Journal of System Simulation

Volume 34 | Issue 6 Article 7

6-16-2022

Application of Improved Q Learning Algorithm in Job Shop Scheduling Problem

Yejian Zhao
School of Artificial Intelligence, Shenyang University of Technology, Shenyang 110027, China;,
zhao_yejian@163.com

Yanhong Wang
School of Artificial Intelligence, Shenyang University of Technology, Shenyang 110027, China;,
wangyh_sut@163.com

Jun Zhang
School of Artificial Intelligence, Shenyang University of Technology, Shenyang 110027, China;

Hongxia Yu
School of Artificial Intelligence, Shenyang University of Technology, Shenyang 110027, China;

See next page for additional authors

Follow this and additional works at: https://dc-china-simulation.researchcommons.org/journal

 Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Application of Improved Q Learning Algorithm in Job Shop Scheduling Problem

Abstract

Abstract: Aiming at the job shop scheduling in a dynamic environment, a dynamic scheduling algorithm based on an improved Q learning algorithm and dispatching rules is proposed. *The state space of the dynamic scheduling algorithm is described with the concept of "the urgency of remaining tasks" and a reward function with the purpose of "the higher the slack, the higher the penalty" is disigned. In view of the problem that the greedy strategy will select the sub-optimal actions in the later stage of learning, the traditional Q learning algorithm is improved by introducing an action selection strategy based on the "softmax" function, which makes the improved Q learning algorithm more equal in the probability of selecting different actions in the early stage.* The simulation results obtained from 6 different test instances show that the performance indicator of the scheduling algorithm is improved by an average of about 6.5% compared to the before and by about 38.3% and 38.9% respectively compared with the IPSO algorithm and PSO algorithm. The indicator is significantly better than conventional methods such as using a single dispatching rule and traditional optimization algorithms.

Keywords

reinforcement learning, Q learning, dispatching rules, dynamic scheduling, job shop scheduling

Authors

Yejian Zhao, Yanhong Wang, Jun Zhang, Hongxia Yu, and Zhongda Tian

Recommended Citation

Yejian Zhao, Yanhong Wang, Jun Zhang, Hongxia Yu, Zhongda Tian. Application of Improved Q Learning Algorithm in Job Shop Scheduling Problem[J]. Journal of System Simulation, 2022, 34(6): 1247-1258.

This paper is available in Journal of System Simulation: https://dc-china-simulation.researchcommons.org/journal/vol34/iss6/7

第 34 卷第 6 期 系统仿真学报© Vol. 34 No. 6
2022 年 6 月 Journal of System Simulation Jun. 2022

改进Q学习算法在作业车间调度问题中的应用

赵也践，王艳红*，张俊，于洪霞，田中大
(沈阳工业大学 人工智能学院，辽宁 沈阳 110027)

摘要： 为解决动态环境下作业车间调度问题，提出了一种基于改进Q学习算法和调度规则的动态调度算法。以“剩余任务紧迫程度”的概念来描述动态调度算法的状态空间；设计了以“松弛越高，惩罚越高”为宗旨的回报函数；通过引入以Softmax函数为主体的动作选择策略来改进传统的Q学习算法，使改进后的Q学习算法在前期选择不同动作的概率更加平等，同时改善了贪婪策略在学习后期还会选择次优动作的现象。仿真结果表明：该调度算法相较于改进前，性能指标平均提升约6.5%；相较于IPSO算法和PSO算法，性能指标平均提升分别约为38.3%和38.9%，调度结果明显优于使用单一调度规则以及传统优化算法等常规方法。

关键词： 强化学习；Q学习；调度规则；动态调度；作业车间调度
中图分类号：TB497; TP278 文献标志码：A 文章编号：1004-731X(2022)06-1247-12
DOI: 10.16182/j.issn1004731x.joss.21-0099

Application of Improved Q Learning Algorithm in Job Shop Scheduling Problem

Zhao Yejian, *Wang Yanhong**, *Zhang Jun*, *Yu Hongxia*, *Tian Zhongda*
(School of Artificial Intelligence, Shenyang University of Technology, Shenyang 110027, China)

Abstract: Aiming at the job shop scheduling in a dynamic environment, a dynamic scheduling algorithm based on an improved Q learning algorithm and dispatching rules is proposed. *The state space of the dynamic scheduling algorithm is described with the concept of "the urgency of remaining tasks" and a reward function with the purpose of "the higher the slack, the higher the penalty" is disigned. In view of the problem that the greedy strategy will select the sub-optimal actions in the later stage of learning, the traditional Q learning algorithm is improved by introducing an action selection strategy based on the "softmax" function, which makes the improved Q learning algorithm more equal in the probability of selecting different actions in the early stage.* The simulation results obtained from 6 different test instances show that the performance indicator of the scheduling algorithm is improved by an average of about 6.5% compared to the before and by about 38.3% and 38.9% respectively compared with the IPSO algorithm and PSO algorithm. The indicator is significantly better than conventional methods such as using a single dispatching rule and traditional optimization algorithms.

Keywords: reinforcement learning; Q learning; dispatching rules; dynamic scheduling; job shop scheduling

引言

作业车间调度问题(job shop scheduling problem, JSSP)是许多现代生产制造环境，诸如装配车间、芯片及半导体工艺制造，以及机械零部件

收稿日期：2021-02-02 修回日期：2021-03-14
基金项目：国家自然科学基金(61803273)；辽宁省重点研发计划(2020JH2/10100041)
第一作者：赵也践(1995-)，男，硕士，研究方向为车间调度与机器学习。E-mail：zhao_yejian@163.com
通讯作者：王艳红(1967-)，女，博士，教授，研究方向为企业综合自动化与生产调度。E-mail：wangyh_sut@163.com

Journal of System Simulation, Vol. 34 [2022], Iss. 6, Art. 7

加工的综合表现形式及简化模型，具有NP-Hard特性[1]，被广泛研究。解决JSSP的普遍方法是将生产制造车间抽象成一个已知各方面生产加工属性的静态环境，作业的生产加工过程只需按照既定的方案即可完成加工任务。然而现实中的生产制造环境总是具有机器故障、订单急插、加工起始时间发生改变等非预见性突发事件，所以在静态环境下，调度系统的应用受到一定程度的限制。因此，学术界和工业界迫切需要研究出一种实时的、在线的调度方法来处理动态作业车间调度问题(dynamic job shop scheduling problem, DJSSP)，从而使企业的生产车间更加高效且稳定地运行。

