

[经营·管理]

DOI:10.3969/j.issn.1005-2895.2024.03.015

面向智能生产的分布式流水车间调度研究

陈俊贤,李仁旺*

(浙江理工大学机械工程学院,浙江杭州 310018)

摘要:为了使传统流水车间的调度模型更灵活和更智能化以适应不同生产环境,课题组提出了基于深度学习的分布式流水车间调度方法。通过学习和分析分布式车间系统中的大量数据,利用策略梯度方法在多次迭代优化后使目标得到近似最优解,获取了更智能、适应性更强的生产计划和调度策略;并通过实验和仿真进行验证。结果表明该方法能提高生产效率和资源利用率,并具有成本控制方面的潜力。该研究为制造业的分布式生产环境提供了一种先进的调度策略,为车间管理者提供更准确、更智能的决策参考。

关键词:生产调度;分布式流水车间;深度学习;调度策略;策略梯度法

中图分类号:TB497;TH186 文献标志码:A 文章编号:1005-2895(2024)03-0100-08

Research on Distributed Flow Shop Scheduling for Intelligent Production

CHEN Junxian, LI Renwang*

(School of Mechanical Engineering, Zhejiang University of Science and Technology, Hangzhou 310018, China)

Abstract: In order to make the traditional flow shop scheduling model more flexible and intelligent to adapt to different production environments, scheduling strategy of distributed flow shop based on deep learning was proposed. By learning and analyzing a large amount of data in the distributed shop floor system, the strategy gradient method was used to obtain the approximate optimal solution after several iterations of optimization, and a more intelligent and adaptable production planning and scheduling strategy was obtained. It was verified by experiments and simulation. The results show that this method can improve production efficiency and resource utilization, and has potential in cost control. The research provides an advanced scheduling strategy for distributed production environment of manufactur industry, and provides more accurate and intelligent decision reference for shop floor managers.

Keywords: production scheduling; distributed flow shop; deep learning; scheduling strategy; strategy gradient method

随着全球制造业的快速发展,生产流水车间的调度问题一直是制造企业面临的重要挑战之一。然而,传统的流水车间调度方法往往无法满足现代制造环境中分布式流水车间的需求,其中涉及到多个地理位置的生产单元和复杂的物流网络^[1]。为了应对这一挑战,研究者们纷纷探索基于深度学习的新型调度方法,以提高生产效率、降低成本,并实现更灵活的生产计划。在现代制造业中,为了更好地适应市场需求的变化和提高整体生产系统的弹性,生产工厂被部署在多个地域上,形成了分布式生产模式。通过在不同地理

位置分散设置工厂,并使它们之间相互协作,可以有效地缓解由于意外情况导致某一工厂无法正常生产的问题。在这种模式下,如果其中一个工厂面临意外状况或生产中断,其他工厂可以分担其生产任务,从而提高整个生产系统的抗风险能力。这种分布式生产模式目前被现代制造企业广泛采用,为其提供了更灵活、更可靠的生产架构,以适应不断变化的市场环境^[2-3]。

分布式流水车间调度问题(distributed flow shop scheduling problem, DFSP)被中外学者广泛研究,并取得了一些进展^[4-5]。分布式流水车间调度问题一开始

收稿日期:2024-03-07;修回日期:2024-04-01

基金项目:浙江省2023年度“尖兵”“领雁”研发攻关计划(2022C01SA111123);国家自然科学基金资助项目(51475434)。

第一作者简介:陈俊贤(1999),男,硕士研究生,主要研究方向为智能制造、车间调度智能优化。通信作者:李仁旺(1971),男,博士,教授,主要研究方向为制造业信息化、产品设计与产品建模技术、企业管理与物流工程。E-mail:2814201821@qq.com

由 Sauer 等^[6]提出,通过模糊理论来求解该调度问题。Naderi 等^[7]建立了相关模型,并通过变领域梯度下降算法求解该模型。杨晓林等^[8]提出分布估计算法求解分布式流水线调度问题,并采用 3 种 Insert 融合搜索方式寻找优质解。Wang 等^[9]针对分布式置换流水车间调度模型,将优化目标定为最小化加工时间和总能耗,提出了一种基于知识的协同算法来解决该模型。Shao 等^[10]基于元启发式算法,求解分布式流水车间调度模型。

