 初始化为 0(第 6 行)。在 T 时从环境中判断空闲机器集合M_s的模是否为 0(第 8~9 行)。若$|Ms| \neq 0$成立依据表 3，智能体根据当前的状态从动作空间中选取一个规则，根据规则将工序分派到机器上，获得$s_t, s_{t+1}, r_t, a_t, E_t$。若$a_t \neq -1$成立，则记录此次的状态、动作和奖励 (第 11~12 行)，这使得智能体有更快的求解速度和学习速度。当记忆池存满后（第 12~13 行），最早的历史信息被新信息替换。智能体开始进行抽样学习，采用 Adam 优化器的梯度下降的方式对动作网络进行更新，为了保持目标网络的稳定性，每隔固定步数（C）对目标网络进行更新（第 14~18 行），重复执行 5~24 行直至训练结束。

2.4 神经网络结构

本文构建了动作网络作为状态与动作的非线性映射器，将状态和动作的关系映射到深度卷积神经网络，目标网络与动作网络具有相同的结构。深度卷积神经网络的结构如图 3 所示。

从环境中获得三通道状态特征作为深度卷积神经网络的输入，输出 20 个动作的值。三通道状态特征的维度为$3 \times n \times m$，其中 n 为工件数目，m 为机器数目。深度神经网络由输入层、输出层和 3 层隐藏层构成。输入层为$3 \times n \times m$的三通道特征。隐藏层由两层卷积层和一层全连接层构成，两层的卷积层的大小分别为$9 \times (n+2) \times (m+2)$、$18 \times (n+4) \times (m+4)$，全连接层由$18 \times (n+4) \times (m+4)$个节点组成，输出层由 20 个节点组成。

3 实验结果和分析

3.1 实验设置与参数设置

为了验证 DRL 算法在 JSSP 当中的有效性，用

Python 编程语言实现，运行于 2.9 GHz Intel i5 处理器，Windows10 操作系统 PC 平台。基于面向对象的思想构建作业车间调度环境类、机器类和工件类，利用 Pytorch 进行智能体的迭代学习，并实现智能体与调度环境的交互。参数的选择对求解的质量具有影响，遵循一般性原则进行参数的设置，如表 5 所示。

表 4　基于 DRL 的作业车间调度优化算法流程

1:	初始化记忆池D容量为B
2:	用随机权重θ初始化动作网络Q
3:	初始化目标网络$\widehat{Q}$权重$\theta^{-}=\theta$
4:	for e=1, 2, ⋯, E do
5:	初始化调度环境E_0，s_0
6:	T ← 0
7:	While 所有工序是否加工完毕 do
8:	Ms ← E_t
9:	if |Ms| ≠ 0
10:	使用表3获得s_t, s_{t+1}, r_t, a_t, E_t
11:	if $a_t \neq -1$
12:	将s_t，a_t，r_t，s_{t+1}存储在记忆池D中，b = b + 1
13:	if b>B
14:	从D中随机抽取一批样本(s_j, a_j, r_j, s_{j+1})
15:	$y_k=\begin{cases} r_j, \text{如果}j+1\text{步完成调度} \\ r_j+\gamma\max\limits_{a_j}\widehat{Q}\left(\left(s_{j+1}\right), a_j; \theta_k^{-}\right), \text{其他} \end{cases}$
16:	计算$\Delta\theta=\alpha(y_k-Q(s_j, a_j;\theta_k))^2\nabla_{\theta_k}Q\left(s_j, a_j;\theta_k\right)$的梯度下降，更新动作值
17:	动作网络Q参数更新$\theta=\theta+\Delta\theta$
18:	每隔C步更新一次目标网络$\widehat{Q}$参数$\theta^{-}=\theta$
19:	end if
20:	end if
21:	else if
22:	更新T为当前状态下工序的最小完工时间，更新调度环境E_t
23:	end if
24:	end while
25:	end for

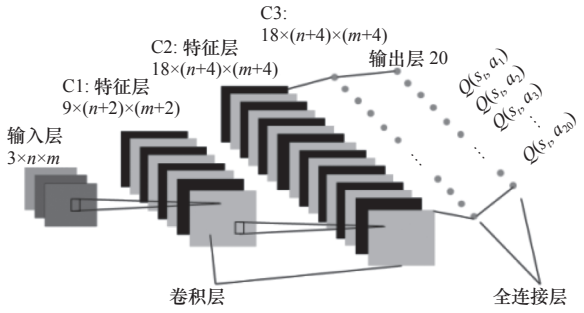

图 3 深度卷积神经网络的结构

表 5　参数设置

参数名称	参数值
迭代次数	3 000
经验池大小	10 000
随机梯度下降采样样本大小	128
学习率	0.001
目标网络的更新频率	200
ε的初始值	0.1
ε最终值	0

