

基于深度学习的绿色柔性作业车间调度研究^{*}

彭国峰 胡 彬 朱晓春

(南京工程学院自动化学院 南京 211167)

摘 要: 在绿色柔性作业车间(GFJSP)中,生产过程复杂导致生产效率低下,有效调度自动导引车(AGV)运输可以确保生产效率,又能保证成本控制。本文提出了多步深度多 Q 网络算法(multi-step D4QN)用于处理绿色车间与 AGV 调度。该方法首先通过马尔科夫决策过程(MDP)设计一个可以提供 AGV 与车间进行交互的数学框架,通过状态特征、动作空间、奖励函数实时调整决策从而实现作业与 AGV 调度。其次设计针对训练决策的算法进行优化改进,通过改进 Q 值的计算方式以及深度网络的训练得到合适解。最后进行两种验证算法学习性能的实验。第 1 种实验为单目标柔性作业车间调度,以最小化完工时间为目标,通过算法对 Brandimarte 算例和 Kacem 算例训练并将实验结果进行多算法对比,结果表明算法的平均时间比其他深度学习算法缩短了 5.1~17.2 s,平均最优差比率减少了 7.5%~21%,证明了算法的优越性和稳定性。第 2 种实验为多目标车间中 AGV 调度实验,以最小化完工时间和 AGV 能耗为目标,在 MK01 算例上计算 AGV 最优数量,实验表明 4 台 AGV 相较于其他数量标准化指数提升了 3%~31.8%,证明其能够更好的实现车间降本增效的效果。

关键词: 深度学习;绿色柔性作业车间;AGV 调度;马尔科夫决策过程

中图分类号: TP18;TN05 **文献标识码:** A **国家标准学科分类代码:** 520.2060

Research on green flexible job shop scheduling based on deep learning

Peng Guofeng Hu Bin Zhu Xiaochun

(School of Automation, Nanjing Institute of Technology, Nanjing 211167, China)

Abstract: In the Green Flexible Job Shop Problem (GFJSP), the complexity of production processes leads to low efficiency. Effective scheduling of Automated Guided Vehicles (AGVs) for transportation can ensure both production efficiency and cost control. This paper proposes a Multi-step Deep Double Q-Network (D4QN) algorithm to address the scheduling of green workshops and AGVs. The method first designs a mathematical framework based on the Markov Decision Process (MDP) to enable interaction between AGVs and the workshop. By adjusting the decision-making in real time through state features, action space, and reward functions, the algorithm coordinates job and AGV scheduling. Next, the algorithm for training decision-making is optimized, improving the calculation of Q-values and deep network training to obtain suitable solutions. Finally, two validation experiments are conducted to evaluate the learning performance of the proposed algorithm. The first experiment involves single-objective flexible job shop scheduling with the objective of minimizing makespan. The algorithm is trained on the Brandimarte and Kacem benchmark problems, and the results are compared with those of other deep learning algorithms. The results show that the proposed algorithm reduced the average processing time by 5.1~17.2 s and decreased the average optimal gap ratio by 7.5%~21%, demonstrating the algorithm's superiority and stability. The second experiment focuses on multi-objective AGV scheduling in a workshop, with the goals of minimizing makespan and AGV energy consumption. The optimal number of AGVs is calculated for the MK01 problem instance, with results showing that four AGVs achieved a 3%~31.8% improvement in the normalized index compared to other quantities, proving its effectiveness in reducing costs and improving efficiency in the workshop.

Keywords: deep learning; green flexible job shop; AGV scheduling; markov decision process

0 引 言

传统车间研究中往往以最小化时间作为单一目标,缺

少了现实生产环境中的能源消耗这一重要约束。随着绿色制造的提出^[1],绿色柔性作业车间(green flexible job-shop scheduling problem, GFJSP)的研究具有重大意义。针对

车间生产中自动导引车(automated guided vehicles, AGV)运输调度和加工工序调度可以视为AGV与车间的多目标车间调度。常见的GFJSP研究中,AGV调度往往会被假设为忽略不计,因此,车间研究所得结果只是工件工序加工时间,并没有工件运输的时间。然而在现实车间生产过程中,AGV、工件与加工机器三者之间相互依赖,AGV作为运输的枢纽,合理的进行AGV调度能够减少时间成本,提升生产效率。对于GFJSP中,增加AGV调度需对AGV进行参数建模及决策设计,这大大提升了研究的复杂性。

