引文格式：郭羽, 唐敦兵, 张泽群. 基于深度强化学习的柔性作业车间调度方法[J]. 航空制造技术, 2024, 67(23/24): 114–120.
GUO Yu, TANG Dunbing, ZHANG Zequn. Flexible job-shop scheduling method based on deep reinforcement learning[J]. Aeronautical Manufacturing Technology, 2024, 67(23/24): 114–120.

基于深度强化学习的柔性作业车间调度方法*

郭　羽，唐敦兵，张泽群
（南京航空航天大学，南京 210016）

[摘要]　受到车间动态扰动的影响，单一调度规则在车间调度问题中无法一直获得较好的调度结果。对此，本文提出了一种基于 D3QN（Dueling double DQN）的调度方法，用于柔性作业车间调度问题。首先通过将调度问题转化为马尔可夫决策过程，构建了强化学习任务数学模型，并依次设计了 18 种生产系统状态特征、9 种用于评价机床和工件的分值动作以及与调度目标相关的奖励函数。然后基于 Dueling double DQN 算法，在机床 Agent、工件 Agent 与车间生产系统的交互过程中，不断训练两个 Agent 在每个调度决策时刻选择最高评分的机床和工件，从而完成工件和机床的资源分配任务。最后通过仿真试验，将所提出的方法与直接选取机床编号和选取调度规则的调度方法进行对比，结果表明该方法能取得更好的调度结果。

关键词：深度强化学习；柔性作业车间调度；神经网络；深度 Q 网络；奖励函数

Flexible Job-Shop Scheduling Method Based on Deep Reinforcement Learning

GUO Yu, TANG Dunbing, ZHANG Zequn
(Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

[ABSTRACT]　Affected by the dynamic disturbance of the workshop, a single scheduling rule cannot consistently obtain good scheduling results in the shop scheduling problem. To this end, a scheduling method based on dueling double DQN (D3QN) is proposed in this paper to solve the flexible job-shop scheduling problem. Firstly, by transforming the scheduling problem into Markov decision process, a mathematical model of reinforcement learning task was constructed, and 18 state features of production system, 9 scoring actions for evaluating machines and jobs, and reward functions related to scheduling objectives are designed respectively. Then, based on dueling double DQN algorithm, during the interaction of machine agent and job agent and workshop production system, the two agents are continuously trained to select the machine and job with the highest score at each scheduling decision-making time, so as to complete the resource allocation task of jobs and machines. Finally, through simulation experiments, the proposed method is compared with the scheduling method which directly selects the machine tool number and selects the scheduling rules. The results show that this method can obtain better scheduling results.

Keywords: Deep reinforcement learning; Flexible job-shop scheduling; Neural networks; Deep Q-network; Reward function
DOI: 10.16080/j.issn1671-833x.2024.23/24.114

随着全球工业化水平的不断提高，提升企业的生产效率、降低生产成本成为制造业发展的方向，车间调度问题作为关键问题也越来越受到人们的关注[1]。传统的作业车间调度本质上属于组合优化问题，是 NP-hard（非确定性多项式）问题。在作业车间调度的定义中，工件的每道工序仅能被同一台机床加工一次，因此需要解决工序的加工顺序问题。但是随着生产技术水平的不断提高，传统的作业车间调度问题已经不符合现在的生产环

* **基金项目：**国家重点研发计划（2018YFE0177000）；国家自然科学基金（52075257）；江苏省重点研发计划（BE2021091）；江苏省卓越博士后计划（2022ZB213）。

境，这使得引入柔性作业车间调度问题成为必然，相较于作业车间调度问题，柔性作业车间调度问题的每道工序既可以在同一台机床加工多次，也允许在多台机床上进行加工，而且不同机床的加工时间通常存在差异，这增加了求解的难度。在求解过程中，机床选择序列和工序选择序列的细微变化都会对结果产生较大的影响，但这更符合实际生产要求，同时也更具有研究意义。目前解决车间调度问题的主要方法有两类[2]，分别为最优化方法和近似方法。最优化方法可以求得全局最优解，但是求解速度无法保证；近似方法可以在短时间得到满意解，但是无法适应车间的动态扰动。

