计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2024,33(3):281−287 [doi: 10.15888/j.cnki.csa.009454]
©中国科学院软件研究所版权所有.
E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于上下文赌博机的自适应实时车间调度①

陈　鸣, 王　闯, 许　政

(中航机载系统共性技术有限公司 预先研究与工程应用研究室, 扬州 225006)
通信作者: 陈　鸣, E-mail: mingjstz@163.com

摘　要: 传统的多 Agent 车间调度方法使用单一调度规则, 忽略了生产环境变化对调度规则适用性的影响, 导致调度结果欠佳. 本文针对该问题提出一种自适应实时车间调度方法, 通过上下文赌博机对工件调度过程进行类比建模. 经过若干回合学习的上下文赌博机模型能够依据生产环境制定调度决策, 获得优异的调度结果. 最后, 通过仿真实验验证了提出方法的有效性.

关键词: 多 Agent 系统; 上下文赌博机; 车间调度; 自适应调度规则

引用格式: 陈鸣,王闯,许政.基于上下文赌博机的自适应实时车间调度.计算机系统应用,2024,33(3):281–287. http://www.c-s-a.org.cn/1003-3254/9454.html

Adaptive Real-time Workshop Scheduling Based on Contextual Bandits

CHEN Ming, WANG Chuang, XU Zheng

(Pre Research and Engineering Application Lab, AVICAS Generic Technology Co. Ltd., Yangzhou 225006, China)

Abstract: The traditional multi-agent workshop scheduling method uses a single scheduling rule, ignoring the influence of production environment changes on the applicability of scheduling rules and resulting in poor scheduling results. This study proposes an adaptive real-time workshop scheduling method to model the workpiece scheduling process by analogy through the contextual bandits. After several rounds of learning, the contextual bandit model can make scheduling decisions according to the production environment and obtain excellent scheduling results. Finally, simulation experiments verify the effectiveness of the proposed method.

Key words: multi-agent system; contextual bandits; workshop scheduling; adaptive scheduling rule

柔性作业车间调度问题 (flexible job shop scheduling problem, FJSP) 是制造企业实际生产中经常遇到的一类问题, 它是作业车间调度问题 (JSP) 的扩展[1], 去除了各工序存在唯一指定机器的限制, 考虑了工序的加工柔性, 更能适应当前动态的车间环境[2]. 在过去的几十年中, 专家学者对 FJSP 进行了大量的研究, 提出了许多有效的解决方案[3]. 这些解决方案主要可以分为集中式的调度方法和分布式多 Agent 的调度方法.

在复杂的实际作业生产车间中, 集中式的调度方法, 如遗传算法 (genetic algorithm, GA)、模拟退火算法 (simulated annealing, SA)、禁忌搜索算法 (tabu search, TS) 等由于其计算时间长, 往往无法很好地应用[4]. 多 Agent 方法由于其实时性好、处理扰动能力强, 能够很好地解决实际车间中的调度问题[5]. 然而, 传统的多 Agent 方法一般采用单一的调度规则, 忽略了调度过程中的环境变化影响, 存在 Agent 自适应能力和调度优化效果较差的缺点.

为了克服多 Agent 方法的这些缺点, 许多文献对其进行了研究. 郭剑等[6]提出了一种 SP-MCTS 搜索方法, 利用马尔科夫决策过程对生产调度过程进行建模, 实现了自适应调度, 提高了调度的性能, 但仅适用于混流车间. 刘亚辉等[7]提出了一种双环深度 Q 网络方法,

① 收稿时间: 2023-08-24; 修改时间: 2023-09-26; 采用时间: 2023-11-24; csa 在线出版时间: 2024-01-19
CNKI 网络首发时间: 2024-01-22

© 中国科学院软件研究所 http://www.c-s-a.org.cn

利用知识图谱实现对车间知识的表示, 并通过感知-认知双系统来对调度进行驱动, 但优化性能一般. 胡晓阳等[8]将调度规则与贪心算法融合, 扩大了调度结果解的搜索空间, 实现了调度结果的优化, 但牺牲了一定实时性能. 贺俊杰等[9]利用多智能体强化学习对纺织面料染色车间调度问题进行了调度优化, 克服了单一调度规则的缺点, 但仅适用于特定的并行调度问题. Qu 等[10]采用强化学习方法来根据环境状态自适应地生成调度规则, 但也仅适用于 JSP.