针对上述问题,邓希等^[2]提出了改进混合遗传算法,以车间加工总时间最短为目标解决多AGV调度问题。罗雄等^[3]提出无法解遗传算法,基于机器的两点交叉和基于工序的单点变异操作减少算法运行时间。Popper等^[4]提出一种深度学习算法,通过优化作业调度与AGV调度解决灵活作业车间问题。Li等^[5]提出改进的多目标麻雀搜索算法,通过训练混合整数线性多目标优化模型得出最小化制造跨度和总碳排放量的最优解。盛煜等^[6]提出了基于改进的竞争深度双Q网络的多目标任务调度算法,通过不同线性加权组合得出帕累托最优解,达到响应时间和能耗最小化。祝正宇等^[7]提出基于组合规则和强化学习的智能调度方法,利用7种特征表征车间状态,设计6种组合式规则作为动作库求解FJSP。曹红倩^[8]提出一种改进的Q-learning算法,通过引入动态选择策略解决传统Q学习算法收敛较慢的问题。Lei等^[9]提出了一个端到端的深度强化框架,以自动学习使用神经网络解决FJSP的策略。Yuan等^[10]提出了一个新的深度强化学习框架来设计状态与动作空间,从而提高其决策能力。Cai等^[11]提出一种考虑稀疏奖励问题的深度强化学习框架解决深度强化学习解决车间调度的稀疏奖励问题。Berterottiere等^[12]提出了一种使用邻域函数的元启发式方法,通过禁忌搜索框架提升FJSP的运输效率。Zhang等^[13]基于近端策略优化(proximal policy optimization, PPO)算法提出新的模型,通过将一些高级状态嵌入到车间中神经网络,使算法获取完整状态。乐明皓等^[14]利用多策略改进麻雀搜索算法,通过增强算法的全局搜索能力解决算法陷入局部最优解。Tremblet等^[15]提出了几种机器学习模型,利用机器学习模型来预测车间容量消耗。对于上述学者提出的求解GFJSP方法,多数选择对以最小化最大完工时间进行研究,而忽略了生产车间中的工件运输环节。而在考虑AGV运输的研究中,则仅仅只是在车间建模等部分添加AGV约束,忽略了现实车间生产中AGV的电量约束、能耗约束以及考虑成本的数量约束。

常见的调度求解方式有:基于规则调度^[16]和元启发式算法^[17]。但这两种方式存在陷入局部最优解、效率过低的问题,而深度强化学习通过智能体与环境的不断交互避免局部最优,并且高效的探索策略也能够显著提高训练效率。因此,本文提出一种改进深度强化学习,利用马尔科夫决策过程设计新的状态、动作与奖励,使AGV与车间能够获取

完整状态和合理动作,随后设计网络框架,通过改进后的算法做出最优决策进行多目标GFJSP问题求解,最后在基准算例上进行多算法数据对比实验,验证该方法的优越性与稳定性,达到车间降本增效的效果。

1 描述与模型建立

GFJSP不同于经典作业车间调度问题,除了对工件的工序进行排序以外,还需对工序的加工机器进行调度排序。首先是建立GFJSP的数学模型。

1.1 问题描述

GFJSP的问题描述为:有 n 个工件在 m 台机器上加工,每个工件有不同数量的工序,每道工序都可以在多台机器上进行加工,工序加工之间需要考虑在机器间的运输时间。运输时间的大小取决于机器之间的距离和运输的工件。工件的优先级相同且工件的各工序加工时间已知。调度目标为最小化完工时间和AGV能耗。此外,根据实际生产情况还需满足以下约束:

- 1)调度车间由仓库、充电站和 m 台机器构成;
- 2)初始状态AGV和工件都在仓库,当工件加工完所有工序返回仓库则调度结束;
- 3)每台机器旁出入缓冲区空间不限;
- 4)AGV完成运输任务后停靠在当前站台等待下一道运输任务;
- 5)同一工件的相邻工序在同一机器上加工不需要占用AGV资源;
- 6)不考虑AGV运输过程中的碰撞和故障;
- 7)不考虑加工过程中的机器故障和紧急任务;
- 8)在零时刻,所有的AGV和机器均可用以及所有工件均可被加工;
- 9)同一时间,1台机器只能加工1个工件,1道工序只能选择1台机器,1台AGV只能运输1个工件;
- 10)工序开始后,加工不能中断。

