第 36 卷 第 2 期 中 国 机 械 工 程 Vol.36 No.2
2025 年 2 月 CHINA MECHANICAL ENGINEERING pp.280-293

考虑峰值功率受限约束的柔性作业车间调度研究

李益兵[1,2] 曹 岩[1] 郭 钧[1,2*] 王 磊[1,2] 李西兴[3] 孙利波[4]

1.武汉理工大学机电工程学院,武汉,430070
2.数字制造湖北省重点实验室,武汉,430070
3.湖北工业大学机械工程学院,武汉,430068
4.天津水泥工业设计研究院有限公司,天津,300400

摘要:针对车间峰值功率受限约束下的柔性作业车间调度面临的作业周期增加、机器负荷增大的问题,建立以最小化最大完工时间和最小化机器最大负载为优化目标、考虑车间峰值功率约束的柔性作业车间调度问题(PPCFJSP)模型。为更好地调度决策,首先将该问题转化为马尔可夫决策过程,基于此设计了一个结合离线训练与在线调度的用于求解 PPCFJSP 的调度框架。然后设计了一种基于优先级经验重放的双重决斗深度 Q 网络(D3QNPER)算法,并设计了一种引入噪声的 ε-贪婪递减策略,提高了算法收敛速度,进一步提高了求解能力和求解结果的稳定性。最后开展实验与算法对比研究,验证了模型和算法的有效性。

关键词:柔性作业车间调度;马尔可夫决策过程;深度强化学习;峰值功率受限

中图分类号:TH165;TP18

DOI:10.3969/j.issn.1004-132X.2025.02.011 **开放科学(资源服务)标识码(OSID)**:

Research on Flexible Job-shop Scheduling Considering Constraints of Peak Power Constrained

LI Yibing[1,2] CAO Yan[1] GUO Jun[1,2*] WANG Lei[1,2] LI Xixing[3] SUN Libo[4]

1.School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan, 430070
2.Hubei Key Laboratory of Digital Manufacturing, Wuhan University of Technology, Wuhan, 430070
3.School of Mechanical Engineering, Hubei University of Technology, Wuhan, 430068
4.Tianjin Cement Industry Design & Research Institute Co., Ltd., Tianjin, 300400

Abstract: Peak power constrained flexible job shop scheduling problem(PPCFJSP) model was established to address the challenges of increased work cycles and increased machine load in flexible job shop scheduling under the constraints of peak power in the workshops. The optimization objectives were to minimize the maximum completion time and the maximum machine loads, taking into account the constraints of peak power in the workshops. For better scheduling decisions, firstly, the problem was transformed into a Markov decision process, then, a scheduling framework combining offline training and online scheduling was designed for solving PPCFJSP. Secondly, a double dueling deep q-network based on priority experience replay(D3QNPER) algorithm was designed based on priority experience replay, and a ε- greedy descent strategy introducing noise was designed to improve the convergence speed of the algorithm, further enhance the solving ability and stability of the solution results. Finally, experimental and algorithmic comparative studies were conducted to verify the effectiveness of the model and algorithm.

Key words: flexible job shop scheduling; Markov decision process; deep reinforcement learning; peak power constrained

0 引言

在当今全球气候变化和环境保护的背景下,“双碳”战略已成为国家发展战略的重要组成部分。随着“双碳”战略的深入实施,各地区开始逐步推行“双控”政策,即对能耗总量和能耗强度进行双重控制,以实现节能降耗和减排目标。这些政策措施对高耗能行业产生了深远影响,近年我国多地发布迎峰度夏冬有序用电指导方案,限电措施也从化工、冶金行业转向全体商业、工业,限电限产已成为一些地区常见的现象。企业在面临停电限荷的情况下,如何优化生产调度以应对突发的电力供应问题,成为企业管理者必须面对的新问题。管理者需要快速作出调整生产计划的决

收稿日期:2024-01-22
基金项目:国家自然科学基金(52305552,52375510)

策，这不仅要求调度决策能够迅速响应变化，还需要对生产过程有深入的理解和预测。同时，停电导致的作业周期增加问题也不容忽视。由于电力供应不稳定，企业可能不得不延长某些产品的生产周期，这直接影响到交货期和客户满意度。此外，由于用电功率限制，车间中全部机器无法同时作业，导致个别机器的负荷增大，会导致机器过度磨损甚至故障，增加了维护成本和生产风险，故设计一种高效、稳定、泛化能力强的车间调度方法具有紧迫性和现实意义。

随着机器学习技术的不断进步，深度强化学习(deep reinforcement learning，DRL)将深度学习的表征学习能力与强化学习的决策求解能力相结合，使得强化学习技术逐渐变得实用化。为了更好地作出决策，DRL 算法被应用在多种组合优化问题的求解中，并在车间调度领域展现出色的性能[1-2]，而且 DRL 弥补了整数规划、基于规则和元启发式方法无法利用历史学习经验预测调度决策的不足。在处理决策响应及预测问题和完工时间与机器负荷平衡问题上，黎声益等[3]提出了一种面向设备负荷稳定的智能车间调度方法，利用 Double DQN 解决半导体车间动态事件下设备负荷的稳定调度。贺俊杰等[4]以加权完工时间和为目标，提出了一种基于长短期记忆近端策略优化(proximal policy optimization with long short-term memory，LSTM-PPO)强化学习的在线调度方法，通过设计融合 LSTM 的智能体记录车间的历史状态变化和调度策略，实现了智能体根据状态信息进行在线调度。LIU 等[5]和 LI 等[6]分别提出了基于深度强化学习的动态柔性车间调度方法，用于处理不确定性和限制资源的情况，取得了良好的性能。WU 等[7]采用深度强化学习方法解决了过程规划中的动态加工资源调度问题，通过蒙特卡罗方法和深度学习算法评估和改进了过程策略。LEE 等[8]和 HE 等[9]提出了基于多智能体强化学习的纺织制造和半导体制造过程优化方法，通过引入深度 Q 网络和多智能体学习实现了多目标优化。郭具涛等[10]提出了一种基于复合规则和强化学习的调度方法用于求解混流装配线的平衡与排序问题，实现了复合规则权值参数的调控优化。刘亚辉等[11]解决了航天结构件生产过程中柔性作业车间面临的动态调度问题，提出了感知-认知双系统驱动的双环深度 Q 网络方法，通过感知和认知系统提高了调度决策对知识图谱的利用。ZHANG 等[12]提出了一种基于多代理图的深度强化学习的柔性作业车间调度模型(deep reinforcement learning with multi-agent graphs，DeepMAG)，通过将不同的智能体关联到每台机器和作业，将 DRL 与多智能体强化学习(multi-agent reinforcement learning，MARL)集成在一起共同作用完成决策。GUI 等[13]针对动态柔性车间调度问题，提出了一种具有复合调度动作的马尔可夫决策过程，设计了由单个调度规则和连续权重变量聚合的复合调度动作，以提供连续的规则空间和单一调度规则权重选择。ZHANG 等[14]针对机器加工时间不确定的动态柔性作业车间调度问题，采用近端策略优化算法对模型进行求解，使用处理信息矩阵作为网络输入，通过图神经网络将一些高级状态嵌入车间中，使得智能体能够学习环境的完整状态。由此可见，DRL 算法在不同领域的调度问题中得到了广泛应用，展现了出色的性能，使得历史学习经验得到充分利用，应用 DRL 的车间调度领域也在不断扩大。

近年来，越来越多的研究人员也将 DRL 用于处理考虑能耗约束的柔性作业车间调度。例如，何彦等[15]针对车间调度中柔性工艺路线对调度能耗的影响特性，使用改进的 Q 学习算法求解节能调度模型并得到 Pareto 解。DU 等[16]设计了 12 个状态特征和 7 个动作来描述调度过程中的特征，使用 DQN 算法对具有起重机运输和安装时间的柔性作业车间调度问题(multiobjective FJSP with crane transportation and setup times，FJSP-CS)进行了有效求解，对完工时间和能耗进行了同时优化。NAIMI 等[17]提出了一种结合能量和生产率目标的机器故障环境下柔性作业车间问题的 Q 学习重调度方法，使得系统能够对意外事件作出快速反应，实现了对制造跨度和能耗变化的同步优化。LI 等[18]在求解具有 2 型模糊处理时间的 FJSP(energy-efficient FJSP with type-2 processing time，ET2FJSP)时，为了更好地模拟绿色柔性车间调度实际生产，设计了一种基于学习的参考向量模因算法(learning-based reference vector memetic algorithm，LRVMA)，实现了对时间约束的不确定性预测。

当前，针对具有能耗约束的柔性作业调度问题，相关研究多将总能耗或总成本作为目标函数，这样可以得到总能耗或总成本与完工时间的“最优前沿解”。然而，在当前迎峰度夏冬有序用电各类管理措施中一般以用电负荷(即总功率)来对能耗进行描述。一旦执行限电要求，如果仍以此前研究中的总能耗作为约束，便无法应对峰值功率

受限的情形，会导致车间功率峰值居高不下、执行限电要求不力，企业面临更大损失，因此，对于考虑峰值功率约束的生产调度问题，还需要进一步研究和关注。为了在峰值功率约束条件下优化生产效率，提高决策响应能力，本文提出了一种考虑峰值功率受限的柔性作业车间调度问题（peak power constrained flexible job shop scheduling problem，PPCFJSP）模型，主要研究内容包括：①建立了一个基于马尔可夫决策过程的符合当前产业环境需求的峰值功率受限柔性车间调度问题模型，设计了一个用于求解 PPCFJSP 的 DRL 调度框架。②设计了一种结合离线训练与在线调度的基于优先级经验重放的双重决斗深度 Q 网络（double dueling deep q-network based on priority experience replay，D3QNPER）算法，用来求解 PPCFJSP 模型。同时，设计了一种引入噪声的 ε-贪婪递减策略，提高了算法收敛速度，进一步提高了求解能力和求解结果的稳定性。③进行仿真实验分析，与不同调度规则和深度强化学习算法进行对比，用实验结果来证明本文算法的有效性。