目前,解决分布式流水车间调度问题的方法主要包括传统的优化算法和基于深度学习的方法^[11]。传统的优化算法如遗传算法^[12]、蚁群算法^[13]和贪婪迭代算法^[14]等在一定程度上能够解决问题,但在处理复杂的、实时的生产环境问题时存在一定的局限性;相比之下,基于深度学习的方法通过学习大量实时数据,能够更准确地捕捉系统的动态变化,并提供更智能、更灵活的调度策略。深度学习技术,尤其是神经网络模型^[15],具有强大的学习和泛化能力,能够更好地适应不断变化的生产环境。基于深度学习的方法不仅可以处理大规模的数据,还能够从数据中学到复杂的非线性

性关系,为分布式流水车间的调度问题提供了一种全新的解决途径。

课题组深入探讨了基于深度学习的分布式流水车间调度方法,通过对目前的求解方法进行分析,以最小最大完工时间为优化目标,采用近端策略优化算法求解这一调度问题,以提高生产效率和降低成本,为制造业在面对分布式生产环境的调度挑战时提供新的思路和参考方案。

1 问题描述及数学模型

1.1 问题描述

经典分布式流水车间调度问题可以描述如下:要将 n 个工件 $J = \{j_1, j_2, \dots, j_n\}$ 分配到 m 条相同的工厂流水线 $F = \{F_1, F_2, \dots, F_m\}$ 上,每条流水线有 r 台加工设备 $M = \{M_1, M_2, \dots, M_r\}$ 。分布式流水车间加工过程如图 1 所示。课题组研究的优化目标是缩短生产周期,即使得所有最大完工时间最小化,因此分布式流水车间调度问题可以分解为 2 个子问题:①如何使工件在多个工厂流水线中分配最为合理;②如何合理安排单个工厂流水线的工件加工顺序。

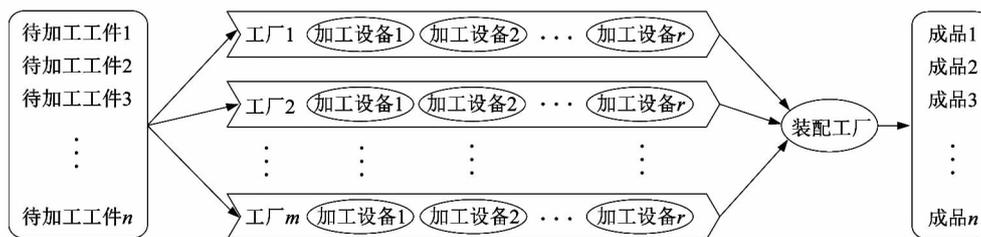


图 1 分布式流水车间加工过程

Figure 1 Distributed flow shop process

1.2 建立数学模型

为了更好地建立模型求解分布式流水车间调度问题,作出以下假设:

1) 同质性假设。假设各车间和机器是同质的,即它们在性能和处理能力上没有差异。

2) 无中断假设。假设在工件加工和机器维护过程中没有中断。一旦工件开始加工,就不能中断加工过程,直到完成;同样,机器的维护过程也不能中断。

3) 工序时间确定性假设。假设每个工件在每个机器上的加工时间是确定的,即不存在变动的加工

时间。

4) 工序顺序假设。假设每个工件的工序顺序是固定的,即工件必须按照预定的顺序在不同的机器上进行加工。

5) 设备稳定性假设。假设设备的稳定程度在加工每个工件之前都是相同的,并且在维护后能够迅速恢复到最大稳定程度。

6) 调度周期假设。假设调度是按照周期性进行的,即在每个调度周期内,所有车间和机器都执行相同的调度计划。

7) 无限制的物料和能量假设。假设在生产过程中不存在物料和能量的限制。

1.2.1 参数定义

相关参数及符号定义如下: n 为工件总数; m 为工厂总数; r 为加工设备总数; l 为工件所在加工设备上的索引标识符; J 为工件集合, $J = \{j_1, j_2, \dots, j_n\}$; F 为工厂集合, $F = \{F_1, F_2, \dots, F_m\}$; M 为加工设备集合, $M = \{M_1, M_2, \dots, M_r\}$; n_q 为在第 q 个工厂中含有的工件数量; $d_{(o,p,q)}$ 为工件 o 在第 q 个工厂的第 p 台加工设备的加工时长; $C_{(o,p,q)}$ 为工件 o 在第 q 个工厂的第 p 台加工设备的完成时间; C_{\max} 为所有工件加工完成的最大完工时间; T_{total} 为所有的工件加工完成后的总时间; $X_{(q,o,l)}$ 为决策变量,判定 l 位置工件 o 是否在工厂 q 处理; $Y_{(q,o)}$ 为决策变量,判定工件 o 是否在工厂 q 处理。