元启发式方法和调度规则是被广泛使用的调度方法。元启发式算法在解决DJSSP时，通常将动态问题转化为一系列的静态子问题，并采用诸如遗传算法(GA)[2]、粒子群算法(PSO)[3]等经典的智能优化算法来解决这些问题。这些算法虽然可以保证较高的求解质量，但是需要耗费更多的求解时间，当条件发生改变时，需要重新调参，甚至修改环境模型。这些缺点直接导致了调度过程中的非时效性，难以满足当今DJSSP中实时调度的要求。

使用调度规则(dispatching rules, DRs)是解决DJSSP的一种有效途径。Iskander等[4]对113条调度规则做了总结和综述；Jones等[5]将调度规则分为简单规则、组合规则和有序规则集；Durasevic等[6]将一些常用的调度规则放在9种调度性能指标和4类调度问题上进行测试后发现，不同的调度规则在不同的条件和性能指标下表现出不同的调度结果；王东军等[7]将整合规划模型应用到实际案例中来研究不同规则对性能指标的影响，结果发现与工件释放时间相关的调度规则具有更好的求解效果；Holthaus等[8]提出没有一条规则可以在所有绩效指标上表现良好，相反，一条规则只能在1个或2个指标上表现最佳。因此，单一使用调度规则既不能保证全局最优，也不能保证局部最优，因此需要一种方法能够在不同的调度时刻，动态地选择调度规则来满足不同时刻的作业任务需求。

随着人工智能研究的不断深入，许多机器学习算法越来越多地应用于DJSSP领域。强化学习[9]作为机器学习中的一个分支，被广泛认定是解决马尔可夫决策问题的有力途径，弥补了以往方法的不足，例如系统仿真[10]等。强化学习主要通过智能体(Agent)在动态环境中使用自己的知识来执行适当的智能操作[11]。强化学习方法已被证明有足够的能力学习求解调度问题的高效策略。Bouazzad等[12]对传统Q学习(Q learning, QL)算法的Q表改进为存储机器选择概率的Q表和特定规则概率的Q表，用于为动态作业车间中的作业分配最合适的加工机器和作业的加工顺序；Shiue等[13]使用多个调度规则(multiple dispatching rules, MDP)机制的强化学习算法以及离线学习模块，用于维护随着车间环境的改变而发生变化的实时调度系统(real-time scheduling system, RTSS)的知识库(knowledge base, KB)，并验证了该方法的调度结果比单个调度规则的机器学习调度算法以及其他元启发式算法更有效；Wang[14]提出了一种动态贪婪搜索策略和一种新的Q函数加权迭代更新算法，采用基于动态贪婪搜索的加权QL算法确定最优调度规则，解决了盲目搜索问题，提高了算法的收敛性和准确性，其仿真结果表明了该方法的适应性和有效性；Shahrabi等[15]使用Q学习算法来寻找可变邻域搜索的最佳参数，用于解决考虑到作业随机到达车间的动态调度问题；Aydi等[16]改进了传统QL算法的学习过程，以构建基于Q-III的RL算法。该算法能够在不同的车间环境下实时选择最佳的DR；Zhao等[17]在作业车间中使用了一种基于QL算法的智能体，用于在机器发生故障的情况下最佳地选择替代机器，实验表明该算法可有效减少频繁动态环境下的延迟时间，验证了该算法求解调度问题的有效性。

为了进一步提高基于QL算法和调度规则的DJSSP调度算法的收敛速度和寻优精度，对传统QL算法进行了改进：①设计了一种新的状态空间和回报函数的表示方法，提高对整个调度过程的

https://dc-china-simulation.researchcommons.org/journal/vol34/iss6/7
DOI: 10.16182/j.issn1004731x.joss.21-0099

Zhao et al.: Application of Improved Q Learning Algorithm in Job Shop Scheduli

描述；②选择合适的调度规则作为动作组，适应本文设定的调度性能指标；③提出了一种基于Softmax函数的改进型动作选择策略，改善传统QL算法在整个学习过程的动作选择概率，使QL算法在学习过程的前期具有更高的动作选择随机性，在后期具有更高概率选择最优动作，提升了QL算法在整个过程中的求解精度。以上将作业车间调度问题转化为强化学习问题，解决了在不同生产情况下使用不同的调度规则的问题，提高了算法在例如DJSSP等复杂多扰动的生产环境下的自主决策能力和对环境变化的自适应能力。

1 DJSSP调度系统模型

1.1 DJSSP的问题描述与数学模型

DJSSP指的是在经典JSSP的基础上，考虑到在加工作业过程中的多种干扰因素，包括紧急订单、机器故障等。针对含有各种干扰因素的生产环境，DJSSP对所有作业的加工进行动态调度。在DJSSP中，设有n个待加工工件$J=\{J_1,J_2,\cdots,J_i,\cdots,J_n\}$需要在$m$台机器$M=\{M_1,M_2,\cdots,M_g,\cdots,M_m\}$上加工，每个工件$J_i$要经过$n_i$道不同的加工顺序，且已知各工序的加工时间和$J_i$在各机器上的加工次序约束。DJSSP的任务目标是在约束条件以及干扰因素下，明确为各个工件的每一道工序选择合适的加工机器，确定其开始加工时间以及每一台加工机器上工件的加工顺序，从而让调度结果满足预期的性能指标。

为了简化当前的作业车间问题，本文设定如下约束：①所有的机器准备时间为0，即所有的工件到达机器后都能立即开始加工；②按照加工工艺规定，每道工序须在它之前的工序加工完成后方可开始加工；③各个工件必须按照工艺路线以指定的次序在机器上加工，并且假定不同工件之间具有相同的优先权；④每一时刻每台机器只能加工一个工件，某一时刻每个工件只能被一台机器加工；⑤工序一旦开始加工不可中断；⑥前一个操作未完成，则后面的操作需要等待；⑦各工件的准备时间和完成时间一起计入加工时间。

在问题描述中涉及到的参数如表1所示。

表1 问题描述参数集
Table 1 Problem description parameter set

符号	符号描述
n	工件总数
i,l	工件索引，$i,l=[1,2,\cdots,n]$
m	机器总数
M_g	第g台机器
g,h	机器索引，$g,h=[1,2,\cdots,m]$
J_i	第i个工件
j	工序索引
O_{ij}	J_i的第j道工序
n_i	J_i的工序数量
t	当前调度时刻(每当一道工序加工完成，或有新工件到达，都视为调度时刻)
PT_{ig}	在t时刻，J_i在M_g上的加工时间
$NPM_i(t)$	在t时刻，J_i流经的机器数

