ISSN 1004-9037,CODEN SCYCE4
Journal of Data Acquisition and Processing Vol. 39,No. 6,Nov. 2024,pp. 1517－1531
DOI:10. 16337/j. 1004-9037. 2024. 06. 019

http://sjcj. nuaa. edu. cn
E-mail:sjcj@nuaa. edu. cn
Tel/Fax:＋86-025-84892742

基于深度强化学习的不确定作业车间调度方法

吴新泉[1]，燕雪峰[1,2]，魏明强[1,2]，关东海[1,2]

(1. 南京航空航天大学计算机科学与技术学院，南京 211106;2. 软件新技术与产业化协同创新中心，南京 210093)

摘　要：作业车间调度是具有非确定性多项式(Non-deterministic polynomial,NP)难的经典组合优化问题。在作业车间调度中,通常假设调度环境信息已知且在调度过程中保持不变,然而实际调度过程往往受到诸多不确定因素影响(如机器故障、工序变化)。本文提出基于混合优先经验重放的近端策略优化(Proximal policy optimization with hybrid prioritized experience replay,HPER-PPO)调度算法,用于求解不确定条件下的作业车间调度问题。将作业车间调度问题建模为马尔科夫决策过程,设计作业车间的状态特征、回报函数、动作空间和调度策略网络。为了提高深度强化学习模型的收敛性,提出一种新的混合优先经验重放模型训练方法。在标准数据集和基于标准数据集生成的数据集上评估了提出的调度方法,结果表明:在静态调度试验中,本文提出的调度模型比现有的深度强化学习方法和优先调度规则取得了更精确的结果。在动态调度试验中,针对作业车间的工序不确定性,本文所提出的调度模型可以在合理的时间内获得更精确的调度结果。

关键词：作业车间调度;深度强化学习;近端策略优化;优先经验重放

中图分类号：TP311　　**文献标志码**：A

Deep Reinforcement Learning Model for Job Shop Scheduling Problems with Uncertainty

WU Xinquan[1], YAN Xuefeng[1,2], WEI Mingqiang[1,2], GUAN Donghai[1,2]

(1. College of Computer Science and Technology, Nanjing University of Aeronautics & Astronautics, Nanjing 211106, China; 2. Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210093, China)

Abstract: Job shop scheduling problem (JSSP) is a non-deterministic polynomial (NP)-hard classical combinatorial optimization problem. In JSSP, it is usually assumed that the scheduling environment information is known and remains unchanged during the scheduling process. However, the actual scheduling process is often affected by many uncertain factors (such as machine failures and process changes). A proximal policy optimization with hybrid prioritized experience replay (HPER-PPO) scheduling algorithm is proposed for solving JSSPs with uncertainties. The JSSP is modeled as a Markov decision process where the state features, reward function, action space, and scheduling policy networks are designed. In order to improve the convergence of the proposed deep reinforcement learning model, a new hybrid prioritized experiential replay training method is proposed. The proposed scheduling method is evaluated on standard datasets and datasets generated based on standard datasets. The results show that in static scheduling experiments, the proposed scheduling model achieves more accurate results than existing

基金项目:“十四五”装备预研项目(JZX7Y20210401001801)。
收稿日期:2023-06-06;**修订日期**:2023-09-28

deep reinforcement learning methods and priority dispatching rules. In dynamic scheduling experiments, the proposed scheduling model can achieve more accurate scheduling results in a reasonable time for JSSP with process order uncertainty.

Key words: job shop scheduling problem; deep reinforcement learning; proximal policy optimization; prioritized experience replay

引　　言

作业车间调度问题(Job shop scheduling problems, JSSP)是一种经典的组合优化问题(Combinatorial optimization problem, COP),旨在确定若干作业在若干机器上的分配顺序,以在特定的约束条件下最小化或最大化预定义的目标函数,它已被证明是非确定性多项式(Non-deterministic polynomial, NP)难问题[1]。几十年来,JSSP广泛应用于半导体制造业[2]、机械制造业[3]、汽车制造业[4]和供应链[5]等领域。然而,在实际的制造业中,制造过程受到不确定因素(如机器故障、紧急订单)的严重影响。近年来,随着人工智能、信息物理系统和物联网等新制造技术的出现,现实世界中越来越多不确定因素被考虑进JSSP模型中[6]。不确定条件下的作业车间调度问题正吸引着学术和工业界的研究兴趣。

近几年,随着深度强化学习(Deep reinforcement learning,DRL)在游戏、自然语言处理(Natural language processing,NLP)、自动驾驶和计算机视觉(Computer vision,CV)领域的成功应用[7],DRL也用于求解组合优化问题[8],例如旅行商问题(Travelling salesman problem, TSP)[9]和车辆路线规划问题(Vehicle routing problem, VRP)[10]。DRL因集成了深度神经网络的表示能力和强化学习的决策能力,在求解大规模和具有连续状态的调度任务中表现出明显的优越性。尽管如此,现有的基于DRL的调度方法着眼于设计复杂的深度神经网络来构建通用的调度策略,经过离线训练后获得预训练的调度策略,从而可以使用该预训练策略实时地处理各种不确定条件下的调度问题。然而,这种预训练策略在训练时难以收敛并且求解的精度太差,甚至不如简单的优先级规则。

本文致力于在实现不确定条件下的实时调度的基础上提高调度模型的求解精度和收敛性。目前,导致基于深度强化学习的调度方法求解精度低和难以收敛的原因为:(1)作业车间调度环境的状态特征较长。一方面,状态特征越多,在计算状态特征时消耗的时间就越多;另一方面,状态特征作为调度策略网络的输入,其维度越高,策略网络的计算越耗时。(2)复杂的深度神经网络虽然表示能力强,但是网络层数越多计算越复杂,计算时间消耗越大,并且由于深度神经网络的不确定性,训练模型时不能保证强化学习模型的收敛性。(3)这些方法在处理作业车间调度环境中的不确定因素时,通常是直接将预训练模型用于调度决策,并且调度策略在执行过程中保持不变,忽略了调度模型随着调度环境变化自适应调整更新调度策略的能力,导致预训练模型的求解精度低。针对上述问题,本文提出了一种基于混合优先经验重放的近端策略优化的调度模型(Proximal policy optimization with hybrid prioritized experience replay,HPER-PPO)来求解不确定条件下的作业车间调度问题。

1　作业车间调度相关工作

在过去的几十年里,解决作业车间调度问题的方法主要包括数学规划、启发式、元启发式、超启发式和强化学习调度方法。用于精确求解的数学规划方法,如整数线性规划[11]和分枝定界算法[12],可以找到小规模问题的最优解,但是由于维度灾难带来的计算成本昂贵且无法在执行过程中进行实时修改,对于解决大规模或动态调度问题是不可行的。

