

DOI: 10.3901/JME.2023.13.246

基于改进Q学习的可重入混合流水车间 绿色动态调度*

吴秀丽 闫晓燕

(北京科技大学机械工程学院 北京 100083)

摘要: 节能减排对于实现可持续发展具有重要意义。考虑了动态扰动事件对生产的影响,研究了可重入混合流水车间绿色动态调度问题,提出了改进的Q学习算法。在可重入混合流水车间中,将各个加工阶段抽象为智能体,搭建了多智能体强化学习模型。选用均值漂移算法对历史状态进行聚类。为实现全局优化,设计了经验共享策略实现各个智能体之间的经验交互,并设计了自适应贪婪策略选取动作。最后进行了数值实验,实验结果表明,在求解可重入混合流水车间绿色动态调度问题时,改进的Q学习算法优于单一的调度规则,可以在提高生产效率的同时保证较低的能耗,并且能够对实际生产环境中的动态扰动因素快速做出反应,能够有效地解决实际问题。

关键词: 节能减排;可重入混合流水车间;绿色动态调度;改进的Q学习算法

中图分类号: TP391

An Improved Q Learning Algorithm to Optimize Green Dynamic Scheduling Problem in a Reentrant Hybrid Flow Shop

WU Xiuli YAN Xiaoyan

(School of Mechanical Engineering, University of Science and Technology Beijing, Beijing 100083)

Abstract: Energy conservation and emission reduction are of great significance to achieve sustainable development. This study considers the influence of dynamic disturbance events on production and studies the reentrant hybrid flow shop green dynamic scheduling problem (RHFS-GD). An improved Q learning algorithm (IQL) is proposed to solve the RHFS-GD problem. In a reentrant hybrid flow shop, each stage is abstracted as an agent, and a multi-agent reinforcement learning model is established. The mean shift algorithm is used to cluster the historical states. To achieve global optimization, an experience sharing strategy is designed to realize the experience interaction among agents, and an adaptive greedy strategy is proposed to select actions. Finally, numerical experiments are carried out, and the experimental results show that the IQL algorithm is superior to single scheduling rules, which can improve production efficiency while ensuring low energy consumption, can quickly respond to dynamic disturbance events in the actual production, and can effectively solve practical problems.

Key words: energy conservation and emission reduction; reentrant hybrid flow shop; green dynamic scheduling; improved Q learning algorithm

0 前言

全球气候变化正在对人类社会构成重大威

胁,节能减排已成为全球共识。世界各国也越来越关注“绿色制造”,碳排放量或能源消耗成为企业考虑的重要指标之一^[1]。2020年,中国宣布了“碳达峰”和“碳中和”的双碳目标,致力于推进“绿色制造”。这里的“绿色制造”指绿色科技创新与制造业转型发展深度融合形成的新

* 国家自然科学基金资助项目(52175499)。20220709收到初稿,20221219收到修改稿

技术、新业态、新模式,是全球新一轮工业革命和科技竞争的重要领域^[2]。“绿色制造”对提高我国制造业资源、能源利用效率,促进我国传统制造业绿色转型和战略新兴产业绿色发展,维护我国经济高质量发展和实现“双碳”战略愿景目标具有重大意义。

近些年来,为降低能耗成本,缓解碳排放的压力,不少学者开始聚焦于“绿色制造”和“绿色调度”。如段建国等^[3]面向绿色制造的半组合式船用曲轴结构件生产车间,提出了快速非支配排序遗传算法求解了最小化运输时间和加工能耗的多目标调度优化模型;耿凯峰等^[4]以最小化最大完工时间、总能耗成本和碳排放为目标,研究了绿色可重入混合流水车间调度问题,提出了一种改进的多目标文化基因算法求解问题;MANSOURI 等^[5]以最小化 makespan 和总能耗为目标,研究了序列相关的双机置换流水车间调度问题,通过统计实验比较了启发式算法与 CPLEX 算法的性能。

上述研究表明,目前关于绿色调度问题的研究中,大多都是针对传统的作业车间或者流水车间,关于可重入生产系统的绿色调度问题还有待进一步的研究。可重入生产系统是学者 KUMAR^[6]在 1993 年研究半导体制造系统的控制过程时提出的。可重入生产系统是不同于作业车间^[7]和流水车间^[8]的第三类生产系统,该类生产系统中,在不同的加工阶段,工件需要多次经过某些机器或某几个工作站进行加工。自可重入的概念被提出后,便被纳入到流水车间调度问题进行研究,形成了可重入流水车间调度问题。其中,绝大多数的研究中都考虑了工作站包含多台并行机的情况,并且将该类问题归结为可重入混合流水车间调度问题(Reentrant hybrid flow shop scheduling problem, RHFS))。目前,对于 RHFS 问题进行单目标优化的研究成果主要有:HUANG 等^[9]研究了发生在模具工厂的带有并行批处理机和两阶段 RHFS 问题,并提出了新的启发式算法进行求解;ZHANG 等^[10]提出了一种差分进化算法以最小化总拖期为目标求解 RHFS 问题。此外,也有很多学者研究 RHFS 的多目标优化问题,如姚远远等^[11]提出了改进的多目标灰

狼优化算法解决最小化最大完工时间和总延误的双目标 RHFS 问题;顾涛等^[12]针对无缝钢管冷拔生产中的周期式退火炉作批处理机的可重入批离散机流水车间调度问题,以最小化工件完工时间与批处理机能源消耗为优化目标,设计了多目标粒子群算法进行求解;YING 等^[13]以最小化 Makespan 和总拖期为目标,提出了迭代 Pareto 贪婪算法求解 RHFS 问题。

目前,制造企业除了面临着上述提到的降低能耗和碳排放的压力以外,实际工业环境还存在着较多的不确定性因素。生产过程中常见的动态扰动事件可以归纳为^[14]:①任务类的扰动,主要包括新任务插入、紧急订单或者订单取消等;②设备类的扰动,主要指机器发生故障;③工艺类的扰动,指工艺路线的改变或者工艺参数的偏差;④时间类的扰动,主要包括加工时间不确定、开工时间的延迟等等;⑤质量类的扰动,主要体现在成品或者中间品的质量不合格,需要重新生产或者再加工的情况。其中,任务类的扰动是属于生产过程中受到的外部扰动,其余四类扰动属于生产过程中的内部扰动。

为了应对这些动态事件,需要实现动态调度。现有的动态调度方法主要有鲁棒调度,预测-反应式调度和反应式调度三类。

鲁棒调度根据不确定事件的信息,添加了可调整时间的调度模式。这样的调度模式可以实现在调度过程中与动态扰动事件进行交互,可以提高整个调度方案的鲁棒性。例如,XIONG 等^[15]研究了考虑机器故障的柔性作业车间动态调度问题,设计了两种代理鲁棒性指标,一种是所有工序总松弛时间的权重和,另一种是考虑随机故障位置信息的前提下总松弛时间的权重和;AL-HINAI 等^[16]以鲁棒性和调度结果的稳定性作为目标,研究了考虑机器故障的动态柔性作业车间调度问题,设计了两阶段混合遗传算法。

预测-反应式调度是动态调度领域中应用最为广泛的方法。该方法的主要思路是在生产活动开始前先产生预调度方案,在后续的生产中再根据动态扰动事件去修改预调度方案。近年来关于预测-反应式调度策略的研究取得了很多成果。如 MEHTA 等^[17]在瓶颈机器上预留了一些空闲

时间段, 牺牲部分 Makespan 指标以换取调度方案缓冲机器故障的潜力; GAO 等^[18]根据新工件插入时间点将动态柔性作业车间调度问题划分成调度和重调度两个阶段, 并且设计了两阶段人工蜂群算法进行求解。

反应式调度也被称为在线调度或者实时调度。该类调度方法只需要在生产过程中根据实时状态或者实时发生的扰动事件做出反应。与预测—反应式调度相比, 反应式调度方法不用预先生成预调度方案, 其优点体现在可以快速生成调度方案, 实用性更强。目前的研究集中于设计一些高效的调度规则或者采用机器学习中的强化学习、深度强化学习等来实现实时动态调度。如 NIE 等^[19]研究了工件推移到达的动态作业车间调度问题, 提出一种基于基因编程表达式算法构建调度规则; 王维祺等^[20]针对作业车间的动态调度问题, 重新设计了 Q 学习算法的要素去求解问题; WANG 等^[21]对于强化学习在生产调度方面的研究进行了综述, 梳理出基于强化学习的生产调度问题, 并综述了强化学习在不同类型调度问题中的应用, 最后分析和总结了现有研究中存在的不足。