1.2 数学建模

FJSP的数学建模所涉及到的各变量的数学符号和定义如表1所示。

GFJSP是以最大完工时间和AGV能耗为优化目标,数学模型的优化目标和约束条件如下:

$$\min(C_{max}) = \min(\max_{i=1, \dots, n} c_i) \quad (1)$$

$$\min E = \sum_{k=1}^K T_k (P_{empty} v z t_{ik} + P_{load} u z t_{ik}) \quad (2)$$

$$\sum_{m=1}^M x_{ijm} = 1 \quad (3)$$

$$at_{ijm} - dt_i \geq 0, at_{ijm} - dt_{im} \geq 0 \quad (4)$$

$$(bt_{ijm} - at_{ijm} - zt_{ijm}) X_{ijm} \geq 0 \quad (5)$$

$$x_{im} y_{mj} (at_{ijk} - t) (1 - Y_{ijt}) \geq 0 \quad (6)$$

$$at_{i_1 j_1 m_1} = \max \{ bt_{i_1 (j_1 - 1) m_2}, bt_{i_2 j_1 m_1} \}, \forall i_1, i_2 \in I, \forall j_1 \in J, \forall m_1, m_2 \in M \quad (7)$$

表 1 数学符号与定义

Table 1 Mathematical symbols and definitions

符号	定义
i	工件数, $i \in I, I = \{1, 2, \dots, n\}$
j	工件的工序数, $j \in J, J = \{1, \dots, n_j\}$
m	机器数, $m \in M, M = \{1, 2, \dots, M\}$
k	AGV 数, $k \in K, K = \{1, 2, \dots, K\}$
o_{ij}	作业 i 的第 j 道工序
m_{ij}	加工 o_{ij} 的机器, $m_{ij} \in M_{ij}$
at_{ijm}	在机器 m 上加工 o_{ij} 的开始时间
bt_{ijm}	在机器 m 上加工 o_{ij} 的结束时间
zt_{ijm}	o_{ij} 在机器 m 上的加工时间
zt_m	机器 m 的总工作时间
dt_i	作业 i 到达系统的时间
dt_{im}	作业 i 到达机器 m 的时间
vat_{ik}	AGV 空载启动时间 t
vbt_{ik}	AGV 空载结束时间 t
uat_{ik}	AGV 负载开始时间 t
ubt_{ik}	AGV 负载结束时间 t
vzt_{ik}	AGV 空载运行时间 t
uzt_{ik}	AGV 负载运行时间 t
c_i	作业 i 完成时间
d_i	作业 i 交货时间
X_{ijm}	0,1 变量, 作业 i 的操作 j 在机器 m 上进行为 1, 否则 0
Y_{ijt}	0,1 变量, 作业 i 的操作 j 在时间 t 正在处理为 1, 否则 0
x_{im}	0,1 变量, 作业 i 分配机器 m 为 1, 否则 0
y_{mt}	0,1 变量, 机器 m 在时间 t 空为 1, 否则 0
H_{ik}	0,1 变量, 作业 i 分配 AGV k 为 1, 否则 0
h_{kt}	0,1 变量, AGV k 在时间 t 空为 1, 否则 0

$$\sum_{k=1}^K H_{ik} = 1 \quad (8)$$

$$at_{ijm} \geq ubt_{ik} \quad (9)$$

$$vat_{i_2k} \geq ubt_{i_1k}, \forall i_1, i_2 \in I, \forall k \in K \quad (10)$$

$$vbt_{ik} \geq vat_{ik} + vzt_{ik} \quad (11)$$

$$ubt_{ik} \geq uat_{ik} + uzt_{ik} \quad (12)$$

$$uat_{ik} \geq \max\{vbt_{ik}, bt_{i(j-1)m}\}, \forall i \in I, \forall (j-1) \in J, \forall k \in K, \forall m \in M \quad (13)$$

$$at_{i_1j_1m} \geq \max\{ubt_{i_1k}, bt_{i_2j_2m}\}, \forall i_1, i_2 \in I, \forall j_1, j_2 \in J, \forall k \in K, \forall m \in M \quad (14)$$