由于这些精确优化方法无法有效地求解动态作业车间调度问题,旨在寻找次优解的近似优化方法

得到了广泛应用。简单优先级调度规则(Priority dispatching rule, PDR),如最短加工时长(Shortest processing time, SPT)和最多剩余工序(Most operations remaining, MOR)优先规则,对动态事件反应迅速,可以在实践中实时地生成有效的解决方案。然而,PDR方法的性能因实例而异,有效PDR的设计需要大量的专业知识。传统的元启发式方法,如禁忌搜索[13]、模拟退火[14]和遗传算法[15],通过引入随机搜索因子来增强探索能力,并且可以在合理的计算时间内提供高质量的解决方案,但是这些方法的超参数难以调试。作为一种新兴的元启发式方法,粒子群优化[16]、蚁群优化[17]和人工蜂群算法[18]等群体智能算法也被广泛用于解决各种调度问题。然而,很难将这些元启发式方法应用于动态调度问题(Dynamic job-shop scheduling problems,DJSP),因为这些方法在调度时需要给定初始条件,当问题的条件发生变化时,就需要重新启动这些方法[19],这在实际的调度环境中是不适用的,因为需要大量的计算时间。

超启发式方法,如基于遗传规划的超启发式(Genetic programming-based hyper-heuristic,GPHH),其目的是在启发式空间而不是解空间中自动设计调度规则,已成为解决DJSP最有效的方法之一[20]。现有的用于求解DJSP的GPHH算法存在的问题是在DJSP的终结符集中存在大量特征。尽管更多的特征可以提供更多的信息来发现更好的调度规则,但它以指数级的方式扩展了搜索空间,这使得很难识别有前景的搜索区域[21]。尽管在GPHH中使用了许多特征选择方法[22-24]来减少生成的调度规则的长度,但现有的特征选择方法通常采用离线选择机制来获得更好的终结符子集,这需要额外的仿真试验来做特征选择,使得GPHH算法太耗时且不实用。此外,GPHH算法的可解释性仍然是一个挑战。

强化学习作为机器学习最重要的分支之一,吸引了众多领域研究人员的兴趣,成为调度文献中最有前景的5种方法之一[25]。在早期的强化学习调度方法中[26-27],调度智能体需要学习状态-动作对作为调度策略,该策略通常用数组或表格表示。然而,实际的调度环境通常包含连续的状态特征,调度的执行过程包含多个步骤,每个步骤中又有多种选择,因此,它们通常具有大的或连续的状态空间和动作空间。存储这个大表需要巨大的存储空间,并且调度智能体需要花费大量时间遍历所有状态才能做出良好的决策。因此,这种表格型的强化学习调度方法将会面临“维度灾难”问题。

DRL融合了深度神经网络强大的状态表示能力和强化学习的决策能力,常用于解决具有连续状态空间和巨大动作空间的作业车间调度问题。基于DRL的调度方法在处理JSSP中的不确定性时,通常利用已知的静态案例或随机生成的案例训练模型得到一个通用的预训练调度策略,当调度系统中的不确定因素发生变化时,利用这个预训练的调度策略执行调度。调度策略通常用深度神经网络表示,如文献[28-31]中的卷积神经网络(Convolutional neural network,CNN),文献[32-35]中的多层感知机(Multi-layer perceptron,MLP)和文献[36-38]中的其他类型的人工神经网络(Artificial neural network,ANN)。这些调度策略网络的最后一层通常是全连接层或分类器(SoftMax),这就要求它们的输入特征的长度固定,这样才能将训练好的调度策略用于未知案例的调度。但是由于不同案例的规模不同,它们的状态特征长度通常也不同,并且动作空间也可能不同。因此,这类调度策略网络只能用于相同规模的未知案例的调度。

为了将预训练的调度策略应用于不同规模的案例,图神经网络(Graph neural network,GNN)和图嵌入(Graph embedding,GE)技术被集成到了强化学习模型中,用于生成适用于规模可变的案例的调度策略[39-44]。在这些DRL方法中,析取图(Disjunctive graph, DG)用来表示作业车间的状态,GNN用于对析取图中的节点进行编码,GE用于为全连接层生成固定维的嵌入向量。后面的步骤与一般的神经网络类似,利用全连接网络来计算嵌入向量的动作得分,并使用SoftMax函数来生成动作的概率分布。尽管这些方法中的调度策略能够应用到不同规模的案例中,但这些调度方法的性能仍远未达到最优,甚至有时不如简单的PDR[34]。

同样地，为了解决调度任务规模的不确定性，Monaci等[45]利用长短期记忆(Long-short term memory，LSTM)网络，将任意长度的输入映射到固定长度的嵌入向量中。然而，将所有信息压缩为固定长度向量的编码-解码机制使其难以处理非常长的输入。为了缓解这种情况，研究人员在深度神经网络中使用了注意力机制，将最重要的特征而不是所有特征编码为单个固定长度的向量。在文献[46-48]中，图注意力网络(Graph attention，GAT)用于从析取图中提取固定长度的向量，其他部分与一般的DRL调度方法类似。Transformer[49]是NLP和CV领域中另一个流行的基于注意力机制模型，也用于解决作业车间调度问题。Zhao等[50]提出简化的Transformer来提取向量形式的状态特征；Chen等[51]提出DG-ERD Transformer模型，首先用图嵌入方法直接从析取图中提取状态特征，然后利用注意力机制来帮助智能体更好地学习长期的依赖关系。尽管Zhao等取得了良好的结果，但他们只测试了小规模的调度案例，而Chen等在大规模的标准数据集上评估了他们的方法，但结果远未达到最优解，甚至无法与PDR方法相匹敌。

目前，基于深度强化学习的调度方法旨在设计通用的调度策略，通过在已知的静态实例或者随机生成的实例上训练获得调度知识，即预训练策略，用于实时地求解各种调度问题并且预训练策略在求解过程中保持不变。这种通用的调度策略通常采用复杂的深度神经网络来实现，而深度神经网络的复杂性和不确定性使得该策略难以训练和收敛。此外，由于作业车间调度问题是NP难的，通用的预训练策略在未知的实例上难以获得高质量的调度解。尽管在不确定条件下的调度问题经常会发生变化，如果调度策略能够利用好这些变化信息，自适应地更新调度策略，可以提高不确定条件下调度问题的求解精度。

2 基于混合优先经验重放的深度强化学习调度方法

本节提出了一种基于HPER-PPO的DRL调度模型，其框架如图1所示，主要由4部分组成：(1)基于状态变量的作业车间状态特征；(2)基于PDR的动作空间；(3)基于机器空闲时间的回报函数；(4)基于PPO的混合优先经验重放的模型训练方法。