1.2.2 数学模型

建立求解分布式流水车间调度问题数学模型。

决策变量:

$$X_{(q,o,l)} = \begin{cases} 1, & \text{工件 } o \text{ 在 } l \text{ 位置上;} \\ 0, & \text{其他。} \end{cases} \quad (1)$$

$$Y_{(q,o)} = \begin{cases} 1, & \text{工件 } o \text{ 在 } q \text{ 厂加工;} \\ 0, & \text{其他。} \end{cases} \quad (2)$$

目标函数:

$$f = \min (C_{\max})。 \quad (3)$$

约束条件:

$$C_{(o,p-1,q)} \leq C_{(o,p,q)} - d_{(o,p,q)} \circ \quad (4)$$

$$C_{(o-1,p,q)} \leq C_{(o,p,q)} - d_{(o,p,q)} \circ \quad (5)$$

$$C_{\max} \geq C_{(o,p,q)} \circ \quad (6)$$

$$C_{(1,1,q)} = d_{(1,1,q)}, q = 1, 2, \dots, m。 \quad (7)$$

$$C_{(o+1,1,q)} = C_{(o,1,q)} + d_{(o+1,1,q)} \circ \quad (8)$$

式中: $o = 1, 2, \dots, n-1; q = 1, 2, \dots, m。$

$$C_{(1,p+1,q)} = C_{(1,p,q)} + d_{(1,p+1,q)} \circ \quad (9)$$

式中: $p = 1, 2, \dots, r-1; q = 1, 2, \dots, m。$

$$C_{(o,p,q)} = \max \{ C_{(o,p-1,q)}, C_{(o-1,p,q)} \} + d_{(o,p,q)} \circ \quad (10)$$

式中: $o = 2, \dots, n; p = 2, \dots, r; q = 1, 2, \dots, m。$

$$C_{\max} = \max \{ C_{(n_q,r,q)} \}, q = 1, 2, \dots, m。 \quad (11)$$

$$T_{\text{total}} = \max \left\{ \sum_{o=1}^{n_q} C_{(o,r,q)} \right\}, q = 1, 2, \dots, m。 \quad (12)$$

式(3)是目标函数,即优化最小最大完工时间;约

束式(4)和约束式(5)表示工件和加工设备都空闲时可以开始加工,也就是说该工件的上一步加工工序已经完成,并且加工设备上没有其他工件在进行加工;约束式(6)表示最后加工完毕所需时间要大于等于全部的工件加工的完成时间;式(7)的含义是要在加工设备1上加工每个工件的第1步;式(8)用来计算每个工厂的第1台加工设备上的工件的加工完成时间;式(9)用来计算工件1在每台加工设备上的加工完成时间;式(10)可以计算接下来每个工件的每一道工序所用的加工时长;式(11)计算工件的最后完成时间、式(12)计算所有工件的总完成时间,同时式(11)和式(12)可以用来衡量目标函数的优劣。

2 深度学习求解 DFSP

在求解分布式流水车间调度问题时,可以使用多种深度学习算法,比较常见的求解算法有以下4种:

1) 深度强化学习(deep reinforcement learning, DRL)^[16]。DRL结合了深度学习和强化学习,使用神经网络来表示智能体的策略或值函数,用于学习复杂的车间状态和调度决策,通过神经网络处理高维的状态信息,提高对系统的建模能力和处理能力,例如可以使用深度Q网络(deep Q-network, DQN)来学习状态-动作值函数,或使用深度确定性策略梯度(deep deterministic policy gradient, DDPG)来学习连续动作的策略。

2) 多智能体强化学习(multi-agent reinforcement learning, MARL)^[17]。MARL适用于多个智能体相互影响的场景,每个智能体都有自己的策略和目标,在分布式流水车间调度中,每个机器可以看作一个智能体,它们需要协同工作以最大化整体生产效率。MARL允许智能体之间学习合作策略,确保全局最优调度。

3) 策略梯度方法。策略梯度方法直接学习策略函数,而不是值函数,适用于连续动作空间的问题。在分布式流水车间调度中,使用近端策略优化(proximal policy optimization, PPO)来学习每个智能体的调度策略,可以更为有效地处理连续的工艺调度决策^[18]。