考虑到在当前客户化制造环境下，按时交付是生产作业管理的重要指标之一，也是实现客户满意和车间生产效率的关键，所以本文选择“最小化拖延时间T”作为DJSSP的调度性能指标，即

$$\min T=\max\{C_i-D_i,0\} \tag{1}$$

D_i为工件J_i的预计加工完成时间，由工件的释放时间和加工时间确定：

$$D_i=A_i+k\sum_{j=1}^{n_i}PT_{ig} \tag{2}$$

按照DJSSP的特性设定约束条件：

任何一道工序的完工时间C_{iM_g}不能小于其加工时间t_{iM_g}：

$$c_{iM_g}\geqslant t_{iM_g} \tag{3}$$

工件J_i前后2道相邻工序在机器M_h和M_g上加工的顺序约束：

$$c_{iM_g}-t_{iM_g}+\beta(1-a_{iM_hM_g})\geqslant c_{iM_h} \tag{4}$$

机器M_g完成一个作业任务后才能开始另一个：

$$c_{lM_g}-c_{iM_g}+\beta(1-b_{lM_g})\geqslant t_{lM_g} \tag{5}$$

数学描述中的参数说明如表2所示。

Published by Journal of System Simulation, 2022

Journal of System Simulation, Vol. 34 [2022], Iss. 6, Art. 7

表2 数学描述参数集
Table 2 Mathematical description parameter set

符号	描述
t_{iM_g}	J_i在M_g上的加工时间
c_{iM_g}	J_i在M_g上的加工完成时间
$a_{iM_hM_g}$	布尔系数，当M_h先于M_g加工J_i时，$a_{iM_hM_g}=1$，否则$a_{iM_hM_g}=0$
b_{iIM_g}	布尔系数，当J_i先于J_l在机器上加工时，b_{iIM_g}=1，否则$b_{iIM_g}=0$
A_i	J_i的释放时间
D_i	J_i的预计加工完成时间
C_i	J_i的实际加工完成时间
β	调节因子，一个很小的正值常数
k	延迟系数，若为负值，表示在加工过程中某些工件已经出现加工滞后现象

1.2 QL算法模型

在动态的生产制造环境下，作业任务随客户的订单需求不断进入车间，因此强化学习算法需要支持调度Agent在不断变化的作业条件下，快速而准确地确定最佳的调度规则。

将调度规则封装到强化学习的分支——QL算法的动作序列中，建立DJSSP的调度模型框架。该框架由状态空间、动作、回报函数表达式和选择策略四要素组成。设动作值函数为$Q(s_t,a)$，其中s_t为系统状态，$s_t\in S$，S为系统的状态集合；a为调度系统选用的调度规则。本文从规则库中选出若干调度规则组成QL算法的行为动作组，该动作组可表示为A:$\{a_1,a_2,\cdots,a_n\}$。QL算法在不同的时刻从动作组中随机选取某一规则与环境进行反复试错，由此得到了在不同状态下采取不同规则的方法，能够使调度系统性能达到最优。调度Agent在某一时刻的状态s_t下根据选择策略从动作组中选择并对调度系统使用一个规则，根据回报函数获得当前状态相应的回报并进入新的状态s_{t+1}，动作值也随之更新，故s_{t+1}只与当前状态s_t有关，与更早状态无关。调度系统最终在学习阶段得到了一个用于存储所有Q值的Q表，它包含了在不同状态下对应的最优调度规则的信息，使得调度Agent在任一时刻都选择动作集合中对应Q值最大的调度规则作为当前的动作。动作值函数的迭代定义为

$$Q(s,a)=Q(s_t,a_t)+\alpha[r_{t+1}+\lambda\max_a Q(s_{t+1},a_t)-Q(s_t,a_t)] \tag{6}$$

式中：α为学习率，决定了函数值的更新程度；λ为折扣系数，若λ趋近于0，智能体倾向于选择立即回报值，若λ趋近于1，则智能体倾向于未来奖励值。$Q(s,a)$通过与环境的持续交互，逐步迭代学习，Q表得以更新，直至囊括了绝大多数的状态动作对–(s,a)的Q表，最终收敛于最优状态动作值函数$Q^*(s,a)$。QL算法学习步骤如下：

QL算法流程

初始化 $s_t\in S,Q(s,a)$为任意值

循 环 初始化s_t

循 环 经验轨迹中的时间步$t=1\sim T$

根据动作值Q，在状态s_t下，使用贪婪策略，以概率ε随机选择动作a_t，否则以$1-\varepsilon$的概率选择Q值最大的动作

执行动作a_t，获得奖励r_{t+1}和下一步的状态s_{t+1}

$$Q(s,a)=Q(s_t,a_t)+\alpha[r_{t+1}+\lambda\max_a Q(s_{t+1},a_t)-Q(s_t,a_t)]$$

状态s进入下一个状态s_{t+1}，记录状态

终止时间步T

输 出 动作值函数$Q(s,a)$

QL算法收敛至最优的(s,a)值，支持调度Agent通过在未知环境中不断探索，在不同状态下选择最佳调度规则。

为了使QL算法在DJSSP中表现得更加优异，需要对QL算法的各个组成部分进行设计，以满足DJSSP在求解的精确性和收敛性方面的需求。

2 求解DJSSP的改进QL算法设计

状态空间、回报函数、动作组，以及动作选择策略是QL算法的重要组成部分。状态空间和回报函数的设计影响着调度系统对于求解调度结果的精确度，求解最优策略所选取的方法则影响着QL算

https://dc-china-simulation.researchcommons.org/journal/vol34/iss6/7
DOI: 10.16182/j.issn1004731x.joss.21-0099

Zhao et al.: Application of Improved Q Learning Algorithm in Job Shop Scheduli

法在学习过程中的最终收敛性。状态空间和回报函数的表示方法，以及对动则选择策略的设计是本文对基于QL算法的DJSSP调度算法的主要改进部分。

根据“最小化拖延时间”这一性能指标，分别设计了一种新的状态空间以及回报函数的表示方法，并提出了一种以Softmax函数为主体的动作选择策略，完成对基于QL算法的DJSSP调度算法进行改进，最后将上述内容整合到改进后的QL算法学习步骤中。

2.1 状态空间

应用QL算法的关键是系统要识别多阶段决策的状态特征，在不同的学习阶段，其描述方式也应不同。判断当前的系统状态是选择动作的关键，选择正确的状态变量可以用于准确地表示当前的系统状态空间。状态空间的设计大部分情况下遵循以下原则[18]：①状态空间能够描述调度环境的主要特点和变化，包括系统全局特征和局部特征；②状态空间可以表示和概括各种不同的调度问题；③状态空间是一种对状态变量的数值表征；④状态空间应易于计算；⑤状态区间的划分应适量，过多的状态空间会导致算法在寻找策略的过程中收敛缓慢，难以达到最优。