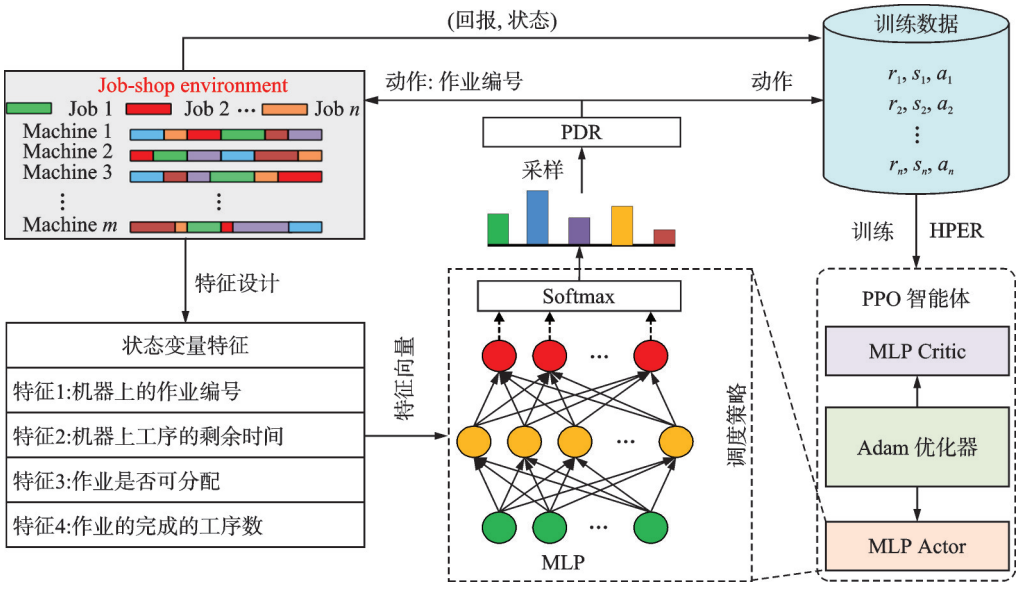

图1 基于近端策略优化的DRL调度框架

Fig.1 Framework of the proposed DRL scheduling method based on PPO

2.1 基于状态变量的状态特征表示

过去基于DRL的调度方法中,状态特征的表示方法主要包括基于析取图、特征矩阵和人工设计的状态变量的方法。然而,无论是析取图中的节点特征还是矩阵特征都是人工设计的,并且特征数量较多,存在特征冗余问题。

为了减少特征冗余,在构建调度环境仿真程序的过程中,利用4个状态变量的变化反映调度环境的变化,这4个状态变量的定义如表1所示。其中,前两个状态变量的长度为作业的数量,后两个的长度为机器的数量。为了归一化这些状态变量的值,将变量current_op_of_job、job_on_machine和left_time_on_machine的值分别除以机器数、最大作业数和最大加工时长作为状态特征。最后,将这些状态特征连接在一起形成1个一维的特征向量,其长度等于作业和机器总数的2倍。

表1 调度环境的状态变量说明

Table 1 Description of state variables in scheduling environment

状态变量	含义	类型	取值
assignable_job	表示作业是否可分配	布尔型	1表示作业可分配,0表示作业不可分配
current_op_of_job	表示作业已完成的工序数	整型	最小值为0,最大值为作业的工序总数
job_on_machine	表示在机器上加工的作业编号	整型	取值为作业的编号,−1表示机器空闲
left_time_on_machine	表示工序在机器上的剩余时间	浮点型	最小值为0,最大值为最大加工时长

在调度领域,作业车间调度问题的标准数据集有固定的格式,如图2所示,第1行表示案例的作业数和机器数,以后每行表示1个作业,从左到右是作业的加工顺序,奇数列表示作业工序所需的机器,偶数列表示该工序在所需机器上的加工时长。随着调度过程向前推进,这4个状态变量的值随之变化。如图2所示,调度开始时所有作业从工序0开始加工并且都可选,所有机器空闲;当选择作业3进行加工时,作业1和作业5在下次选择中则不可分配,因为它们占用的机器1被作业3占用,同时作业3被安排在机器1上以及机器1上的剩余加工时间的信息分别被记录到job_on_machine和left_time_on_machine中。

图2 调度案例(ft06)及其特征变化示意图

Fig.2 An example of scheduling instance (ft06) and its changes of state features

2.2 基于优先调度规则的动作空间

本文选择了6个PDR来构造动作空间:其中4个直接从文献[39]中选取,包括SPT、最长剩余加工时间(Most work remaining,MWKR)、已完成工序与最剩余工序的最小比率(Flow due date to MWKR,FDD/MWKR)和MOR优先规则;当前工序的最长剩余加工时间(Longest remaining machine time,

LRM)从文献[28]中选择,因为该PDR的表现较好;最后一个是先进先出(First in first out,FIFO),它广泛应用于各种调度问题,它们的数学定义如下

$$\begin{cases} \text{SPT: } \min[Z_{i,j}=p_{i,j}] \\ \text{MWKR: } \max[Z_{i,j}=\sum_{j}^{n_i}p_{i,j}] \\ \text{FDD/MWKR: } \min[Z_{i,j}=\sum_{1}^{j}p_{i,j}/\sum_{j}^{n_i}p_{i,j}\sum_{j}^{n_i}p_{i,j}][Z_{i,j}=\sum_{j}^{n_i}p_{i,j}] \\ \text{MOR: } \max[Z_{i,j}=n_i-j+1] \\ \text{LRM: } \max[Z_{i,j}=\sum_{j-1}^{n_i}p_{i,j}] \\ \text{FIFO: } \max[Z_{i,j}=t-Re_i] \end{cases} \tag{1}$$

式中:$Z_{i,j}$为工序$O_{i,j}$的优先级指数;$p_{i,j}$为工序$O_{i,j}$的加工时长;n_i为作业J_i的总工序数;j为作业J_i已完成的工序数;t为当前时间;Re_i为作业J_i的释放时间。

2.3 基于机器空闲时间的回报函数

受调度面积[34]的启发,本文提出了一种基于机器空闲时间的回报函数。在调度过程中,每一步的回报为调度分配前后所有机器的空闲时间之和的负值,即有

$$\text{reward}(s,a)=-\sum_{m\in M}\text{vacancy}_m(s,s') \tag{2}$$

式中:s和s'分别为调度前和调度后的状态;vacancy_m表示这期间在机器m上的空闲时间;a为调度动作;M为机器集合。

如图3所示,总的调度面积由所有作业的总加工时长(阴影区域)和所有机器上的空闲时间(白色区域)组成。总的回报(R)和最大完成时间(makespan)之间的关系为

$$R=-b=c-(c+b)=c-S=c-M\times\text{makespan} \tag{3}$$

式中:b为所有机器的空闲时间之和;c为所有工序的加工时长之和,其为常量;S为加工过程在所有机器上消耗的时间;$|M|$为机器的数量;M_1~M_4为机器编号。

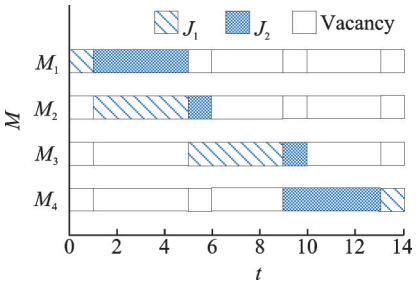

图3 调度面积示意图

Fig.3 An example of scheduling area

2.4 调度策略网络与基于混合优先经验重放的模型训练方法

调度策略是将调度环境的状态特征映射到动作空间的函数,通常由深度神经网络表示,其参数通过强化学习更新。在本文的方法中,首先将状态特征向量输入到MLP网络以获得每个动作的得分,然后通过Softmax函数输出得分的分布为