在某一场景下, 调度规则选择与此场景的特征密切相关, 不同的特征下需要的调度规则也是各异的. FJSP 比 JSP 更加复杂, 调度过程中的变化因素更多, 相应需要的调度规则也更为多变. 如果仅使用单一调度规则而不依据调度场景特征来个性化的选择, 那么将无法获得优异的调度方案. 本研究针对该问题, 将 FJSP 以机器选择和缓冲区工件调度两阶段来表示[11], 利用上下文赌博机 (contextual bandits) 来类比建模. 上下文赌博机在若干回合的训练后, 能够识别场景特征与所需调度规则的模式关系. 利用该模式, 工件 Agent 在不同的场景下会应用上下文赌博机模型的计算结果制定调度策略, 实现整体调度结果的优化. 这种调度方式相比原始的单一调度规则, 具有更强的自适应性, 能够满足 FJSP 复杂场景的需求.

1 问题描述

本研究以最大完工时间 (Makespan) 为优化目标. 最大完工时间是所有加工工件最后完成时刻, 反映所有加工任务的整体完成速度, 体现了车间生产效率. 对最大完工时间目标进行优化, 可以实现对车间生产效率的提高.

同时, 依据实际车间场景作出如下假设[12].

(1) 工件分批抵达生产车间, 每批抵达的工件数量是随机的.

(2) 工序加工机器不唯一, 有若干台机器可供使用.

(3) 每台机器在同一时刻仅可以加工一个工件.

(4) 准备时间包含在加工时间当中, 运输时间不予考虑.

(5) 工序一旦开始加工, 直到加工完成前都不能被打断.

1.1 符号定义

根据以上假设, 定义符号如下.

J: 所有工件组成的集合

M: 所有机器组成的集合

C_i: 工件J_i的完工时间

O_{ij}: 工件i的第j道工序

st_{ij}: 工件i的第j道工序的开始加工时间

ot_{ij}: 工件i的第j道工序的加工结束时间

sm_{mi}: 机器m加工第i道工序的开始时间

om_{mi}: 机器m加工第i道工序的结束时间

t_{ijm}: 工件i的第j道工序对应机器m的加工时间

$$X_{ijm}=\begin{cases}1, & O_{ij}\text{在机器}m\text{上加工}\\ 0, & \text{否则}\end{cases}$$

1.2 FJSP 优化模型

根据以上假设及符号定义, 可以建立 FJSP 数学模型如下.

优化指标选择 Makespan, 记为C.

$$C=\max_{1\leqslant i\leqslant N}(C_i) \tag{1}$$

目标函数为:

$$\min(C) \tag{2}$$

约束函数为:

$$\sum_{m=1}^{M}X_{ijm}=1 \tag{3}$$

$$st_{i(j+1)}\geqslant ot_{ij} \tag{4}$$

$$sm_{m(i+1)}-om_{mi}\geqslant 0 \tag{5}$$

$$ot_{ij}-st_{ij}=\sum_{m=1}^{M}(X_{ijm}\cdot t_{ijm}) \tag{6}$$

其中, 式 (3) 表示工件工序加工机器选择的唯一性约束; 式 (4) 表示工艺路线顺序约束; 式 (5) 表示工件工序加工先后顺序约束; 式 (6) 表示工序加工时间与机器加工时间需要具备一致性.

2 调度策略

2.1 自适应实时调度方法

在经济全球化的浪潮下, 市场需求变化不断加快, 生产制造企业亟需一个具有高柔性、高实时性、高敏捷性的制造系统来迎接这种挑战. 在该种需求下, 多 Agent 车间调度方法崭露头角, 以其独有的智能性、实时性、可重构性为现代制造系统应对挑战提供了可能.

© 中国科学院软件研究所 http://www.c-s-a.org.cn

然而, 传统的多 Agent 调度方法一般采用单一的调度规则, 忽略了调度过程中场景变化影响, 使得调度结果及 Agent 的自适应能力较差. 本研究通过上下文赌博机对车间调度过程建模, 利用场景特征进行规则选择, 从而改善了整体的表现.