设计状态空间的关键是定义合适的状态变量，状态空间依赖于状态变量的表示形式。为了更加准确地描述调度系统的加工状态以及针对“最小化拖延时间”的性能指标，本文对传统状态变量的定义进行了改进，分别设计了$EAST(t)$和$EART(t)$这2个新的状态变量，并使用参数$NPM_i(t)$及PT_{ig}表示这2个变量。参数与状态变量如表3所示。

表3 状态空间参数与变量集

Table 3 State space parameters and variable sets

参数与变量	描述(所有工件)
$NPM_i(t)$	在t时刻，J_i流经的机器数
PT_{ig}	在t时刻，J_i在M_g上的加工时间
$EAST(t)$	在t时刻，剩余工序预计平均松弛时间
$EART(t)$	在t时刻，预计平均剩余加工时间

$EAST(t)$与$EART(t)$的数学表达式分别为

$$EAST(t)=\frac{1}{n}\sum_{i=1}^{n}\left(\sum_{k=1}^{n_i-NPM_i(t)}PT_{ig}-(D_i-t)\right) \tag{7}$$

$$EART(t)=\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{n_i-NPM_i(t)}PT_{ig} \tag{8}$$

其中，$n_i-NPM_i(t)$可理解为J_i的剩余工序数。

$EAST(t)$和$EART(t)$强化了状态变量对整个系统生产状态的描述，更加准确地体现了不同生产状态下调度系统的性能。

进一步提出“剩余加工紧迫程度”的思想，用于具体描述状态空间的划分，更加准确地描述调度系统的状态空间，增强了基于QL算法的动态调度算法的精确性。随着调度动作的逐一执行而变化，$EAST(t)$不断增加，$EART(t)$会逐渐减少，工件的剩余加工时间越少，加工进程越紧迫。为此，以“剩余加工紧迫程度”为主要依据，通过对$EAST(t)$与$EART(t)$2个状态变量大小的比较，得到大小不同的6个区间，分别对应6个系统状态，如表4所示。h是一个可调参数，通常为1的整数倍数，用于将连续的状态空间离散化为任意区间，并赋予状态特征相应的权重。h的取值不宜过大，否则会出现QL算法收敛缓慢，难以达到最优的现象。

表4 状态空间集

Table 4 State space set

状态序列	状态区间	动作1	动作2	动作3	…
0	$EAST(t)\leqslant 0$	$Q(0,0)$	$Q(0,1)$	$Q(0,2)$	
1	$0\leqslant EAST(t)<EART(t)\cdot h$	$Q(1,0)$	$Q(1,1)$	$Q(1,2)$	
2	$h\cdot EART(t)\leqslant EAST(t)<EART(t)\cdot 2h$	$Q(2,0)$	$Q(2,1)$	$Q(2,2)$	
3	$2h\cdot EART(t)\leqslant EAST(t)<EART(t)\cdot 3h$	$Q(3,0)$	$Q(3,1)$	$Q(3,2)$	
4	$3h\cdot EART(t)\leqslant EAST(t)<EART(t)\cdot 4h$	$Q(4,0)$	$Q(4,1)$	$Q(4,2)$	
5	$4h\cdot EART(t)\leqslant EAST(t)$	$Q(5,0)$	$Q(5,1)$	$Q(5,2)$	

Published by Journal of System Simulation, 2022

Journal of System Simulation, Vol. 34 [2022], Iss. 6, Art. 7

2.2 回报函数

设计回报函数应该将调度性能指标作为主要参考依据。回报函数的选择应依据以下原则[19]：①反映动作的即时影响，与动作的即时回报联系密切；②累计的总回报值反应目标函数值；③回报函数能够应用于不同规模的问题。

回报函数中变量的选取影响着调度决策系统在当前调度状态的奖励值与惩罚值。在以往的研究中，回报函数的定义大部分取决于状态空间在作业加工过程中的优劣，例如，好的状态空间的回报值为正值，坏的状态空间的回报值为负值，其他情况的回报值为0。虽然这样能较为广泛地描述各个状态空间的实际生产情况，但这种回报函数往往在某一加工状态空间形成以后才开始实施，并不能描述在状态空间的形成过程中的回报值，既不能充分地体现整个加工过程中状态空间的优劣变化，又存在一定的滞后性。

为了解决上述问题，本文针对“最小化拖延时间”的性能目标，提出了“松弛时间一旦出现，系统立即受到惩罚”的回报函数设计思路，并将$EAST(t)$设定为回报函数变量。回报函数r定义为

$$r = c - EAST(t) \tag{9}$$

式中：c为正值常数。

调度系统出现拖延时间会影响性能指标的优化，这里选取惩罚值与$EAST(t)$成正比，即$EAST(t)$越大，系统得到的惩罚值越高。式(9)表明，$EAST(t)$的值越大，得到的回报值越小，即对调度系统的惩罚越严厉。当$EAST(t) \leqslant 0$时，回报值为正数，可视为该工件被正常加工完毕，无拖延时间。本文提出的回报函数体现了QL算法在作业车间调度问题上拖延时间越长，回报值越低这一动态调度特性，同时设定一个动作标签，记录该状态下执行的动作“不太适用于这一生产状态”，引导算法继续探索下一个动作。

2.3 定义动作组

在DJSSP中，调度Agent根据加工机器的加工情况以及加工工件的生产属性，为每一个待加工工件计算一个加工优先数值，以对待加工工件进行排序，再择优通过加工机器加工完成调度任务。调度规则是计算工件加工优先数值的有力工具。本文选取在“最小化拖延时间”这一性能指标下常用的6种调度规则：