3.2　算例分析

为了验证算法的有效性，使用 OR Library 中的标准案例进行实验验证，其中标准案例 ft06 的训练过程展示如图 4 所示。从图中可以看出，随着智能体学习过程的推进，调度结果逐渐收敛。在学习迭代的后期结果会有一定的波动，主要是在任务选择上仍然有一定的随机性。该训练过程表明了本文提出的算法具有较强的学习能力和收敛性。

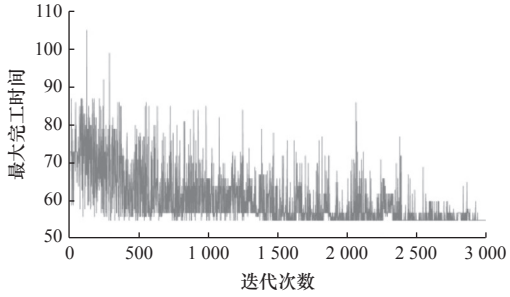

图 4 ft06 训练过程

本文以最小化最大完工时间为目标，复合启发式调度规则作为算法的动作空间，因此选用动作空间中最优复合启发式调度规则（best rule）以及遗传算法（GA）与基于 DRL 的作业车间调度（job shop scheduling based DRL, JSSBDRL）进行比较。算例的对比情况如表 6 所示，其中最优结果加粗表示。分析表 6 可知，本算法在标准算例上的表现明显优于复合启发式调度规则，相比较于遗传算法也具有较大的优势。

本文引用文献 [19] 中的调度分数对优化性能进行比较。其中，调度分数= $C_{LB}/C_{alg}\times100$，C_{LB}为下界对应的最大完工时间，而C_{alg}为算法对应的最大完工时间。将 DRL 与复合启发式调度规则和遗传算法的调度得分比较如图 5 所示。分析图 5 可知本文算法调度结果中有 15.8% 达到最优结果，有 31.6% 调度得分高于 95，并且调度得分全部高于 80。

尽管本文算法没有全部达到最优解，但 DRL 在上述案例中与复合启发式调度规则比较性能平均提升了 8.8%，与 GA 算法比较性能平均提升了 4.6%。

表 6　实验结果对比

案例	LU	UB	best rule	GA	JSSBDRL
ft06(6x6)	55	55	57	55	55
ft10(10x10)	930	930	1 142	1 095	1 002
ft20(20x5)	1 165	1 165	1 430	1 278	1 170
abz5(10x10)	1 234	1 234	1 451	1 352	1 280
abz7(20x15)	656	656	802	854	780
orb01(10x10)	1 059	1 059	1 296	1 250	1 195
orb02(10x10)	888	888	1 060	991	940
la01(10x5)	666	666	694	666	680
la02(10x5)	655	655	799	729	705
la06(15x5)	926	926	945	926	930
la07(15x5)	890	890	947	893	893
la11(20x5)	1 222	1 222	1 256	1 222	1 222
la12(20x5)	1 039	1 039	1 129	1 039	1 039
la21(15x10)	1 046	1 046	1 295	1 283	1 195
la22(15x10)	927	927	1 102	1 110	1 089
la26(20x10)	1 218	1 218	1 439	1 404	1 356
la27(20x10)	1 235	1 235	1 450	1 340	1 302
la36(15x15)	1 268	1 268	1 510	1 528	1 401
la37(15x15)	1 397	1 397	1 680	1 648	1 398

3.3　泛化性分析

为了验证 JSSBDRL 的泛化性，对 ft06(6×6)，ft10(10×10)，la21(15×10)，la26(20×10) 进行分析，按照工艺路线随机将部分工序设定为已加工工序，建立不为零的初始调度环境。设置已调度比σ=30%、40%、50%，同时为每个已调度比生成 60 个随机案例作为测试集，对训练好的智能体进行泛化性分析，计算结果如图 6 所示。