2.1.1 上下文赌博机基本原理

上下文赌博机是一种强化学习模型, 每个回合包含一个状态, 并且能够影响即时奖励[13]. 可以将上下文赌博机模型由$\{A,S,R\}$三元素构成, 其中, 元素A表示模型具备的行为空间, 元素S表示模型具备的状态空间, 元素R表示模型所获得的奖励. 图 1 描述了上下文赌博机运行的逻辑, 在回合e, Agent 会依据此刻所处的环境s_e通过计算值从行为空间A中选择行为a_e. 当 Agent 将行为a_e作用于环境s_e后, 环境s_e将反馈给它奖励r_e. 奖励r_e是与环境状态s_e和行为a_e都相关的变量[14], 这也意味着变化的环境状态被量化为上下文信息, 以帮助在上下文相关的、高度动态的、复杂的系统中进行决策[15]. Agent 通过若干回合的学习, 可以识别到环境状态与奖励之间的模式, 利用该模式即可制定使奖励值最大的策略.

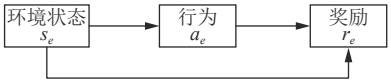

图 1 上下文赌博机基本原理

2.1.2 车间调度决策过程建模

柔性作业车间调度问题比较复杂, 涉及工件、加工工序、加工机器, 是一个三元组合优化问题并已被证明为 NP-hard. 对于该问题, 直接使用上下文赌博机模型进行优化势必会导致状态空间爆炸而无法收敛. 但是, 在采用多 Agent 车间调度方法的 FJSP 中, 工件 Agent 的决策是单步决策, 并且仅包含两个阶段: 机器选择阶段和缓冲区调度阶段. 如图 2 所示, 在每一决策回合, 工件 Agent 仅需要选择对应的机器选择规则和缓冲区调度规则, 随后便可基于该类规则实施加工进程. 在这种情形下, 上下文赌博机只需对工件 Agent 的单步决策过程进行优化, 仅涉及单道工序与多个调度规则组合的优化问题, 大大降低了状态空间维度并使得模型优化能够收敛.

车间调度决策过程建模为上下文赌博机的描述如下.

(1) 状态空间

状态空间包含从生产场景中提取的关键特征, 在 Agent 进行调度时, 上下文赌博机模型会依据这些特征实时计算每个行为收益并进行选择. 本研究提取的场景关键特征如下: 同时调度的各工序数量, 加工机器缓冲区的工件数量, 加工机器缓冲区的总加工时间, 加工机器所需加工时间.

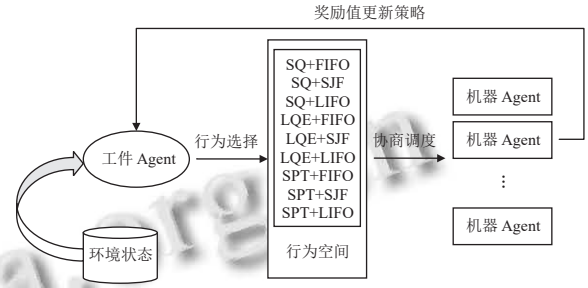

图 2 基于上下文赌博机的 Agent 调度决策过程模型

(2) 行为空间

行为空间由机器选择规则和缓冲区调度规则的组合构成. 机器选择规则包括: 最短队列优先 (shortest queue, SQ), 队列工件最少优先 (less queued element, LQE), 处理时间最短优先 (shortest processing time, SPT). 缓冲区调度规则包括: 先入先出 (first in first out, FIFO), 工件加工时间最短优先 (shortest job first, SJF), 后入先出 (last in first out, LIFO). 通过将以上各机器选择规则和缓冲区调度规则进行两两组合, 形成包含 9 种行为的自适应实时车间调度行为空间, 分别为 SQ+FIFO, SQ+SJF, SQ+LIFO, LQE+FIFO, LQE+SJF, LQE+LIFO, SPT+FIFO, SPT+SJF, SPT+LIFO. 在工件需要调度时, 上下文赌博机会基于实时环境状态从调度规则构成的行为空间中选择最合适的规则用于当前工件的调度, 从而是实现了对调度规则的个性化选择.

(3) 奖励

在 Agent 进行了行为选择后, 奖励值通过平均等待时间 (mean wait time, MWT) 变化量来计算, 计算方法见式 (7) 和式 (8). 若当前平均等待时间较决策前变小, 计算的奖励将为正值, 表示这是一个好的决策; 否则给予负奖励值, 表示这是一个坏的决策. 策略学习算法将根据奖励值不断地进行学习, 最后获得能使奖励值最大的策略. 奖励值计算公式如下:

$$MWT_t = \frac{\sum_{j=1}^{n} WT_{j,t}}{n} \tag{7}$$

$$r_t = MWT_{t-1} - MWT_t \tag{8}$$

© 中国科学院软件研究所 http://www.c-s-a.org.cn

其中, $WT_{j,t}$为t时刻工件j到完成加工还需等待的时间, n为当前工件的总数量.