1 问题描述及数学模型

本文提出的 PPCFJSP 模型主要研究 n 个工件在设有用电负荷上限的车间 m 台机器上加工，每个工件均有多道工序，同一工件的各道工序的先后关系不能发生改变。同时，还需要满足以下约束：①某一时刻一个工件只能有一道工序被加工；②工件的任一工序在同一时刻只能被一台机器加工；③车间存在峰值功率限制，在任一时刻运行设备叠加功率不能超过峰值功率限制；④任一工件的工序在加工过程中不能被中断；⑤认为车间中各机器的加工功率不随外部条件变化。

由于加工所需的最大完工时间和机器负载情况均会随机器的选择而变化，同时考虑到车间设有用电峰值功率上限，还会导致车间中个别功率小的机器负载情况加重，影响机器使用寿命，故本文以考虑车间功率峰值约束时最小化最大完工时间和最小化机器的最大负载为优化目标。本文所使用的符号定义见表 1。

本文的优化目标有两个：一是最小化最大完工时间 C_{T}，二是最小化最大机器负载 W_{T}，即

$$C_{\mathrm{T}} = \min \max(C_i) \tag{1}$$

$$W_{\mathrm{T}} = \min \max_{k\in\{1,2,\cdots,m\}} \sum_{i=1}^{n}\sum_{j=1}^{j_i}(C_{ijk}X_{ijk}) \tag{2}$$

在不考虑待机功耗且加工过程的功率不变的情况下，最小化最大机器负载可以用机器处于加工状态的总耗时来表示。本文的输出结果为两个优化目标的 Pareto 前沿上取得最优解的集合。约束条件表示如下：

表 1 本文使用的符号定义

Tab.1 Definitions of symbols used in this paper

符号	含义
m	设备数量
n	待加工件总数
i	工件编号，$i \in \{1,2,\cdots,n\}$
j	工序编号，$j \in \{1,2,\cdots,j_i\}$
k	机器编号，$k \in \{1,2,\cdots,m\}$
j_i	第 i 个工件的总工序数
O_{ij}	第 i 个工件的第 j 道工序
X_{ijk}	判断 O_{ij} 是否在机器 k 上加工，$X_{ijk}=\begin{cases}1 & O_{ij}\text{ 在机器 }k\text{ 上加工}\\0 & O_{ij}\text{ 不在机器 }k\text{ 上加工}\end{cases}$
M_{ij}	第 i 个工件的第 j 道工序使用的设备集
m_{ij}	第 i 个工件的第 j 道工序所使用的设备
L_{ijt}	t 时刻第 i 个工件的第 j 道工序所使用的设备编号
S_{ij}	第 i 个工件的第 j 道工序的开工时间
F_{ij}	第 i 个工件的第 j 道工序的完工时间
S_{ijk}	第 i 个工件的第 j 道工序在机器 k 上的开工时间
T_{ijk}	第 i 个工件的第 j 道工序在机器 k 上的加工时间
F_{ijk}	第 i 个工件的第 j 道工序在机器 k 上的完工时间
C_i	第 i 个工件的末道工序的完工时刻
C_{ijk}	第 i 个工件的第 j 道工序在机器 k 上的时间
W_i	第 i 个设备的加工总时间
W_k	机器 k 的总负荷
P_{W_k}	机器 k 的加工功率
P_t	t 时刻运行机器的峰值功率总和
P_u	车间功率上限

$$S_{ij} \geqslant F_{(i-1)j}\ (i>1) \tag{3}$$

$$L_{ijt_1} \neq L_{ijt_2} \rightarrow t_1 \neq t_2 \tag{4}$$

$$P_t = \sum_{k=1}^{m} P_{W_k} \cdot l(\exists\, i \in 1,2,\cdots,n, \exists\, j \in 1,2,\cdots,j_i : (S_{ijk} \leqslant t < F_{ijk})) \tag{5}$$

$$P_t \leqslant P_u \tag{6}$$

$$F_{ijk} - S_{ijk} = T_{ijk} \tag{7}$$

$$C_i = \max_{jk}(F_{ijk}) \tag{8}$$

$$S_{ij}, F_{ij} \geqslant 0 \tag{9}$$

其中，式(3) 表示某一时刻一个工件只能有一道工序被加工；式(4) 表示工件的任一工序在同一时刻只能被一台机器加工，L_{ijt_1} 为 t_1 时刻执行 O_{ij} 的设备编号，L_{ijt_2} 为 t_2 时刻执行 O_{ij} 的设备编号；式(5) 表示 t 时刻的车间峰值功率，l 为指示函数；式(6) 表示车间峰值功率不能超过限制，且待机功率忽略不计；式(7) 表示最大完工时间大于或等于任一工件的末道工序的完工时间；式(8) 表示 C_i 为同一工件的 F_{ijk} 中最大值；式(9) 为非负性约束。

2 求解 PPCFJSP 问题的 DRL 调度框架

为了更好地求解 PPCFJSP 问题，本文构建

了基于马尔可夫决策过程的深度强化学习 DRL 的调度框架，如图 1 所示，主要包含三部分：调度环境层、数据处理层与测试应用层。

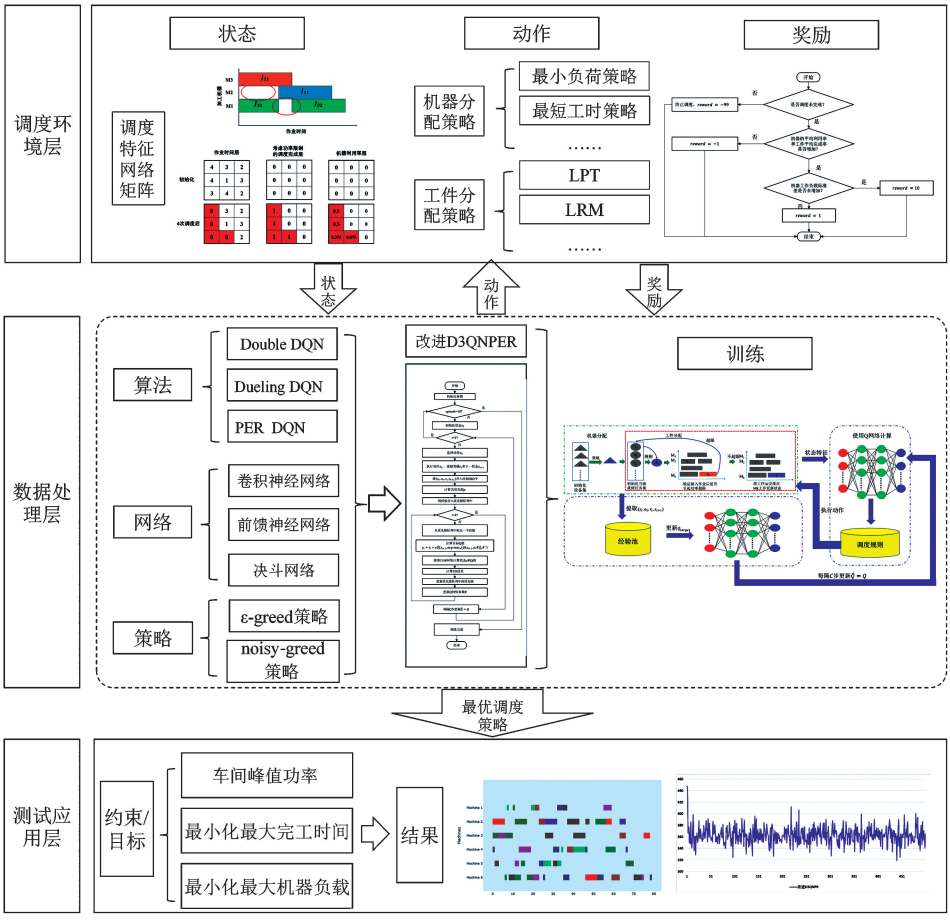


图 1　基于马尔可夫决策过程的 DRL 调度框架示意图

Fig.1　Schematic diagram of DRL scheduling framework based on Markov decision process

在调度环境层，本文将 PPCFJSP 问题转化为马尔可夫决策过程，结合 PPCFJSP 问题中对峰值功率的解释，对马尔可夫决策过程的状态、动作、奖励函数进行了设计，包括使用调度特征网络矩阵表示状态，使用不同的机器分配策略和工件分配策略组成动作空间，使用三个特征数值来指导奖励函数。

在数据处理层，本文设计了一种改进的 D3QNPER 算法用于训练调度环境中生成的调度数据。该算法融合了双重深度 Q 网络（double deep q-network，Double DQN）、决斗深度 Q 网络（dueling deep q-network，Dueling DQN）、优先级经验回放深度 Q 网络（prioritized experience replay DQN，PER DQN）三种深度强化学习算法在数据处理上的优势，将卷积神经网络、前馈神经网络、决斗网络进行有效结合，在探索和利用策略上设计了一种结合 ε-greed 和 noisy-greed 的探索策略。通过调度环境与数据处理的不断迭代交互，最终得到最优的调度策略。