(1) 先到先加工规则FCFP (first come first processed)：在工件的待加工序列中，选择释放时间r_i最小的的工件进行加工，各个待加工工件的优先级为

$$P = \min r_i \tag{10}$$

(2) 工序加工时间最短规则SPT(the shortest processing time)：优先选择当前工序加工时间p_{ij}最短的工件，各个待加工工件的优先级为

$$P = \min p_{ij} \tag{11}$$

(3) 工序调度松弛量最小规则SL(the smallest slack)：优先选择工序松弛量最小的工件，各个待加工工件的优先级为

$$P = D_i - s' - \sum_{j=1}^{m} p_{ij} \tag{12}$$

(4) 剩余工序数量最少规则LOPNR(the least number of operations remaining)：优先选择剩余工序数量最少的工件，各个待加工工件的优先级为

$$P = n_i - NPM_i(t) \tag{13}$$

(5) 剩余加工时间最长规则MWKR(the most work remaining)：优先选择优先选择剩余加工时间最长的工件，各个待加工工件的优先级为

$$P = -\sum_{j=j'}^{n_i} (PT_{ig}) \tag{14}$$

式中：j'为当前工序。

(6) 随机规则Random：随机选择一个工件进行加工。

选取能用到的各个调度规则后，将这些调度规则封装为改进QL算法的动作组，以在调度时刻从动作组中选择调度规则作用于工件的加工过程。

2.4 求解最优策略

QL算法在不同时刻选择不同的动作，最终目

https://dc-china-simulation.researchcommons.org/journal/vol34/iss6/7
DOI: 10.16182/j.issn1004731x.joss.21-0099

Zhao et al.: Application of Improved Q Learning Algorithm in Job Shop Scheduli

的是让调度Agent将回报值积累至最大化，回报值的最大化决定了每个调度时刻所选择的最优动作。QL算法在求解最优策略时需要在探索与利用之间权衡，探索指的是在每一次需采取动作的决策中，算法在已知最优动作的条件下，继续寻找其他可能获得更高回报值的其他动作，而不局限于当前最优动作；利用指的是选择当前已经发现的最优动作来获得最大的回报值。在传统的QL算法中，动作选择策略一般基于贪婪策略($\varepsilon-greedy$)。在QL算法中，贪婪策略以概率$P=\varepsilon$随机选择一个动作a_t，否则根据$a_t=\max Q(\varphi_t,a)$来选择动作a_t。

贪婪策略的缺点是在后期还会选择非最优策略为最优策略，在学习的过程中随着学习的深入选择动作的概率基本不变，导致学习初期存在次优动作值大于最优动作Q值的现象，在学习后期也会以较大概率选择非最优动作，因此使学习的结果不准确，学习过程收敛慢，造成一些不必要的资源浪费。为了解决这一问题，对传统的QL算法进行改进，引入Softmax函数，提出一种基于Softmax函数的动作选择策略：

$$P(s_t,a_i)=\frac{\exp(\mu Q(s_t,a_i))}{\sum_{a\in A}\exp(\mu Q(s_t,a,))} \tag{15}$$

式中：μ为标量。

由式(15)可知，当初始状态的Q值都为0时，各个动作被选择的概率相等，从而使得前期的动作选择更加随机，探索的更合理。对Q值的更新更准确；当学习进行至一定阶段时，各个动作的回报值不同，Q值的更新情况也不同；动作的回报值越大，Q值越大，动作的回报值越小，Q值越小。随着学习的深入，系统更倾向于回报值更高的动作。

2.5 更新Q学习算法流程

将2.1~2.4节中所设计的状态空间、回报函数、行为动作组，以及动作选择策略整合到原始QL算法中，得到了改进QL算法的学习步骤：

改进QL算法流程

初始化$s_t\in S,Q(s,a)$为任意值

循　环　初始化s_t

　循　环　经验轨迹中的时间步$t=1\sim T$

　　初始化状态$s_t=\{EAST(t),EART(t)\}$，根据动作值Q，使用2.4中的基于“Softmax”函数的动作选择策略函数$P(s_t,a_i)$从规则库$Z=\{FCFP,SPT,SL,LOPNR,MWKR,Random\}$中随机选择一个规则$a_i$进行调度

　　执行a_i，根据式(9)获得奖励r_{t+1}，并更新下一时间步的状态s_{t+1}

$$Q(s,a)=Q(s_t,a_t)+\alpha[r_{t+1}+\lambda\max_a Q(s_{t+1},a_t)-Q(s_t,a_t)]$$

　　状态s进入下一个状态$s_{t+1}=\{EAST(t+1),EART(t+1)\}$，记录状态

　　终止时间步T

输　出　动作值函数$Q(s,a)$

3 仿真研究与结果分析

为验证改进QL算法解决DJSSP的有效性，本文设置了多组基于经典JSSP算例的计算机仿真实验。所有仿真实验均在加速频率为3.6 GHz的Intel i5-9400处理器、运行内存为16 G的Windows10 PC平台进行，其中改进QL算法采用Python编程语言在Pycharm软件中编写并调试。实验中，分别采用了经典的JSSP算例：LA01(10×5)、LA05(10×5)、LA06(15×5)、LA10(15×5)、LA11(20×5)、LA12(20×5)，其中10×5代表10个工件在5台机器的算例规模，以此类推。在仿真过程中，作业到达机器上开始处理的时间符合一定程度的均匀分布。

为了统一仿真标准，算法的基础参数如表5所示。每个算例均采用FCFP、SPT、SL、LOPNR、MWKR和Random规则作为算法的动作组，规则的具体说明见表6。

Published by Journal of System Simulation, 2022

Journal of System Simulation, Vol. 34 [2022], Iss. 6, Art. 7

表5 仿真参数取值
Table 5 Values of simulation parameters

参数	取值
学习率α	0.01
折扣率λ	0.9
回报函数参数c	1
状态空间参数h	1
k取值范围	{0.2, 0.4, 0.6, 0.8, 1.0}
Episode	1 000
Softmax 策略u	1
均匀分布	[1,10]

表6 每台机器的动作候选集
Table 6 Action candidate set for each machine

规则名称	规则描述
FCFP	优先选择最先到达机器的工件
SPT	优先选择工序加工时间最短的工件
SL	优先选择调度松弛量最小的工件
LOPNR	优先选择剩余工序数量最少的工件
MWKR	优先选择剩余加工时间最长的工件
Random	随机选择一个工件

3.1 仿真结果

3.1.1 紧急加工系数k的取值对调度结果的影响

k的取值反映了工件需要加工的紧急程度，k的值越小，需要加工工件的优先程度越高。在6个算例中，通过改变k的值，可以获取改进Q学习算法的调度结果，并可将其与单一调度规则的调度结果相比较，如表7~12所示。由此可知，改进QL算法在学习结束后，能够在不同的调度时刻动态地使用调度规则，进而对DJSSP进行求解，且所得调度结果优于使用单一调度规则的调度结果。