其中, 式(1)为目标函数, 最小化作业完成时间(Makespan); 式(2)为目标函数, AGV 能耗; 式(3)表示表示 1 台机器 m 一次只能加工一道工序 j ; 式(4)表示工件 i 的工序 j 开始时间不早于该工序到达系统的时间, 工序 j 的开始时间不早于到达机器 m 的时间; 式(5)表示工件 i 的工序 j 结束时间不早于该工序在机器 m 的开始时间与加工时间之和, 即工件工序开始加工便不能中断; 式(6)表示工

件 i 的工序加工需要分配给空闲机器 m 或者等待机器当前加工的工序完成; 式(7)表示作业 i_1 在机器 m_1 上操作 $o_{i_1j_1}$ 的开始时间不小于作业 i_1 在机器 m_2 上前操作 $o_{i_1j_1-1}$ 的结束时间和前操作 $o_{i_2j_2}$ 在机器 m_1 上操作的结束时间; 式(8)表示作业 i 在任何操作中最多由一辆 AGV 运输; 式(9)表示在给定时间, AGV 只能处理一个作业, 作业开始运行的时间大于加载 AGV 的结束时间; 式(10)表示一个 AGV 在给定时间只能运输一个作业, 且空载 AGV 作业 i_2 的开始时间不早于负载 AGV 上一个作业 i_1 的结束时间; 式(11)和式(12)表示空/负载 AGV 结束时间不小于空/负载 AGV 开始时间与空/负载运行时间之和; 式(13)表示前往运输工件 i 的空载 AGV k 需要到达机器 m 并等工件 i 的前一次操作 o_{ij-1} 结束才能开始运输; 式(14)表示操作 $o_{i_1j_1}$ 的开始时间不早于加载 AGV k 的结束时间和机器 m 当前操作 $o_{i_2j_2}$ 的结束时间。

2 马尔科夫决策过程

在 GFJSP 中, 除了工件与机器的加工调度以外, AGV 的调度也可以描述为一个阶段性的工件-AGV 匹配问题。将整个工件-机器-AGV 匹配过程建模为马尔可夫决策过程(MDP)。调度系统在所有决策阶段间歇地与环境交互, 然后动态地将工件分配给可用的 AGV 并运输到可用的机器执行, 直到所有任务完成。这样一个完整的过程被称为一个插曲。

在每个决策阶段 t , 调度系统感知到环境状态变量 S_t , 调度系统根据策略 $\pi(a|s)$ 对交互状态变量 S_t 做出相应的动作 A_t 。此时, 工件与可用 AGV 和机器实现了成功匹配。在当前的工件运输加工结束后, 环境产生一个数字奖励 R_t 作为决策的有效反馈。根据上述调度过程定义状态特征、动作规则、奖励函数和策略定义。

2.1 状态特征

MDP 的状态能够帮助算法了解当前车间与 AGV 的状态。为了得到完整状态需对当前环境进行完整描述。状态特征如表 2 所示。

根据与优化目标相关的状态特征, 设计状态矩阵 S_t 由式(15)表示。

$$[it_{wait}, mt_{wait}, LR_k, UR_m, UR_k, o_{ixj_y}, szt_m, s_k^i, t_k^i, t_k^l] \quad (15)$$

2.2 动作空间

智能体通过交互获取车间信息后在动作空间中选择合适的调度规则, 而 DRL 智能体的动作表示在决策点的调度行为, 动作由向量 $A_t = (am_t, ak_t)$ 表示。其中, am_t 表示决策 t 时刻机器加工工序所选择的规则, ak_t 表示决策 t 时刻 AGV 运输工序所选择的规则。常见的规则有: 最短加工时间优先(shortest processing time, SPT)、最早开始时间优先(earliest start time, EST)、先进先出(first in first out, FIFO)、剩余处理时间最小优先(shortest remaining processing