在测试应用层，对约束和优化目标进行调控，使用调度环境与数据处理交互训练过程中得到的最优策略代入案例，从而完成对 PPCFJSP 问题的求解，最终得到满足约束、符合调度目标的调度结果。

2.1　调度问题的马尔可夫决策过程转化

深度强化学习应用于车间调度问题的关键和难点是将车间调度问题转化为马尔可夫决策过程（markov decision processes，MDP）。MDP 由一组状态 S 和操作 A 组成，针对 PPCFJSP 的最小化最大完工时间与机器最大负载两个优化目标，本文设计了以下 MDP 的状态空间表示、动作空间和奖励函数。

2.1.1　状态空间表示

在状态空间表示上，依据文献[19]提出的表述原则，本文将调度过程需要的机器、工序、功率、加工时间等调度特征信息以网络矩阵的形式作为输入图像的通道直接输入深度神经网络中训练。每个通道包含不同的调度特征，使用卷积、池化等操作来捕捉调度问题中的空间局部性和特征关联性，从而提高模型的表达能力和性能。

本文将作业号编码为图像的高度和宽度，从而保留作业之间的空间关系。在作业时间层，第 1 行第 1 列数据表示第 1 个工件的第 1 道工序，依此类推；在考虑功率限制的调度完成层，第 1 行第 1 列数据表示第 1 个工件的第 1 道工序是否在峰值功率未超限时进行操作，依此类推；在机器利用率层，第 1 行第 1 列数据表示在当前调度时刻第 1 个工件的第 1 道工序在作业完成后其使用的加工机器的机器利用率，其值越接近 1 表示该机器负载越大。以 3×3 调度为例，其特征状态与状态空间的转化如图 2 所示。图中，圈出部分为考虑功率限制而采取的延时和更换操作，该操作将在动作空间设计中具体说明。

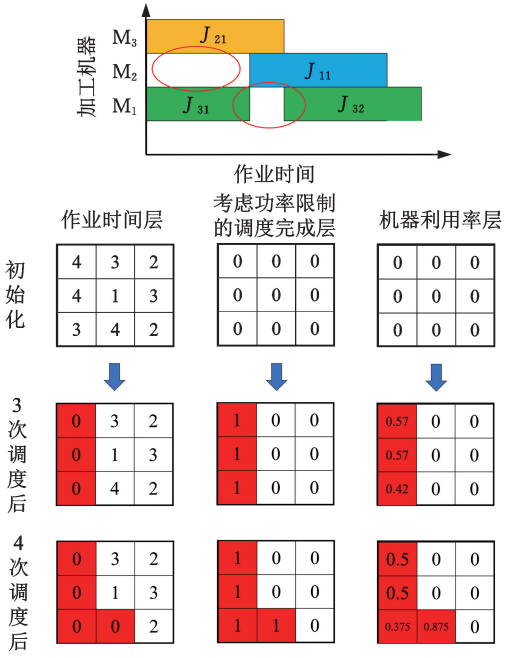


图 2　特征状态与状态空间的转化示意图

Fig.2　Schematic diagram of the transformation between feature state and state space

以第 4 次调度为例，假设在执行第 4 次调度前已经完成了第 1 个工件的第 1 道工序、第 2 个工件的第 1 道工序、第 3 个工件的第 1 道工序，第 4 次调度选择第 3 个工件的第 2 道工序进行加工，根据对应索引找到加工机器为 1 号的机器。因为车间有峰值功率的限制，此时发现不能直接安排生产作业，需要采取一定的措施避免峰值功率超限后再安排生产。为此延时实行 1 个时间单位，于是得到第 4 次调度的结束时刻为 8。调整后，在考虑功率限制的调度完成层中将第 3 个工件的第 2 道工序对应位置设置为 1，表示已完成该工序。由于没有采用更换操作，此时 1 号机器利用率为加工状态总用时/机器开机时间，即(3+4)/(3+4+1)=0.875，其他机器的利用率均为 0.5。

2.1.2　动作空间设计

相较于常规柔性作业车间调度问题，由于考虑了峰值功率约束，故还需要对动作的合法性进行判断。一般情况下，认定选定操作执行后会引起车间峰值功率超过上限的操作为非法操作。假设某车间有 6 台加工机器，加工机器功率 $P_{W_k} \in \{3, 2, 3, 2.4, 1.8, 3\}$，单位为 kW，当车间没有用电负荷约束时，车间各类加工机器的车间峰值功率为 15.2 kW 且可以同时工作；而当企业收到限电限产通知，如要求企业用电负荷降为原来车间峰值功率的一半(即 7.6 kW)时，生产运作安排就要考虑设备的用电负荷，否则极易出现用电负荷超限而导致断电停产或设备供电不足等问题，例如当车间中已开启第 1、2 号机器时，如果再选择第 3 号机器，那么就会导致车间峰值功率超限，此时选择第 3 号机器进行加工的操作定义为非法操作。为此，本文设计了两种处理操作来避免非法动作的产生，即延时操作和更换操作。

延时操作，即在选定非法动作后，将该操作延时到最早满足峰值功率限制的时刻执行，其操作如图 3 所示。当执行选定动作后，反馈的调度决策为工件 1 的第 1 道工序在机器 2 上加工，此时由于存在功率限制，导致该动作执行后会使车间峰值功率超限。若没有可以替换的柔性执行机器，此时便需要采用延时操作来执行，延时至最早可执行操作的时刻，即工件 3 第 1 道工序的结束时刻 t_1。

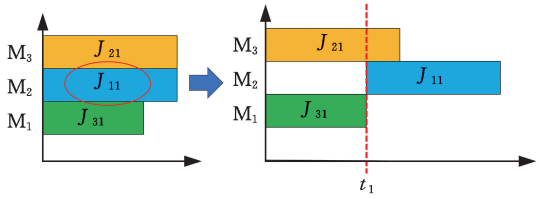


图 3　处理非法动作的延时操作

Fig.3　Delay operation for handling illegal actions

更换操作，即在选定非法动作后，由于该工序的加工机器具有柔性，可以选择其他能够满足峰值功率限制的机器来执行，其操作如图 4 所示。当执行选定动作后，反馈的调度决策为工件 3 的第 2 道工序在机器 1 上加工，此时由于存在功率限制，导致该动作执行后会使车间峰值功率超限。假设此时有可以替换的柔性执行机器 3 且替换后不会导致车间峰值功率超限，便采用更换操作来

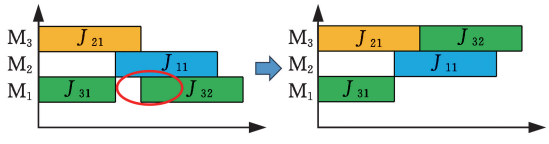


图 4　处理非法动作的更换操作

Fig.4　Replacing operations for handling illegal actions

执行，更换机器 3 作为可执行操作的机器。

此外，为更好地求解 PPCFJSP 问题的调度过程，包括以下 5 种机器分配策略、18 种工件分配规则，与两种非法动作处理操作共同构成动作空间。其中，18 种工件分配策略由文献[20]提到的 16 种分配策略和该文献未提到的与 SRM、SRPT 相对的 LRM、LRPT 共同组成。在初始阶段，机器分配策略与工件分配策略均为等概率随机选择。5 种机器分配策略如下：①最小机器负荷优先，优先选择待机序列中加工时间最少的机器；②最短加工时间优先，优先选择该工序可选加工机器中加工时间最短的机器；③最少作业数量优先，优先选择加工作业数量最少的机器；④最小功率优先，优先选择空闲机器中功率最小的机器；⑤完全随机分配，即随机选择机器。18 种工件分配规则见表 2。

表 2　工件分配规则

Tab.2　Workpiece allocation rules

序号	规则名称	说明
1	SPT	加工时间越短越优先
2	LPT	加工时间越长越优先
3	LRM	优先选择剩余加工时间(不包括当前工序加工时间)最长的工件
4	LRPT	优先选择剩余加工时间最长的工件
5	LSO	优先选择下一工序加工时间最长的工件
6	SRM	优先选择剩余加工时间(不包括当前工序加工时间)最短的工件
7	SRPT	优先选择剩余加工时间最短的工件
8	SSO	优先选择下一工序加工时间最短的工件
9	SPT+SSO	优先选择当前工序与下一工序加工时间之和最小的工件
10	SPT×TWK	优先选择工序加工时间与总工时之积最小的工件
11	SPT×TWKR	优先选择工序加工时间与总剩余工时之积最小的工件
12	LPT+LSO	优先选择当前工序与下一工序加工时间之和最大的工件
13	LPT×TWK	优先选择工序加工时间与总工时之积最大的工件
14	LPT×TWKR	优先选择工序加工时间与总剩余工时之积最大的工件
15	LPT/TWK	优先选择工序加工时间与总工时之比最大的工件
16	LPT/TWKR	优先选择工序加工时间与总剩余工时之比最大的工件
17	SPT/TWK	优先选择工序加工时间与总工时之比最小的工件
18	SPT/TWKR	优先选择工序加工时间与总剩余工时之比最小的工件