针对深度强化学习算法在调度问题中的应用，国内外诸多学者进行了大量研究。对于调度问题的执行方案主要分为两种，第 1 种是对于组合调度规则进行选取，第 2 种是对于机床编号进行选取。Bouazza 等[3]将部分柔性作业车间分解成分配机床和分配工序两种子问题，针对这两种子问题各采用 4 种调度规则，利用 Q 学习算法对规则的选择概率进行计算。Luo[4] 利用深度 Q 网络，将车间状态信息作为神经网络的输入，提出了 6 种复合调度规则，在工件加工完成和新工件到达时将待加工工件分配到可行的机床上。Liu 等[5]考虑了环境中的突发事件，将车间调度视为贯序决策问题，利用 Actor–Critic 算法构建出演员网络和评论家网络，将状态矩阵作为输入，简单调度规则作为输出。Han 等[6]对 DQN 算法进行改进，加入了优势网络和优先经验回放，并基于析取图模型对车间约束进行建模。Zhou 等[7]在 DQN 算法的基础上提出一种复合奖励机制，构造新型价值网络，以高维度状态数据作为输入，借助学习状态动作值函数，实现精准高效的实时决策。Zhou 等[8]针对动态车间，在强化学习算法中使用预测网络和目标网络分别学习预测值和目标值，以当前时刻最大等待时间为优化目标，用两种不同任务间隔概率的案例来说明所提出方法的有效性。Wang 等[9]将作业车间调度问题建模为马尔可夫决策过程（Markov decision process，MDP），利用近端策略优化（Proximal policy optimization，PPO）算法来解决作业车间动态调度问题，考虑了机床故障、工件再加工等动态及不确定因素，以最小化最大完工时间为优化目标。Kardos 等[10]利用多 Agent Q 学习算法将每个产品作为一个 Agent，当到达决策点时选择合适的机床进行加工，同时将产品和机床的状态作为输入。Zhou 等[11]针对小批量在线调度问题，将车间中的设备实体作为 Agent，利用多 Agent 强化学习算法，使每个设备都可以通过学习以往调度经验来与其他 Agent 进行协作。

本文同时训练两种 Agent 对每个决策点的机床和工件进行评分，分别解决柔性作业车间调度问题中的工序排序子问题和机床排序子问题。通过设计通用的车间环境状态空间，将对工件和机床的评分值离散化后作为动作输出，根据优化目标设定奖励值函数，训练工件 Agent 和机床 Agent，在每个决策点选取评分值最高的工件和机床作为当前调度方案。利用 Agent 与环境的不断交互，使 Agent 可以根据环境的变化即时做出反应，且同时具备一定的抗干扰能力。

1 柔性作业车间调度模型的构建

1.1 柔性作业车间调度约束模型

对于柔性作业车间调度问题的描述如下。假设有 n 个工件（$J_1, J_2, \cdots, J_n$）要在 m 台机床（$M_1, M_2, \cdots, M_m$）上加工，对于每一个工件，包含 h_i 道工序（$O_{i1}, O_{i2}, \cdots, O_{ih_i}$）；其中每道工序可以在多台同种类型的机床上加工；由于机器之间的差异性，每道工序在不同机床上的完工时间互不相同；为了使整个系统的特定性能指标达到最佳，需要完成机床和工件工序的安排，将每道工序放置在当前最适宜加工的机床上，并确定机床上每一道工序的加工序列和开始加工的时间。所以柔性作业车间包含两个子问题：确定工件的加工机床（机床选择子问题）和各机床上工件的加工顺序（工序排序子问题）。而对于作业车间调度问题，工件每道工序仅可以安排在一台机床上，所以柔性作业车间调度问题更加复杂，但更符合实际车间加工环境。

柔性作业车间加工过程中的约束条件：（1）每台机床在每一时刻只能加工一个工件；（2）每一个工件的每一个工序在同一时刻只能被一台机床加工；（3）每个工件一旦开始加工，到此道工序结束，中间不能暂停；（4）不同的工件之间的优先级相同；（5）同一个工件的工序先后顺序不能改变，只有当前一道工序加工完成后方可加工下一道工序；（6）在开始时刻，所有的工件都可以被加工。具体约束条件公式如下。

$$S_{ij}+x_{kij}\times t_{kij}\leqslant e_{ij} \tag{1}$$

$$e_{ij}\leqslant s_{i(j+1)},\ j=1, 2, 3, \cdots, h_i-1 \tag{2}$$

$$e_{ih_i}\leqslant C_{\max} \tag{3}$$

$$s_{ij}+t_{kij}\leqslant s_{pl}+L(1-y_{kijpl}) \tag{4}$$

$$e_{ij}\leqslant s_{i(j+1)}+L(1-y_{kpli(j+1)}) \tag{5}$$

$$\sum_{k=1}^{m_{ij}} x_{kij}=1 \tag{6}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{h_i} y_{kijpl}=x_{kpl} \tag{7}$$

$$\sum_{p=1}^{n}\sum_{l=1}^{h_p} y_{kijpl}=x_{kij} \tag{8}$$

$$s_{ij}\geqslant 0,\ e_{ij}\geqslant 0 \tag{9}$$

各变量符号含义见表 1。式（1）和（2）表示同一个

工件的后一道工序必须在前一道工序加工完成后才可以开始；式（3）表示所有工件完成加工的时间要小于等于总工件的最大完工时间；式（4）和（5）表示每台机床在每一时刻只能加工一个工件的其中一道工序；式（6）表示同一时刻同一道工序仅可以被一台机床加工；式（7）和（8）表示每一台机床有循环排列的操作；式（9）表示参数变量的值必须为非负数。