动态调度一直强调实时性、在线性以及自适应性, 要求可以对出现的动态扰动事件快速做出反应, 不影响整个调度过程, 而反应式调度方法相比其余两种可以更好地满足实时和快速反应的要求。反应式调度的应用中, 由于强化学习可以实现与环境的不断交互, 通过在交互中不断地探索和学习获取更大的累积奖励, 相比单一调度规则而言, 具有更长远的眼光, 在实现实时调度上更有优势。因此, 强化学习在调度问题研究领域得到了广泛的应用^[20-22]。强化学习算法中 Q 学习算法作为一类经典算法, 因其反应速度快、易实现的优点, 近些年被广泛的应用于求解调度问题。如曹红倩^[23]将 Q 学习算法应用到柔性作业车间调度问题中, 并对动作选择策略进行了改进, 提升了 Q 学习算法在解决该问题时的速度和精度; 韩忻辰等^[24]基于 Q 学习算法, 提出了高速铁路列车动态调度方法, 并通过仿真实验验证了 Q 学习算法用于高铁动态调度的有效性。从现有研究中发现, Q 学习算法在求解调度问题时具有一定的优势, 因此, 本文将结合 RHFS 问题的特点, 设计改进的 Q 学习算法进

行求解。

由于 RHFS 问题本身具有高度重入、多工序、多并行机等特点, 实际生产中会受到动态扰动事件的影响, “绿色制造”又增加了关于能耗的优化目标和约束条件, 所以联合形成的问题更为复杂。本文将针对可重入混合流水车间绿色动态调度问题 (Reentrant hybrid flow shop green dynamic scheduling problem, RHFS-GD) 进行研究。本文主要针对多阶段的 RHFS-GD 问题, 搭建了多智能体的强化学习模型, 并设计了改进的 Q 学习算法 (An improved Q-learning algorithm (IQL)) 求解; 设计了经验共享策略来实现多智能体之间的经验交互; 提出了自适应贪婪策略来辅助智能体选取动作。

1 RHFS-GD 问题描述

1.1 问题描述

RHFS-GD 问题可以描述为: 可重入生产系统中共有 W 个工作站, 有 N 个工件等待加工, 每个工件有 N_i 道工序。如图 1 所示, 每个工作站至少有一台机器, 且至少有一个工作站有两台或两台以上并行机。工件需要依次经过每个工作站来完成加工, 在完成第一道次加工后, 需要重入系统 $(L-1)$ 个道次完成加工, 即工件总的加工道次为 L 。这里的道次指的是工件重复在系统循环加工的次数。由于工件需要重入某些机器进行加工, 加工过程中需要消耗较多能耗。因此, 为了降低能源消耗, 在求解过程中除了优化生产效率如 makespan 和总拖期两个性能指标以外, 也优化了总能耗指标。调度的任务是以优化上述性能指标为目的, 确定每个工作站前缓冲区中工件的加工顺序并为每个工件的每道工序分配加工机器, 调度过程中考虑实际发生的动态扰动事件。

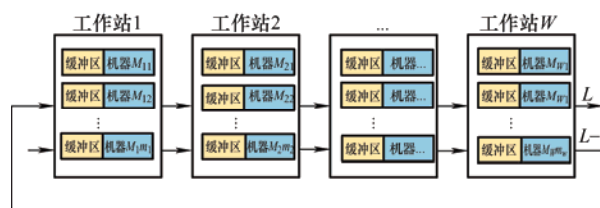


图1 可重入混合流水车间示意图

1.2 符号说明

本文用到的符号如表 1 所示。

表 1 符号说明

符号	定义
N	工件数
n_i	工件 i 的工序数
W	工作站数
M	机器总数
L	道次总数
m	加工阶段或智能体数
h	状态特征总数
k	机器索引, $k=1,2,\dots,M$
O_{ij}	工件 i 的第 j 道工序
i	工件索引, $i=1,2,\dots,N$
w	工作站索引, $w=1,2,\dots,W$
M_w	工作站 w 包含的并行机数
j	加工阶段(工序或智能体)索引, $j=1,2,\dots,m$
u	状态向量索引, $u=1,2,\dots,h$
o	当前阶段或状态, $o=1,2,\dots,m$
t	当前时刻
$S_{jt}^{(u)}$	状态集中第 u 个状态向量
S_{jt}'	智能体 j 在时刻 t 的状态
S_{jt}	智能体 j 在时刻 t 的聚类状态
a_{jt}	智能体 j 在当前状态采取的动作
a_{jt}^*	在当前状态下智能体 j 的最佳的动作
r_{jt}	智能体 j 实施动作后所获的奖赏值
$Q^i(S_{jt}, a_{jt})$	在状态 S_{jt} 下的动作 a_{jt} 在智能体 j 的 Q 表中对应的 Q 值
$Q^i(S_{jt}, a_{jt}^*)$	在状态 S_{jt} 下所有动作在智能体 j 的 Q 表中的最大 Q 值
$Q^c(S_{jt}, a_{jt})$	在状态 S_{jt} 下的动作 a_{jt} 在公共 Q 表中的 Q 值
q_o	当前智能体的缓冲区前等待加工的工件总数
C_i	工件 i 的完工时间
B_{ij}	工序 O_{ij} 的开始时间
C_{ij}	工序 O_{ij} 的结束时间
p_{ij}	工序 O_{ij} 的加工时间
d_i	工件 i 的交货期
wt_{io}	工件 i 在当前智能体 o 前产生的等待时间
at_{io}	工件 i 到达智能体 o 的时间
rt_{io}	工件 i 完成第 o 个加工阶段后的剩余加工时间
st_{io}	工序 O_{ij} 的松弛时间, $st_{io} = d_i - t - rt_{io}$
h_k	机器 k 上安排的工序数
b	机器 k 上安排的工序索引, $b=1,2,\dots,h_k$
P_{idle}^k	机器 k 的闲置功率
$P_{process}^k$	机器 k 的加工功率
$E_{process}^k$	机器 k 产生的加工能耗
E_{idle}^k	机器 k 产生的闲置能耗
$E_{process}^t$	在 t 时刻执行动作 a_{jt} , 机器产生的加工能耗
E_{idle}^t	在 t 时刻执行动作 a_{jt} , 机器产生的闲置能耗
$F_{(t-1)}$	转移到状态 $S_{j(t-1)}$ 的时刻
F_t	转移到状态 S_{jt} 的时刻
C_{max}	最大完工时间
E	总能耗
D	总拖期
X_{ijk}	如果工序在机器 k 上加工, $X_{ijk}=1$; 否则, $X_{ijk}=0$
Y_{hgij}	如果 O_{hg} 是与 O_{ij} 相邻的前一道工序, $Y_{hgij}=-1$; 如果 O_{hg} 是与 O_{ij} 相邻的后一道工序, $Y_{hgij}=1$; 否则, $Y_{hgij}=0$

1.3 假设条件

为便于研究 RHFS-GD 问题, 做出如下假设。

- (1) 工件之间相互独立, 即不同的工件之间不存在加工顺序的约束。
- (2) 初始时刻工件和机器都是可用的。
- (3) 工件开始加工后不允许中断。
- (4) 任意一个工件只有完成上一道工序, 才可以开始下一道工序的加工; 任意工件只有完成上一道次所有工序的加工后才可以进入下一道次。
- (5) 每一道工序的加工时间已知。
- (6) 每道工序只能由一台机器完成加工。
- (7) 每台机器同一时刻只能加工一个工件。

2 RHFS-GD 调度优化模型

为求解 RHFS-GD 问题, 以最小化最大完工时间、总能耗和总拖期为目标, 建立了调度优化模型。

$$f = \min(C_{\max}, E, D) \quad (1)$$

$$C_{\max} = \max(C_i) \quad (2)$$

$$E = \sum_{k=1}^m (E_{idle}^k + E_{process}^k) \quad (3)$$

s.t.