2.1.3　奖励设计

因调度目标结果均在全部工序安排完成后才能知晓，如果将调度目标结果直接作为奖励函数的参数，会导致奖励函数的反馈变得稀疏，且输出结果为一组动作的综合奖励，无法判断是否陷入局部最优，因此，需要将调度目标进行合理转化，使得智能体执行一个动作后，根据当前状态和执行的动作立即得到奖励值作为反馈，使得调度过程的每一步都尽可能采取最优策略，从而避免陷入局部最优。为此，额外定义三个变量 $C_k(t)$、$O_i(t)$ 和 $U_k(t)$：

$$U_k(t)=(C_k(t))^{-1}\sum_{i=1}^{n}\sum_{j=1}^{O_i(t)}\sum_{k=1}^{m}(C_{ijk}X_{ijk}) \tag{10}$$

$$J_a=\frac{1}{n}\sum_{i=1}^{n}\frac{O_i(t)}{J_i} \tag{11}$$

$$U_a=\frac{1}{m}\sum_{k=1}^{m}U_k(t) \tag{12}$$

$$W_a=\frac{1}{m}U_k(t)C_k(t) \tag{13}$$

$$W_{aa}=\sqrt{\frac{1}{m}\sum_{k=1}^{m}\left[\sum_{i=1}^{n}\sum_{j=1}^{O_i(t)}(C_{ijk}X_{ijk})-W_a\right]^2} \tag{14}$$

其中，$C_k(t)$ 表示在 t 时刻机器 k 上已完成的最后一道工序的完工时间；$O_i(t)$ 表示在 t 时刻工件 i 已完成的工序数量；$U_k(t)$ 表示在 t 时刻机器 k 的利用率；式(11) 表示在每一动作执行后，该调度时刻下工件的工序平均完成率；式(12) 表示在每一动作执行后，该调度时刻下机器的平均利用率；式(13) 表示在每一动作执行后，该调度时刻下机器的平均工作负载；式(14) 表示在每一动作执行后，该调度时刻下机器工作负载的标准差。

由式(10) ～ 式(12) 可以发现，式中的指标均与最大完工时间直接或间接相关，所以最小化最大完工时间可以描述为使得机器利用率、工件完成率尽可能大。由于峰值功率约束直接影响到机器能否被选择，而选择延时或更换操作来处理非法操作均大概率会导致等待时间增加，故最小化最大机器负载可以描述为使得工作负载均匀分布在各个机器上的同时机器工作负载的标准差尽可能小。

调度未完成时，每执行一个动作后计算对应的 U'_a、J'_a、W'_{aa}，通过比较前一状态下的 U_a、J_a、W_{aa} 进行赋奖励值($reward$)。本文奖励值的设置参考文献[19]中的设置方法，在执行一个动作后如果机器平均利用率、工件平均完成率增加的同时机器平均工作负载标准差没有增加，这种情况说明机器分布更加均匀，这一动作是能够使得两个优化目标均减小的动作，故给予一个较大奖励 10；如果机器平均利用率、工件平均完成率增加的同时机器平均工作负载标准差增加，这种情况下对优化最大完工时间是有益的，但不能完全认为是最大机器工作负载的增加导致机器平均工

作负载增加或考虑了均匀分布负载但由于该工序加工时间较长导致的机器平均工作负载增加，故给予一个较小的奖励 1；如果机器平均利用率减小，这种情况下不能完全认为是由最大机器工作负载增加导致的机器平均利用率减小或延时操作的存在而导致的机器平均利用率减小，故给予一个较小的惩罚 −1。奖励设计伪代码如下。

伪代码 1：奖励设计伪代码

```
if 调度过程未完成
    if U'_a − U_a > 0, J'_a − J_a > 0
        if W'_aa − W_aa ≤ 0
            reward = 10
            else
                reward = 1
        else
            reward = −1
else
    终止调度, reward = −99
end if
```

2.2 改进的 D3QNPER 算法设计

D3QNPER 算法是在 DQN(Deep Q-Network) 算法的基础上发展起来的。由于 DQN 算法在求解过程中存在积极性偏差、高方差、非静态目标影响等问题，故在将 PPCFJSP 问题转化为 MDP 问题后，需要进一步对算法进行改进，改进 D3QNPER 算法的主要内容包括：

1) 引入 Double DQN 算法改善积极性偏差。通过不同网络解耦动作选择与评估，使用两个独立的神经网络来分别估计当前状态下的动作值函数和目标动作值函数。其中一个网络用于选择动作，另一个网络用于评估选择的动作的价值。这种解耦的方式具体体现为使用 θ_t 决定的网络选择动作 a，再用 θ_t^- 决定的网络计算 Q 值，这样的改动可以减少动作价值的高估，从而减小积极性偏差，提高 Q-learning 算法的稳定性和性能。此时目标网络的目标函数变为

$$Y_t \equiv r_{t+1} + \gamma \hat{Q}(s_{t+1}, \operatorname{argmax}_a Q(s_{t+1}, a; \theta_t), \theta_t^-) \quad (15)$$

式中：Y_t 为目标网络的目标函数；r_{t+1} 为下一动作的奖励；$Q(s, a, \theta)$ 为计算 Q 值的函数。

2) 引入 Dueling DQN 优化网络结构来缓解神经网络的高方差问题。将动作值函数分解为状态值函数和优势函数。状态值函数表示在给定状态下不同动作的平均价值，而优势函数表示每个动作相对于平均值的优势。通过 Dueling DQN 的优化，神经网络可以更有效地学习状态的价值和动作的优势，从而提高了对动作价值的估计效果，提高了算法的性能和效率。此时动作值函数为

$$Q(s, a; \theta, \alpha, \beta) = V(s; \theta, \beta) + A(s, a; \theta, \alpha) - \frac{1}{|A|} \sum_{a'} A(s, a'; \theta, \alpha) \quad (16)$$

式中：θ 为共享参数；α 为优势函数 A 的参数；β 为状态值函数 V 的参数；$|A|$ 为动作空间的大小。

3) 引入 PER DQN 设定样本优先级，减小非静态目标的影响。通过引入优先级队列，根据样本的 TD 误差(temporal difference error) 来赋予样本优先级，TD 误差可以被视为样本的重要性指标。通过优先级采样，网络更多地关注那些对于当前参数下预测不准确的样本，从而提高了训练的效率和收敛速度。改进后的损失函数为

$$L_i(\theta_i) = E(\omega_i (Y_t - Q(s, a; \theta_i)))^2 \quad (17)$$

式中：E 为期望值函数；ω_i 为重要性参数。

D3QNPER 算法结合了 Double DQN、Dueling DQN 和 PER DQN 算法的优点，进一步提高了学习的效率和稳定性，其算法流程如图 5 所示，其中，每个 *episode* 表示一次完整调度过程。

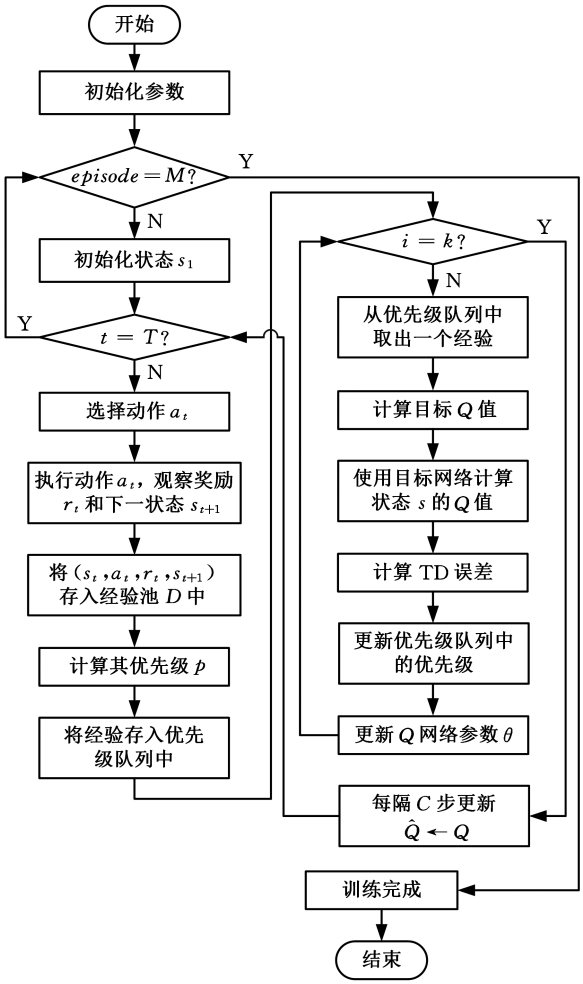


图 5 D3QNPER 算法流程图

Fig.5 D3QNPER algorithm flowchart

4) 此外，设计一种引入噪声的 ε- 贪婪递减策略来增加算法的探索性，从而帮助算法更充分地探索环境。为平衡探索和利用，本文综合 noisy-greed 和 ε-greed 两种探索策略，在训练前期

通过随机策略和 noisy-greed 策略提高智能体的探索能力，在训练后期则更多地考虑利用已知信息中最优的行为。探索和利用策略可以表示为

$$a_t=\begin{cases}\operatorname{argmax}\tilde{Q}(s,a) & P=1-\varepsilon_2\\ \operatorname{argmax}(\tilde{Q}(s,a)+\sigma\varepsilon_2) & P=\varepsilon_2-\varepsilon_1\\ random & P=\varepsilon_1\end{cases}\tag{18}$$

$$\varepsilon_2=\max(\varepsilon_{\min},(1-\mu_2)\varepsilon_2)\tag{19}$$

$$\varepsilon_1=\max(\varepsilon_{\min},(1-\mu_1)\varepsilon_1)\tag{20}$$

式中：P 为选择对应策略的概率；$random$ 为随机一个动作 a，$\sigma\sim N(0,1)$；$\varepsilon_{\min}$ 为递减策略中最小 ε 值；μ_1、μ_2 为递减速率。