1.2 强化学习与马尔可夫决策过程

本文将作业车间调度问题抽象成 MDP 并进行数学建模。一个 MDP 一般由状态空间、动作空间、状态转移函数、奖励值函数等组成，由一个四元组来表示（S, A_s, P, R），其中 S 表示环境中状态空间的集合；A_s 表示 Agent 可以采用的动作空间集合；P 表示状态转移函数，当一个 Agent 对当前所观测到的环境采取动作 a 后，状态从 s 变到 s+1 的状态转移概率；R 则表示动作执行后环境反馈的奖励值。

在强化学习中，本文将 Agent 动作执行后得到的奖励值定义为（$r_1, r_2, \cdots, r_n$），将折扣回报定义为各时刻奖励值的折扣累加和，即

$$U_t = r_t + \gamma \times \sum_{k=t+1}^{n} \gamma^{k-t-1} r_k \tag{10}$$

式中，γ 为折扣率；r_t 为时间差分目标；r_k 为执行动作后得到的奖励值。

表 1 变量符号含义

Table 1 Variable sign meaning

符号	含义
n	工件总数
m	机床总数
i, p	工件索引，$i, p \in \{1, 2, \cdots, n\}$
j, l	工序索引，$j, l \in \{1, 2, \cdots, h_i\}$
k	机床索引，$k \in \{1, 2, \cdots, m\}$
h_i	第 i 个工件的工序总数
O_{ij}	第 i 个工件的第 j 道工序
M_{kij}	第 i 个工件的第 j 道工序在机床 k 上加工
m_{ij}	第 i 个工件的第 j 道工序的可选加工机床数
t_{kij}	第 i 个工件的第 j 道工序在机床 k 上加工的时间
s_{ij}	第 i 个工件的第 j 道工序加工的起始时间
e_{ij}	第 i 个工件的第 j 道工序加工的结束时间
C_i	每个工件的完成时间
$C_{\max}$	最大完工时间
x_{kij}	如果工序 O_{ij} 在机床 k 上加工为 1，否则为 0
y_{kijpl}	在机床 k 上如果工序 O_{ij} 在 O_{pl} 之前加工为 1，否则为 0
L	足够大的正数

定义动作价值函数和最优动作价值函数为

$$Q_\pi(s_t, a_t) = \mathbb{E}[U_t | S_t = s_t, A_t = a_t] \tag{11}$$

式中，S_t、s_t 为 t 时刻的状态及其观测值；A_t、a_t 为 t 时刻的动作及其观测值。

$$Q_*(s_t, a_t) = \max_\pi Q_\pi(s_t, a_t), \forall s_t \in S, a_t \in A \tag{12}$$

通过以上公式可以推导出最优贝尔曼方程：

$$Q_*(s_t, a_t) = \mathbb{E}_{s_{t+1} \sim p(s_t, a_t)}[R_t + \gamma \times \max_{a \in A} Q_*(s_{t+1}, A) | S_t = s_t, A_t = a_t] \tag{13}$$

Agent 在执行动作 a_t 后，环境将根据 $p(s_{t+1}|s_t, a_t)$ 计算新状态 s_{t+1}，并且观测到这次动作的奖励值 r_t，将结果记录到四元组中（s_t, a_t, r_t, s_{t+1}），于是可得近似结果为

$$Q_*(s_t, a_t) \approx r_t + \gamma \times \max_{a \in A} Q_*(s_{t+1}, a) \tag{14}$$

将式（14）中得到的最优动作价值函数替换成神经网络的形式，得到

$$Q(s_t, a_t; \omega) \approx r_t + \gamma \times \max_{a \in A} Q(s_{t+1}, a; \omega) \tag{15}$$

式中，ω 为神经网络参数。

本文采取的是基于 DQN 的强化学习算法，需要考虑算法存在的高估问题，这种高估对 Agent 的所有动作是非均匀的，会严重影响算法的性能，所以通过使用 Double DQN 加上 Dueling network 来进一步提高算法的准确性。Agent 的第 1 步是进行选择（式（15）），第 2 步计算求值时使用目标网络（式（16））。

$$a^* = \operatorname{argmax}_{a \in A} Q(s_{t+1}, a; \omega) \tag{16}$$

$$\hat{y}_t = r_t + Q(s_{t+1}, a^*; \omega^-) \tag{17}$$

同时使用 $D(s, a; \omega^D)$ 作为最优优势函数 $D_*(s, a)$ 的近似，$V(s; \omega^V)$ 作为最优状态价值函数 $V_*(s)$ 的近似，可以将最优动作函数进一步写为

$$Q(s, a; \omega) \triangleq V(s; \omega^V) + D(s, a; \omega^D) - \text{mean}_{a \in A} D(s, a; \omega^D) \tag{18}$$