$$p(a_t|s_t)=\text{Softmax}(\text{MLP}_{\pi_\theta}(s_t)) \tag{4}$$

式中:$p(a_t|s_t)$表示在t时刻,在状态s_t下选择动作a_t的概率;θ为调度策略π的参数。

在基于随机梯度的DRL方法中,使用经验重放使样本满足独立同分布的假设。传统的经验重放方法在训练过程中对经验数据进行平均采样,而不考虑其重要性。优先经验回放(Prioritized experience replay, PER)是根据经验数据的重要性来重放经验样本并且将有用的经验用于多次更新,这些经验样本通常使用时间差分误差(TD-error)来衡量优先级。文献[52]中引入了比例优先随机抽样方法,即

$$P(i)=\frac{p_i^\alpha}{\sum_k p_k^\alpha} \tag{5}$$

式中:$P(i)>0$表示样本的优先级;指数α决定了优先级程度,当$\alpha=0$时表示均匀分布;$p_i=|\delta|+\varepsilon$,$|\delta|$为时间差分误差,ε为一个小正数,其目的是为了使时间差分误差为零的经验样本也能够被采样。在本文中δ指的是优势函数,即目标值和估计值之间的差异,通过累积回报与价值函数值之差实现。该方法抽样时的概率介于纯贪婪优先和均匀随机抽样之间,并使用了重要性采样权重来校正样本分布,如式(6)所示。因为随着训练次数的增加,样本估计的分布会逐渐偏离样本的真实分布。

$$w_i=\left(\frac{1}{N}\frac{1}{P_i}\right)^\beta \tag{6}$$

式中:w_i表示样本的重要性权重;N为样本缓冲区的长度;β为从其初始值变为1的参数;P_i表示样本的优先级。

本文提出了一种融合均匀和优先经验重放的混合经验重放方法。在每次样本学习时,首先均匀地重放所有经验样本,并根据式(5)计算经验样本的优先级。然后,执行多次优先经验重放并更新经验样本的优先级。在优先经验重放时在Critic价值网络的均方误差函数MSE中使用了重要性权重来校正价值网络的损失函数,以减少价值网络的实际训练目标产生的偏移,即

$$L_1(\theta)=\mathrm{MSE}(r_d,\ v)w \tag{7}$$

式中:θ表示价值网络的参数;r_d为折扣回报;v为价值网络的输出值;w为样本的重要性采样权重。

但是,在Actor决策网络中没有使用重要性采样权重,是因为决策网络使用的是裁剪的PPO损失函数,即使数据的分布有所偏移也可以通过裁剪策略消除这种影响,表达式为

$$L_2(\theta)=E_t[\min(r_t(\theta)A_t,\ \mathrm{clip}(r_t(\theta),\ 1-\varepsilon,\ 1+\varepsilon)A_t)] \tag{8}$$

式中:$r_t(\theta)=\dfrac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\mathrm{old}}}(a_t|s_t)}$,$\pi_\theta$和$\pi_{\theta_{\mathrm{old}}}$分别为新旧调度策略;$A_t$为在时间步$t$时刻优势函数的估计;$\varepsilon$为超参数,其控制裁剪的范围。

算法1描述了详细的训练过程,包括两个方面:数据收集(第4~16行)和经验重放(第17~26行)。在收集训练数据时,生成T个独立的完整调度轨迹,并将其存储在内存缓冲区M中。在经验重放阶段,首先将M中数据随机分成多个批次,每个批次的大小为b,利用每个批次的样本优化并更新决策网络和价值网络的参数,同时根据优势函数计算经验样本的优先级;然后,根据经验样本的优先级重采样b组数据样本并计算其重要性权重;最后,利用这些样本再次优化和更新决策网络和价值网络的参数。重复以上步骤直到达到最大训练次数或者结果收敛亦或者达到最大训练时间限制。本文定义如果30个连续的调度结果都相同,则模型收敛,模型的最大训练时长为1 h。此外,由于优先经验重放的样本和均匀重放的样本都是由相同的策略生成的,所以满足策略梯度定理所要求的同轨条件。

算法1 基于混合优先经验重放的近端策略优化模型的训练方法

(1)初始化最大训练次数N,折扣因子γ,采样轨迹的数量T,初始化单条轨迹缓冲区B,总数据缓冲区M

(2)初始化批次大小b,单次学习次数K和重采样次数C,初始化决策网络和价值网络及其优化器

(3)for $e=1$ to N do

(4) for $t=1$ to T do

(5) 重置调度环境并观测初始状态s_0

(6) while True do

(7) 根据当前调度策略选出调度动作a_t并返回该动作的概率$p(a_t)$

(8) 在调度环境中执行动作a_t并返回执行动作后的状态s_{t+1}和回报r_t
(9) 保存数据(s_t, a_t, r_t, $p(a_t)$) 到缓冲区B中
(10) if done then
(11) 利用γ计算累积折扣回报并用该回报替换r_t
(12) 将缓冲区B中的数据添加到总的缓冲区M中,并清空缓冲区B
(13) break
(14) end if
(15) end while
(16) end for
(17) for k=1 to K do
(18) 将缓冲区M中的数据随机分成多个批次,每个批次的大小为b
(19) for 每批数据 do
(20) 分别优化和更新决策网络和价值网络的参数,并根据式(5)计算每个样本的优先级
(21) end for
(22) for c=1 to C do
(23) 根据样本的优先级重采样b组数据样本并根据式(6)计算重要性权重
(24) 利用重采样的样本优化和更新决策网络和价值网络的参数并更新样本的优先级
(25) end for
(26) end for
(27) end for

3 实验分析

3.1 实验数据

在公开的数据集(如表2所示)和基于公开数据集生成的具有工序不确定性的案例上分别进行了静态和动态调度实验。基于案例:ft10(10×10)、la26(20×10)、la31(30×10)和ta41(30×20),随机选择案例中的1个作业并且随机交换该作业中的2个工序,交换工序的数量占总工序数的比例分别设置为20%、40%、60%、80%和100%。

表2 静态实验数据集

Table 2 Datasets for static experiments

数据集	数据源
ft10(10×10)	Fisher[53]
la01(10×5), a06(15×5),la11(20×5),la16(10×10), la21～la25(15×10),la26～la30(20×10), la31～la35(30×10),la36～la40(15×15)	Lawrence[54]
ta21～ta22(20×20), ta31～ta32(30×15), ta41～ta42(30×20), ta51～ta52(50×15),ta61～ta62 (50×20),ta571～ta72(100×20)	Taillard[55]

3.2 实验设置

在提出的模型中,PPO的Actor网络和Critic网络都由具有1个隐藏层的MLP网络实现。每个隐藏层网络的维度等于调度案例的总工序数。在训练过程中,对于每个调度案例,最大训练次数设置为