对比使用引入噪声的 ε- 贪婪递减策略前后 D3QNPER 算法的 *reward* 值变化（图 6）可以发现，使用该策略改进 D3QNPER 算法网络收敛速度和稳定程度明显提高，改进后的算法奖励函数曲线明显优于改进前，结果拥有更高的均值及稳定性。

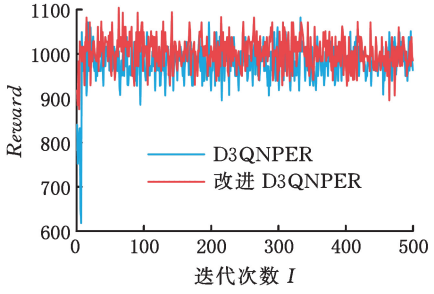


图 6　使用引入噪声的 ε-贪婪递减策略前后 D3QNPER 算法 *reward* 值变化

Fig.6　The reward change of D3QNPER algorithm before and after using greedy descent strategy with introduced noise

在训练阶段，本文将描述作业时间、考虑功率限制的调度结果和机器利用率三通道图像作为深度卷积神经网络的输入和输出。伪代码 2 描述了改进 D3QNPER 算法求解 MDP 流程。在求解过程中，首先需要对环境进行初始化，包括初始化估计网络和目标网络、经验池和优先级队列。然后进行多轮训练。在每一轮训练中，根据调度方案的初始状态，在每个时间步中以一定的概率选择动作。概率通过 ε 的值来调整。当 ε 较小时，会尽可能选择当前 Q 值函数估计的最优动作；当 ε 较大时，会更多地进行探索。执行选择的动作后，观察下一个状态并计算奖励。然后将得到的经验元组存入经验池，并计算样本的优先级。当满足条件时，从经验池中采样一批样本，根据计算得到的目标值和当前 Q 值计算 TD 误差。然后更新样本的优先级，并根据累积的权重更新量来执行梯度下降，从而更新神经网络参数。在每一轮训练结束后，周期性地更新目标网络的参数，将当前的 Q 值函数的参数复制给目标网络。整个过程循环执行，直到达到预定的训练次数。最后将训练好的 Q 网络作为输出结果。

伪代码 2：改进 D3QNPER 算法求解 MDP 流程伪代码

初始化神经网络 $Q(s,a;\theta)$ 和目标网络 $\hat{Q}(s,a;\theta^-)$
初始化经验池 D 和优先级队列 P
初始化参数，误差 $\Delta=0$，样本优先级 $p_1=1$
对于 $episode=1:M$
　重置调度方案，生成初始化状态 s_1
　对于 $t=1:T$
　　以一定的概率 ε 选择动作 a_t

$$\text{动作 } a_t=\begin{cases}\operatorname{argmax}\tilde{Q}(s,a) & P=1-\varepsilon_2\\ \operatorname{argmax}(\tilde{Q}(s,a)+\sigma\varepsilon_2) & P=\varepsilon_2-\varepsilon_1\\ random & P=\varepsilon_1\end{cases}$$

　　执行动作 a_t，观察下一个状态 s_{t+1}，计算奖励 r_t
　　将 (s_t,a_t,r_t,s_{t+1}) 存入经验池 D，优先级 $p_t=\max_{j<t}p_j$
　　如果 k 能够被 t 整除（k 为最小批量）
　　对于 $i=1:k$
　　从 D 中采样

$$\text{令 } y_i=\begin{cases}r_i & \text{当前步数调度结束}\\ r_i+\gamma\hat{Q}(s_{i+1},\operatorname{argmax}_a(Q(s_{i+1},a;\theta)),\theta^-) & \text{其他}\end{cases}$$

　　计算当前 Q 值：$Q_{\text{cur}}=Q(s_i,a_i;\theta)$
　　计算 TD 误差：$\delta=(y_i-Q_{\text{cur}})^2$
　　更新优先级：$p_i\leftarrow\delta$
　　累积权重更新量 $\Delta\leftarrow\Delta+w_i\delta\nabla_\theta Q(s_i,a_i)$
　　执行梯度下降更新神经网络参数：
　$\theta\leftarrow\theta+\eta\Delta$（$\eta$ 为步长）
　$\Delta=0$
　$s=s'$
　每隔 C 步更新 $\hat{Q}=Q$
返回 Q 网络

2.3　基于改进 D3QNPER 算法的调度训练过程

在应用改进 D3QNPER 算法进行 PPCFJSP 问题的 MDP 求解训练时，需要对动作进行合法性判断，即在完成机器分配工件分配后，检查调度环境中峰值功率是否超过设定上限，如果超过功率上限则需要执行更换或延时操作，此后再循环此操作直至所有工序都已安排完毕；如果没有超限则认为此动作合法，将状态中对应位置置为 1。

在调度任务分配流程的基础上，基于改进 D3QNPER 算法的调度过程可以分为训练、算法、测试三个层面。首先应用本文设计的调度框架将车间的状态、任务的特征、资源的可用性等信息传递到训练层构成训练和验证过程使用的数据集。然后将训练集用于训练改进 D3QNPER 网络，验证集用于调整网络的超参数和监控训练进度，期间重复算法训练流程，进一步优化调度策略。最后在训练完成后，使用测试集对训练得到的调度策略进行评估和验证。改进 D3QNPER 求解 PPCFJSP 问题的过程如图 7 所示。

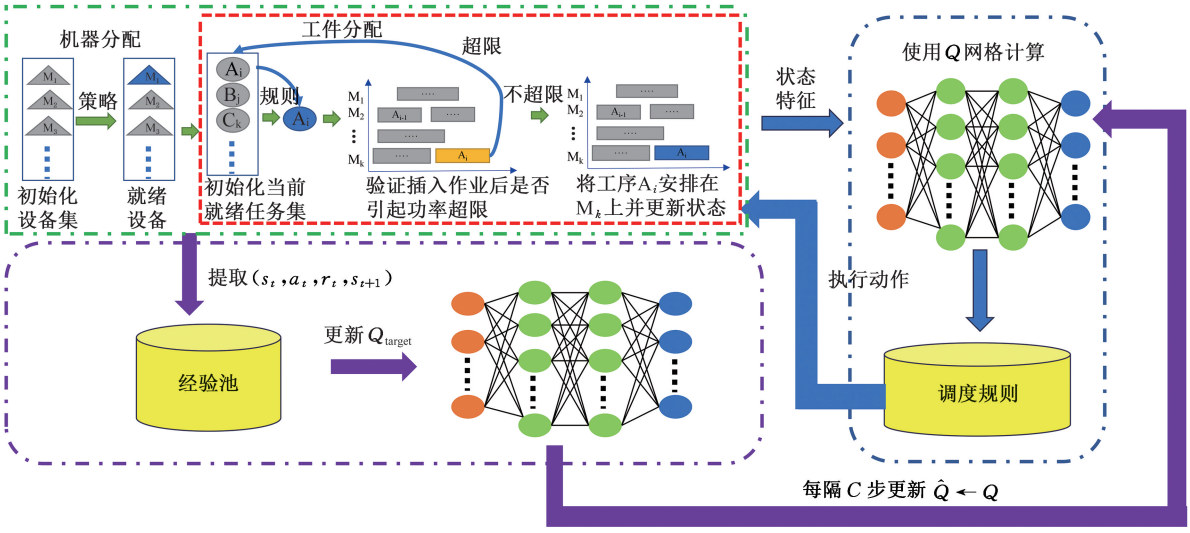


图 7　改进 D3QNPER 求解 PPCFJSP 的调度训练过程

Fig.7　Improving the scheduling training process of D3QNPER for solving PPCFJSP

3　实验设计与结果分析

为验证 D3QNPER 算法在求解 PPCFJSP 问题时的优越性，本文结合 Benchmark 标准算例，为其中 Mk01～Mk15 这 15 组拥有不同工件数、工序数、机器数的柔性作业车间调度问题标准测试集引入机器的加工功率集（表 3），此时生成的测试集规模和参数见表 4。

表 3　各机器的加工功率集

Tab.3　Processing power set of each machine　kW

机器	M_1	M_2	M_3	M_4	M_5
Pw_k	3	2	3	2.4	1.8
机器	M_6	M_7	M_8	M_9	M_{10}
Pw_k	3	2	2.6	1.8	3
机器	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}
Pw_k	3	2	3	2.4	2

表 4　DMK01-15 测试集规模及参数

Tab.4　DMK01-15 test set scale and parameters

实例	工件数	机器数	机器峰值功率总和/kW	峰值功率上限/kW
DMk01	10	6	15.2	7.6
DMk02	10	6	15.2	7.6
DMk03	15	8	19.8	9.9
DMk04	15	8	19.8	9.9
DMk05	15	4	10.4	5.2
DMk06	10	15	24.6	12.3
DMk07	20	5	12.2	6.1
DMK08	20	10	24.6	12.3
DMK09	20	10	24.6	12.3
DMK10	20	15	37.0	18.5
DMK11	30	5	12.2	6.1
DMK12	30	10	24.6	12.3
DMK13	30	10	24.6	12.3
DMK14	30	15	37.0	18.5
DMK15	30	15	37.0	18.5

对得到的新数据集 DMk01-DMk15 进行等权重实验。实验程序在 Windows 11 64 位的个人计算机（CPU：AMD R7-6800H、内存：16 GB）上运行。语言环境基于 Python 3.8.16，问题环境基于 OpenAIGym，深度网络基于 Torch 2.1.0 和 Numpy 1.24.1 编写。