$Q(s, a; \omega^D)$ 和 $V(s; \omega^V)$ 共享部分全连接层，先将输出分为每个动作的优势值和状态的价值，最终合并输出每个动作的动作价值。

2 基于 Dueling double DQN（D3QN）的车间调度算法设计

2.1 状态空间

对于车间状态来说，需要从总体和局部两方面综合考虑，以便 Agent 能够全面地观测到车间的所有状态。本文综合考虑了 18 种状态，涉及工件、机床、订单队列和缓冲区的状态信息，由于这些状态之间的数据范围和量纲都不相同，如果不加处理直接作为算法输入可能会不适用于未经过训练的数据。所以为了提高算法的泛化性和通用性，采用归一化的方式将这些状态信息映射到 [0,1] 的区间，将这些状态信息组合成状态向量 $\boldsymbol{s}$=[s_1,

s_2, …, s_{18}]，具体状态信息如表 2 所示。

2.2 动作空间

动作空间为 Agent 可选的动作集合，本文将整个调度问题分解为两个子问题，工件选择当前工序可以加工的机床和最早可用的机床选择缓冲区内的工件。本文算法的训练目的是对待选择机床和工件分别进行评分，从候选结果中选择得分较高的机床和工件作为下一步要执行的动作。设定分值都为正数，将分值从 10~90 平均分成 9 段，每隔 10 分为一段，划分出 9 个动作，a=[a_1, a_2, a_3, …, a_9]，分值作为对待选择机床或工件的评分值，分值越高代表选择当前机床或工件的调度结果越好，越能满足调度指标。

2.3 奖励值函数

奖励函数的设定与算法最终的优化目标有着密切的关联，本文要优化的调度目标为最小化最大完工时间，是衡量调度性能的最根本指标，可以体现车间的生产效率，可表示为

$$f_1 = \min(\max_{1\leqslant i\leqslant n}(C_i)) \tag{19}$$

在解决两个子问题时，需要采取不同的奖励函数，首先在工件选择待加工机床时，要充分考虑各台机床的负荷情况，瓶颈设备就是负荷最大的机床，要使机床的负荷平衡且尽量小，即时奖励值为

$$p_t = \frac{\max_{1\leqslant k\leqslant m}\sum_{i=1}^{n_k}\sum_{j=1}^{h_{ki}} t_{kij}x_{kij}}{\sum_{k=1}^{m}\sum_{i=1}^{n_k}\sum_{j=1}^{h_{ki}} t_{kij}x_{kij}} \tag{20}$$

$$r_t = -(P_t - P_{t-1}) \tag{21}$$

式中，n_k 表示已加工的工件数；h_{ki} 表示已加工工件的工序数。

在机床选择缓冲区内的工件时，采用动作执行之后的平均机床利用率之差来计算 [12]，平均机床利用率为

$$U_t = \frac{1}{m}\sum_{k=1}^{m}\frac{\sum_{i=1}^{n_k}\sum_{j=1}^{h_{ki}} t_{kij}}{t_{\max}} \tag{22}$$

式中，$t_{\max}$ 表示到达当前决策时间点的最长时间，则即时奖励函数定义为

$$r_t = U_t - U_{t-1} \tag{23}$$

2.4 探索和利用

在强化学习中不仅要利用已经学习到的经验，从已知动作中选择下一步动作，最大限度地提高累计回报值，还要尝试探索更多的环境信息，尝试未试过的行为，以免陷入局部最优结果。大多数的强化学习算法中会使用 ε-greedy 算法，即

$$a = \begin{cases} \arg\max_a Q(a), & p=\varepsilon \\ \text{random}, & p=1-\varepsilon \end{cases} \tag{24}$$

式中，p 为在 0~1 之间生成的随机数值，以 ε（探索的概率）的概率选取当前价值最大的动作，以 1–ε（利用的概率）的概率从动作空间中随机选取动作，作为当前 Agent 的动作。

2.5 车间调度框架设计及模型更新流程

本文使用 Dueling double DQN 算法对车间调度问题进行求解，同时训练两个 Agent，分别对待选工件和机床进行评分，在所有候选集中选择分值最高的机床和工件，作为下一步的调度方案，从而完成工件和机床的资源分配任务，算法的更新流程如下。