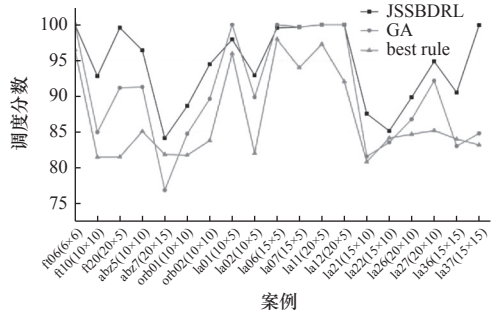

图 5 不同算法调度得分

由图 6 可知：（1）对于 ft06(6×6) 和 ft10(10×10) 标准案例 DRLOA 与 GA 优化性能基本一样，好于复合启发式调度规则。（2）对于 la21(15×10) 和 la26(20×10) 标准案例 JSSBDRL 性能强于 GA 和复合启发式调度规则。（3）随着已调度比的增加 3 种算法性能有所恶化，但 JSSBDRL 在优化性能上依旧优于 GA 和复合启发式调度规则。（4）对 3 种算法同时使用 Python 编程语言进行计算求解，各案例的已调度比的测试时间汇总于表 7，针对 la21 和 la26 标准案例 JSSBDRL 的测试时间相比 GA 算法测试时间缩短了 1~5 倍，对于不同的已调度比 GA 算法均需要重新编码，会增加额外的工作量，而 JSSBDRL 算法仅需要将状态调整为已调度比下的状态即可。相比复合启发式调度规则 JSSBDRL 测试时间是其 6~90 倍，但 JSSBDRL 的计算时间也都在 1 s 左右，这个计算时间可以满足实际需求，平衡了求解质量和计算时间。

表 7　不同调度比下测试时间对比

案例	σ	JSSBDRL/s	best rule/s	GA/s
ft06(6×6)	0.3	1.00	0.028	1.20
	0.4	0.95	0.022	1.14
	0.5	0.90	0.016	1.08
ft10(10×10)	0.3	1.10	0.072	2.24
	0.4	1.00	0.066	2.14
	0.5	0.90	0.060	2.04
la21(15×10)	0.3	1.20	0.142	4.00
	0.4	1.05	0.132	3.86
	0.5	0.95	0.122	3.72
la26(20×10)	0.3	1.36	0.198	5.60
	0.4	1.18	0.182	5.44
	0.5	1.00	0.166	5.28

4　结语

本文将深度强化学习与组合调度规则相结合来解决车间调度问题。加工时间矩阵、工件需求矩阵和完成时间矩阵的 3 个通道状态特征用于描述每个调度时间的输入状态。动作空间由 20 种组合调度规则组成。为了提高算法的学习效率，提出了动作有效性的探索和利用策略。通过不同的基准实例实验，JSSBDRL 的求解质量总体上优于组合调度规则和 GA，并且在计算时间上也很好。经过训练的代理对具有不同处理率的实例具有很好的泛化能力。还有一些细节需要进一步研究，例如克服手工设计奖励和多目标优化的不足。