3.1　改进 D3QNPER 算法与单一调度规则对比

在实验中，依据我国各地迎峰度夏冬负荷管理方案的相关要求，本文将企业生产车间最大功率限制为全部加工机器功率总和的一半。为了便于比较本文方法与单一调度规则的优劣，以等权重对最大完工时间和机器最大负载两个指标的加权值作为评估值，选择动作空间中 35 种基于规则的调度方法与改进的 D3QNPER 方法进行对比，再将改进的 D3QNPER 方法与和单一方式 DQN 优化方法进行对比，在全随机机器分配策略下为独立运行 20 次的最优结果，见表 5。可以发现，改进 D3QNPER 与任意调度规则相比较，均可得到较好结果。

3.2　改进 D3QNPER 算法与优化 DQN 对比

首先使用等权重评估值对算法效果进行对比。表 6 表明，相同的 MDP 下，单一优化 DQN 算法在各算例的训练测试结果一般优于单一调度规则方法，但也存在样本数据规模较大时训练结果劣于单一调度规则方法，而本文设计的改进 D3QNPER 算法在每个算例下均优于单一调度规则方法且表现最优。

为了更详细地比较各 DQN 优化方法差异性与本文设计算法的优越性，以五个不同规模算例 DMK03、DMK07、DMK10、DMK13、DMK15 为例，以评估值作为指标可以得到训练迭代测试结果变化，如图 8 所示。可以观察到，改进 D3QNPER 算法的收敛速度更快，输出结果更稳定，其求解性能优于单一 DQN 及 DQN 改进算法。

表 5　改进 D3QNPER 和调度规则求解 PPCFJSP 结果

Tab.5　Improve D3QNPER and scheduling rules to solve PPCFJSP results

实例	机器分配策略	SPT	LPT	SRM	LRM	SPT* TWKR	LPT* TWKR	SPT+SSO	改进 D3QNPER
DMk01	最小负荷	104	90.5	99.5	**77**	93.5	90	90.5	**65.5**
	最短时间	91.5	87	100	79.5	96.5	90	90.5	
	最少作业	103	98	112	84.5	113	97	97	
	最小功率	171.5	163	175	151.5	168	156	164.5	
	全随机	103.5	110	120	94	114.5	112.5	110	
DMk02	最小负荷	99.5	94	119.5	76	102	89	97.5	**54.5**
	最短时间	72.5	70	78	62	71.5	**59.5**	62.5	
	最少作业	88	94.5	97.5	66	79	74	72.5	
	最小功率	185.5	183.5	185	183	188.5	183	186.5	
	全随机	109.5	98.5	117.5	79	103.5	87.5	91	
DMk03	最小负荷	594.5	507.5	779.5	412	716	497.5	504.5	**319**
	最短时间	443.5	396	484	**340**	445	375.5	391	
	最少作业	621.5	570.5	784.5	452.5	696.5	565.5	629.5	
	最小功率	554.5	582	575	479.5	556	520	523.5	
	全随机	521.5	579	700	441.5	655.5	475.5	543.5	
DMk04	最小负荷	156	124.5	200	**109**	164	132.5	169.5	**106**
	最短时间	212	193.5	207.5	188	217.5	191	207.5	
	最少作业	142	140	183.5	129.5	178.5	136.5	140.5	
	最小功率	189	198	195.5	140	170	180.5	167	
	全随机	141	163	192	139	180.5	171.5	184.5	
DMk05	最小负荷	437	398.5	427.5	**361.5**	438	364	487	**350.5**
	最短时间	447	434.5	444.5	397	447	395	444.5	
	最少作业	449.5	425	449	380.5	449	385	440	
	最小功率	486.5	456.5	498	410.5	488.5	411.5	493.5	
	全随机	461.5	438.5	471.5	400	462.5	423.5	457	
DMk06	最小负荷	280.5	224.5	307.5	144.5	317	213	230.5	**99**
	最短时间	114.5	115.5	122	**101**	119.5	103.5	118	
	最少作业	272	244.5	354	159.5	332	197.5	236	
	最小功率	315.5	294	316.5	280	315	295	306	
	全随机	188	204	251.5	157.5	263.5	226.5	231	
DMk07	最小负荷	489	466.5	585	449.5	545	444	525	**290.5**
	最短时间	361.5	354	381	332	386	**329.5**	354	
	最少作业	528	490.5	590	464	585	500.5	541.5	
	最小功率	613	601	625.5	558	595	603.5	602.5	
	全随机	504.5	483.5	620.5	438	559.5	496.5	471.5	
DMK08	最小负荷	1372.5	1128	1372	665	930.5	770.5	800.5	**611**
	最短时间	1125	1184	1200	677	844.5	743	895	
	最少作业	1081.5	1133	1360.5	**654.5**	917.5	755	1060	
	最小功率	1170.5	1039	1158	690.5	1173	751	1071	
	全随机	1246.5	1203	1438	717.5	985.5	844	1064.5	
DMK09	最小负荷	935.5	755	1338.5	561.5	1255.5	627	915.5	**486**
	最短时间	851	874.5	1047.5	**515**	1007	549.5	881.5	
	最少作业	1074.5	829	1443	585.5	1168	724.5	1043.5	
	最小功率	973	849	1135.5	650	1053	727.5	1094.5	
	全随机	1045.5	953.5	1361	668	1176.5	802.5	869	
DMK10	最小负荷	609.5	602.5	703.5	362.5	695.5	412	642	**317**
	最短时间	491.5	429	445	**341**	452.5	347	408	
	最少作业	511.5	825.5	753.5	376.5	530	462.5	514.5	
	最小功率	960	921.5	986	857	936.5	861	945	
	全随机	539.5	680	615	438.5	625.5	496.5	601.5	
DMK11	最小负荷	1453	1519.5	1879	**1331**	1715	1347.5	1557	**1243.5**
	最短时间	1629	1571	1829.5	1388	1659.5	1362.5	1599.5	
	最少作业	1555	1550.5	1891.5	1346	1777	1344.5	1505.5	
	最小功率	1697.5	1595	1864.5	1418.5	1867.5	1387	1641	
	全随机	1640	1574	1759.5	1353	1457	1349	1603	
DMK12	最小负荷	1186.5	1112.5	1764.5	859	1256	**849**	1284.5	**779.5**
	最短时间	1147.5	1271	1531	888	1257.5	927	1200	
	最少作业	1227.5	1050.5	1674.5	845	1272	853.5	1337	
	最小功率	1460.5	1298.5	1492	989.5	1178.5	1178.5	1167.5	
	全随机	1221.5	1107	1624	869	1167	949	1086	
DMK13	最小负荷	1250	1393	2297.5	951.5	1587.5	1034.5	1371.5	**819.5**
	最短时间	1278.5	1390.5	1683.5	**868.5**	1263.5	905	1089	
	最少作业	1348.5	1524	2137.5	931.5	1589	1020.5	1138.5	
	最小功率	1344	1348	1412.5	1254.5	1367	1270.5	1367.5	
	全随机	1484.5	1541	2131	1103	1945.5	1145.5	1462	
DMK14	最小负荷	1654.5	1546	2386	1058	1699.5	1042.5	1560.5	**853**
	最短时间	1315.5	1448.5	1817	1117	1988	1157.5	1481.5	
	最少作业	1453.5	1505	2537.5	**865**	1692.5	1042	1449	
	最小功率	1539	1614	1580	1491	1618	1503.5	1558.5	
	全随机	1550	1498.5	2323	1183	2011	1199	1299	
DMK15	最小负荷	1440	1201	2312.5	785.5	1606.5	853	1444.5	**668.5**
	最短时间	1120.5	909.5	1432	**708.5**	1115	760	971.5	
	最少作业	1362.5	1529	2057	758.5	1462	907.5	1349	
	最小功率	955.5	913.5	1348	832.5	1014.5	831	952	
	全随机	1381	1349.5	2465	861	1616	986	1436	

表 6　不同优化 DQN 算法求解 PPCFJSP 结果

Tab.6　Different optimized DQN algorithms for solving PPCFJSP results

实例	DQN	Double DQN	Dueling DQN	PER DQN	改进 D3QNPER	实例	DQN	Double DQN	Dueling DQN	PER DQN	改进 D3QNPER
DMk01	69.5	66	65.5	67	65.5	DMK09	524.5	489	501	587.5	486
DMk02	56.5	54.5	54.5	55	54.5	DMK10	335.5	331	327.5	349	317
DMk03	333	328	330.5	336.5	319	DMK11	1274	1266.5	1257	1433	1243.5
DMk04	113.5	113	108	117	106	DMK12	800	801	803	856.5	779.5
DMk05	355.5	356.5	356	362.5	350.5	DMK13	847	836.5	820	832.5	819.5
DMk06	100.5	99.5	99.5	118.5	99	DMK14	913	933.5	931.5	1078.5	853
DMk07	297	302.5	295	314.5	290.5	DMK15	674	669.5	689	817.5	668.5
DMK08	632	628.5	633	663	611						