$$E_{idle}^k = P_{idle}^k (B_{1k} + \sum_{b=2}^{h_k} (B_{bk} - C_{(b-1)k})) \quad \forall k, h_k > 1 \quad (4)$$

$$E_{process}^k = P_{process}^k (\sum_{i=1}^n \sum_{j=1}^m p_{ijk} X_{ijk}) \quad \forall k \quad (5)$$

$$D = \sum_{i=1}^N \max(0, C_i - d_i) \quad (6)$$

$$M_w \geq 1 \quad \forall w \quad (7)$$

$$M_w \geq 2 \quad \exists w \quad (8)$$

$$B_{ij} \geq C_{i(j-1)} \quad \forall i, j > 1 \quad (9)$$

$$B_{i1} \geq C_{im} \quad \forall i \quad (10)$$

$$\sum_{k=1}^m X_{ijk} = 1 \quad \forall i, j \quad (11)$$

$$(C_{ij} - C_{hg} - p_{ij}) X_{hgk} X_{ijk} \left(\frac{Y_{hgij}}{2} \right) (Y_{hgij} - 1) +$$

$$(C_{hg} - C_{ij} - p_{hgk})X_{hgk}X_{ijk}(\frac{Y_{hgij}}{2})(Y_{hgij} + 1) \geq 0$$

$$\forall i, j, h, g, k \quad (12)$$

$$B_{ij} > 0 \quad \forall i, j \quad (13)$$

$$C_{ij} > 0 \quad \forall i, j \quad (14)$$

$$C_{ij} = p_{ij} + B_{ij} \quad \forall i, j \quad (15)$$

$$C_{\max} > C_i \quad \forall i \quad (16)$$

$$X_{ijk} \in \{0, 1\} \quad \forall i, j, k \quad (17)$$

$$Y_{hgij} \in \{-1, 0, 1\} \quad \forall i, j, h, g \quad (18)$$

式(1)列出了最小化 makespan、总能耗和总拖期的目标函数。式(2)定义了 makespan 的计算方式。式(3)-(5)给出了能耗的计算公式。式(6)为总拖期的计算公式。式(7)表示每个工作站至少包含一台并行机。式(8)表示至少有一个工作站包含两台或两台以上并行机。式(9)表示工序的开始加工时间不得早于紧前工序的完工时间。式(10)约束了下一道次第一道工序的开始时间。式(11)表示一道工序只能选一台机器完成加工。式(12)表示每台机器在同一时刻只能加工一道工序。式(13)和(14)确保了工序的开始加工时间完工时间不为负。式(15)计算了每道工序的结束时间。式(16)是最大完工时间的约束。式(17)-(18)确定了决策变量的取值。

3 求解 RHFS-GD 问题的 IQL 算法

3.1 强化学习模型

3.1.1 多智能体强化学习模型

强化学习是人工智能领域中机器学习的关键技术，其主要特点是与环境不断地交互学习，该方法具有很强的自适应能力和实时学习的能力。强化学习的目标主要在于与环境的试探性交互中学习相应的行为策略来获取最大的长期奖赏。如图 2 描述了强化学习的流程，强化学习的主体分别是智能体以及智能体所处的环境。环境意味着多样的复杂状态， t 时刻的状态可以表示为 S_{jt} ，当智能体接收到 t 时刻的状态 S_{jt} 时，智能体将根据当前状态并结合动作选取策略从可选的动作集

合中选取一个动作来执行，同时环境状态从 S_{jt} 转换成 $S_{j(t+1)}$ 。以此循环，根据学习到的策略，不断尝试并调整行为来获取最大的长期奖赏。

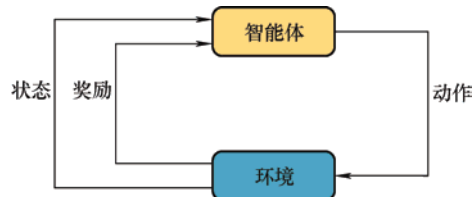


图 2 强化学习过程

利用强化学习求解 RHFS-GD 问题，第一个要解决的问题是如何将 RHFS-GD 问题转换为强化学习的模型。本文在使用 IQL 算法求解 RHFS-GD 问题时，将每个加工阶段作为一个智能体，调度过程就可以抽象成多智能体与环境交互的过程。当车间生产状态发生改变或出现扰动事件时，多智能体强化学习模型根据系统当前的状态选择合理的动作，动作是如表 3 所示的调度规则，并按照调度规则确定对应的工件进行加工。在整个过程中多智能体强化学习模型通过与实际生产车间的不断交互来提高生产系统的适应能力。如图 3 所示，展示了 IQL 算法在求解 RHFS-GD 问题时的多智能体交互模型。

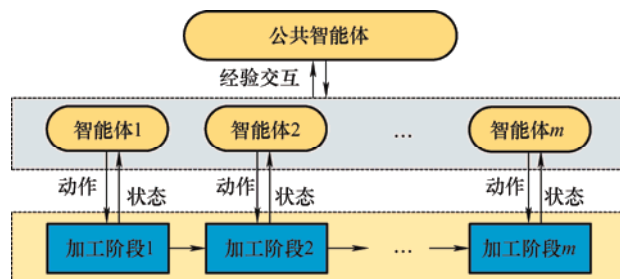


图 3 多智能体交互模型

3.1.2 状态特征向量

设置合适的状态能够实时反映系统状态的变化情况，如机器的状态、工件的加工信息、各个工件的加工进度以及已完成工件和等待工件的数量等，同时也能够反映出生产调度的实时情况。针对 RHFS-GD 问题的特点，触发状态转移的事件主要有工序的加工完成、新工件的到达和机器发生故障等。将 RHFS-GD 问题中的每个加工阶段作为一个智能体，每一个智能体在 t 时刻对应的状态集 S_{jt} ，表示为 $S_{jt} = \{s_{jt}^{(1)}, s_{jt}^{(2)}, s_{jt}^{(3)}, \dots, s_{jt}^{(h)}\}$ ，其中 $s_{jt}^{(u)}$ 表示车间中第 j 个加工阶段的第 u 个状态特征。如表 2 所示，7 个状态特征描述了 t 时刻每个智能体的系统状态 S_{jt} 。

表 2 状态特征

状态特征	描述	计算方法
$s_{jt}^{(1)}$	当前缓冲区前工件的平均等待时间	$\frac{\sum_{i=1}^{q_o} wt_{io}}{q_o}, q_o \neq 0$
$s_{jt}^{(2)}$	缓冲区中工件松弛时间均值	$\frac{\sum_{i=1}^{q_o} (d_i - t - \sum_{j=a}^m p_{ij})}{q_o}, q_o \neq 0$
$s_{jt}^{(3)}$	缓冲区中所有工件当前工序加工时间的最大值和均值的比值	$\frac{\max_{0 \leq i \leq q_o} (p_{io})}{\frac{\sum_{i=1}^{q_o} p_{io}}{q_o}}, q_o \neq 0, \forall o$
$s_{jt}^{(4)}$	缓冲区中所有工件当前工序加工时间的最小值和均值的比值	$\frac{\min_{0 \leq i \leq q_o} (p_{io})}{\frac{\sum_{i=1}^{q_o} p_{io}}{q_o}}, q_o \neq 0, \forall o$
$s_{jt}^{(5)}$	工件的平均剩余加工时间	$\frac{\sum_{i=1}^{q_o} \sum_{j=a}^m p_{ij}}{q_o}, q_o \neq 0, \forall o$
$s_{jt}^{(6)}$	工件的最小交货期	$\max(d_i), \forall i$
$s_{jt}^{(7)}$	工件的最大交货期	$\min(d_i), \forall i$

对于每个智能体来说,当系统状态发生变化时,智能体需要根据当前状态采取下一步动作。为了便于将状态值离散到对应的 Q 表中,搜集历史状态数据并对其进行聚类,之后接收到系统状态后,根据聚类结果确定对应的聚类状态,最后根据聚类状态确定动作。

3.1.3 动作

将每一个智能体当前可选的调度规则作为可选

的动作。调度规则主要用来将等待加工的工件或加工任务分配给机器去完成加工。当工件到达缓冲区时,根据“哪台机器先空闲选择哪台机器”的规则来选择加工机器,除此外,还需要为每台机器确定要加工的工件,本文选取了现有的 7 种调度规则以及不选取工件的操作作为每个智能体可选的动作集合,具体如表 3 所示。

表 3 调度规则

调度规则	描述	计算方法
—	不选取任何工件	—
FIFO	选取最先到达缓冲区的工作	$i^* = \arg \min(at_{io} i \in QN)$
FILO	选择最晚到达缓冲区的工作	$i^* = \arg \max(at_{io} i \in QN)$
SPT	选择当前工序加工时间最短的工件,若相同,选择到达时间最早的工件	$i^* = \arg \min(p_{io} i \in QN)$
LPT	选择当前工序加工时间最长的工件,若相同,选择到达时间最早的工件	$i^* = \arg \max(p_{io} i \in QN)$
SPRT	选择剩余加工时长最短的工件,若相同,选择到达时间最早的工件	$i^* = \arg \min(rt_{io} i \in QN)$
EDD	选择交货期最早的工件	$i^* = \arg \max(rt_{io} i \in QN)$
SLT	选择松弛时间最小的工件	$i^* = \arg \min(st_{io} i \in QN)$

3.2 动态调度方法总框架

RHFS-GD 问题中,工件根据实际加工需求多次重入某些工作站,再加上实际生产环境的不确定性和绿色制造的约束,使得该问题的优化极为复杂,而 Q 学习算法具有较强的自适应能力和实时学习的能力,可以通过与环境的不断交互学习选取行为策略。因此,本文基于经典的 Q 学习算法,结合 RHFS-GD 问题的特点,提出了 IQL 算法。图 4 为