8 000,每次包含5个独立的轨迹(即一个案例的完整调度数据),设置批次大小b等于案例的总工序数。对于PPO,将更新网络的学习次数K设置为10,裁剪参数epsilon设置为0.2,折扣因子γ设置为0.999,Actor和Critic网络的学习率分别设置为1e−3和3e−3,使用Adam优化器。在优先经验重放时,设置α为0.6,β的初始值为0.4、最大值为1,优先经验重放的次数C为2,收敛训练次数设置为5 000。其中Actor网络的轨迹数量T、批量大小b、优先经验重放的次数C、隐藏层的维度和收敛训练次数超参数都经过仔细选择,其他参数在文献中被广泛使用或遵循PyTorch中的默认设置,本文公开了模型的源代码(https://github.com/sx1616039/simple-order-uncertainty)。实验在配备Windows 10 64操作系统8 GB RAM、Intel Core i7-9750H 2.60 GHz CPU的笔记本电脑上运行。

3.3 静态调度实验

本文在表1的公开数据集上进行了静态调度实验,对比了Chen等[51]提出的基于Transformer的调度模型、Han等[28]提出的基于DQN的调度模型、Park等[41]提出的基于GNN的多智能体强化学习调度模型以及2.2节出现的6种PDR调度方法,结果如表3所示,其中在Min_PDR列中只列出了PDR中最好的调度结果。OPT列表示已知的最佳调度结果,本文所提方法的调度结果给出了5次独立运行结果的平均值和标准差,并且方差分析表明这5次运行结果没有明显差异。案例的大小由作业数量×机器数量给出,对比的几种方法中最佳的调度结果以粗体显示。

由于目标或者算法不同,很难公平地比较不同的DRL调度方法。基于DRL的调度方法主要有两种训练模式获得调度策略。一种训练模式是,调度智能体在随机生成的案例上进行训练并在新的案例上进行测试,旨在获得通用的调度策略;另一种类似于传统的基于特定案例的优化方法,训练和测试在同一个案例上进行。Chen等[51]和Park等[41]遵循前一种训练模式,而Han等[28]使用后一种训练模式,本文使用了后一种训练模式。

从表3可以看出,本文所提方法在36个案例中有34个的调度结果小于(优于)文献[28]的调度结果,并且在所有的案例上优于其他两种DRL方法和所有6种PDR方法。值得注意的是,Chen等[51]的调度方法比PDR方法还要差,这表明DRL调度方法并不总是优于PDR方法。文献[28,41,51]使用了复杂的深度神经网络作为调度策略,分别为Transformer(包含3个注意力网络)、ScheduleNet(包含6个MLP神经网络)、卷积神经网络(包含4个卷积层),而本文提出的调度策略使用了仅有1个隐藏层的MLP神经网络,这说明调度网络不需要设计很深也能够获得较好的调度结果。

此外,本文还展示了所提调度方法的收敛性能,包括训练所需轨迹数以及训练时间。如图4所示,本文的调度模型在小、中、大规模的案例上(la01~ta42)可以在1 h内训练收敛,在超大规模的案例上(ta51~ta72)由于时间的限制而无法收敛。从平均收敛时间上看,小规模案例(10×5、15×5、20×5和10×10)小于500 s,中等规模案例(15×10、15×15、20×10、30×10和20×20)小于1 000 s,大规模案例(30×15和30×20)小于3 000 s。在超大规模案例中,随着案例规模的增加,在1 h内产生的训练轨迹数逐渐减少,从而说明收集轨迹和更新策略所花费的时间随规模明显增加。

3.4 动态调度实验

针对作业车间调度系统中的不确定性,基于DRL的调度方法通过重调度以适应这些不确定性因素的变化,主要有3种重调度模式:第一种是利用预训练的调度策略在新的案例上执行调度;第二种是利用训练好的策略在新的案例上继续训练得到1个重用的调度策略并在该新的案例上执行调度;最后一种是直接在新案例上重新训练1个的新调度策略,然后在该新的案例上执行调度。

本文考虑工序不确定的调度环境,对比了提出的DRL模型的3种重调度策略和PDR重调度策略。通过重复独立运行5次这3种重调度策略,得到调度结果的平均值,而PDR方法则取6种PDR调度策略的最

表 3 静态调度结果对比

Table 3 Comparison for static scheduling results

规模	案例	OPT	Chen等[51]	Han等[28]	Park等[41]	Min_PDR	Ours(μ/σ)
10×5	la01	666	—	666	692	680	**666 / 0.0**
15×5	la06	926	—	926	971	926	**926 / 0.0**
20×5	la11	1 222	—	1 222	1 319	1 254	**1 222 / 0.0**
10×10	la16	945	—	**980**	1 054	1 047	1 041 / 10.8
15×10	la21	1 046	—	1 162	1 261	1 230	**1 138.4 / 8.5**
15×10	la22	927	—	1 021	1 207	1 060	**983.6 / 8.7**
15×10	la23	1 032	—	1 053	1 145	1 152	**1 038.8 / 8.1**
15×10	la24	935	—	1 029	1 088	1 085	**1 025.2 / 10.0**
15×10	la25	977	—	1 067	1 117	1 112	**1 046.2 / 18.9**
20×10	la26	1 218	—	1 327	1 458	1 386	**1 272.8 /12.7**
20×10	la27	1 235	—	1 397	1 516	1 415	**1 340.6 / 27.9**
20×10	la28	1 216	—	1 386	1 357	1 472	**1 291.8 / 35.2**
20×10	la29	1 152	—	1 323	1 320	1 343	**1 267.4 / 12.7**
20×10	la30	1 355	—	1 417	1 490	1 534	**1 393.8 / 10.9**
30×10	la31	1 784	—	1 854	1 906	1 810	**1 784 / 0.0**
30×10	la32	1 850	—	1 900	1 850	1 884	**1 850 / 0.0**
30×10	la33	1 719	—	1 782	1 731	1 794	**1 719 / 0.0**
30×10	la34	1 721	—	1 880	1 784	1 856	**1 723.6 / 5.2**
30×10	la35	1 888	—	1 941	1 969	2 039	**1 888 / 0.0**
15×15	la36	1 268	—	**1 355**	1 449	1 396	1 374.2 / 19.1
15×15	la37	1 397	—	1 540	1 653	1 584	**1 511.6 / 14.6**
15×15	la38	1 196	—	1 348	1 444	1 358	**1 326.6 / 15.4**
15×15	la39	1 233	—	1 357	1 430	1 405	**1 356.2 / 3.9**
15×15	la40	1 222	—	1 336	1 350	1 358	**1 318.6 / 4.3**
20×20	ta21	1 642	2 145.63	1 952	1 921	1 964	**1 846.8 / 6.4**
20×20	ta22	1 600	2 015.89	1 870	1 844	1 868	**1 757.4 / 17.9**
30×15	ta31	1 764	2 382.63	1 986	2 055	2 134	**1 956.8 / 18.3**
30×15	ta32	1 784	2 458.52	2 135	2 268	2 163	**2 012.8 / 17.6**
30×20	ta41	2 005	2 541.22	2 450	2 572	2 499	**2 323.2 / 12.5**
30×20	ta42	1 937	2 762.26	2 351	2 397	2 401	**2 197.4 / 15.6**
50×15	ta51	2 760	3 762.60	3 263	3 382	3 442	**3 069.2 / 72.5**
50×15	ta52	2 756	3 511.20	3 229	3 231	3 263	**2 891 / 31.3**
50×20	ta61	2 868	3 633.48	3 367	3 202	3 335	**3 102.6 / 23.4**
50×20	ta62	2 869	3 712.30	3 489	3 339	3 274	**3 217.2 / 22.6**
100×20	ta71	5 464	6 321.22	5 908	5 879	5 839	**5 687 / 27.6**
100×20	ta72	5 181	6 232.22	5 746	5 456	5 462	**5 320.8 / 26.4**