表7 LA01调度结果
Table 7 LA01 scheduling result

规则	k=0.2	k=0.4	k=0.6	k=0.8	k=1.0
FCFP	547.42	523.17	446.46	389.08	312.00
SPT	527.12	510.67	459.66	334.18	303.10
SL	538.02	549.82	470.76	418.88	323.60
LOPNR	563.72	605.42	447.86	405.78	299.30
MWKR	542.62	531.17	475.76	381.48	301.20
Random	533.12	546.12	486.96	379.58	327.00
QL	**524.02**	**507.62**	**442.26**	**312.78**	**292.30**

表8 LA05调度结果
Table 8 LA05 scheduling result

规则	k=0.2	k=0.4	k=0.6	k=0.8	k=1.0
FCFP	447.54	423.75	337.52	289.06	248.00
SPT	468.04	456.05	321.52	277.36	230.70
SL	463.74	487.63	394.72	325.66	276.50
LOPNR	457.64	437.63	356.62	292.46	214.30
MWKR	447.54	463.65	344.12	313.66	264.30
Random	483.84	445.13	359.22	293.36	274.70
QL	**445.84**	**440.85**	**293.42**	**277.26**	**212.30**

表9 LA06调度结果
Table 9 LA06 scheduling result

规则	k=0.2	k=0.4	k=0.6	k=0.8	k=1.0
FCFP	702.24	720.68	522.39	426.07	485.93
SPT	693.04	679.18	565.25	488.83	494.06
SL	759.51	745.68	623.65	572.83	529.67
LOPNR	733.84	730.18	605.79	583.73	513.67
MWKR	687.44	660.88	547.65	482.63	492.06
Random	713.64	692.68	577.99	506.63	480.93
QL	**684.44**	**653.68**	**521.52**	**423.36**	**474.93**

表10 LA10调度结果
Table 10 LA10 scheduling result

规则	k=0.2	k=0.4	k=0.6	k=0.8	k=1.0
FCFP	763.13	733.23	574.2	507.93	435.00
SPT	765.87	726.63	566.2	512.13	463.93
SL	789.2	745.63	622.47	601.4	526.00
LOPNR	736.67	711.53	589.37	538.74	528.16
MWKR	721.17	704.13	603.2	510.93	487.63
Random	750.47	730.43	615.07	522.53	532.87
QL	**704.73**	**692.83**	**515.33**	**494.8**	**426.73**

表11 LA11调度结果
Table 11 LA11 scheduling result

规则	k=0.2	k=0.4	k=0.6	k=0.8	k=1.0
FCFP	834.49	810.22	708.32	665.91	589.65
SPT	840.44	815.32	719.22	660.31	607.80
SL	1 010.44	923.82	878.37	846.31	780.10
LOPNR	847.64	845.12	787.67	735.31	642.30
MWKR	835.44	820.62	764.62	814.56	619.00
Random	886.29	876.18	736.37	759.46	703.25
QL	**833.24**	**802.42**	**754.82**	**659.36**	**585.70**

https://dc-china-simulation.researchcommons.org/journal/vol34/iss6/7
DOI: 10.16182/j.issn1004731x.joss.21-0099

表 12　LA12 调度结果
Table 12　LA12 scheduling result

规则	k=0.2	k=0.4	k=0.6	k=0.8	k=1.0
FCFP	675.59	663.43	680.27	574.71	471.20
SPT	723.34	715.45	628.42	584.46	498.30
SL	921.04	876.25	798.87	702.76	632.10
LOPNR	747.84	708.75	644.17	635.71	458.40
MWKR	689.54	664.35	650.82	596.61	479.30
Random	767.14	756.63	687.42	609.46	466.20
QL	**670.94**	**643.15**	**627.97**	**565.71**	**452.60**

3.1.2 改变k的值对改进QL算法收敛性的影响

以 LA10 为例，将 k 分别设为 0.2、0.4、0.6、0.8、1.0。训练步数设为 1 000，改进 QL 算法的迭代对比图如图 1 所示，直观地展现了改进 QL 算法的学习过程。在该学习过程中，由于选取的调度规则不同，每个调度规则分别产生的回报值也不同，导致了改进 QL 算法的学习轨迹产生了一定程度的波动。曲线的波动是由于动作选择策略反复选取不同调度规则所致，体现了改进 QL 算法在不同的状态下选择调度规则的动态性，同时波动也可以帮助改进 QL 算法找到更好的调度结果。

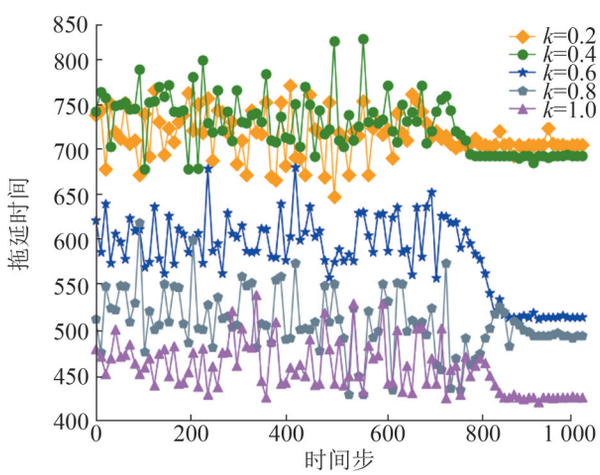

图 1　LA10 中不同的 k 值对改进 QL 算法的迭代曲线的影响
Fig. 1　Influence of different k values on the iterative curve of improved QL algorithm in LA10

3.1.3 Softmax 函数与贪婪策略对 QL 算法收敛性的分析

本文对 QL 算法的改进主要是基于动作选择策略的改进。为了验证所提的基于 Softmax 函数的动作选择策略相较于贪婪策略的优越性，本文设置了对照组实验：将算法中 Softmax 选择策略替换为探索率 ε = 0.01 的贪婪策略，即贪婪策略以 1% 的概率随机选择调度规则，以 99% 的概率选择价值最大的调度规则。2 种实验同时选择 LA10、k=0.6 为算例，观察 QL 的迭代计算曲线，如图 2 所示。从图 2 可以看出，基于 Softmax 函数的动作选择策略使 QL 算法具有更快的收敛速度以及更优的调度结果，充分验证了本文所提基于 Softmax 动作选择策略的改进 QL 算法的有效性与优越性。