4) 强化学习与深度学习的结合。在车间调度问题中,可以将Q-learning融入深度神经网络中以解决决策问题。在深度神经网络的结构中,将Q-learning

算法集成到训练过程中,具体来说就是将深度神经网络的输出层设计为值函数或策略函数的逼近器,通过在训练过程中使用 Q-learning 算法逐步优化神经网络的参数,以使其能够更准确地估计状态值或选择最优策略。例如使用卷积神经网络(convolutional neural network, CNN)处理图像状态^[19],可以更好地捕捉车间状态的复杂性;使用循环神经网络(recurrent neural network, RNN)处理序列数据,能够考虑到作业之间的时序关系^[20]。神经网络相邻的输入、输出层的神经元都是由上至下连接的网络,如图 2 所示。隐藏层可以有多个,每个隐藏层包含多个节点,每个节点与前一层的节点相连接,每个连接都有一个权重,表示了前一层节点对后一层节点的影响程度,通过激活函数处理加权和。隐藏层节点数和层数是神经网络结构的一部分,可以根据任务和性能需求进行调整。

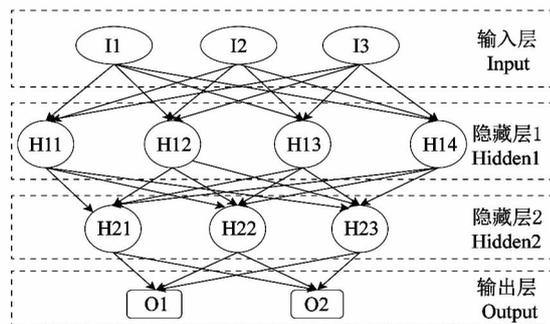


图 2 神经网络模型
Figure 2 Neural network

2.1 基于深度学习求解 DFSP 问题的策略梯度总体架构

采用策略梯度求解 DFSP 问题的总体框架如图 3 所示。课题组通过优化策略来寻找最优的决策策略使得系统在不同状态下能够做出能最大化累积奖励的动作,以解决复杂的、多变的分布式流水车间调度问题,求出最优解。

如图 3 所示,先定义状态(State),每个工序都视为一种状态,其次依据调度规则,将选取工序和加工设备定义为动作(Action),然后再依据算法获取奖励(Reward)。在本框架中,首先进行策略初始化,也就是初始化参数,再通过更新迭代等优化与改进 N 次。初始化状态在每次迭代开始前,每次迭代都要调度 H

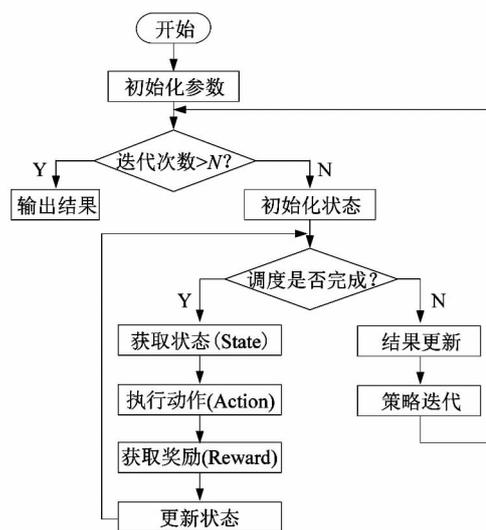


图 3 策略梯度求解 DFSP 问题的总体框架
Figure 3 Overall framework of strategy gradient solving DFSP problem

次。在每次调度之前,系统需要获取当前的状态信息,基于当前状态,使用预先定义的策略选择一个动作,即确定一个调度规则;接着根据选择的动作,执行调度操作;执行调度后,系统会获得一个奖励。这个过程在每次迭代回合内会重复进行多次,直到完成所有工序的调度。在每个迭代回合结束时,采用深度强化学习算法对策略进行更新,以提高在不同状态下做出决策的性能。这种循环迭代的过程有助于系统逐渐学习和优化调度策略。

2.2 算法设计

基于近端策略优化(PPO)算法求解 DFSP 问题的算法思想如图 4 所示。

基于 PPO 算法求解 DFSP 问题的伪代码如下:

输入:DFSP 实例

输出:最小化最大完工时间。

- 1: 初始化状态-动作表, $Q(s, a)$ 的值为 0
- 2: 初始化 $\min(C_{\max})$
- 3: for episode 1 to N (N 为迭代次数) do
- 4: 状态 State 初始化为“0”
- 5: 状态-动作集初始化为空
- 6: for $t = 1$ to H (H 为状态总数) do
- 7: 根据策略 $\pi(a|s)$ 与状态-动作表选择动作 a
- 8: 动作 a 的调度执行