2.1.3 策略学习算法 LinUCB

本研究采用 LinUCB 算法对上下文赌博机进行训练, 该算法使用一维线性模型拟合环境状态和奖励两者潜在的模式关系, 实现高准确、高泛化的预测目标. 该算法在每一回合t会计算行为集合中所有行为的期望奖励值, 并选择期望奖励值最大的行为作为 Agent 本轮的决策, 计算公式如下:

$$E\left[r_{t,a}\left|x_{t,a}\right.\right]=x_{t,a}^{\mathrm{T}}\theta_a^* \tag{9}$$

其中, $r_{t,a}$是回合t选择行为a的期望奖励值, θ_a^*是行为a的线性规划参数, $x_{t,a}\in R^d$是状态空间设定的关键特征向量.

在线性模型中, θ_a^*决定了模型预测的准确度. LinUCB 算法通过在每一回合中的学习, 可以不断地对θ_a^*进行在线优化, 使其符合当前的环境状态. 具体而言, LinUCB 算法会记录m回合的$x_{t,a}$与$r_{t,a}$, 构造历史经验矩阵$G_a\in R^{m\times d}$和$c_a\in R^m$, 然后使用岭回归[16]对当前的θ_a^*进行优化, 计算公式如下:

$$\hat{\theta}_a=\left(G_a^{\mathrm{T}}G_a+I_d\right)^{-1}b_a \tag{10}$$

其中, 为简化公式的表示, 记$b_a=G_a^{\mathrm{T}}c_a$.

此外, 为了防止陷入局部最优, LinUCB 在上述计算值的基础上设置了置信区间. 在模型进行行为选择时, 将会选择置信区间上界最大的行为. 即在回合t, 选择:

$$a_t:=\underset{a\in A}{\arg\max}\,\hat{\mu}_a=\underset{a\in A}{\arg\max}\left(x_{t,a}^{\mathrm{T}}\hat{\theta}_a+\hat{\sigma}_a\right) \tag{11}$$

其中, $\hat{\sigma}_a=\alpha\sqrt{x_{t,a}^{\mathrm{T}}A_a x_{t,a}}$, $A_a:=G_a^{\mathrm{T}}G_a+I_d$, α是控制探索程度的参数.

算法 1 介绍了 LinUCB 算法的计算流程.

2.1.4 自适应实时车间调度策略

图 3 详细描述了本研究提出的基于上下文赌博机的自适应实时车间调度方法. 其详细描述如下.

(1) 机床 Agent 和已到达的工件 Agent 在管理 Agent 处进行注册.

(2) 有工件释放工序时, 工件 Agent 向管理 Agent 提出加工请求, 管理 Agent 发送可供选择的加工机床清单.

(3) 根据设定的状态空间关键特征, 工件 Agent 使用 LinUCB 算法, 计算出当前最优的机器选择规则和缓冲区调度规则.

算法 1. 在线 LinUCB

初始化参数$\alpha\in\mathbb{R}_+$
for $e=1\to E$ **do**
  观察所有行为的状态特征向量$x_{e,a}\in R^d$, $a\in A$
  for all $a\in A$ **do**
  if a 第 1 次被选 **then**
    $A_a\leftarrow I_d$ (d 维单位矩阵)
    $b_a\leftarrow 0_{d\times 1}$ (d 维零向量)
  end if
  $\hat{\theta}_a\leftarrow A_a^{-1}b_a$
  $\hat{\mu}_a\leftarrow\hat{\theta}_a^{\mathrm{T}}x_{e,a}+\alpha\sqrt{x_{e,a}^{\mathrm{T}}A_a^{-1}x_{e,a}}$
  end for
  选择行为$a_e=\underset{a\in A}{\arg\max}\,\hat{\mu}_a$并观察真实的奖励$r_{e,a_e}$
  $A_{a_e}\leftarrow A_{a_e}+x_{e,a_e}x_{e,a_e}^{\mathrm{T}}$
  $b_{a_e}\leftarrow b_{a_e}+r_{e,a_e}x_{e,a_e}$
end for

(4) 工件 Agent 向清单中所有机床 Agent 发起招标.

(5) 机床 Agent 向工件 Agent 投标, 提供选出的机器选择规则需要的信息.

(6) 工件 Agent 根据选出的机器选择规则计算标值, 选择标值最大的机床, 并向对应的机床 Agent 发送“接受”消息.