该算法的总框架。

(1) 根据均值漂移聚类算法^[25]将历史状态数据进行聚类,并记录聚类结果。

(2) 初始化经验共享概率 $k = 0.9$, 贪婪选择概率 $\varepsilon = 1$, 各个智能体的 Q 值表和公共 Q 值表。

(3) 观察智能体的当前状态 S'_{jt} , 并根据步骤 1 得到的聚类结果确定当前系统状态对应的聚类状态 S_{jt} 。

(4) 经验共享, 用 $Q^c(S_{jt}, a_{jt}^*)$ 以概率 $1-k$ 替换对应智能体的 Q 值 $Q^i(S_{jt}, a_{jt}^*)$ 。

(5) 根据贪婪搜索策略来选取动作。以概率 $1-\varepsilon$ 从对应 Q 值表中选择 Q 值最大的动作 $a = \arg \max_a Q(s, a)$, 否则随机选取动作。

(6) 执行动作并计算奖励 r 。

(7) 更新参数 $k = 0.9 \times k$, 并更新 ε , 并计算新的状态值。

(8) 更新 Q 值表 Q^i 公共 Q 值表 Q^c 。

(9) 判断是否所有工件都调度完毕? 是, 进行下一代学习; 否, 循环步骤(3)~(8)。

(10) 判断是否满足终止条件? 是, 输出结果; 否, 返回步骤(3)~(9)。

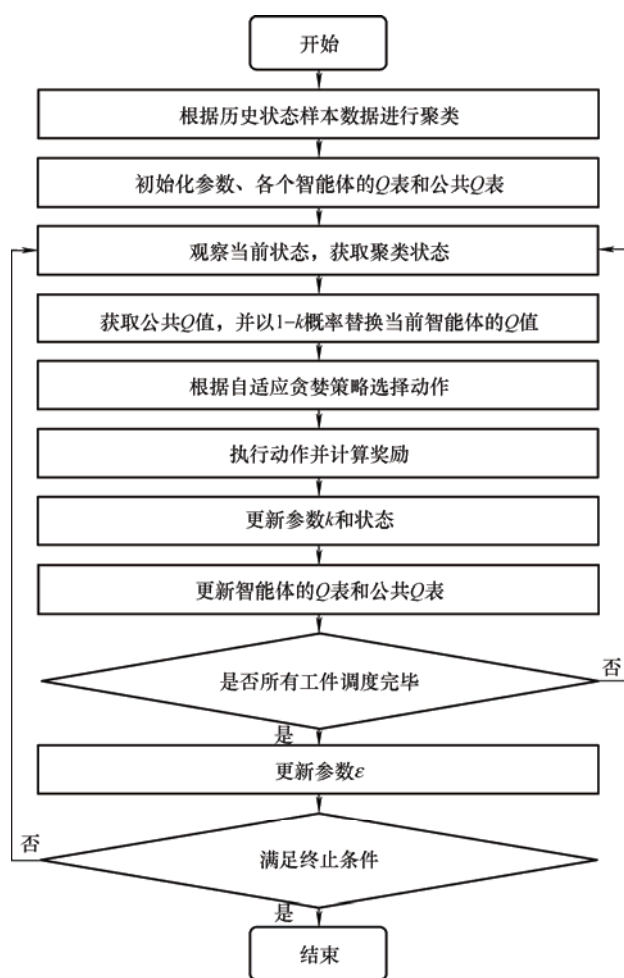


图 4 IQL 算法流程图

3.3 算法详细步骤

3.3.1 均值漂移聚类算法

在求解 RHFS-GD 问题时, 每一道工序的完成或者动态扰动事件的发生都会触发一次新状态的产生。每个状态都包含 7 个状态向量, 为了便于将状态离散到 Q 表中, 降低算法的搜索空间, 本文参考

现有文献中的处理方法^[26], 采用了聚类算法来对历史状态数据进行聚类。

均值漂移算法是无监督学习中常被使用的一种强大的聚类算法。其核心思想是通过将数据点移向最高密度的数据点(即群集质心), 不断迭代将数据点分配给群集。与 K-means 聚类相比, 这种算法不需要选择簇的数量, 均值漂移会自动确定聚类的簇数, 状态空间的维度可以根据历史状态数据的聚类结果直接确定; 另外, 均值漂移聚类算法是基于密度的聚类算法, 受异常值的影响小。所以, 本文选用该算法实现对历史状态的聚类。

均值偏移算法的主要步骤如下所述。

(1) 在没有被分类的数据点中随机地选取一个点作为中心点。

(2) 确定半径, 距离中心点的距离小于等于半径的点记作集合 M , 集合内的这些点属于当前的簇 c 。

(3) 计算整个圆形空间内所有向量的平均值, 得到偏移向量。

(4) 中心点沿着偏移向量的方向移动, 移动的距离等于偏移向量的模。

(5) 重复步骤(2)~(4), 直到偏移向量的大小满足初始设定的阈值要求, 并记录中心点。

(6) 重复步骤(1)~(5), 直到所有的点完成归类。

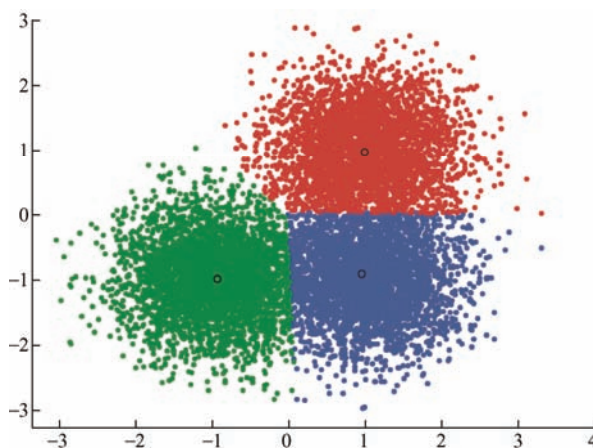


图 5 均值漂移算法示例图

3.3.2 经验共享策略

在 Q-learning 求解调度问题过程中, 所有智能体都具有相同的动作集合和奖赏函数。经验交互就是在决策时考虑其他的智能体所获得经验。在学习过程中, 每个智能体迭代更新自己的 Q 值表如公式(19)所示。

$$Q^i(S_{jt}, a_{jt}) = Q^i(S_{jt}, a_{jt}) + \alpha[r^i(S_{jt}, a_{jt}) + \gamma \max_a Q^i(S_{j(t+1)}, a_{jt}) - Q^i(S_{jt}, a_{jt})] \quad (19)$$

为实现各个智能体之间的经验交互, 本文根据现有研究中经验交互的思想^[26], 对其进行了改进, 设计了经验共享策略。由于经验交互的目的是促使各个智能体在选取动作时, 综合考虑动作对整体产生的影响, 所以公共 Q 表在更新过程中要考虑所有智能体。公共 Q 表具体更新的步骤如下: 首先, 当智能体接收到新状态 $S_{j(t+1)}$ 时, 在公共 Q 表中找出最大的 Q 值 $Q^c(S_{j(t+1)}, a_{j(t+1)}^*)$, $a_{j(t+1)}^*$ 表示最大 Q 值对应的动作; 之后, 找出每个智能体对应的 $Q^i(S_{j(t+1)}, a_{j(t+1)}^*)$, m 表示智能体的总数; 最后, 按照式(20)来更新公共 Q 值表。

$$Q^c(S_{j(t+1)}, a_{j(t+1)}^*) = \frac{\sum_{i=1}^m Q^i(S_{j(t+1)}, a_{j(t+1)}^*)}{m} \quad (20)$$

各个智能体将参考自身的 Q 值表选取动作, 为实现多智能体之间的经验交互, 选取动作前以 $1-k$ 的概率用 $Q^c(S_{j(t+1)}, a_{j(t+1)}^*)$ 来更新当前智能体的 $Q^i(S_{j(t+1)}, a_{j(t+1)}^*)$ 。

3.3.3 自适应贪婪策略

在学习过程中, 智能体通过与环境不断交互来尝试选取一个动作。通过获得累计奖励, 智能体开始倾向于选择奖励值较大的动作。为避免采取传统的贪婪搜索策略导致算法陷入局部最优, 设计了适应贪婪策略来实现动作的选取。

以概率 $1-\varepsilon$ 从对应 Q 值表中选择 Q 值最大的动作, 否则随机选取动作进行调度。其中 $\varepsilon = \frac{1}{\zeta^2}$, 初始 $\zeta = 1$, 每迭代一次, ζ 按照如下的伪代码更新。

伪代码 1 自适应探索策略
输入: 迭代到当前代数的适应度值集合
输出: 参数 ζ
1: if $y_e = \min(\mathbf{y}) \quad e \geq 2$, then
2: $\zeta = \zeta + 0.6$
3: else if $y_e < y_{e-1} \quad (e \geq 2)$, then
4: $\zeta = \zeta + 0.3$
5: else
6: $\zeta = \zeta + 0.1$
7: end if