算法 1：基于 Dueling double DQN 算法的柔性作业车间调度算法。

初始化：经验回放池大小 N、奖励折扣因子 γ、更新目标状态动作值函数的延迟步长 C、贪婪策略中的 ε、批量抽取大小 k、迭代次数 M。

使用权重 ω 初始化动作价值函数 $Q(s, a; \omega)$。

使用权重 ω^- 初始化动作价值函数 $Q(s, a; \omega^-)$。

for episode = 1 to M do

初始化车间环境，清除调度结果，获取观测值 s 并进行归一化预处理。

for t =1 to T do

根据 ε 的概率选择动作 a_t=argmax$_a Q(a)$，否则从

表 2 车间环境的状态特征

Table 2 State characteristics of job-shop environment

状态特征	含义
s_1	剩余待加工的工件总数
s_2	剩余待加工的工序总数
s_3	剩余待加工的工序总时间
s_4	工件平均完成率
s_5	工件完成率的标准差
s_6	工件当前工序最小加工时间 / 加工总时间
s_7	工件当前工序平均加工时间 / 加工总时间
s_8	工件当前工序最大加工时间 / 加工总时间
s_9	已加工工件 / 总工件数
s_{10}	机床的平均利用率
s_{11}	机床的利用率标准差
s_{12}	待分配机床的利用率
s_{13}	待分配机床当前工序加工时间
s_{14}	待分配机床剩余加工时间
s_{15}	缓冲区中工件平均完成率
s_{16}	缓冲区中工件利用率标准差
s_{17}	待分配工件当前工序的加工时间
s_{18}	待分配工件剩余加工工序数

动作空间随机选择动作。

在调度环境中执行动作 a_t，选择评分最高的工件或机床，将工件放入机床缓冲区或选择工件加工。

观测奖励值 r_t 和新状态 s_{t+1} 并进行归一化预处理。如果 episode 结束，$d_t=1$，否则 $d_t=0$。

将经验值 $(s_t, a_t, r_t, s_{t+1}, d_t)$ 存入经验回放池中。

从经验池中随机采样批量为 k 的状态转移数据。

如果 $d_t=0$：

$y_t=r_t+\gamma Q(s_{t+1}, \text{argmax}_a Q(s_{t+1}, a; \omega), \omega^-)$

否则：

设置 $y_t=r_t$

执行梯度下降并进行更新。

每隔 C 步对目标网络 $Q(s, a; \omega^-)$ 进行更新 $\omega^- \leftarrow \omega$

如果 episode 结束，跳出循环。

end for

end for

车间调度流程如图 1 所示，在流程开始后，首先初始化车间状态和其他超参数，将机床加工记录清除，并把订单中的工件加入生产订单队列，等待分配加工设备。然后判断是否达到迭代设定阈值，如果达到则算法流程结束，保存训练模型，否则继续执行训练流程。在车间调度过程中，算法的核心是如何同时训练工件评分和机床评分两个 Agent，两个 Agent 同时参与执行调度任务，如果各自没有达到决策点，Agent 会阻塞当前进程，直到出现下一个决策点。当机床前的缓冲区有空余的位置时，唤醒阻塞的机床 Agent 对订单队列里下一个工件当前工序的所有可加工机床进行评分，选出动作 a_t，比较所有候选机床的分值，选定分值最大的机床，将工件分配到该机床上，将当前经验值 (s_t, a_t, r_t, s_{t+1}) 存储到经验回放池中，如果出现分数相同的情况，则随机选取。当机床结束当前工序的加工任务时，唤醒工件 Agent，对机床缓冲区前的工件进行评分，同样选取分值最大的工件进行下一步加工，将经验值存入回放池中。最后当经验池达到足够多的样本后，从中随机抽取出固定批量的样本，计算样本的时序差分误差，对神经网络参数执行梯度下降法更新，当达到设定的 C 步后，重设 Target 网络的参数值。本文使用经验回放的方式来打破序列之间的相关性，因为在训练 Agent 对工件和机床进行评分的过程中，搜集到的两个相邻四元组 (s_t, a_t, r_t, s_{t+1}) 和 $(s_{t+1}, a_{t+1}, r_{t+1}, s_{t+2})$ 之间有很强的相关性，所以加入经验回放来消除相关性对算法训练结果的影响。