(7) 被接受的机床 Agent 回应确认, 如果当前机床空闲则直接进行加工; 否则存入缓冲区, 并使用选出的缓冲区调度规则对自己的缓冲区进行优先级排序; 最后, 计算平均等待时间的变化值, 作为奖励值提交给工件 Agent.

(8) 工件 Agent 根据收到的奖励值, 更新对应工序算法的参数.

(9) 机床加工完成后, 如果工件完成所有加工则存入仓库并注销工件 Agent, 否则工件 Agent 释放下一道工序, 重复步骤 (2)–步骤 (8); 之后, 空闲的机床从缓冲区中选择优先级最高的工件继续加工.

3 实验与分析

本节描述了验证基于上下文赌博机的自适应实时车间调度方法有效性的仿真实验. 实验环境为: Windows 10 64 位操作系统, 主频 3.10 GHz 的 Intel Core i9-9900 CPU、128 GB 内存, 编程语言为 Python, 开发环境为 PyCharm.

© 中国科学院软件研究所 http://www.c-s-a.org.cn

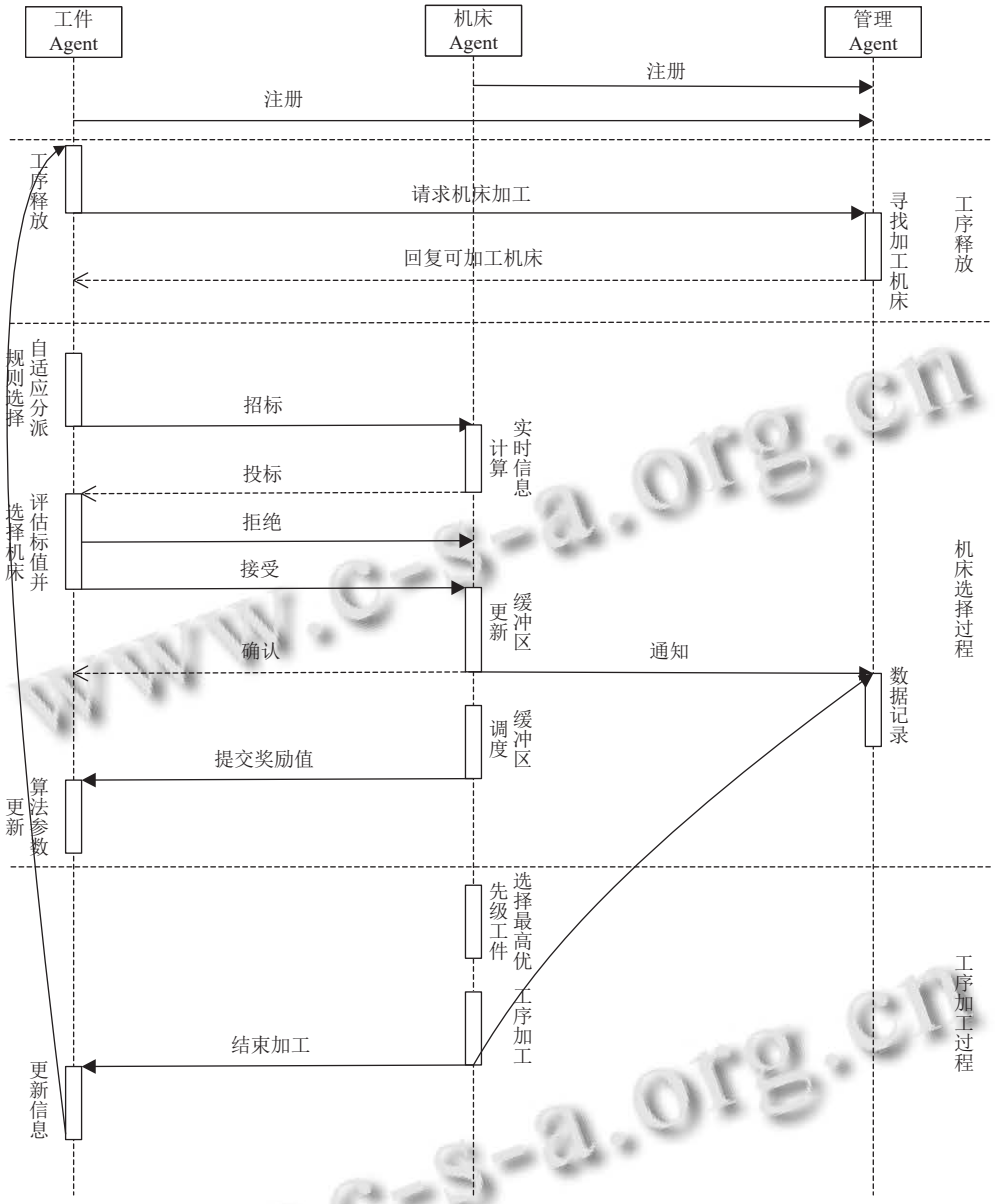

图 3 自适应调度规则实时调度策略 UML 序列图

3.1 算例设计

本研究设计的算例共包含 10 台机床, 编号为 M1 到 M10. 同时包含 10 种类型的工件, 编号为J_1到J_{10}, 详细加工信息如表 1 所示. 表 1 中的数字是工序在对应机器上的加工时间, 而“—”代表此道工序不可以在此机器上加工.

在许多离散型制造企业中, 尤其是在面向订单的离散型制造企业中, 工件通常是成批地、间断地投放到车间的[17]. 因此, 在本研究的模拟算例中, 工件成批地到来, 每批工件的数量从数值 5–10 中随机产生, 工件的类型从表 1 的 10 种工件中随机选择, 各批次工件组成如表 2 所示. 为了更好地比较调度的 Makespan 指标, 选择 5 批随机产生的工件, 到达时间间隔设定为 5, 然后采用单一调度规则和自适应调度规则分别计算完成 5 批工件加工的 Makespan 进行比较.

3.2 仿真结果与分析

针对随机产生的 5 批工件进行调度, 调度的规则包括 9 种单一的调度规则以及采用上下文赌博机的自适应实时车间调度方法. 自适应实时车间调度方法首先进行 800 个回合的学习, 图 4 显示了最大完工时间的变化趋势. 从图 4 可见, 通过不断地学习, 最大完工时间逐渐下降, 最终稳定在 127.5–130 之间. 因此, 上下

© 中国科学院软件研究所 http://www.c-s-a.org.cn

文赌博机模型使用 LinUCB 算法能够识别到环境状态与规则选择之间的映射关系, 并能够利用该模型优化调度结果.

表 1 工件加工信息

零件	工序	机床加工时间									