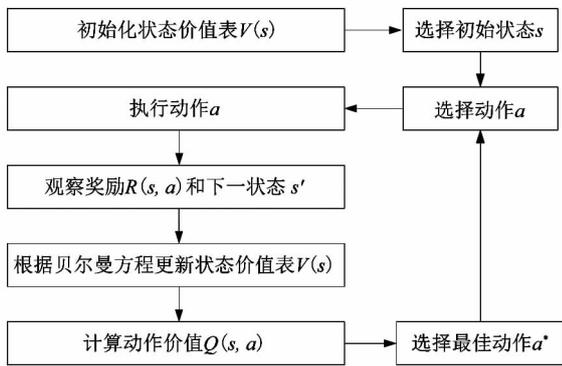


图 4 基于近端策略优化算法求解 DFSP 问题的算法思想
Figure 4 Idea of solving DESP problem based on PPO algorithm

```

9:      把当前状态-动作加入状态-动作集
10:     更新状态
11:     end for
12:     计算奖励 R, 更新 min (C_max)
13:     for Q(s, a) in 状态-动作集 do
14:       Q_pi(s, a) = Q_pi(s, a) + R
15:     end for
16: end for
17: return min(C_max)
  
```

算法的第 1、2 行是初始化参数,第 3 行表示迭代开始,第 4 行是状态初始化,第 5 行是初始化状态-动作集,第 6 行是调度开始,第 7 行根据策略 $\pi(a|s)$ 与状态-动作表选择动作,第 6—11 行表示完成一次调度,第 3—15 行表示整个的迭代过程。

2.2.1 建立状态价值表

状态价值 $V(s)$ 表示在状态 s 下的长期回报期望。

2.2.2 定义动作价值函数

动作价值函数 (Q -function) 表示在特定状态下采取某个动作后的长期回报期望,通过状态价值表,可以轻松计算动作价值。对于状态 s 和动作 a ,动作价值为:

$$Q(s, a) = V(s')$$

式中, s' 是采取动作 a 后的下一个状态。

在强化学习的训练过程中,状态价值表会不断更新,更新使用贝尔曼方程 (Bellman equation) 来进行,

该方程描述了状态价值之间的关系,对于状态 s ,更新规则可以表示为:

$$\max_a (R(s, a) + \gamma \sum_{s'} P(s' | s, a) V(s')) \rightarrow V(s)$$

式中: $R(s, a)$ 是在状态 s 采取动作 a 后的即时奖励; $P(s' | s, a)$ 是从状态 s 采取动作 a 后转移到下一个状态 s' 的概率; γ 是折扣因子。

2.2.3 计算动作价值

通过状态价值表,可以计算每个状态下所有可能动作的动作价值。

2.2.4 利用动作价值进行决策

在决策时,可以基于动作价值来选择最佳动作,动作价值的函数有:

$$Q(s, a) = R(s, a) + \gamma \sum_{s'} P(s' | s, a) V(s')$$

最后进行价值决策,最佳动作 a^* 有:

$$a^* = \arg \max_a Q(s, a)$$

通过维护状态价值表,可以在强化学习中实现值迭代算法,使代理能够学习到在不同状态下采取不同动作的长期效果。

3 实验与分析

3.1 实验设备

采用了 Windows 11 操作系统作为实验平台的基础。实验运行环境主要依赖于—台主频为 3.2 GHz 的 Intel Core i7 处理器;GPU 为 RTX 3060,满足图形处理相关的计算任务要求;系统内存为 16 GiB 的 RAM,满足实验中的内存需求;同时使用了 MATLAB 2019b 作为实验的运行环境进行数值计算和实验数据的分析,并利用 Origin 2018 进行主要数据的分析与相关图表的绘制。

3.2 实验设计

实验的原始数据集源自于 OR-Library 的 Scheduling 下的 Flow shop,其中有各类调度问题的标准测试数据集,该数据集包含了一系列 Flow shop 调度问题的实例,每个实例包括了作业数、工作站数以及每个作业在每个工作站上的处理时间等信息,这些数据被广泛用于研究和解决调度问题,涵盖了不同规模和复杂度的实例,可以用于评估和比较不同调度算法的性能。