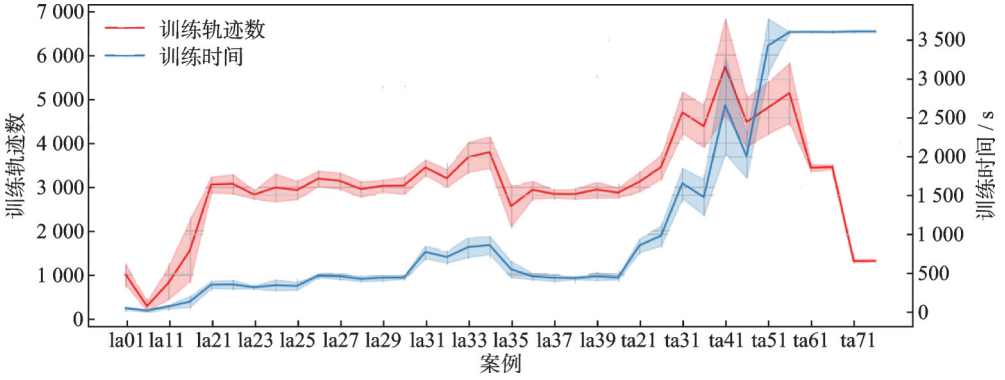

图4 本文提出的DRL调度模型的训练轨迹数和训练时间

Fig.4 Training trajectories and training time of our DRL scheduling model

小值。为了更清楚地对比调度结果差异,使用了相对调度结果,即所有调度结果减去它们中的最小值。

如图5所示,4种重调度策略的调度结果从好到差依次排序为重训练策略(红色实线)、重用策略(黑色实线)、预训练策略(黄色实线)和PDR策略(蓝色实线),而调度结果与随机交换的工序数量没有明显的关系。由于动态实验生成的案例不同,难以直接和静态实验中对比的DRL方法相比较,结合静态实验通过与PDR对比,间接地反映比其他DRL方法的优越性:本文的重调度策略全面超越了Chen等[51]的调度策略,在至少一半的案例上比Park等[41]的调度策略好。由于本文的训练模式和文献[28]的相同,在动态实验上的调度性能可以参考在静态实验上的调度性能差异。重训练策略和重用策略的调度结果之所以更好,是因为这两种模型都"看"到了新的调度问题的信息并利用该信息更新了调度策略。而预训练的调度策略虽然是在旧的调度问题上训练获得的,但是旧案例与新案例存在相似性,因此在旧案例中获得的调度知识依然可用于指导新案例执行调度。

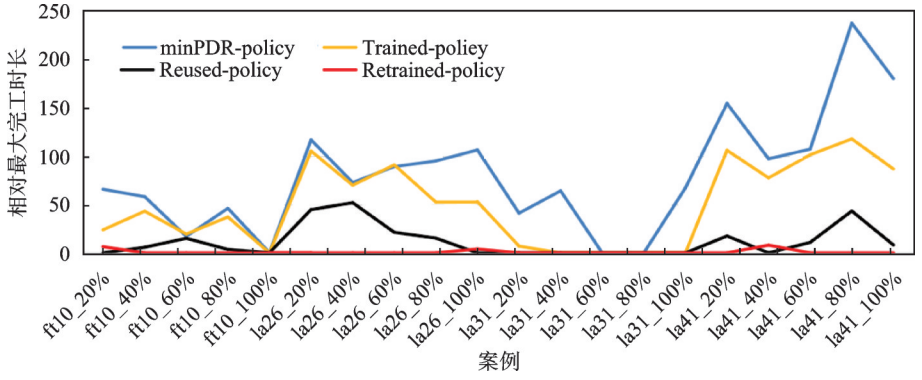

图5 4种重调度策略的调度结果对比

Fig.5 Rescheduling results of four kinds of rescheduling policies

此外,本文还分析了重调度策略的决策时间。Chen等[51]和Park等[28]的调度策略和预训练策略一样,决策时间都少于1 s,PDR方法比这些方法稍快。而重用策略和重训练策略的决策时间和训练时间相同,二者的训练时间对比如图6所示,从图中可以看出重用策略能够明显地减少训练时间,并且随着问题规模的增加重用策略减少的时间越多。然而对于一个不确定的调度环境,不确定因素的变化是持续的,为了达到调度精度和调度时间的折中,重用的调度策略是最佳的选择。一方面,重用的调度策略可以随时给出调度结果(在0时刻退化为预训练的策略);另一方面重用的调度策略可以持续学习调度环境的变化并将学到的新知识用于调度。此外,对于作业车间调度环境中的其他不确定性(如加工时