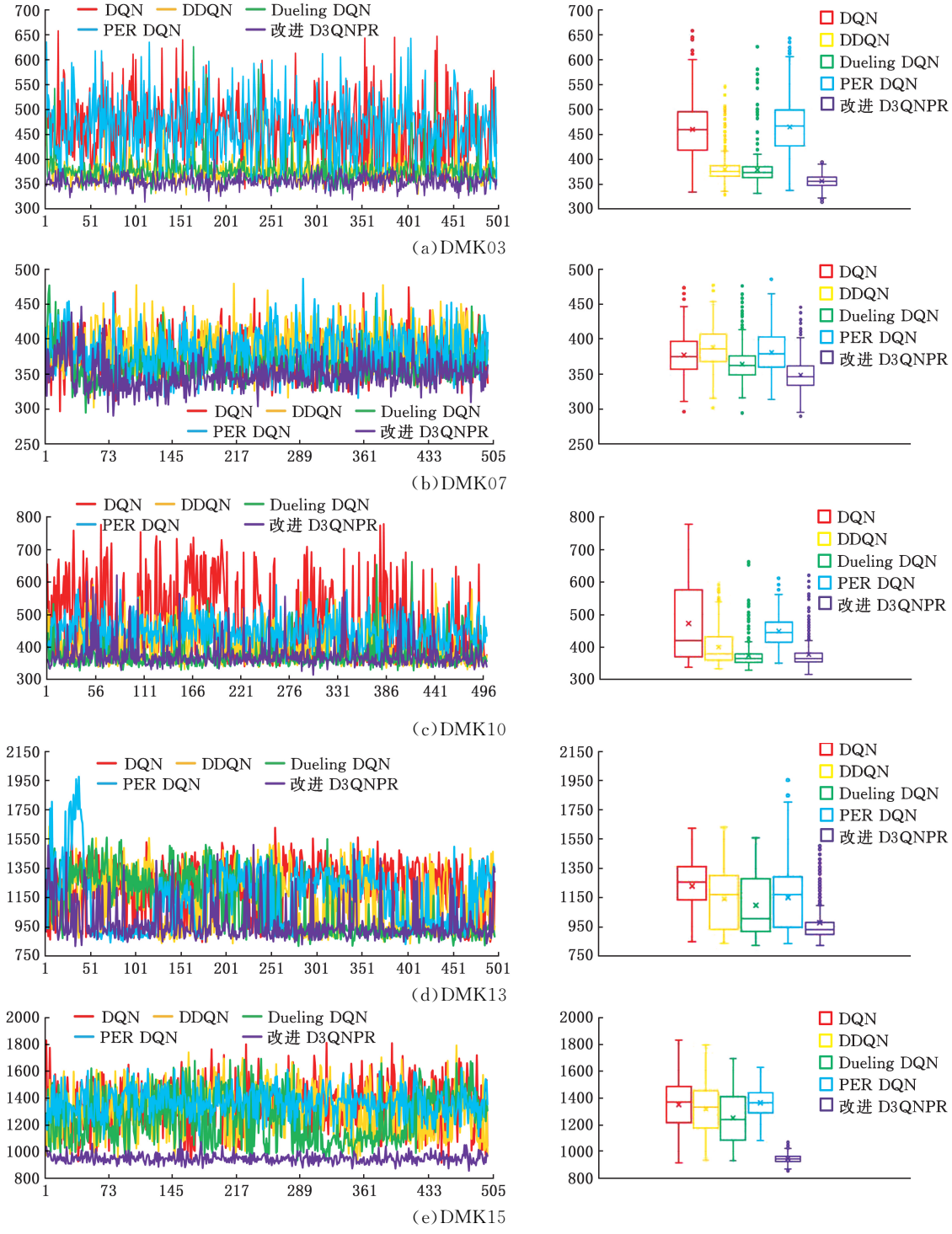


图 8　各 DQN 优化方法在 DMK03、DMK07、DMK10、DMK13、DMK15 算例迭代测试结果

Fig.8　Results of iterative testing of various DQN optimization methods in DMK03, DMK07, DMK10, DMK13 and DMK15 case studies

同时，依据 Pareto 最优理论将所得数据转化为二维散点，可以得到三个算例的散点图以及 Pareto 前沿，如图 9 所示。可以发现使用改进 D3QNPER 算法得到的结果分布明显更加靠近由各算法 Pareto 前沿组成的各测试算例真实的 Pareto前沿，离散程度更低，与前文以评估值作为指标得到的训练迭代测试结果变化一致，说明改进的 D3QNPER 算法得到的 Pareto 解更优，更能满足本文的优化目标。

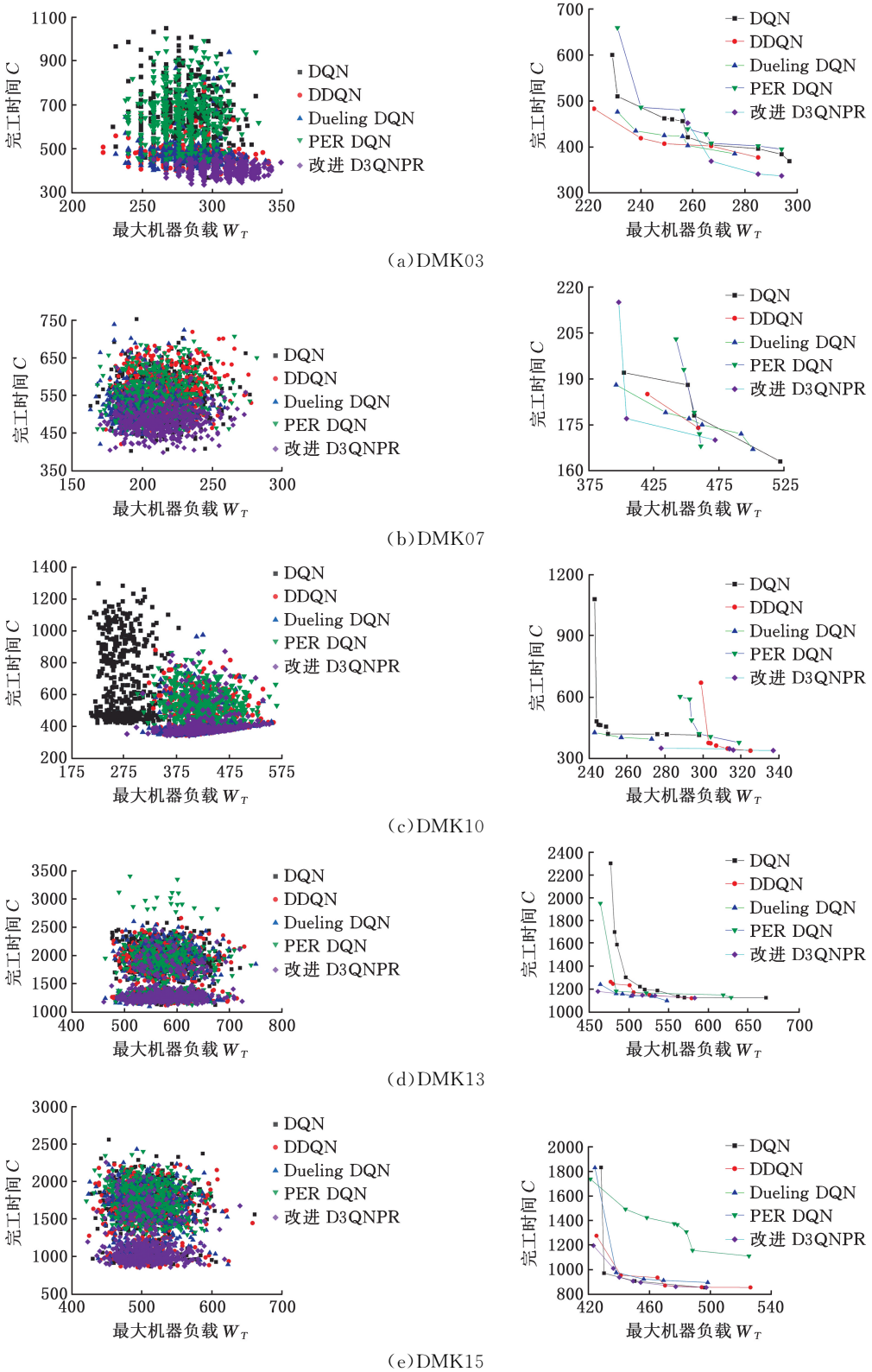
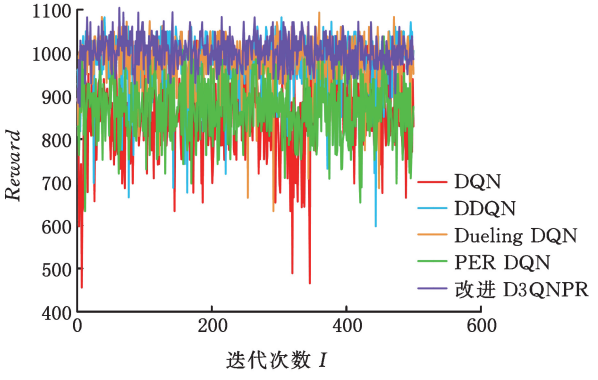


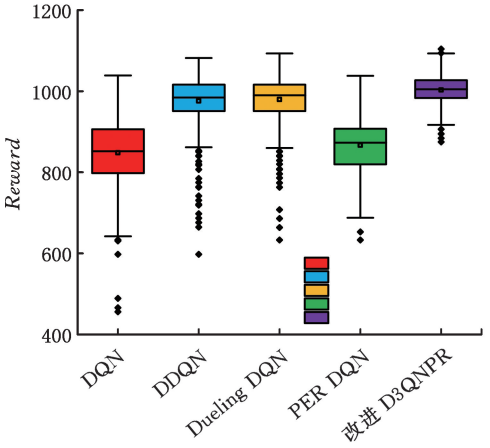
图 9　各 DQN 优化方法在 DMK03、DMK07、DMK10、DMK13、DMK15 算例数据散点图及 Pareto 前沿