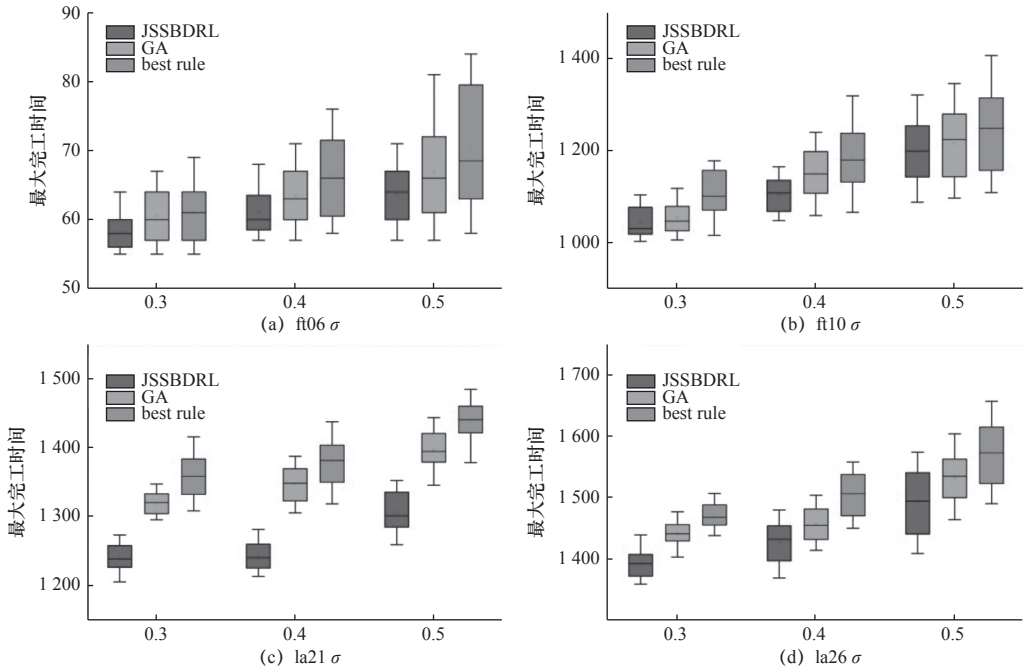

图 6 各算法在不同调度比下的调度结果

参 考 文 献

[1] Meeran S, Morshed M S. A hybrid genetic tabu search algorithm for solving job shop scheduling problems: a case study[J]. Journal of Intelligent Manufacturing, 2011, 23(4): 1063-1078.

[2] Ahmadian M M, Khatami M, Salehipour A, et al. Four decades of research on the open-shop scheduling problem to minimize the makespan[J]. European Journal of Operational Research, 2021, 295(2): 399-426.

[3] 王艳红, 赵也践, 刘文鑫. 数据挖掘算法在作业车间调度问题中的应用[J/OL]. 计算机集成制造系统, https://kns.cnki.net/kcms/detail/11.5946.TP.20211214.1816.004.html.

[4] 王成龙, 李诚, 冯毅萍, 等. 作业车间调度规则的挖掘方法研究[J]. 浙江大学学报:工学版, 2015, 49(3): 421-429,438.

[5] Fuendeling C U, Trautmann N. A priority-rule method for project scheduling with work-content constraints[J]. European Journal of Operational Research, 2010, 203(3): 568-574.

[6] 范华丽, 熊禾根, 蒋国璋, 等. 动态车间作业调度问题中调度规则算法研究综述[J]. 计算机应用研究, 2016, 33(3): 648-653.

[7] Durasevic M, Jakobovic D. A survey of dispatching rules for the dynamic unrelated machines environment[J]. Expert Systems with Application, 2018, 113: 555-569.

[8] 郑倩, 奚立峰. 飞机移动生产线作业调度问题的启发式算法[J]. 工业工程与管理, 2015, 20(2): 116-121.

[9] Mouelhi-Chibani W, Pierreval H. Training a neural network to select dispatching rules in real time[J]. Computers & Industrial Engineering, 2010, 58(2): 249-256.

[10] Vinyals O, Babuschkin I, Czarnecki W M, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning[J]. Nature, 2019, 575(7782): 350-354.

[11] Wang Q, Tang C. Deep reinforcement learning for transportation network combinatorial optimization: a survey[J]. Knowledge-Based Systems, 2021, 233: 1-22.

[12] Guo W, Tian W, Ye Y, et al. Cloud resource scheduling with deep reinforcement learning and imitation learning[J]. IEEE Internet of Things Journal, 2021, 8(5): 3576-3586.

[13] Zhao Y, Zhang H. Application of machine learning and rule scheduling in a job-shop production control system[J]. International Journal of Simulation Modelling, 2021, 20(2): 410-421.

[14] 王凌, 潘子肖. 基于深度强化学习与迭代贪婪的流水车间调度优化[J]. 控制与决策, 2021, 36(11): 2609-2617.

[15] 肖鹏飞, 张超勇, 孟磊磊, 等. 基于深度强化学习的非置换流水车间调度问题[J]. 计算机集成制造系统, 2021, 27(1): 192-205.

[16] Luo S. Dynamic scheduling for flexible job shop with new job insertions by deep reinforcement learning[J]. Applied Soft Computing, 2020, 91: 1-44.

[17] Christiano P F, Leike J, Brown T, et al. Deep reinforcement learning from human preferences[J]. Advances in Neural Information Processing Systems, 2017, 30: 1-10.

[18] Miller T, Niu J. An assessment of strategies for choosing between competitive marketplaces[J]. Electronic Commerce Research and Applications, 2012, 11(1): 14-23.

[19] Han B A, Yang J J. Research on adaptive job shop scheduling problems based on dueling double DQN[J]. IEEE Access, 2020, 8: 186474-186495.

第一作者：乔东平，男，1978年生，博士研究生，副教授，研究方向为智能优化算法、车间调度。E-mail：444517536@qq.com

通信作者：段绿旗，男，1997年生，硕士研究生，研究方向为数字化设计与制造。E-mail：715405101@qq.com

（编辑 李 静）

（收稿日期：2022-11-22）

文章编号：20230424

如果您想发表对本文的看法，请将文章编号填入读者意见调查表中的相应位置。