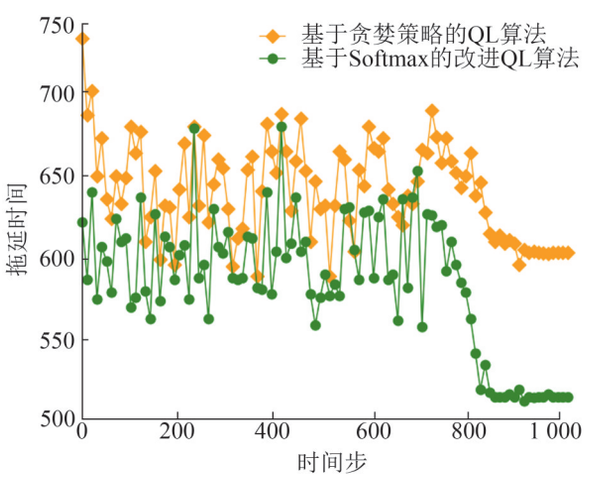

图 2　LA10 中 QL 算法改进前后性能对比
Fig. 2　Performance comparison of QL algorithm before and after improvement in LA10

3.2 仿真对比分析

为了验证改进 QL 算法在 DJSSP 中的有效性，本文分别对比了不同算法所得到的性能指标结果。

首先，为了验证使用 Softmax 动作选择策略的改进 QL 算法得到的结果优于使用贪婪策略的传统 QL 算法，本文与文献[20]进行对比，在 QL 算法相关参数保持一致且保持文献[20]算法其他部分不变的情况下，同样使用 FCFS、SPT、SL、LOPNR、MWKR 和 Random 作为调度时刻的动作，同样以 LA01、LA05、LA06、LA10、LA11、LA12 为测试算例、k=0.6 的情况下，得到对比结果，如表 13 所示。

Journal of System Simulation, Vol. 34 [2022], Iss. 6, Art. 7

表 13 改进 QL 算法与传统 QL 算法的仿真结果对比
Table 13 Comparison of simulation results between improved QL algorithm and traditional QL algorithm

测试算例	算例规模	改进 QL 算法	传统 QL 算法
LA01	10×5	**442.26**	488.56
LA05	10×5	**293.42**	334.87
LA06	15×5	**521.52**	573.45
LA10	15×5	**515.33**	520.64
LA11	20×5	**754.82**	806.75
LA12	20×5	**627.97**	680.68

由表 13 可知，改进 QL 结果优于传统 QL，验证了所提算法相较于传统 QL 算法的更优越。图 3 中的曲线分别展现了以 LA10，k=0.6 的情况下，本文与文献[20]中所提算法的收敛过程对比，改进 QL 算法的收敛与求解结果明显优于文献[20]中算法所得到的结果。

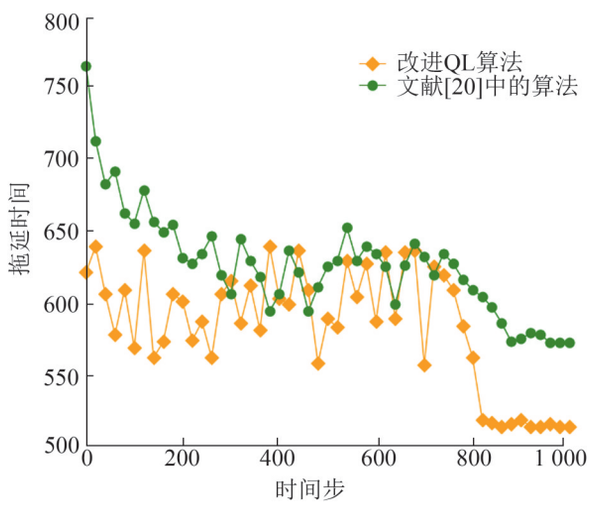

图 3 算法结果对比分析
Fig. 3 Comparative analysis of algorithm results

其次，验证改进 QL 算法与粒子群算法(PSO)、改进粒子群算法(IPSO)等传统优化方法在解决 JSSP 上的优劣性。本文分别选取文献[21-22]中的 IPSO、PSO 算法进行仿真对比。实验中，IPSO 和 PSO 的相关参数与文献[21-22]保持一致，并使用 LA01、LA05、LA06、LA10、LA11、LA12 作为测试算例，调度性能指标为“最小化拖延时间”。将本文的调度规则与改进 QL 相结合的调度算法与文献[21-22]中的算法所得到的调度结果相互比较，实验中，改进 QL 算法中的 k 设为 0.6，结果如表 14 所示。

表 14 改进 QL 算法与传统优化算法的仿真结果对比
Table 14 Comparison of simulation results between improved QL algorithm and traditional optimization algorithms

算例	算例规模	改进 QL 算法	文献[21]的 IPSO	文献[22]的 PSO	文献[22]的 IPSO
LA01	10×5	**442.26**	736.5	736.5	736.5
LA05	10×5	**395.02**	765	775	792
LA06	15×5	**521.52**	849.3	850.0	867.0
LA10	15×5	**515.33**	944	975	944
LA11	20×5	**754.82**	1 025	1 025	1 025
LA12	20×5	**627.97**	895	923	895

实验结果表明，改进后的 QL 算法在求解过程中通过大量试错，能够在全局内找到更优的计算结果，在不同规模的 JSSP 算例下，所得到的性能指标结果均优于 PSO、IPSO 等传统的智能优化方法，体现出了改进 QL 算法在求解 DJSSP 的有效性。

4 结论

本文提出了一种基于改进 QL 算法与调度规则的动态调度算法来解决 DJSSP，将“最小化拖延时间”作为性能指标，依据该指标分别提出了具有“紧迫程度”的状态空间和“拖延惩罚”的汇报函数，并设计了一种在整个学习过程具有“前期更加随机”与“后期更加精确”性质的动作选择策略，从而通过在每个调度时刻动态地分配最合适的调度规则的方式来求解 DJSSP。实验结果表明：该算法相较于未改进前的 QL 算法，性能指标至少有 6% 的提升，相较于 IPSO、PSO 至少有 38% 的提升，同时远远优于使用单个调度规则的调度结果，总体体现出了更好的动态性，达到了更好的动态调度效果，证实了改进 QL 算法在求解 DJSSP 的有效性与优越性。

未来，在 DJSSP 中的研究中会考虑以下几个方面：①为了详细描述 DJSSP 的工作步骤或工作流程，可将 DJSSP 在行为层面分为 2 部分，一是

https://dc-china-simulation.researchcommons.org/journal/vol34/iss6/7
DOI: 10.16182/j.issn1004731x.joss.21-0099

Zhao et al.: Application of Improved Q Learning Algorithm in Job Shop Scheduli

将待加工工件通过适当的调度规则分配在不同加工状态的机器上等待加工。二是将这些工件通过适当的调度规则来排序加工，从而达到某一或某些调度性能指标，这是未来该方向的一个研究热点；②当前的强化学习调度策略仅仅使用了一种回报函数，如果能在上述的2层调度行为中分别设计回报函数，也许会大大增加调度系统的精度，是未来的研究工作中值得深入探索的内容；③当调度系统需要定义大量的输入状态时，QL算法的Q表的状态维数会不断增大。使用Q表存储有限的Q值转化为使用神经网络中的众多神经元来拟合Q值，这样可以避免Q表维数过大。

参考文献：

[1] Applegate D, Cook W. A Computational Study of the Job-Shop Scheduling Problem[J]. Orsa Journal on Computing (S0899-1499), 1991, 3(2): 149-156.