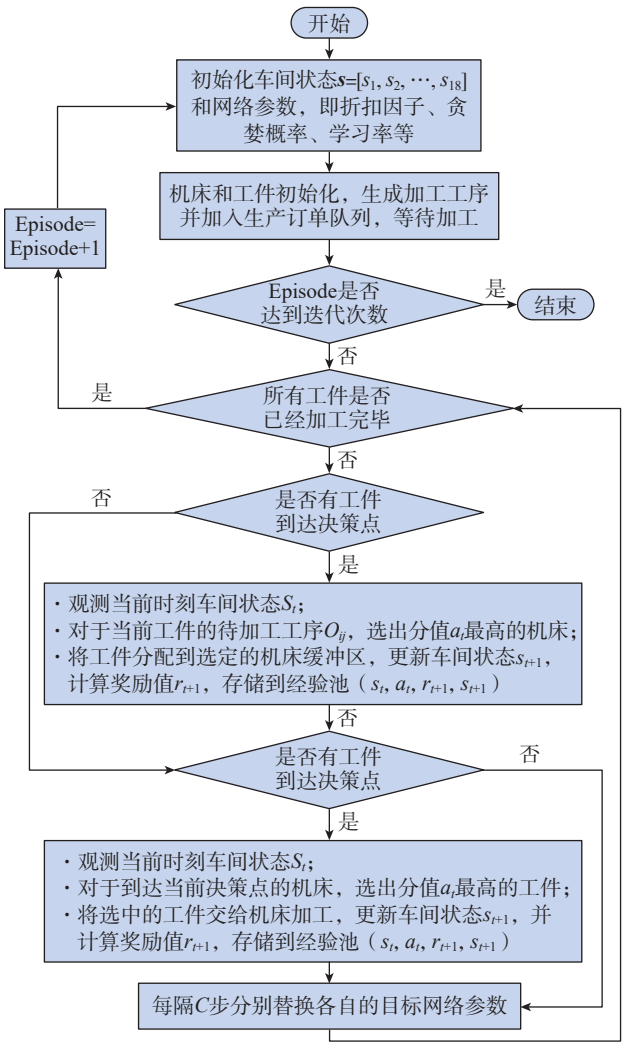


图 1　调度算法流程图

Fig.1　Flow chart of scheduling algorithm

3　实例验证

3.1　算法参数设置

本文使用 Python 3.6 与 Tensorflow 2.0 框架作为深度强化学习算法试验环境，硬件配置为 Intel（R）Core（TM）i5–6300HQ CPU @2.30 GHz，RAM 16 GB，NVDIA GeForce GTX 960 M，部分算法参数见表 3。

本文的神经网络使用了全连接神经网络，包含 3 个隐藏层，节点数为 100、50、30，在第 1 层输出后加入 Sigmoid 激活函数，后两层的输出都加入 Relu 激活函数，

表 3　算法参数设置（部分）

Table 3　Algorithm parameters setting (partly)

参数	设定值
迭代次数	8000
经验池大小	100000
折扣因子	0.95
ε 贪婪概率	0.9
网络更新学习率	0.0001
批量采样大小	512
目标网络更新频率	300

价值网络和优势动作网络共享相同的全连接神经网络层，神经网络结构图如图 2 所示。

3.2 案例分析和试验对比

本文使用的车间试验环境包含 3 种加工设备，分别为车床、铣床和钻床，每种机床各 4 台，共 12 台机床，工件的同种工序在不同机床上的加工时间各不相同，设定工序额定加工时间乘以加工系数为机床实际加工时间。加工的工件种类有 3 种，每种工件有 3 种或 4 种工序，设定工件的到达时间相同，且各机床的缓冲区容量都为 4。车间环境参数如表 4 所示，算法的迭代次数设定为 1000 次，3 种工件个数分别随机生成，共 30 个工件进行初步训练，机床 Agent 和工件 Agent 的累计回报值的结果平滑曲线如图 3 所示。可以看出，机床评分 Agent 在大约 750 次迭代之后达到稳定，工件评分 Agent 在大约 700 次迭代后收敛。由于训练过程中的贪婪概率始终存在，在 Agent 做选择时有一定概率会随机从动作空间中选择，所以后期的迭代依然存在波动，但并不影响算法整体的收敛性能。最大完工时间的收敛结果如图 4 所示，在 600 次迭代后趋于平缓，说明两个 Agent 都学习到了如何有效分配车间的资源，验证了本文算法的有效性。

为了分析算法在不同规模订单下的求解情况，在上述工件的工序数量和种类不变的情况下，将工件的规模设定为 10 个、20 个、30 个、50 个、100 个和 200 个并分别进行 10 次独立的仿真试验，对最终的完工时间取平均值，并与其他两种基于 DQN 算法的调度方案进行对比。其中，方法 1 对于组合调度规则进行（SDR）：选取 SPT、LPT、FIFO、LIFO、SRPT、LRPT 6 种调度规则作为动作空间；方法 2 对于机床编号进行选取（SMN），将机床的编号作为动作空间，通过选取编号完成机床分配工件，机床则采取先进先出的规则来选择要加工的工