为了验证 PPO 算法的有效性,课题组设计了多种

情况下的数据集进行测试,工厂流水线数分别为 3,4 和 5,对应的工件数分别为 30,60 和 80,对应的加工设备台数分别为 8,10 和 12,对应的工序道数分别为 5,10 和 15。设置了不同的奖励系数来评估模型的性能,奖励系数分别为: $r_1 = 0.1$; $r_2 = 0.3$; $r_3 = 0.5$ 。这些奖励系数影响着模型对于不同目标的优化程度,可以帮助我们了解 PPO 算法在不同激励下的表现。

针对不同规模的问题训练了不同的模型,针对不同的模型各生成数量不等的数据集进行深度学习训练,为

了保证实验的充分性,将每个实验模型训练的次数设定为 20 000,确保了模型在不同数据集和参数设置下得到充分的训练和学习,以便进行有效的性能评估。

3.3 结果分析

在本次实验中,课题组主要采用 PPO 算法进行求解,同时以 Q-learning, DQN 和 DDPG 作为对照算法进行比较。在相同的实验条件下,经过多次实验,得到了各算法求解后的最小最大完工时间,具体数据如表 1 所示。

表 1 PPO 算法及其他深度学习算法实验结果比较

Table 1 Comparison of experimental results between PPO algorithm and other deep learning algorithms

组别	流水线数	工件数	设备数	工序数	最小最大完工时间 $\min(C_{\max})/s$			
					PPO	Q-learning	DQN	DDPG
1	3	30	8	5	866	898	872	889
2	3	30	8	5	901	1 006	997	873
3	3	30	8	5	799	879	864	813
4	3	30	8	5	748	961	822	992
5	3	30	8	5	957	1 197	924	977
6	4	60	10	10	1 749	2 003	1 974	1 863
7	4	60	10	10	1 822	2 356	1 803	2 164
8	4	60	10	10	1 689	2 147	1 994	2 466
9	4	60	10	10	1 633	2 641	1 877	1 733
10	4	60	10	10	2 313	2 339	2 330	2 326
11	5	80	12	15	3 384	3 567	3 488	3 747
12	5	80	12	15	3 755	4 479	4 352	3 996
13	5	80	12	15	3 667	5 057	4 007	5 253
14	5	80	12	15	4 102	7 753	3 989	5 561
15	5	80	12	15	3 995	5 197	5 026	4 369

由表 1 可以看出:在这 15 组算例中,PPO 算法求解得到的最小最大完工时间有 11 组比其他算法得到的值更小,DQN 算法和 DDPG 算法在表现上各有优劣;总体相较于 DQN 算法、DDPG 算法和 Q-learning 算法,PPO 算法表现较优。因此,在解决 DFSP 问题的众多算法中,近端策略优化算法有较强的竞争力,课题组选择近端梯度优化算法而非其他几种深度强化学习算法。

3.4 收敛性验证

为了验证算法的收敛性,课题组以表 1 中第 1 组和第 10 组算例为例进行收敛验证,得到的 PPO 算法、Q-learning 算法、DQN 算法和 DDPG 算法对比的迭代次数与最小最大完工时间的关系,如图 5 和图 6 所示。

从图中可以看到:这 4 种算法的最小最大完工时间都随着迭代次数增加而减小,说明收敛效果都不错;PPO 算法的最小最大完工时间明显小于其他几种算法,且收敛速度也相对较快,因此课题组提出的近端策略优化 PPO 算法及初始化条件是有效的。

4 结论

课题组提出了一种近端策略优化的深度学习算法,用来求解分布式流水车间调度问题,以最小最大完工时间为优化目标。近端策略优化算法能够直接优化策略,而非依赖于对问题的精确建模,在分布式流水车间背景下,这意味着近端策略优化算法能够灵活地适应各种车间之间的差异,同时处理不确定性和变化,这种灵活性使得算法能够在实时调度和动态环境中表现

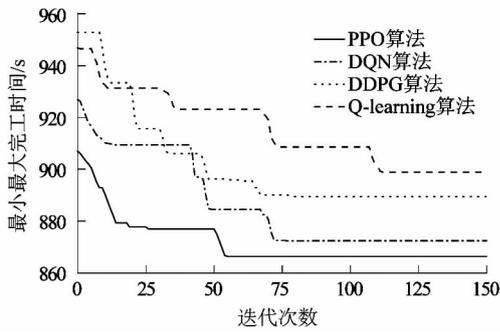


图 5 第 1 组算例的 4 种算法最小最大完工时间迭代曲线

Figure 5 Minimum and maximum completion time iteration curves of four algorithms in first group examples