3.3.4 奖赏函数

多智能体在车间执行动作后, 会收到车间反馈给它的奖励。奖励函数的定义应该在较大程度上贴近性能指标或目标函数。针对本文考虑的几个性能指标, 奖赏函数的设置如公式(21)所示。

$$r = (F_{(t-1)} - F_t) + \frac{1}{(E_{idle}^t + E_{process}^t)} \quad (21)$$

其中 $F_{(t-1)} - F_t$ 表示了状态转移前后的时间差, 可以优化 makespan 和总拖期; $\frac{1}{(E_{idle}^t + E_{process}^t)}$ 为状态发生转移后产生的总能耗的倒数, 能够优化总能耗。

因状态转移前后的时间差和能耗值量纲相差较大, 按照公式(22)对其进行处理。

$$X = \frac{X - \min}{(\max - \min)} \quad (22)$$

将每次状态转移后的时间差值和能耗值储存到不同的集合中, 根据公式(22)对两个指标值进行归一化处理。其中 X 表示当前获得的指标值, \min 为集合中的最小值, \max 表示集合中的最大值。

4 数值实验

4.1 实验设计

4.1.1 环境设置

所有算法均在 11th Gen Intel(R) Core(TM) i5-11320H @ 3.20 GHz, 16.00 GB RAM, Win11 64 位操作系统和 Python3.7 编程环境下编译运行。

4.1.2 实验数据

采用表 4 的方法, 随机生成 20 组算例, 其中工件数为 15 和 25 的各生成 10 组, 用来测试 IQL 在求解不同规模算例下的表现, 并且根据表 5 设置调度过程中发生的动态扰动事件。

表 4 算例设置

参数	规模
工件个数	15, 25
阶段数	7
可重入次数	U [1, 6]
工作站包含的并行机数	U [1, 4]
工序的加工时间	U [1, 30]

表 5 动态扰动事件设置

事件	设置
故障	故障机器: 随机; 故障时间: U [10, 30]
订单到达	到达时间: 随机; 到达工件数: U [1, 5]

4.1.3 实验目的

(1) 为了确定算法参数, 设计了参数正交试验。

(2) 为了验证 IQL 算法在求解 RHFS-GD 问题时的表现, 将其与经典调度规则进行对比。

4.2 参数正交试验

对于 IQL 算法,除了自适应调整的参数以外,其余两个需要设置的关键参数是折扣率和学习率,为了确定算法参数,选用 makespan 的平均值为评价指标,选用其中一个算例进行了参数正交试验。正交表如表 6 所示。

表 6 正交试验表

因素	γ	α	makespan 均值
1	0.05	0.8	373.10
2	0.1	0.85	376.07
3	0.15	0.9	375.43
4	0.05	0.9	374.21
5	0.1	0.8	372.08
6	0.15	0.85	370.31
7	0.05	0.85	372.03
8	0.1	0.9	371.29
9	0.15	0.8	373.52
k_1	373.11	372.90	$\alpha=0.15$
k_2	373.15	372.80	$\gamma=0.85$
k_3	373.09	373.64	

结果显示当学习率 $\alpha=0.15$,折扣率 $\gamma=0.85$ 时 IQL 表现最好。

4.3 IQL 算法性能分析实验

IQL 最大的优势是能够实现对系统的实时控制,能够及时对动态扰动事件做出反应。为验证 IQL 算法的收敛性,选用其中一个算例进行了实验。

如图 6 给出了累积奖赏值的收敛图。图 7~9 分别是 makespan,总拖期和总能耗三个指标的收敛图。实验结果显示,IQL 算法的收敛性能较好,迭代过程中没有大幅度的振荡。

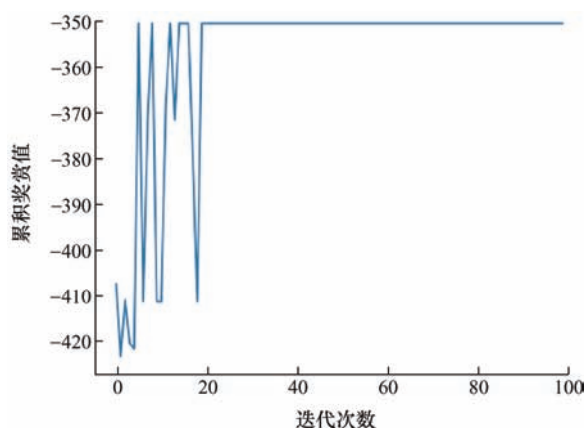


图 6 累积奖赏值收敛曲线

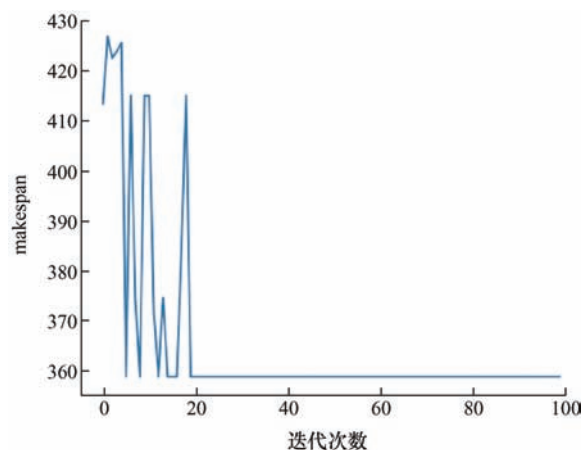


图 7 Makespan 收敛曲线

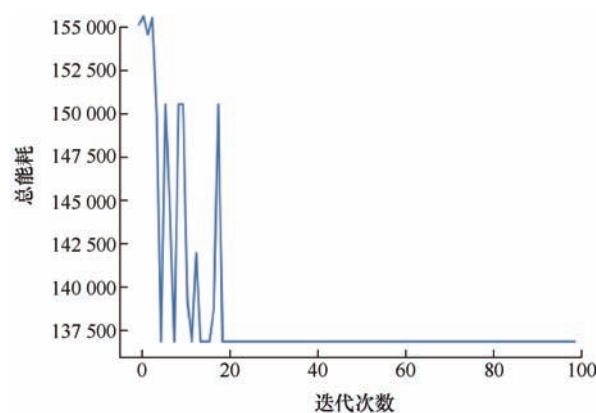


图 8 总能耗收敛曲线

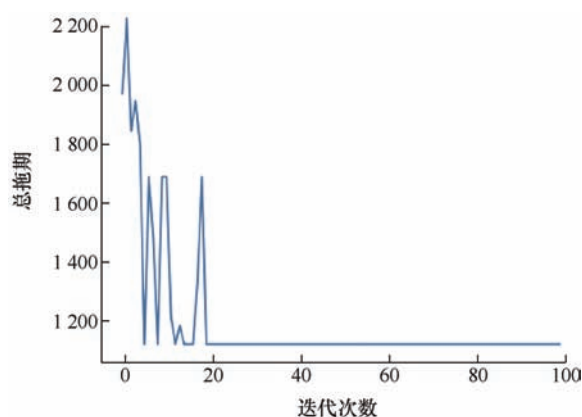


图 9 总拖期收敛曲线

为验证 IQL 算法可以实时地对动态扰动事件做出反应,进行了如下实验。随机设计了机器故障和新订单到达两类动态扰动事件,图 10 和图 11 分别是动态扰动事件发生前后的 IQL 算法求解得出的调度甘特图。横轴为时间,纵轴是机器。例如 M11 代表第一个工作站的第一台并行机,图中每个矩形条代表一道工序的加工时长,矩形条上