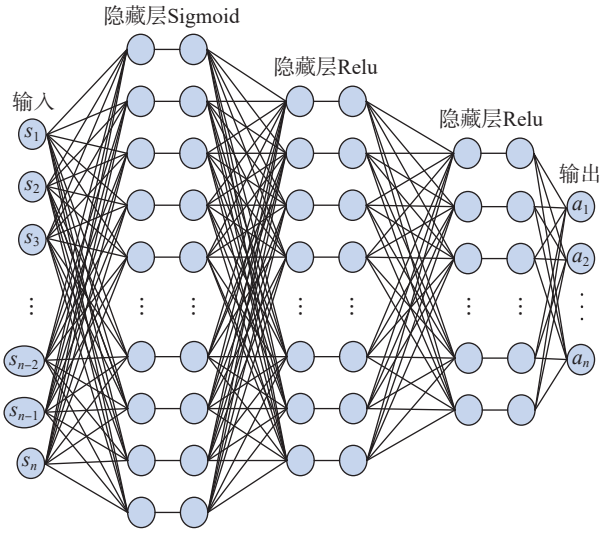


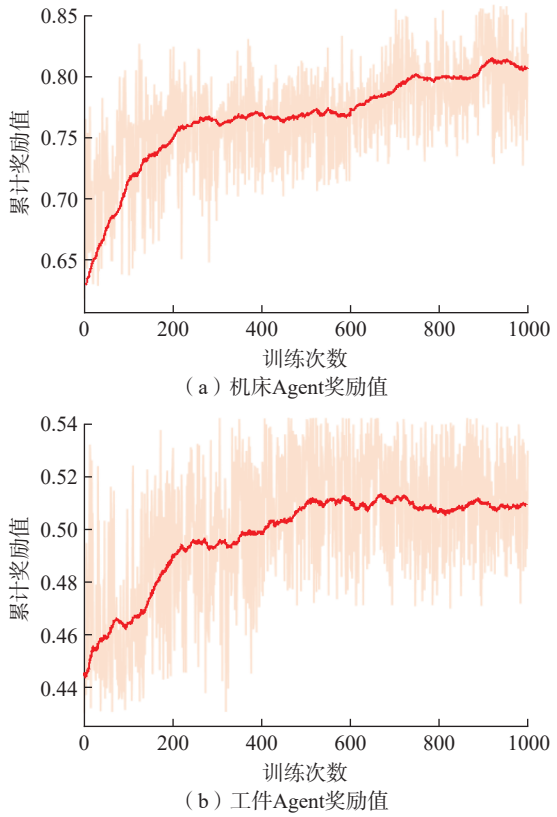
图 2 神经网络结构图

Fig.2 Structure diagram of neural network

表 4 车间环境参数

Table 4 Job-shop environment parameters

参数	值
工件种类 J_0 工序列表及时间	$\{O_1: 150, O_2: 22, O_3: 34, O_4: 100\}$
工件种类 J_1 工序列表及时间	$\{O_1: 37, O_3: 300, O_2: 20\}$
工件种类 J_2 工序列表及时间	$\{O_2: 125, O_1: 20, O_3: 70, O_2: 150\}$
车床加工系数	$\{M_1: 1.0, M_2: 1.2\}$
铣床加工系数	$\{M_3: 0.8, M_4: 1.0\}$
钻床加工系数	$\{M_5: 1.1, M_6: 1.2\}$



（a）机床Agent奖励值

（b）工件Agent奖励值

图 3 机床 Agent 和工件 Agent 奖励值曲线

Fig.3 Reward value curves of machine tool Agent and workpiece Agent

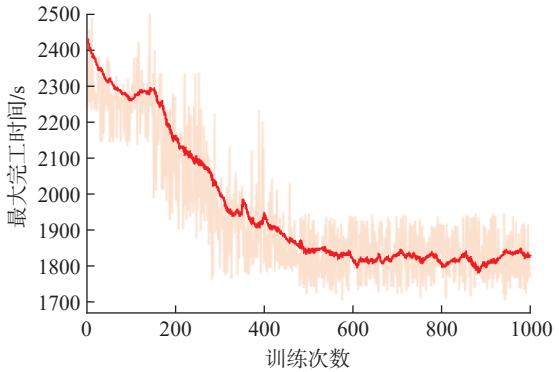


图 4 最大完工时间收敛曲线

Fig.4 Convergence curve of maximum completion time

件。最终得到的结果如图 5 所示，可以看出在相同的试验环境中，本文算法通过对工件和机床的组合分配，可以达到更优的解，在不同规模的问题上都表现较好。

本文的算法考虑到在工件调度到机床后，并不一定立刻开始加工，而是先放到缓冲区，然后根据当前车间环境状态选择此刻适宜加工的工件，这样就会有更灵活的选择方案，从而得到更优的调度结果。同时本文也考虑到车间中会出现机床故障，导致加工无法进行的情况，同样在机床发生故障概率为 10%~60% 的情况下进行 10 次试验，选定工件的数量为 30 个，并设定机床的恢复时间相同，一旦恢复即可开始加工，得到的结果取平均值，如图 6 所示。结果表明本文算法在车间发生动态扰动的情况下可以快速产生反应并得到一个较优的结果。