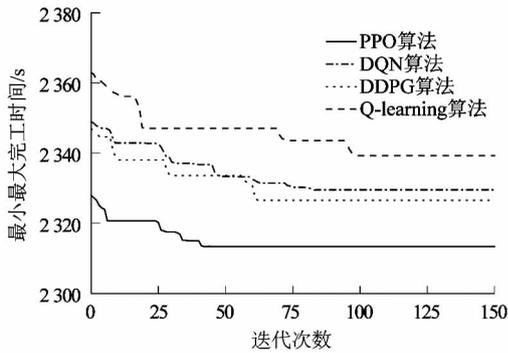


图 6 第 10 组算例的 4 种算法最小最大完工时间迭代曲线

Figure 6 Minimum and maximum completion time iteration curves of four algorithms in tenth group examples

出色,从而更好地应对制造过程中的挑战。在 PPO 算法与其他深度强化学习算法求解 DFSP 问题的对比中,可以看到 PPO 算法在相同条件下效率有了较大提升,同时收敛性也较好,但是在求解质量上并不显著,可以在后续融入其他算法进一步研究。

总之,在解决分布式流水车间调度问题上近端策略优化算法较为有效,同时这种方法可以为制造业提供了一种创新的、适应性强的解决方案,随着对这一方法的深入研究和不断优化,将看到更多基于策略梯度算法的前沿应用,为提高制造业的生产效率和智能化水平贡献力量。

参考文献:

[1] 张静,宋洪波,林剑. 分布式装配换流水车间调度问题研究综述[J]. 计算机工程与应用,2024,60(6):1-9.

[2] NADERI B,RUIZ R. The distributed permutation flowshop scheduling problem[J]. Computer & Operation Research,2010,37:754-768.

[3] 王凌,邓瑾,王圣尧. 分布式车间调度优化算法研究综述[J]. 控制与决策,2016,31(1):1-11.

[4] RUIZ R,PAN Q K,NADERI B. Iterated Greedy methods for the distributed permutation flowshop scheduling problem [J]. Omega, 2018,83(1):213-222.

[5] LIN S W,YING K C. Minimizing makespan for solving the distributed no-wait flow shop scheduling problem [J]. Computers & Industrial Engineering,2016,99:202-209.

[6] SAUER J,SUELMANN G,APPELRATH H J. Multi-site scheduling with fuzzy concepts [J]. International Journal of Approximate Reasoning,1998,19(1/2):145-160.

[7] NADERI B,AZAB A. Modeling and heuristics for scheduling of distributed job shops[J]. Expert Systems with Applications,2014,41(17):7754-7763.

[8] 杨晓林,胡蓉,钱斌,等. 增强分布估计算法求解低碳分布式流水线调度[J]. 控制理论与应用,2019,36(5):803-815.

[9] WANG J J,WANG L. A knowledge-based cooperative algorithm for energy-efficient scheduling of distributed flow-shop [J]. IEEE Transactions on Systems Man and Cybernetics Systems,2018,50(5):1805-1819.

[10] SHAO W S,SHAO Z S,PI D C. A multi-neighborhood-based multi-objective memetic algorithm for the energy-efficient distributed flexible flow shop scheduling problem [J]. Neural Computing and Applications,2022,34(24):22303-22330.

[11] 杨媛媛,胡蓉,钱斌,等. 深度强化学习算法求解动态流水车间实时调度问题[J/OL]. 控制理论与应用:1-9[2024-01-15]. <http://kns.cnki.net/kcms/detail/44.1240.TP.20231122.1359.012.html>.

[12] ZHANG X,LI T X,YIN H M. An enhanced genetic algorithm for the distributed assembly permutation flowshop scheduling problem[J]. International Journal of Bio-Inspired Computation,2020,15(2):113-124.

[13] HUANG J P,PAN Q K,MIAO Z H, et al. Effective constructive heuristics and discrete bee colony optimization for distributed flowshop with setup times [J]. Engineering Applications of Artificial Intelligence,2021,97:104016.

[14] RIBAS I,COMPANYS R,TORT-MARTORELL X. An iterated greedy algorithm for the parallel blocking flow shop scheduling problem and sequence-dependent setup times [J]. Expert Systems with Applications,2021,184:115535.