Fig.9　Scatter plots and Pareto front plots of DMK03, DMK07, DMK10, DMK13 and DMK15 case data for various DQN optimization methods

对比 DMK07 算例各算法 *reward* 变化(图 10)可以观察到,Double DQN 算法比 DQN 算法结果更具稳定性,数据波动明显改善,收敛更加快速,但它对噪声干扰的处理能力较差,因此导致训练后期出现较大偏差值;Dueling DQN 与 Double DQN 算法效果无较大差别,但它对噪声干扰的处理明显优于 Double DQN 算法;PER DQN 算法能够利用重要的经验样本,因而探索空间获得的收益较 DQN 算法有明显提升,但也存在探索能力减小,从而在算例中结果表现不尽如人意;而改进的 D3QNPER 算法综合了三者优点,在输出效果整体上优于其他算法。



(a)*reward* 迭代变化曲线



(b)*reward* 迭代变化箱线

图 10　使用对比优化方法在 DMK07 算例测试的 *reward* 变化

Fig.10　Reward changes tested using various DQN optimization methods in the DMK07

4　结语

本文面向车间峰值功率受限这一特定约束,构建了峰值功率受限单约束的柔性作业车间调度问题模型,提出了基于深度强化学习的调度框架,设计了改进的 D3QNPER 算法求解该模型。其中包括设计了两个用于应对峰值功率超限的调度策略,设计了引入噪声的 ε -贪婪递减策略来提高算法的探索和利用能力。通过对比引入噪声的 ε -贪婪递减策略改进前后的回报值可以发现,改进后的方法收敛更快、回报值更高。同时,使用带有峰值功率约束的 Benchmark 标准算例的实验结果表明,改进 D3QNPER 算法在求解 PPCFJSP 问题时,其求解能力优于单一调度规则方法和单一 DQN 优化方法。

本文方法为求解峰值功率受限的柔性作业车间调度双目标优化问题提供了有效解决方案。在实际生产中,生产调度往往需要面临更为复杂的调度目标和条件约束。后续研究可以进一步考虑成本、排放等经济或绿色指标或者其他条件约束等,或者探索动态环境下柔性作业车间调度问题模型的算法设计与改进策略等。

参考文献:

[1] 李凯文,张涛,王锐,等. 基于深度强化学习的组合优化研究进展[J]. 自动化学报,2021,47(11):2521-2537.
LI Kaiwen, ZHANG Tao, WANG Rui, et al. Research Reviews of Combinatorial Optimization Methods Based on Deep Reinforcement Learning[J]. Acta Automatica Sinica, 2021, 47(11):2521-2537.

[2] 李颖俐,李新宇,高亮. 混合流水车间调度问题研究综述[J]. 中国机械工程,2020,31(23):2798-2813.
LI Yingli, LI Xinyu, GAO Liang. Review on Hybrid Flow Shop Scheduling Problems[J]. China Mechanical Engineering, 2020, 31(23):2798-2813.

[3] 黎声益,马玉敏,刘鹃. 基于双深度 Q 学习网络的面向设备负荷稳定的智能车间调度方法[J]. 计算机集成制造系统,2023,29(1):91-99.
LI Shengyi, MA Yumin, LIU Juan. Smart Shop Floor Scheduling Method for Equipment Load Stabilization Based on Double Deep Q-learning Network [J]. Computer Integrated Manufacturing Systems, 2023, 29(1):91-99.

[4] 贺俊杰,张洁,张朋,等. 基于长短期记忆近端策略优化强化学习的等效并行机在线调度方法[J]. 中国机械工程,2022,33(3):329-338.
HE Junjie, ZHANG Jie, ZHANG Peng, et al. Related Parallel Machine Online Scheduling Method Based on LSTM-PPO Reinforcement Learning[J]. China Mechanical Engineering, 2022, 33(3):329-338.

[5] LIU Renke, PIPLANI R, TORO C. Deep Reinforcement Learning for Dynamic Scheduling of a Flexible Job Shop[J]. International Journal of Production Research, 2022, 60(13):4049-4069.

[6] LI Yuxin, GU Wenbin, YUAN Minghai, et al. Real-time Data-driven Dynamic Scheduling for Flexible Job Shop with Insufficient Transportation Resources Using Hybrid Deep Q Network[J]. Robotics and Computer-Integrated Manufacturing, 2022, 74: 102283.

[7] WU Wenbo, HUANG Zhengdong, ZENG Jiani, et al. A Fast Decision-making Method for Process Planning with Dynamic Machining Resources via Deep Reinforcement Learning[J]. Journal of Manufacturing Systems, 2021, 58:392-411.

[8] LEE Y H, LEE S. Deep Reinforcement Learning Based Scheduling within Production Plan in Semiconductor Fabrication[J]. Expert Systems with Applications, 2022, 191:116222.

[9] HE Zhenglei, TRAN K P, THOMASSEY S, et al. Multi-objective Optimization of the Textile Manufacturing Process Using Deep-Q-network Based Multi-agent Reinforcement Learning[J]. Journal of Manufacturing Systems, 2022, 62:939-949.

[10] 郭具涛，吕佑龙，戴铮，等. 基于复合规则和强化学习的混流装配线调度方法[J]. 中国机械工程，2023，34(21):2600-2606.
GUO Jutao, LYU Youlong, DAI Zheng, et al. Compound Rules and Reinforcement Learning Based Scheduling Method for Mixed Model Assembly Lines[J]. China Mechanical Engineering, 2023, 34(21):2600-2606.

[11] 刘亚辉，申兴旺，顾星海，等. 面向柔性作业车间动态调度的双系统强化学习方法[J]. 上海交通大学学报，2022，56(9):1262-1275.
LIU Yahui, SHEN Xingwang, GU Xinghai, et al. A Dual-system Reinforcement Learning Method for Flexible Job Shop Dynamic Scheduling[J]. Journal of Shanghai Jiao Tong University, 2022, 56(9):1262-1275.

[12] ZHANG Jiadong, HE Zhixiang, CHAN W H, et al. DeepMAG: Deep Reinforcement Learning with Multi-agent Graphs for Flexible Job Shop Scheduling[J]. Knowledge-Based Systems, 2023, 259: 110083.

[13] GUI Yong, TANG Dunbing, ZHU Haihua, et al. Dynamic Scheduling for Flexible Job Shop Using a Deep Reinforcement Learning Approach[J]. Computers & Industrial Engineering, 2023, 180: 109255.

[14] ZHANG Lu, FENG Yi, XIAO Qinge, et al. Deep Reinforcement Learning for Dynamic Flexible Job Shop Scheduling Problem Considering Variable Processing Times[J]. Journal of Manufacturing Systems, 2023, 71:257-273.

[15] 何彦，王乐祥，李育锋，等. 一种面向机械车间柔性工艺路线的加工任务节能调度方法[J]. 机械工程学报，2016，52(19):168-179.
HE Yan, WANG Lexiang, LI Yufeng, et al. A Scheduling Method for Reducing Energy Consumption of Machining Job Shops Considering the Flexible Process Plan[J]. Journal of Mechanical Engineering, 2016, 52(19):168-179.

[16] DU Yu, LI Junqing, LI Chengdong, et al. A Reinforcement Learning Approach for Flexible Job Shop Scheduling Problem with Crane Transportation and Setup Times[J]. IEEE Transactions on Neural Networks and Learning Systems, 2024, 35(4):5695-5709.

[17] NAIMI R, NOUIRI M, CARDIN O. A Q-learning Rescheduling Approach to the Flexible Job Shop Problem Combining Energy and Productivity Objectives[J]. Sustainability, 2021, 13(23):13016.

[18] LI Rui, GONG Wenyin, LU Chao, et al. A Learning-based Memetic Algorithm for Energy-efficient Flexible Job-shop Scheduling with Type-2 Fuzzy Processing Time[J]. IEEE Transactions on Evolutionary Computation, 2023, 27(3):610-620.

[19] 张凯，毕利，焦小刚. 集成强化学习算法的柔性作业车间调度问题研究[J]. 中国机械工程，2023，34(2):201-207.
ZHANG Kai, BI Li, JIAO Xiaogang. Research on Flexible Job-shop Scheduling Problems with Integrated Reinforcement Learning Algorithm[J]. China Mechanical Engineering, 2023, 34(2):201-207.

[20] 陈睿奇，黎雯馨，王传洋，等. 基于深度强化学习的工序交互式智能体 Job shop 调度方法[J]. 机械工程学报，2023，59(12):78-88.
CHEN Ruiqi, LI Wenxin, WANG Chuanyang, et al. Interactive Operation Agent Scheduling Method for Job Shop Based on Deep Reinforcement Learning[J]. Journal of Mechanical Engineering, 2023, 59(12):78-88.

（编辑　陈　勇）

作者简介：李益兵，男，1978 年生，教授。研究方向为车间调度与优化等，发表论文 50 余篇。E-mail：ahlyb@whut.edu.cn。
郭　钧*（通信作者），男，1982 年生，副教授。研究方向为制造系统决策与优化，发表论文 20 余篇。E-mail：Junguo@whut.edu.cn。

本文引用格式：
李益兵，曹岩，郭钧，等. 考虑峰值功率受限约束的柔性作业车间调度研究[J]. 中国机械工程，2025，36(2):280-293.
LI Yibing, CAO Yan, GUO Jun, et al. Research on Flexible Job-shop Scheduling Considering Constraints of Peak Power Constrained[J]. China Mechanical Engineering, 2025, 36(2):280-293.