		M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
J_1	O_{11}	10	8	—	—	—	—	—	—	—	—
	O_{12}	—	—	—	—	—	—	—	—	5	6
	O_{13}	—	—	—	—	4	6	—	—	—	—
J_2	O_{21}	8	10	—	—	—	—	—	—	—	—
	O_{22}	—	—	—	—	—	—	—	—	4	5
	O_{23}	—	—	—	—	—	—	5	3	—	—
	O_{24}	—	—	—	—	4	6	—	—	—	—
J_3	O_{31}	12	9	—	—	—	—	—	—	—	—
	O_{32}	—	—	—	—	8	6	—	—	—	—
	O_{33}	—	—	—	—	—	—	3	6	—	—
	O_{34}	—	—	—	—	—	—	—	—	8	6
J_4	O_{41}	—	—	5	8	—	—	—	—	—	—
	O_{42}	—	—	—	—	8	6	—	—	—	—
	O_{43}	—	—	—	—	—	—	—	—	7	9
	O_{44}	—	—	—	—	—	—	4	3	—	—
J_5	O_{51}	—	—	6	8	—	—	—	—	—	—
	O_{52}	9	5	—	—	—	—	—	—	—	—
	O_{53}	—	—	—	—	2	3	—	—	—	—
	O_{54}	—	—	—	—	—	—	—	—	7	5
J_6	O_{61}	—	—	6	4	—	—	—	—	—	—
	O_{62}	—	—	—	—	10	12	—	—	—	—
	O_{63}	—	—	—	—	—	—	—	—	7	5
J_7	O_{71}	—	—	10	8	—	—	—	—	—	—
	O_{72}	4	6	—	—	—	—	—	—	—	—
	O_{73}	—	—	—	—	—	—	6	8	—	—
	O_{74}	—	—	—	—	3	7	—	—	—	—
	O_{75}	—	—	—	—	—	—	—	—	6	8
J_8	O_{81}	—	—	13	11	—	—	—	—	—	—
	O_{82}	—	—	—	—	—	—	6	8	—	—
	O_{83}	3	5	—	—	—	—	—	—	—	—
	O_{84}	—	—	—	—	—	—	—	—	6	8
J_9	O_{91}	9	6	—	—	—	—	—	—	—	—
	O_{92}	—	—	4	5	—	—	—	—	—	—
	O_{93}	—	—	—	—	—	—	8	7	—	—
	O_{94}	—	—	—	—	—	—	—	—	4	9
J_{10}	O_{101}	—	—	3	8	—	—	—	—	—	—
	O_{102}	—	—	—	—	9	4	—	—	—	—
	O_{103}	—	—	—	—	—	—	3	5	—	—
	O_{104}	—	—	—	—	—	—	—	—	4	9