表 2 状态特征
Table 2 Status characteristics

状态特征描述	定义
工件等待时间 it_{wait}	$it_{wait} = at_{ijm} - dt_{im}$
机器空闲时间 mt_{wait}	$mt_{wait} = zt_m - \sum_{i=1}^n \sum_{j=1}^n zt_{ijm}$
AGV 负载率 LR_k	$LR_k = \frac{uzt_{ik}}{uzt_{ik} + vzt_{ik}}$
机器利用率 UR_m	$UR_m = \frac{\sum_{i=1}^n \sum_{j=1}^n zt_{ijm}}{zt_m}$
AGV 利用率 UR_k	$UR_k = \frac{uzt_{ik} + vzt_{ik}}{\max_{i=1, \dots, n} c_i}$
剩余待加工工件 工序 $o_{i_x j_y}$	$o_{i_x j_y} = \{i_x j_y, \dots, i_n j_n\},$ x, y 为当前工序
机器剩余工件工序的 加工时间 szt_m	$szt_m = \sum_{i=x, j=y}^n zt_{ijm}$
AGV k 运输工件 i	s_k^i
AGV k 运输工件 i 选择的路径	l_k^i
AGV k 在路径 l 上的时间	t_k^l

time, SRPT)、最早完成时间优先 (earliest finish, EF)。动作空间如表 3 所示。

表 3 动作空间
Table 3 Action space

序号	规则	定义
1	SPT	最短加工时间优先
2	EST	最早开始时间优先
3	FIFO	先进先出规则
4	SRPT	剩余处理时间最小优先
5	EF	最早完成时间优先
6	SPT+EF	优先选择 SPT, 若出现相同 处理时间则按 EF 选择
7	SRPT+EF	优先选择 SRPT, 若出现相同 处理时间则按 EF 选择
8	SRPT+SPT	优先选择 SRPT, 若出现相同 处理时间则按 SPT 选择

探索是强化学习中的一个关键概念,通过随机探索所有未知行为,为智能体找到所有可能的最优解。贪婪选择策略是基于行为价值的最大化,以高概率选择当前状态下的最优行为。基于 ϵ -贪心的动作选择如式(16)所示, ϵ 为探索率。

$$\alpha = \begin{cases} \operatorname{argmax}_a Q(a'), & 1 - \epsilon \\ \text{随机}, & \epsilon \end{cases} \tag{16}$$

选择较大的 ϵ 值,随着训练逐步减少 ϵ 值,最终趋向于较小的 ϵ ,以保证最终大部分的行为是利用当前的知识进行选择。

2.3 奖励函数

GFJSP 调度不仅要考虑单个 AGV 的行为,还需要兼顾多 AGV 和加工机器之间的协作。这个奖励函数设计必须综合考虑任务的完成时间、资源利用率、运输效率等因素。

每个 AGV 和机器的主要目标是完成调度任务。任务完成情况是最重要的奖励因素。设计任务完成奖励 R_1 , 当任务完成时, AGV 和机器将获得正奖励。 R_1 由式(17)表示。

$$R_1 = \begin{cases} +100, & \text{任务在预期时间内完成} \\ -50, & \text{任务延迟超过可接受的阈值} \end{cases} \tag{17}$$

AGV 与机器之间的协作是提高整体效率的关键,设计协作奖励 R_2 , 鼓励 AGV 与机器之间的协作。 R_2 由式(18)表示。

$$R_2 = +10 \times (\text{协作成功的步骤}) \tag{18}$$

车间的资源利用率(如 AGV 的使用率、机器的使用率)直接影响整体调度效果。奖励函数应鼓励合理调度,以减少空闲时间和资源浪费。设计资源利用奖励 R_3 以此来鼓励高效的资源利用。 R_3 由式(19)表示。

$$R_3 = \begin{cases} +50, & \text{AGV 或机器得到有效利用} \\ -30, & \text{AGV 或机器闲置} \end{cases} \tag{19}$$

在 AGV 的路径选择和作业分配过程中, AGV 应该选择最短路径并合理分配作业。设计路径选择奖励 R_4 和任务分配奖励 R_5 。 分别由式(20)和(21)表示。

$$R_4 = -L(s, a), L \text{ 为所选择的路径距离} \tag{20}$$

$$R_5 = +10 \times (\text{高效的任务分配}) \tag{21}$$

车间的目标之一是提高调度效率,减少任务执行的总时间。时间的长短直接影响调度性能。设计时间奖励 R_6 , 由式(22)表示。

$$R_6 = -Makespan \tag{22}$$

综合考虑上述各个因素,设计总体奖励函数 R , 由式(23)表示。

$$R = x_1 \cdot R_1 + x_2 \cdot R_2 + x_3 \cdot R_3 + x_4 \cdot R_4 + x_5 \cdot R_5 + x_6 \cdot R_6 \tag{23}$$