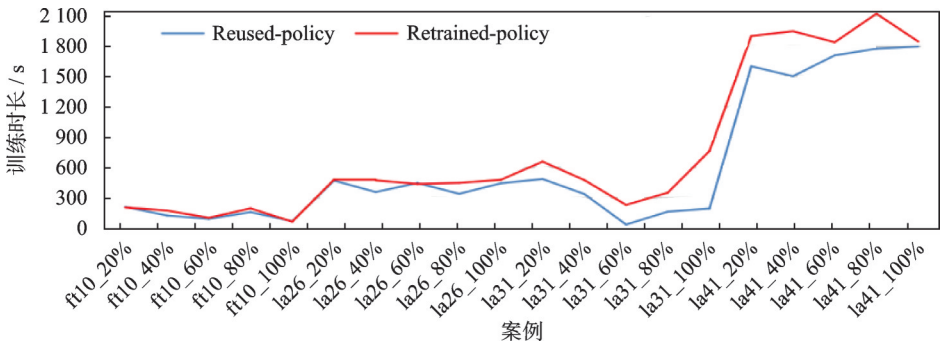

图6 重用策略和重训练策略的训练时间对比

Fig.6 Training time comparison of reused policy and the retrained policy

长不确定),只在案例生成的方法上有所区别,本文所提调度方法同样适用。

最后,对比了使用优先经验重放与不使用优先经验重放的训练时长,如图7所示,在不损失求解精度的情况下,使用优先经验重放能够明显地减少训练时长。

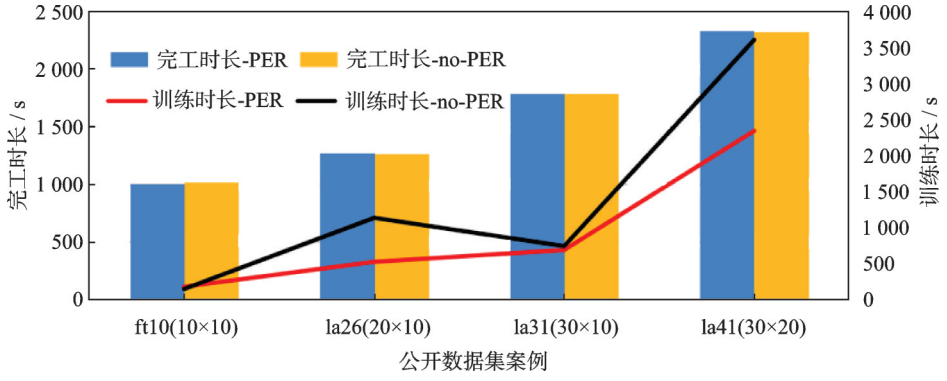

图7 使用优先经验重放前后模型性能比较

Fig.7 Performance of scheduling model before and after using HPER

4 结束语

作业车间调度问题是经典的组合优化问题,广泛存在于各种制造领域中。针对作业车间中的不确定性,本文提出了一种基于DRL的动态调度方法。针对DRL中特征冗余和模型收敛慢的问题,设计了调度环境的状态特征、动作空间和回报函数,同时构建了调度策略网络并通过混合优先经验重放方法加快模型收敛的速度。通过静态和动态调度实验以及与多种作业车间调度方法比较,验证了本文所提出的作业车间调度方法不仅可以获得更精确的调度解,还可以提高训练模型收敛的速度。本文的决策网络为仅包含1个隐藏层的MLP神经网络,从而说明调度网络不需要设计很深也能够获得较好的调度结果。

参考文献:

[1] GAREY M R, JOHNSON D S, SETHI R. The complexity of flowshop and jobshop scheduling[J]. Mathematics of Operations Research, 1976, 1(2): 117-129.

[2] PARK I B, HUH J, KIM J, et al. A reinforcement learning approach to robust scheduling of semiconductor manufacturing facilities[J]. IEEE Transactions on Automation Science and Engineering, 2019(99): 1-12.

[3] LI Y, HE Y, WANG Y, et al. An optimization method for energy-conscious production in flexible machining job shops with dynamic job arrivals and machine breakdowns[J]. Journal of Cleaner Production, 2020, 254(9/10/11/12):120009.

[4] YANG Y Z, GU X S. Pareto-based complete local search and combined timetabling for multi-objective job shop scheduling problem with no-wait constraint[J]. Journal of Donghua University (English Ed.), 2016,33 (4): 601-624.

[5] SHI Q L, KOZAN E. A hybrid metaheuristic algorithm to optimise a real-world robotic cell[J]. Computers & Operations Research, 2017, 84: 188-194.

[6] MOHAMMADI S, AL-E-HASHEM S, REKIK Y. An integrated production scheduling and delivery route planning with multi-purpose machines: A case study from a furniture manufacturing company[J]. International Journal of Production Economics, 2020, 219: 347-359.

[7] WANG X,WANG S, LIANG X,et al. Deep reinforcement learning: A survey[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022. DOI: 10.1109/TNNLS.2022.3207346.

[8] BENGIO Y, LODI A , PROUVOST A. Machine learning for combinatorial optimization: A methodological tourd'horizon[J]. European Journal of Operational Research,2020. DOI: 10.1016/j.ejor.2020.07.063.

[9] OUYANG W, WANG Y, HAN S, et al. Improving generalization of deep reinforcement learning-based TSP solvers[C]// Proceedings of 2021 IEEE Symposium Series on Computational Intelligence (SSCI). [S.l.]: IEEE, 2021.

[10] YU J J Q, YU W, GU J. Online vehicle routing with neural combinatorial optimization and deep reinforcement learning[J]. IEEE Transactions on Intelligent Transportation Systems, 2019,20(10): 3806-3817.

[11] GOMORY R E, Solving linear programming problems in integers[J]. Combinatorial Anal, 1960, 10: 211-215.

[12] CARLIER J, PINSON E . An algorithm for solving the job-shop problem[J]. Management Science, 1989, 35(2): 164-176.

[13] GEIGER C D, KEMPF K G, UZSOY R . A Tabu search approach to scheduling an automated wet etch station[J]. Journal of Manufacturing Systems, 1997, 16(2): 102-116.

[14] YIM S J, LEE D Y. Scheduling cluster tools in wafer fabrication using candidate list and simulated annealing[J]. Journal of Intelligent Manufacturing, 1999, 10(6): 531-540.

[15] NAKANO R, YAMADA T. Conventional genetic algorithm for job shop problems[C]//Proceedings of the 4th International Conference on Genetic Algorithms. San Diego, CA, USA: [s.n.], 1991: 474-479.

[16] LIN T L, HORNG S J, KAO T W, et al. An efficient job-shop scheduling algorithm based on particle swarm optimization[J]. Expert Systems with Applications, 2010, 37(3): 2629-2636.

[17] HUANG K L, LIAO C. Ant colony optimization combined with taboo search for the job shop scheduling problem[J]. Computers and Operations Research,2008, 35(4): 1030-1046.

[18] ASADZADEH L. A parallel artificial bee colony algorithm for the job shop scheduling problem with a dynamic migration strategy[J]. Computers & Industrial Engineering, 2016,102: 359-367.

[19] BAYKASOGLU A, KARASLAN F S. Solving comprehensive dynamic job shop scheduling problem by using a GRASP-based approach[J]. International Journal of Production Research, 2017, 55(11/12): 3308-3325.

[20] BRANKE J, NGUYEN S, PICKARDT C W, et al. Automated design of production scheduling heuristics: A review[J]. IEEE Transactions on Evolutionary Computation, 2016, 20(1): 110-124.

[21] MEI Y, NGUYEN S, XUE B, et al. An efficient feature selection algorithm for evolving job shop scheduling rules with genetic programming[J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2017, 1(5): 339-353.

[22] SHADY S, KAIHARA T, FUJII N, et al. A hyper-heuristic framework using GP for dynamic job shop scheduling problem [C]//Proceedings of the 64th Annual Conference of the Institute of Systems, Control and Information Engineers (ISCIE). Kobe, Japan: [s.n.], 2020.

[23] MEI Y, NGUYEN S, XUE B, et al. An efficient feature selection algorithm for evolving job shop scheduling rules with genetic programming[J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2017, 1(5): 339-353.