表 2 仿真实验算例

批号	包含的工件类型
批号 0	$J_1, J_2, J_3, J_4, J_5, J_6, J_7, J_8, J_9, J_{10}$
批号 1	$J_1, J_2, J_3, J_4, J_5, J_6, J_7, J_8$
批号 2	$J_1, J_2, J_3, J_4, J_5, J_6, J_9, J_{10}$
批号 3	$J_1, J_2, J_5, J_6, J_9, J_{10}$
批号 4	$J_3, J_4, J_5, J_6, J_7, J_8, J_9, J_{10}$

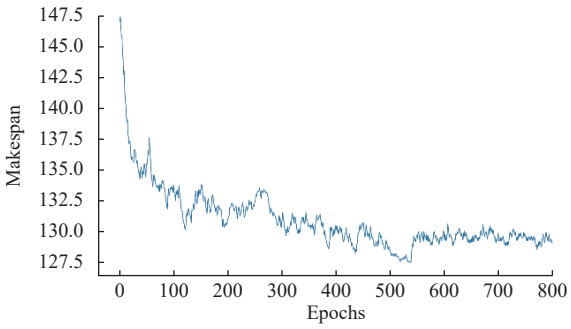

图 4 最大完工时间训练结果

自适应实时车间调度方法的结果采用平均值计算, 并与 9 种单一调度规则方法获得的调度结果进行对比. 由图 5 可见, 采用本研究基于上下文赌博机的自适应实时车间调度方法能够获得比所有单一调度规则方法更好的调度结果. 可见, 利用上下文赌博机模型在不同环境下选择不同的调度规则, 比使用单一调度规则更优. 该结果证明了采用自适应实时车间调度方法的有效性.

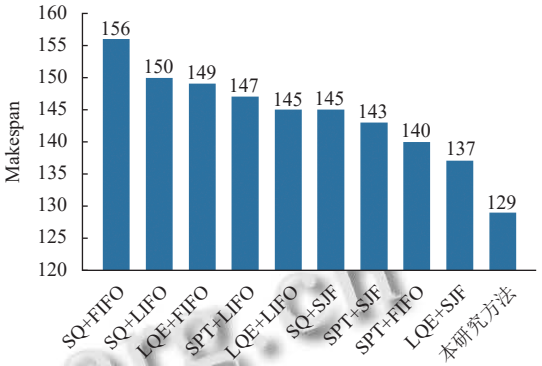

图 5 各调度方法结果比较

在整个仿真实验中, 当工件需要生产调度时, 上下文赌博机会基于生产车间环境特征从调度规则库中选择合适的调度规则, 从而使得不同调度时刻所使用的调度规则不同. 图 6 对各种调度规则的使用频次进行了统计, 显示了在采用自适应实时车间调度方法时各种调度规则被选取的分布, 与采用单一调度规则方法不同, 每种调度规则都在生产调度过程中被选用. 可见, 规则集合中的各元素在上下文赌博机模型的决策下, 适应了动态变化的环境状态, 最终实现了性能更加优秀的调度结果.

4 结论

传统的多 Agent 车间调度方法使用单一调度规则, 难以适应复杂 FJSP 的动态变化, 引起调度结果的欠优

© 中国科学院软件研究所 http://www.c-s-a.org.cn

化. 本研究针对该问题提出了一种新的自适应实时车间调度方法, 采用强化学习中上下文赌博机对工件的调度过程进行类比建模, 并使用 LinUCB 算法进行策略学习, 使得工件 Agent 能够根据环境状态自主选择调度规则, 增强了工件 Agent 的自适应能力, 获得了更好的优化性能.