显示的数字如(6, 3)表示第 6 个工件的第 3 道工序, 该数字重复出现代表该工件重复访问了某工作站。实验过程中, 采用随机产生扰动的方法进

行测试, 设置的扰动事件有机器故障和新订单到达, 如图 11, 灰色斜线条表示机器发生故障, 虚线表示新订单的到达。

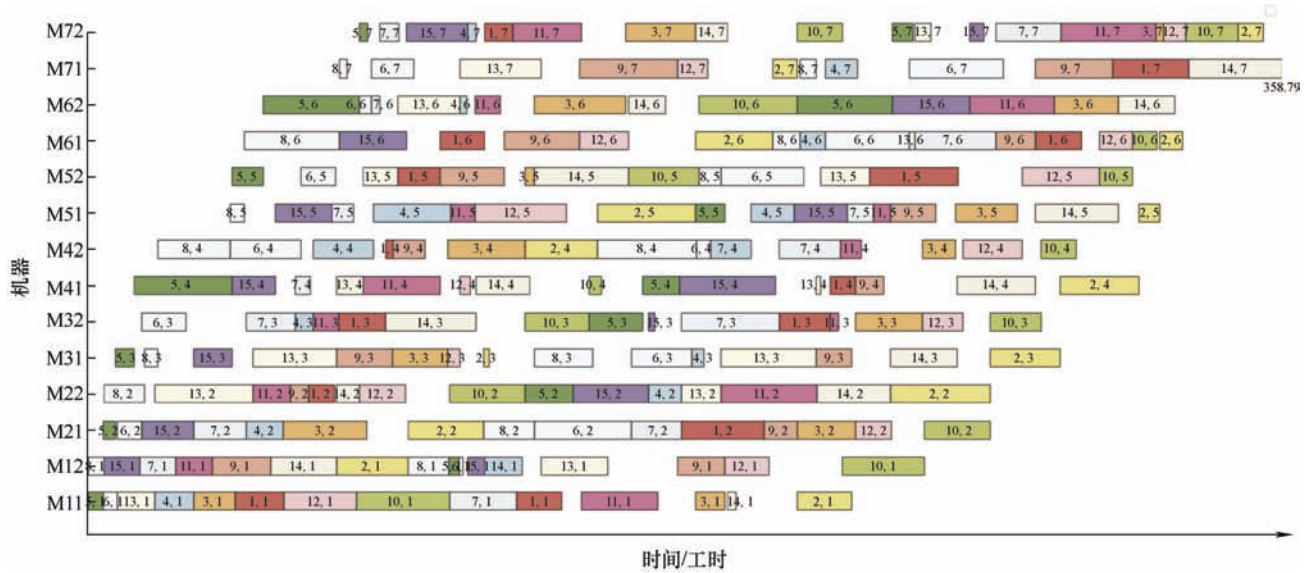


图 10 无动态扰动实时调度甘特图

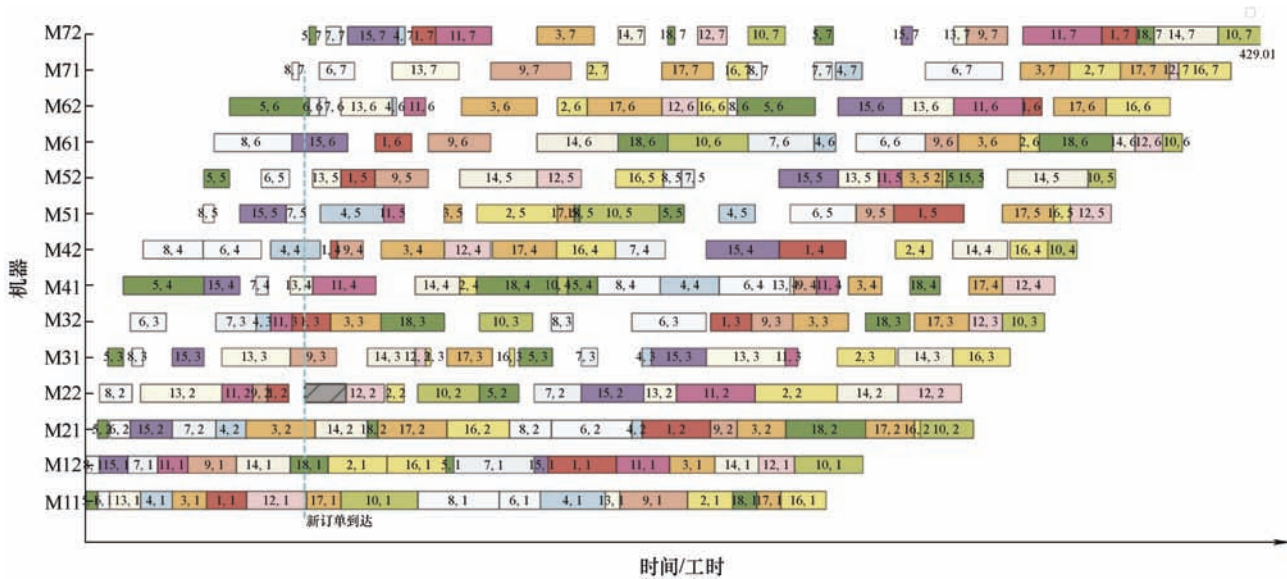


图 11 发生动态扰动后实时调度甘特图

为了验证 IQL 算法的学习和搜索能力, 将 IQL 算法与表 3 中的调度规则进行对比。如图 12-图 14 分别展示了 3 个性能指标下求解小规模问题时 IQL 算法与调度规则的对比柱状图, 图 15-图 17 展示了 3 个性能指标下求解大规模问题时 IQL 算法与调度规则的对比柱状图。实验结果显示, 在 makespan 指标下, 除了算例 2, IQL 算法

与 SPT 规则结果一样, 其余算例下, 均优于单一的调度规则。针对总能耗的表现, 除了算例 8 的实验结果显示 IQL 算法与 SPT 规则表现一致, 其余结果均是 IQL 算法表现最好。针对总拖期指标, 除了求解算例 2 时 IQL 算法与 SPT 规则结果一样以外, 其余算例 IQL 算法均优于单一的调度规则。