4 结论

本文针对柔性作业车间提出了基于 Dueling double DQN 的强化学习车间调度算法，将调度问题转化为马尔可夫决策过程，同时训练机床 Agent 和工件 Agent 对当前待选工件和机床进行评分，从而完成机床和工件的分配。设计了 18 种通用的车间环境状态并归一化处理，评分值作为动作空间，以最小化最大完工时间作为最终目标来设计奖励函数。经过多次试验表明，本文算法在不同规模的订单下都可以取得很好的效果，结果都优于选择组合调度规则和直接选择加工机床两种调度方式。进一步的研究工作主要集中在对分值进行更细粒度地划分、采用基于策略的强化学习算法（如 A–C、PPO）等；本文以最小化工件的最大完工时间作为优化目标，可以针对多种目标和动态调度等方面进行下一步的研究。

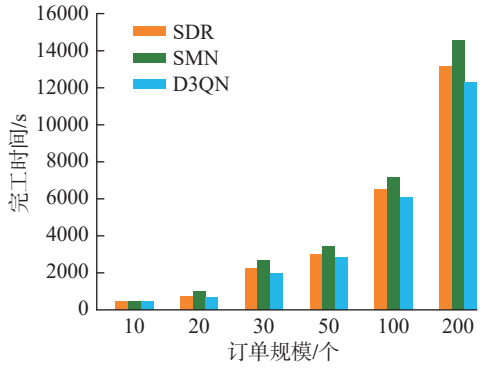


图 5 不同订单规模下的完工时间

Fig.5 Completion time under different order sizes

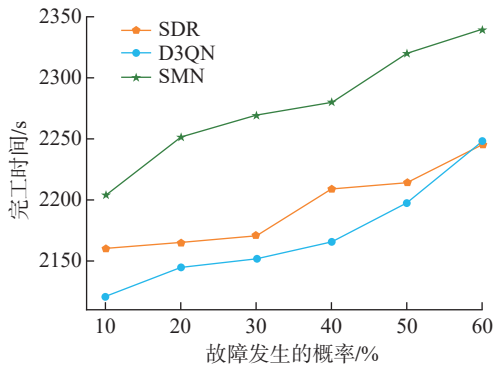


图 6 不同故障概率下的完工时间

Fig.6 Completion time under different failure probabilities

参 考 文 献

[1] 罗梓珲, 江呈羚, 刘亮, 等. 基于深度强化学习的智能车间调度方法研究[J]. 物联网学报, 2022(1): 53–64.

LUO Zihui, JIANG Chengling, LIU Liang, et al. Research on deep reinforcement learning based intelligent shop scheduling method[J]. Chinese Journal on Internet of Things, 2022(1): 53–64.

[2] 钟敬伟, 石宇强. 基于DQN的智能工厂作业车间调度[J]. 现代制造工程, 2021(9): 17–23, 93.

ZHONG Jingwei, SHI Yuqiang. Job shop scheduling based on DQN algorithm in intelligent factory[J]. Modern Manufacturing Engineering, 2021(9): 17–23, 93.

[3] BOUAZZA W, SALLEZ Y, BELDJILALI B. A distributed approach solving partially flexible job-shop scheduling problem with a Q-learning effect[J]. IFAC-PapersOnLine, 2017, 50(1): 15890–15895.

[4] LUO S. Dynamic scheduling for flexible job shop with new job insertions by deep reinforcement learning[J]. Applied Soft Computing, 2020, 91: 106208.

[5] LIU C L, CHANG C C, TSENG C J. Actor–Critic deep reinforcement learning for solving job shop scheduling problems[J]. IEEE Access, 2020, 8: 71752–71762.

[6] 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.

[7] ZHOU T, TANG D B, ZHU H H, et al. Reinforcement learning with composite rewards for production scheduling in a smart factory[J]. IEEE Access, 2020, 9: 752–766.

[8] ZHOU L F, ZHANG L, HORN B K P. Deep reinforcement learning-based dynamic scheduling in smart manufacturing[J]. Procedia CIRP, 2020, 93: 383–388.

[9] WANG L B, HU X, WANG Y, et al. Dynamic job-shop scheduling in smart manufacturing using deep reinforcement learning[J]. Computer Networks, 2021, 190: 107969.

[10] KARDOS C, LAFLAMME C, GALLINA V, et al. Dynamic scheduling in a job-shop production system with reinforcement learning[J]. Procedia CIRP, 2021, 97: 104–109.

[11] ZHOU T, TANG D B, ZHU H H, et al. Multi-agent reinforcement learning for online scheduling in smart factories[J]. Robotics and Computer-Integrated Manufacturing, 2021, 72: 102202.

[12] 李宝帅, 叶春明. 深度强化学习算法求解作业车间调度问题[J]. 计算机工程与应用, 2021, 57(23): 248–254.

LI Baoshuai, YE Chunming. Job shop scheduling problem based on deep reinforcement learning[J]. Computer Engineering and Applications, 2021, 57(23): 248–254.

通讯作者：唐敦兵，教授，研究方向为智能制造系统、制造系统与自动化、数字化设计与制造。

（责编 七七）