- [15] 郎峻,顾幸生.多目标协同正弦优化算法求解分布式流水车间调度问题[J/OL].控制理论与应用;1-8[2024-01-15].http://kns.cnki.net/kcms/detail/44.1240.tp.20230324.0837.002.html.
- [16] WANG X R, LI T K, WANG B L, et al. DQN-based online scheduling algorithm for hybrid flow shop to minimize the total tardiness[C]//2022 15th International Symposium on Computational Intelligence and Design (ISCID). Hangzhou:IEEE,2022;66-69.
- [17] BAO H Z, PAN Q K, RONG M, et al. A Multi-action reinforcement learning algorithm for energy-efficiency blocking flow-shop scheduling problem[C]//2023 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Rio de Janeiro, Brazil:IEEE,2023;492-497.
- [18] 徐宁.基于强化学习的柔性作业车间调度算法研究[D].上海:华东师范大学,2022;10-16.
- [19] 李兴洲,李艳武,谢辉.基于CNN的深度强化学习算法求解柔性作业车间调度问题[J/OL].计算机工程与应用;1-12[2024-01-15].http://kns.cnki.net/kcms/detail/11.2127.TP.20230822.1102.002.html.
- [20] 任剑锋,叶春明.嵌入指针网络的深度循环神经网络模型求解作业车间调度问题[J].计算机应用研究,2021,38(1):120-124.

(上接第91页)

- [3] 路凯华,耿斌斌,何立东,等.基于ISFD的滑动轴承转子系统不平衡振动抑制研究[J].机电工程,2023,40(11):1768-1775.
- [4] 李翁衡,祝长生.主动电磁轴承-柔性转子系统过多个临界转速区的同频振动抑制[J].中国电机工程学报,2023,43(14):5653-5663. DOI:10.13334/j.0258-8013.pcsee.220647.
- [5] 金福艺,臧朝平,邢广鹏,等.基于变刚度支承的转子系统振动控制策略[J/OL].上海交通大学学报,1-19[2023-12-17].https://doi.org/10.16183/j.cnki.jsjtu.2023.248.
- [6] ALEYAASIN M. Simply structured controllers for vibration suppression in long rotors[J]. International Journal of Dynamics and Control,2022,10(1):122-137.
- [7] CHEN Y, CUI J W, SUN X. A vibration suppression method for the multistage rotor of an aero-engine based on assembly optimization[J]. Machines,2021,9(9):189.
- [8] JUNGBLUT J, HAAS J, RINDERKNECHT S. Active vibration control of an elastic rotor by using its deformation as controlled variable[J]. Mechanical Systems and Signal Processing,2022,165:108371.
- [9] ISHIDA Y, INOUE T. Vibration suppression of nonlinear rotor systems using a dynamic damper[J]. Journal of Vibration and Control,2007,13(8):1127-1143.
- [10] 徐庆华.试采用FFT方法实现加速度、速度与位移的相互转换[J].振动、测试与诊断,1997,17(4):30-34. DOI:10.16450/j.cnki.issn.1004-6801.1997.04.005.
- [11] 陈琨.低通数字滤波器的MATLAB实现[J].扬州职业大学学报,1999,3(2):21-24.

(上接第99页)

- [12] JIANG P Y, ERGU D, LIU F Y, et al. A review of YOLO algorithm developments[J]. Procedia Computer Science, 2022, 199:1066-1073.
- [13] SHAFIEE M J, CHYWL B, LI F, et al. Fast YOLO: a fast you only look once system for real-time embedded object detection in video[J]. Journal of Computational Vision and Imaging System, 2017, 3(1):171.
- [14] LAROCA R, SEVERO E, ZANLORENSI L A, et al. A robust real-time automatic license plate recognition based on the YOLO detector[C]//2018 International Joint Conference on Neural Networks (IJCNN). Rio de Janeiro, Brazil:IEEE,2018;1-10.
- [15] ZHU L, WANG X J, KE Z H, et al. BiFormer: vision transformer with bi-level routing attention[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Vancouver:IEEE,2023;10323-10333.
- [16] TAN M X, PANG R M, LE Q V. Efficientdet: scalable and efficient object detection[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle:IEEE,2020;10781-10790.
- [17] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition. Alaska:IEEE,2016;770-778.
- [18] LIN T Y, DOLLAR P, GIRSHICK R, et al. Feature pyramid networks for object detection[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition. Hawaii:IEEE,2017;2117-2125.
- [19] LIU S, QI L, QIN H F, et al. Path aggregation network for instance segmentation[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City:IEEE,2018;8759-8768.