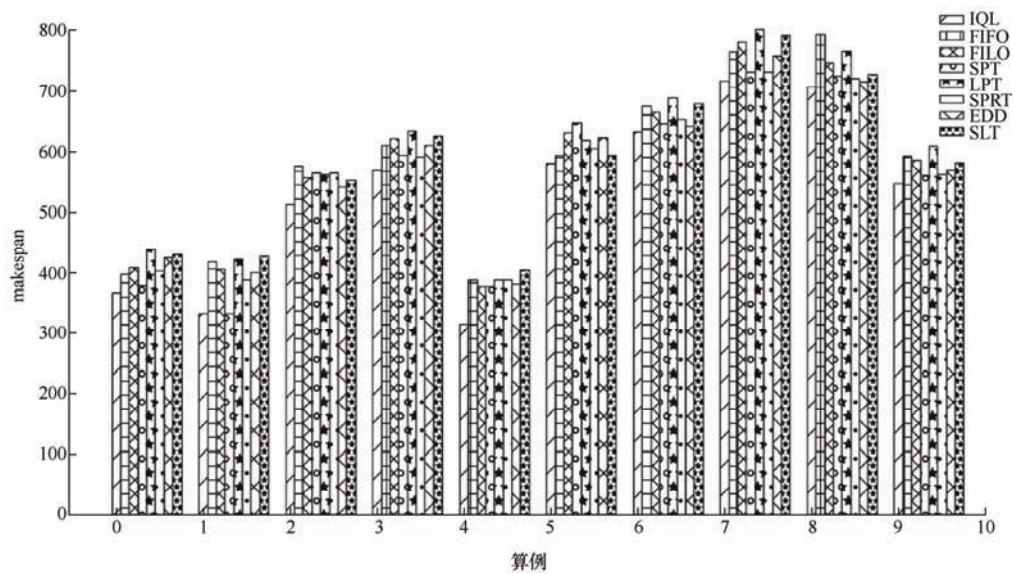


图 12 小规模算例下 IQL 算法与 7 种调度规则的 makespan 对比

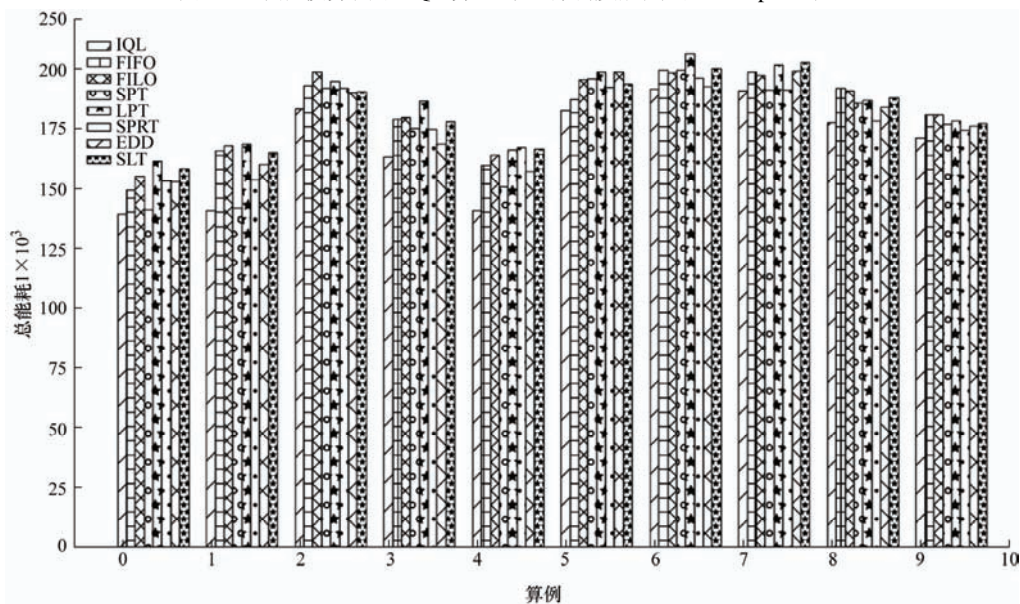


图 13 小规模算例下 IQL 算法与 7 种调度规则的总能耗对比图

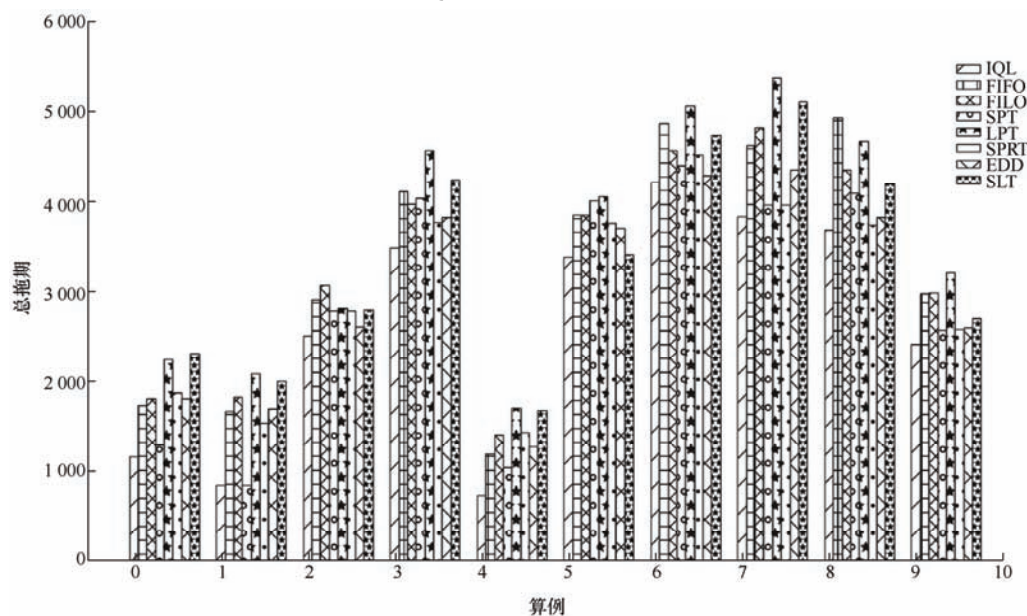


图 14 小规模算例下 IQL 算法与 7 种调度规则的总拖期对比图

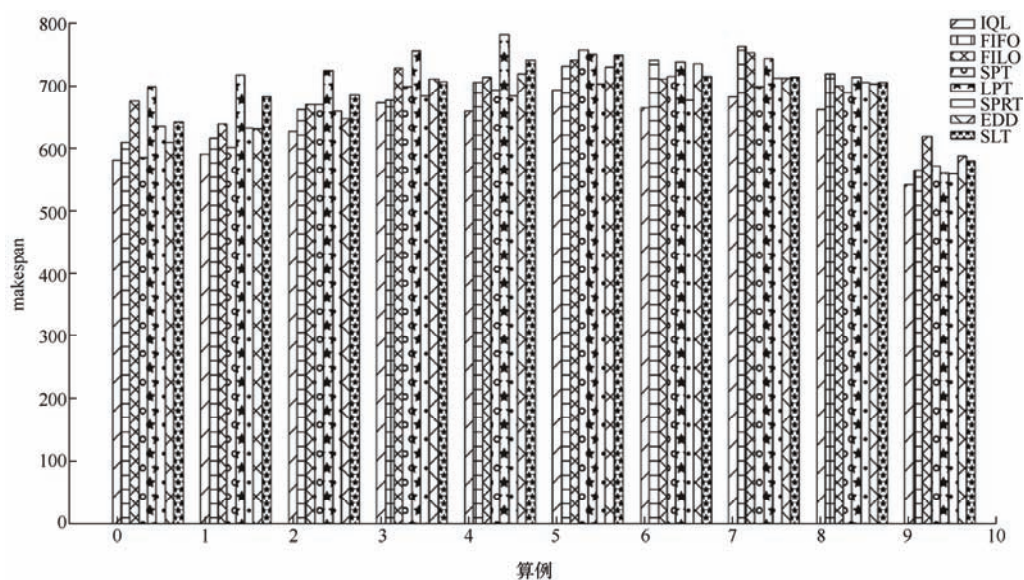


图 15 大规模算例下 IQL 算法与 7 种调度规则的 makespan 对比图

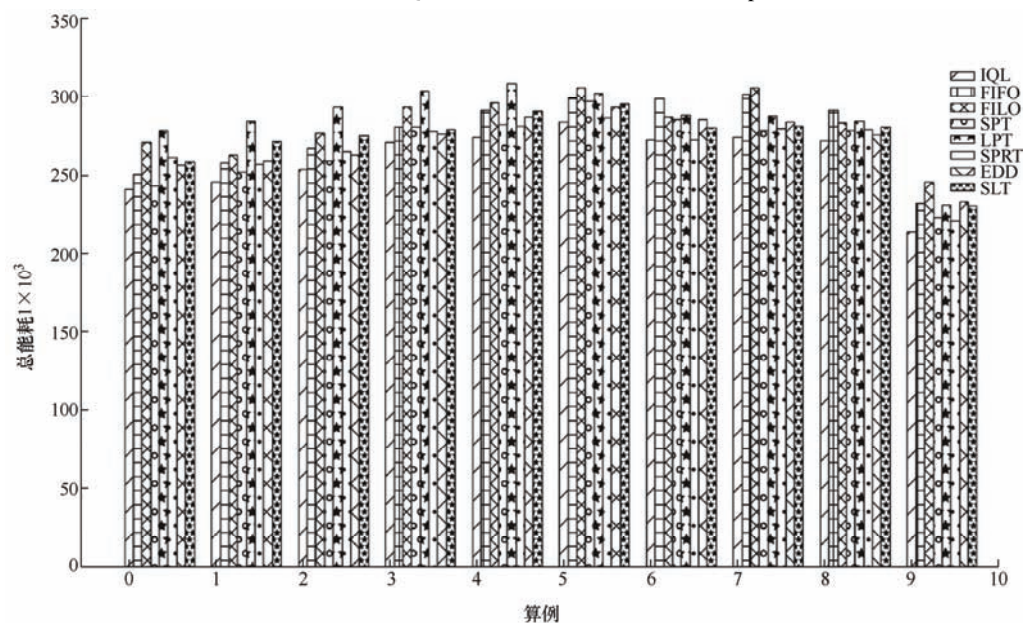


图 16 大规模算例下 IQL 算法与 7 种调度规则的总能耗对比图

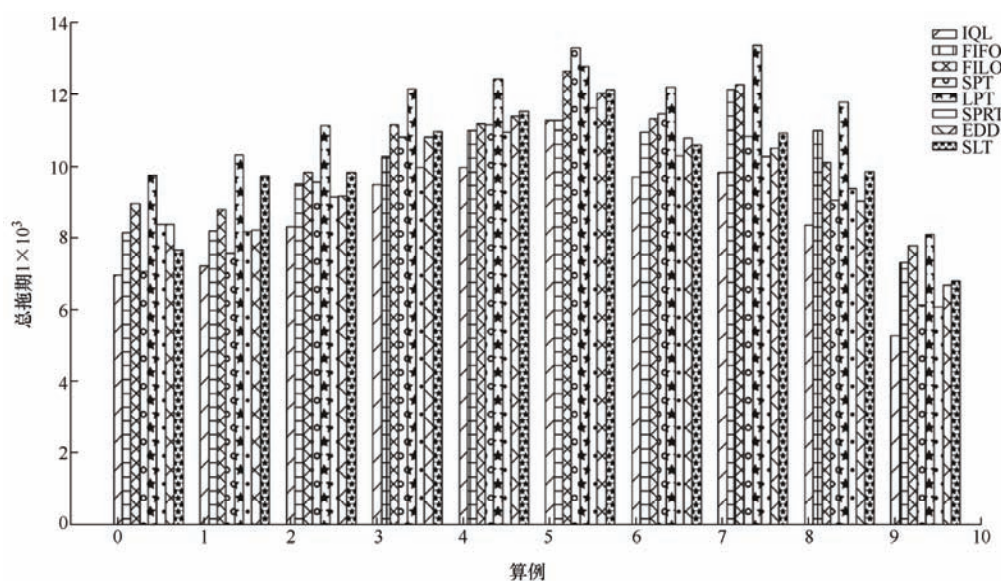


图 17 大规模算例下 IQL 算法与 7 种调度规则的总拖期对比图

5 结论

本文针对可重入混合流水车间绿色动态调度问题,提出了一种改进的Q学习算法。首先构建了多智能体强化学习模型,选取经典的调度规则作为动作,并设置了能反应实际生产状况的状态,为实现完工时间、总拖期和总能耗的同时优化,设置了对应的奖赏函数。通过均值漂移聚类算法对历史状态数据进行聚类,设置了经验共享策略实现多智能体之间的经验交互,为避免算法过早收敛,提出了自适应贪婪策略选取动作。根据实验结果可以得出结论:改进Q学习算法在求解可重入混合流水车间绿色动态调度问题时,可以实现同时优化完工时间、总拖期以及总能耗的目标,针对实际环境中的动态扰动事件,可以快速、实时的做出响应。而且相比于经典调度规则,可以获取更好的调度方案,对实际生产车间有着更好的指导作用。