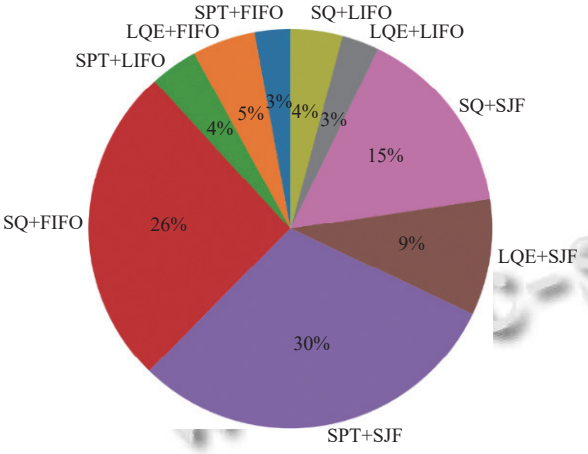

图 6 自适应实时车间调度方法调度规则选取分布

参考文献

1 Pezzella F, Morganti G, Ciaschetti G. A genetic algorithm for the flexible job-shop scheduling problem. Computers & Operations Research, 2008, 35(10): 3202–3212.

2 董蓉, 何卫平. 求解 FJSP 的混合遗传—蚁群算法. 计算机集成制造系统, 2012, 18(11): 2492–2501.

3 Cinar D, Topcu YI, Oliveira JA. A taxonomy for the flexible job shop scheduling problem. Proceedings of the 2014 Optimization, Control, and Applications in the Information Age. Cham: Springer, 2014. 17–37.

4 Joo CM, Kim BS. Hybrid genetic algorithms with dispatching rules for unrelated parallel machine scheduling with setup time and production availability. Computers & Industrial Engineering, 2015, 85: 102–109.

5 Rajabinasab A, Mansour S. Dynamic flexible job shop scheduling with alternative process plans: An agent-based approach. The International Journal of Advanced Manufacturing Technology, 2011, 54(9–12): 1091–1107. [doi: 10.1007/s00170-010-2986-7]

6 郭剑, 史耀耀, 胡昊, 等. 基于工业物联网的混流车间机器人自适应调度. 航空制造技术, 2021, 64(5): 42–51.

7 刘亚辉, 申兴旺, 顾星海, 等. 面向柔性作业车间动态调度的双系统强化学习方法. 上海交通大学学报, 2022, 56(9): 1262–1275.

8 胡晓阳, 姚锡凡, 黄鹏, 等. 改进迭代局部搜索算法求解多 AGV 柔性作业车间调度问题. 计算机集成制造系统, 2022, 28(7): 2198–2212.

9 贺俊杰, 张洁, 张朋, 等. 基于多智能体强化学习的纺织面料染色车间动态调度方法. 计算机集成制造系统, 2023, 29(1): 61–74.

10 Qu SH, Wang J, Shivani G. Learning adaptive dispatching rules for a manufacturing process system by using reinforcement learning approach. Proceedings of the 21st IEEE International Conference on Emerging Technologies and Factory Automation. Berlin: IEEE, 2016. 1–8.

11 Trentesaux D, Pach C, Bekrar A, *et al*. Benchmarking flexible job-shop scheduling and control systems. Control Engineering Practice, 2013, 21(9): 1204–1225. [doi: 10.1016/j.conengprac.2013.05.004]

12 Nouiri M, Bekrar A, Jemai A, *et al*. An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem. Journal of Intelligent Manufacturing, 2018, 29(3): 603–615. [doi: 10.1007/s10845-015-1039-3]

13 Li XB, Liu JJ, Yan L, *et al*. Relay selection in underwater acoustic cooperative networks: A contextual bandit approach. IEEE Communications Letters, 2017, 21(2): 382–385. [doi: 10.1109/LCOMM.2016.2625300]

14 Kacem I, Hammadi S, Borne P. Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews), 2002, 32(1): 1–13. [doi: 10.1109/TSMCC.2002.1009117]

15 Schulz E, Konstantinidis E, Speekenbrink M. Putting bandits into context: How function learning supports decision making. Journal of Experimental Psychology: Learning, Memory, and Cognition, 2018, 44(6): 927–943.

16 Li LH, Chu W, Langford J, *et al*. A contextual-bandit approach to personalized news article recommendation. Proceedings of the 19th International Conference on World Wide Web. Raleigh: ACM, 2010. 661–670.

17 熊禾根, 李建军, 孔建益, 等. 考虑工序相关性的动态 job shop 调度问题启发式算法. 机械工程学报, 2006, 42(8): 50–55. [doi: 10.3321/j.issn:0577-6686.2006.08.008]

(校对责编: 牛欣悦)

© 中国科学院软件研究所 http://www.c-s-a.org.cn