[2] 郑先鹏, 王雷. 面向作业车间调度问题的遗传算法改进[J]. 河北科技大学学报, 2019, 40(6): 496-502.
Zheng Xianpeng, Wang Lei. Improved Genetic Algorithm for Job Shop Scheduling[J]. Journal of Hebei University of Science and Technology, 2019, 40(6): 496-502.

[3] Ge H W, Sun L, Liang Y C, et al. An Effective PSO and AIS-Based Hybrid Intelligent Algorithm for Job-Shop Scheduling[J]. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans(S1083-4427), 2008, 38(2): 358-368.

[4] Iskander P W. A Survey of Scheduling Rules[J]. Operations Research(S0030-364X), 1977, 25(1): 45-61.

[5] Jones A, Rabelo L C, Sharawi A T. Survey of Job Shop Scheduling Techniques[M]// Wiley Encyclopedia of Electrical and Electronics Engineering. John Wiley & Sons, Inc, 1999.

[6] Durasevic M, Jakobovic D. A Survey of Dispatching Rules for the Dynamic Unrelated Machines Environment [J]. Expert Systems with Applications(S0957-4174), 2018, 113: 555-569.

[7] 王东军, 刘翱, 刘克, 等. 基于优先规则的复杂并行机调度问题研究[J]. 系统工程理论与实践, 2016, 36(3): 779-786.
Wang Dongjun, Liu Ao, Liu Ke, et al. Priority Rule-Based Complex Indentical Parallel Machines Scheduling [J]. System Engineering Theory and Practice, 2016, 36 (3): 779-786.

[8] Holthaus O, Rajendran C. Efficient Dispatching Rules for Scheduling in a Job Shop[J]. International Journal of Production Economics(S0925-5273), 1997, 48(1): 87-105.

[9] Sutton R S, Barto A G. Reinforcement Learning: An Introduction[J]. IEEE Transactions on Neural Networks (S1045-9227), 1998, 9(5): 1054.

[10] Jahangirian M, Eldabi T, Naseer A, et al. Simulation in Manufacturing and Business: A Review[J]. European Journal of Operational Research(S0377-2217), 2010, 203 (1): 1-13.

[11] Dorigo M, Colombetti M. Robot Shaping: Developing Autonomous Agents Through Learning[J]. Artificial Intelligence(S0004-3702), 1994, 71(2): 321-370.

[12] Bouazza W, Sallez Y, Beldjilali B. A Distributed Approach Solving Partially Flexible Job-Shop Scheduling Problem with a Q-learning Effect[J]. Ifac Papersonline(S2405-8963), 2017, 50(1): 15890-15895.

[13] Shiue Y R, Lee K C, Su C T. Real-Time Scheduling for a Smart Factory Using a Reinforcement Learning Approach [J]. Computers & Industrial Engineering(S0360-8352), 2018, 125: 604-614.

[14] Wang Yufang. Adaptive Job Shop Scheduling Strategy Based on Weighted Q-Learning Algorithm[J]. Journal of Intelligent Manufacturing (S0956-5515), 2020, 31(2): 417-432.

[15] Shahrabi J, Adibi M A, Mahootchi M. A Reinforcement Learning Approach to Parameter Estimation in Dynamic Job Shop Scheduling[J]. Computers & Industrial Engineering(S0360-8352), 2017, 110: 75-82.

[16] Aydin M E, Ztemel E. Dynamic Job-Shop Scheduling Using Reinforcement Learning Agents[J]. Robotics & Autonomous Systems(S0921-8890), 2000, 33(2/3): 169-178.

[17] Zhao M, Li X, Gao L, et al. An Improved Q-Learning Based Rescheduling Method for Flexible Job-Shops with Machine Failures[C]//2019 IEEE 15th International Conference on Automation Science and Engineering (CASE). Vancouver, Canada: IEEE, 2019: 331-337.

[18] 张智聪, 郑力. 基于增强学习的制造系统调度[M]. 北京: 科学出版社, 2016.
Zhang Zhicong, Zheng Li. Manufacturing System Scheduling Based on Reinforcement Learning[M]. Beijing: Science Press, 2016.

[19] 肖鹏飞, 张超勇, 孟磊磊, 等. 基于深度强化学习的非置换流水车间调度问题[J]. 计算机集成制造系统, 2021, 27(1): 192-205.
Xiao Pengfei, Zhang Chaoyong, Meng Leilei, et al. Non-Permutation Flow Shop Scheduling Problem Based on Deep Reinforcement Learning[J]. Computer Intergarted

Published by Journal of System Simulation, 2022

Journal of System Simulation, Vol. 34 [2022], Iss. 6, Art. 7

Manufacturing Systems, 2021, 27(1): 192-205.

[20] Wei Yingzi, Zhao Mingyang. A Reinforcement Learning-Based Approach to Dynamic Job-Shop Scheduling[J]. Acta Automatica Sinica(S0254-4156), 2005, 31(5): 765-771.

[21] 刘洪铭, 曾鸿雁, 周伟,等. 基于改进粒子群算法作业车间调度问题的优化[J]. 山东大学学报(工学版), 2019, 49(1): 57-62.
Liu Hongming, Zeng Hongyan, Zhou Wei, et al. Optimization of Job Shop Scheduling Problem Based on Improved Particle Swarm Algorithm[J]. Journal of Shandong University (Engineering Edition), 2019, 49(1): 57-62.

[22] 杨恒. 基于改进粒子群算法的作业车间调度优化[J]. 机械设计与制造工程, 2019, 48(2): 77-80.
Yang Heng. Optimization of Job Shop Scheduling Based on Improved Particle Swarm Algorithm[J]. Mechanical Design and Manufacturing Engineering, 2019, 48(2): 77-80.

https://dc-china-simulation.researchcommons.org/journal/vol34/iss6/7
DOI: 10.16182/j.issn1004731x.joss.21-0099