参考文献

- [1] MACHADO C G, WINROTH M P, RIBEIRO da SILVA E H D. Sustainable manufacturing in Industry 4.0: An emerging research agenda[J]. *International Journal of Production Research*, 2020, 58(5): 1462-1484.
- [2] 刘培基, 刘飞, 王旭, 等. 绿色制造的理论与技术体系及其新框架[J]. *机械工程学报*, 2021, 57(19): 165-179.
LIU Peiji, LIU Fei, WANG Xu, et al. The theory and technology system of green manufacturing and their new frameworks[J]. *Journal of Mechanical Engineering*, 2021, 57(19): 165-179.
- [3] 段建国, 李豪晨, 张青雷. 面向绿色制造的半组合式船用曲轴结构件生产车间多目标调度优化[J]. *计算机集成制造系统*, 2021, 27(6): 1714-1727.
DUAN Jianguo, LI Haochen, ZHANG Qinglei. Green manufacturing-oriented multi-objective scheduling optimization for half built-up marine crank shaft component workshop[J]. *Computer Integrated Manufacturing Systems*, 2021, 27(6): 1714-1727.
- [4] 耿凯峰, 叶春明, 吴绍兴, 等. 分时电价下多目标绿色可重入混合流水车间调度[J]. *中国机械工程*, 2020, 31(12): 1469-1480.
GENG Kaifeng, YE Chunming, WU Shaoxing, et al. Multi-objective green re-entrant hybrid flow shop scheduling under time-of-use electricity tariffs[J]. *China Mechanical Engineering*, 2020, 31(12): 1469-1480.
- [5] MANSOURI S A, AKTAS E, BESIKCI U. Green scheduling of a two-machine flowshop: Trade-off between makespan and energy consumption[J]. *European Journal of Operational Research*, 2016, 248(3): 772-788.
- [6] KUMAR K P. Re-entrant lines[J]. *Queueing System*, 1993, 13(1-2): 87-110.
- [7] 吴秀丽, 肖晓, 赵宁. 考虑装卸的柔性作业车间双资源调度问题[J]. *控制与决策*, 2020, 35(10): 2475-2485.
WU Xiuli, XIAO Xiao, ZHAO Ning. Flexible job shop dual resource scheduling problem considering loading and unloading[J]. *Control and Decision*, 2020, 35(10): 2475-2485.
- [8] 吴秀丽, 崔琪. 考虑可再生能源的多目标柔性流水车间调度问题[J]. *计算机集成制造系统*, 2018, 24(11): 2792-2807.
WU Xiuli, CUI Qi. Multi-objective flexible flow shop scheduling problem with renewable energy[J]. *Computer Integrated Manufacturing Systems*, 2018, 24(11): 2792-2807.
- [9] HUANG J D, LIU J J, CHEN Q X, et al. Minimizing makespan in a two-stage flow shop with parallel batch-processing machines and reentrant jobs[J]. *Engineering Optimization*, 2019, 49(6): 1010-1023.
- [10] ZHANG X Y, CHEN L. A re-entrant hybrid flow shop scheduling problem with machine eligibility constraints[J]. *International Journal of Production Research*, 2018, 56(16): 5293-5305.
- [11] 姚远远, 叶春明, 杨枫. 双目标可重入混合流水车间调度问题的离散灰狼优化算法[J]. *运筹与管理*, 2019, 28(8): 190-199.
YAO Yuanyuan, YE Chunming, YANG Feng. Solving bi-objective reentrant hybrid flow shop scheduling problems by a hybrid discrete grey wolf optimizer[J]. *Operations Research and Management Research*, 2019, 28(8): 190-199.
- [12] 顾涛, 李苏建, 林莹璐, 等. 周期式退火炉作批处理机的可重入批离散机流水车间调度[J]. *机械工程学报*, 2020, 56(2): 220-232.
GU Tao, LI Sujian, LIN Yinglu, et al. Research on the re-entrant batch discrete flow shop scheduling for periodic annealing furnace as batch processor[J]. *Journal of Mechanical Engineering*, 2020, 56(2): 220-232.
- [13] YING K H, LIN S W, WAN S Y. Bi-objective reentrant hybrid flow shop scheduling: an iterated Pareto greedy algorithm[J]. *International Journal of Production Research*, 2014, 52(19): 5735-5747.

- [14] 赵梓焱, 李思怡, 刘士新, 等. 钢铁生产过程动态调度综述[J]. 冶金自动化, 2022, 46(2): 65-79.
ZHAO Ziyang, LI Siyi, LIU Shixin, et al. Review on dynamic scheduling of steel production process[J]. Metallurgical Automation, 2022, 46(2): 65-79.
- [15] XIONG J, XING L, CHEN Y. Robust scheduling for multi-objective flexible job-shop problems with random machine breakdowns[J]. International Journal of Production Economics, 2013, 141(1): 112-126.
- [16] AL-HINAI N, ELMEKKAWY T. Robust and stable flexible job shop scheduling with random machine breakdowns using a hybrid genetic algorithm[J]. International Journal of Production Economics, 2011, 132(2): 279-291.
- [17] MEHTA S V, UZSOY R M. Predictable scheduling of a job shop subject to break-downs[J]. IEEE Robotics & Automation Magazine, 1998, 14(3): 365-378.
- [18] GAO K Z, SUGANTHAN P N, CHUA T J, et al. A two-stage artificial bee colony algorithm scheduling flexible job-shop scheduling problem with new job insertion[J]. Expert Systems with Applications, 2015, 42(21): 7652-7663.
- [19] NIE L, GAO L, LI P. Reactive scheduling in a job shop where jobs arrive over time[J]. Computers & Industrial Engineering, 2013, 66(2): 389-405.
- [20] 王维祺, 叶春明, 谭晓军. 基于 Q 学习算法的作业车间动态调度[J]. 计算机系统应用, 2020, 29(11): 218-226.
WANG Weiqi, YE Chunming, TAN Xiaojun. Job shop dynamic scheduling based on Q-Learning algorithm[J]. Application of Computer System, 2020, 29(11): 218-226.
- [21] WANG L, PAN Z X, WANG J J. A review of reinforcement learning based intelligent optimization for manufacturing scheduling[J]. Complex System Modeling and Simulation, 2021, 1(4): 257-270.
- [22] ARVIV K, STERN H, EDAN Y. Collaborative reinforcement learning for a two-robot job transfer flowshop scheduling problem[J]. International journal of production research, 2016, 54(4): 1196-1209.
- [23] 曹红倩. 应用改进 Q-learning 算法解决柔性作业车间调度问题[J]. 国外电子测量技术, 2022, 41(4): 164-169.
CAO Hongqian. Application of improved Q-learning algorithm to solve flexible job shop scheduling problem[J]. Foreign Electronic Measurement Technology, 2022, 41(4): 164-169.
- [24] 韩忻辰, 俞胜平, 袁志明, 等. 基于 Q-learning 的高速铁路列车动态调度方法[J]. 控制理论与应用, 2021, 38(10): 1511-1521.
HAN Xincheng, YU Shengping, YUAN Zhiming, et al. High-speed railway dynamic scheduling based on Q-learning method[J]. Control Theory and Applications, 2021, 38(10): 1511-1521.
- [25] 周芳芳, 樊晓平, 叶榛. 均值漂移算法的研究与应用[J]. 控制与决策, 2007, 22(8): 841-847.
ZHOU Fangfang, FAN Xiaoping, YE Zhen. Mean shift research and applications[J]. Control and Decision, 2007, 22(8): 841-847.
- [26] 杨能俊, 郭宇, 方伟光, 等. 实时数据驱动的离散制造车间自适应调度方法[J]. 组合机床与自动化加工技术, 2020, 9(9): 175-184.
YANG Nengjun, GUO Yu, FANG Weiguang, et al. Real-time data driven adaptive scheduling method of discrete manufacturing workshops[J]. Modular Machine Tool & Automatic Manufacturing Technique, 2020, 9(9): 175-184.

作者简介: 吴秀丽(通信作者), 女, 1977 年出生, 博士, 教授, 硕士研究生导师。主要研究方向为生产计划与调度、机器学习和智能算法。

E-mail: wuxiuli@ustb.edu.cn

闫晓燕, 女, 1997 年出生。主要研究方向为生产计划与调度、机器学习和智能算法。

E-mail: 18148252183@163.com