[24] ZHANG F, MEI Y, SU N, et al. Evolving scheduling heuristics via genetic programming with feature selection in dynamic flexible job-shop scheduling[J]. IEEE Transactions on Cybernetics, 2020(99): 1-15.

[25] KAYHAN B M, YILDIZ G. Reinforcement learning applications to machine scheduling problems: A comprehensive literature review[J]. Journal of Intelligent Manufacturing, 2023. DOI: 10.1007/s10845- 021-01847-3.

[26] LIU C C, JIN H Y, TIAN Y, et al. Reinforcement learning approach to re-entrant manufacturing system scheduling[C]//Proceedings of 2001 International Conferences on Info-Tech and Info-Net. Beijing, China: [s.n.], 2001: 280-285.

[27] WANG Y C, USHER J M. Application of reinforcement learning for agent-based production scheduling[J]. Engineering Applications of Artificial Intelligence, 2005, 18(1): 73-82.

[28] 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.

[29] 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.

[30] FENG Y, ZHANG L, YANG Z, et al. Flexible job shop scheduling based on deep reinforcement learning[C]//Proceedings of 2021 5th Asian Conference on Artificial Intelligence Technology (ACAIT). Haikou, China: IEEE, 2021: 660-666.

[31] PALOMBARINI J A, MARTÍNEZ E C. End-to-end on-line rescheduling from Gantt chart images using deep reinforcement learning[J]. International Journal of Production Research, 2022, 60(14): 4434-4463.

[32] TURGUT Y, BOZDAG C E . Deep Q-network model for dynamic job shop scheduling problem based on discrete event simulation[C]//Proceedings of 2020 Winter Simulation Conference (WSC). Orlando, FL, USA: IEEE, 2020: 1551-1559.

[33] LUO B, WANG S, YANG B , et al. An improved deep reinforcement learning approach for the dynamic job shop scheduling problem with random job arrivals[J]. Journal of Physics: Conference Series, 2021, 1848(1): 0120298.

[34] TASSEL P, GEBSER M, SCHEKOTIHIN K. A reinforcement learning environment for job-shop scheduling[EB/OL]. (2021-04-08)[2023-03-08]. https://doi.org/10.48550/arXiv.2104.03760.

[35] YI Z, ZHU H H, TANG D B, et al. Dynamic job shop scheduling based on deep reinforcement learning for multi-agent manufacturing systems[J]. Robotics and Computer-Integrated Manufacturing, 2022, 78: 102412.

[36] LIU R K, 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.

[37] LUO S, ZHANG L, FAN Y, Real-time scheduling for dynamic partial-no-wait multiobjective flexible job shop by deep reinforcement learning[J]. IEEE Transactions on Automation Science and Engineering, 2022, 19(4): 3020-3038.

[38] BURGGRÄF P, WAGNER J, SAßMANNSHAUSEN T, et al. Multi-agent-based deep reinforcement learning for dynamic flexible job shop scheduling[J]. Procedia Cirp, 2022, 112: 57-62.

[39] ZHANG C, SONG W, CAO Z, et al. Learning to dispatch for job shop scheduling via deep reinforcement learning[C]//Proceedings of the 34th Conference on Neural Information Processing Systems. Vancouver, Canada: [s.n.], 2020.

[40] PARK J, CHUN J, KIM S H, et al. Learning to schedule job-shop problems: Representation and policy learning using graph neural network and reinforcement learning[J]. International Journal of Production Research, 2021(4):1-18.

[41] PARK J, BAKHTIYAR S, PARK J. ScheduleNet: Learn to solve multi-agent scheduling problems with reinforcement learning[EB/OL]. (2021-06-06)[2023-03-06]. https://arxiv.org/pdf/2106.03051v1.

[42] ELSAYED E K, ELSAYED A K, ELDAHSHAN K A. Deep reinforcement learning-based job shop scheduling of smart manufacturing[J]. Computers, Materials & Continua, 2022(12): 18.

[43] LEI K, PENG G, ZHAO W C, et al. A multi-action deep reinforcement learning framework for flexible job-shop scheduling problem[J]. Expert Systems with Applications, 2022(205): 117796.

[44] ZENG Z, LI X, BAI C. A deep reinforcement learning approach to flexible job shop scheduling[C]//Proceedings of 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC). Prague, Czech Republic: IEEE, 2022: 884-890.

[45] MONACI M, AGASUCCI V, GRANI G. An actor-critic algorithm with deep double recurrent agents to solve the job shop[J]. European Journal of Operational Research, 2024, 312(3): 910-926.

[46] YANG S G. Using attention mechanism to solve job shop scheduling problem[C]//Proceedings of 2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE). Guangzhou, China: IEEE, 2022: 59-62.

[47] ZENG Z Q, LI X X, BAI C B. A deep reinforcement learning approach to flexible job shop scheduling[C]//Proceedings of IEEE International Conference on Systems, Man and Cybernetics. [S.l.]: IEEE, 2022: 884-890.

[48] SONG W, CHEN X, LI Q, et al. Flexible job shop scheduling via graph neural network and deep reinforcement learning[J].

IEEE Transactions on Industrial Informatics, 2023,19(2): 1600-1610.

[49] VASWANI A, SHAZEER N,PARMAR N, et al. Attention is all you need[C]//Proceedings of Advances in Neural Information Processing Systems. Long Beach, California, USA: Curran Associates Inc., 2017: 5999-6009.

[50] ZHAO L, SHEN W, ZHANG C, et al. An end-to-end deep reinforcement learning approach for job shop scheduling[C]//Proceedings of 2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD).[S.l.]: IEEE, 2022: 841-846.

[51] CHEN R, LI W, YANG H. A deep reinforcement learning framework based on an attention mechanism and disjunctive graph embedding for the job shop scheduling problem[J]. IEEE Transactions on Industrial Informatics, 2022. DOI: 10.1109/TII.2022.3167380.

[52] SCHAUL T, JOHN Q, ANTONOGLOU I, et al. Prioritized experience replay[EB/OL]. (2015-09-23)[2023-04-12]. https://arXiv.org/1511.05952.

[53] FISHER H. Probabilistic learning combinations of local job-shop scheduling rules[J]. Industrial Scheduling, 1963,1: 225-251.

[54] LAWRENCE S. Supplement to resource constrained project scheduling: An experimental investigation of heuristic scheduling techniques[D]. Carnegie-Mellon: Carnegie-Mellon University, 1984.

[55] TAILLARD E D. Benchmarks for basic scheduling problems[J]. European Journal of Operational Research 1993. DOI: 10.1016/0377-2217(93)90182-M.

作者简介:

吴新泉(1993-),男,博士研究生,研究方向:系统建模与仿真,E-mail:2690402618@qq.com。

燕雪峰(1975-),**通信作者**,男,教授,研究方向:智能建模、大数据、复杂系统建模与仿真。

魏明强(1985-),男,教授,研究方向:深度学习、计算机视觉和计算机图形学。

关东海(1981-),男,副教授,研究方向:机器学习、大数据和社会计算。

(编辑:刘彦东)