其中, $x_1, x_2, x_3, x_4, x_5, x_6$ 为各个奖励因素的权重系数。

3 多步深度多 Q 网络

在传统 Q 学习(Q-Learning)中,随着 Q 值的更新, Q 值估计可能会变得不稳定。而深度 Q 网络(DQN)通过引入经验回放和目标网络来缓解这一问题,确保了学习过程的稳定性。DQN 的 Q 值计算公式如式(24)所示。

$$Q(s_t, a_t) = r_t + \gamma \max_a Q^*(s_{t+1}, a) \quad (24)$$

尽管引入了经验回放和目标网络, DQN 的训练过程仍可能需要非常长的时间才能收敛。并且 DQN 在训练过程中需要平衡探索与利用, 而在训练初期, 因为没有足够的探索可能容易陷入局部最优解。

3.1 双深度 Q 网络

双深度 Q 网络(double deep Q-network, DDQN)能够解决传统 DQN 中 Q 值的过估计问题。该方法通过引入两个 Q 网络: 一个用于选择动作, 另一个用于评估动作, 来降低 Q 值的估计偏差, 从而提高策略的稳定性。DDQN 的 Q 值计算公式由式(25)表示。

$$Q(s_t, a_t) = r_t + \gamma Q^*(s_{t+1}, \arg \max_a Q(s_{t+1}, a)) \quad (25)$$

为实现以上目的, DDQN 需要维护两个 Q 网络(主网络和目标网络)。每次更新时, 需要在两个网络间同步和计算目标, 这会增加计算成本。除此之外, DDQN 主要解决了 Q 值过估计的问题, 但并没有直接解决探索的问题。

3.2 基于优先级经验池回放的深度 Q 网络

基于优先级经验池回放的深度 Q 网络(prioritized experience replay deep Q-network, PER DQN)通过引入优先经验回放机制来提高训练效率和性能。PER-DQN 根据 TD 误差为经验池中不同的经验分配不同的采样优先级, 优先选择那些对于 Q 值更新更有影响的经验, 从而加速训练过程、提升样本效率。时序差分误差 TD 如式(26)所示。

$$TD = r_{t+1} + \gamma \max_a Q^*(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \quad (26)$$

PER-DQN 分配采样优先级时需要对每个经验样本进行计算, 这大大增加了算法的计算难度。

3.3 决斗深度 Q 网络

决斗深度 Q 网络(dueling deep Q-network, Dueling DQN)通过引入 Dueling 网络架构, 将 Q 值函数分解为两个部分: 一个是状态价值函数 $V(s)$, 另一个是优势函数 $A(s, a)$ 。这种架构可以帮助智能体更好地评估动作的重要性, 从而提高学习效率和策略表现, 如式(27)所示。

$$Q(s, a) = V(s) + A(s, a) - \max_a A(s, a) \quad (27)$$

Dueling DQN 中, 通过额外的网络结构使智能体能够更好地评估每个动作相对于其他动作的优势, 但也使得训练缓慢。

3.4 多步学习

多步学习(Multi-Step Learning)利用多个时间步的信息(即多个未来状态的奖励), 可以在每次更新时获得更多的信息, 从而减少学习的时间。对于 m 步的 Multi-Step learning 计算方式如式(28)所示:

$$Q(s, a) = \sum_{i=0}^{m-1} \gamma^i \cdot r_{t+i} + \gamma^m \cdot \max_a Q^*(s_{t+m}, a) \quad (28)$$

多步学习需要容纳足够多历史信息经验回放缓冲区。如果缓冲区不足, 算法可能无法获得足够多的多步信息, 进而影响性能。

3.5 本文算法

经过上述的总结, 本文提出多步深度多 Q 网络算法(Multi-Step D4QN), 该算法集成 DDQN、PER DQN、Dueling DQN、Multi-Step Learning, 并且能够解决各算法所存在的问题。本文算法利用多个时间步的信息加快训练速度解决训练较慢问题, 通过评估动作重要性解决动作探索问题。表 4 为算法的伪代码。

表 4 算法伪代码

Table 4 Algorithm pseudocode

伪代码

初始化当前网络 Q_θ

初始化目标网络 Q_{θ^-}

初始化优先经验回放池 D

初始化 N 步的回报缓存 C

对于每个训练回合:

 通过与环境交互初始化状态 s_0 中的 n 个特征表述为向量 s_t

 在回合结束之前执行以下步骤, 直到达到最大时间步或回合结束:

 根据当前状态 s_t 以概率 ϵ 选择一个随机动作 a_t

 否则动作 $a_t = \arg \max_a Q_\theta(s_t, a)$

 执行动作 a_t , 观察奖励 r_t 和下一个状态 s_{t+1}

 将经验 (s_t, a_t, r_t, s_{t+1}) 存储到经验回放池 D 中

 根据每个样本的 TD 误差进行加权采样并采样一个批次的经验

 对于每个批次中的经验计算多步回报:

$$G_t = r_t + \gamma r_{t+1} + \dots + \gamma^n Q_{\theta^-}(s_{t+n}, \arg \max_a Q_\theta(a', s_{t+n}))$$

 计算目标 Q 值:

$$y_t = r_t + \gamma Q_{\theta^-}(s_{t+1}, \arg \max_a Q_\theta(a', s_{t+1}))$$

 损失函数计算 Q 网络的误差: $\mathcal{L}(\theta) = \mathbb{E}[(Q_\theta(a_t, s_t) - y_t)^2]$

 通过梯度下降算法更新参数 $\theta: \theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta)$

 每隔 T 步将 Q 网络参数 θ 复制到目标网络 $\theta^-: \theta^- \leftarrow \theta$

 根据每次经验回放的误差更新回放池中的优先级

 重复进行回合和步骤, 直到达到停止条件

利用 MDP 所设计的状态特征、动作空间、奖励函数, 提供一个可以使 AGV 与车间进行交互的数学框架, 并采用 multi-step D4QN 算法利用全连接网络描述 Q 网络的网络架构进行求解。图 1 所示为算法网络架构。

4 实验与分析

为了验证基于 Multi-Step D4QN 调度方法的性能, 本文设计了两类实验。首先是测试算法应用于单目标柔性作业车间的有效性, 以最小完工时间为目标函数, 在 Br-data

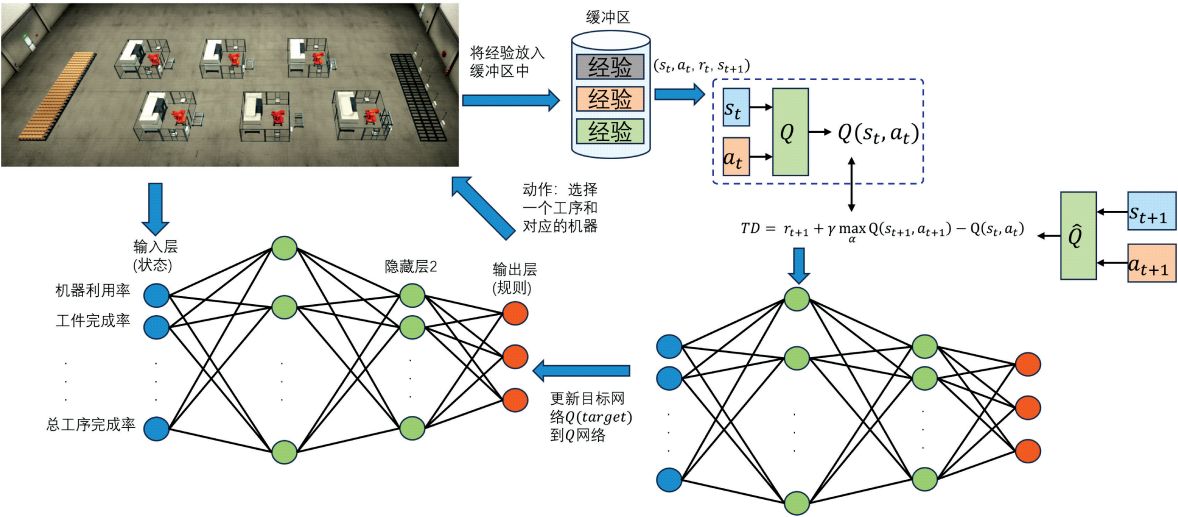


图 1 算法网络架构图

Fig. 1 Algorithm network architecture diagram

中的 MK01~MK10 算例上进行实验。其二,应用于考虑 AGV 运输的多目标柔性作业车间,以最小化完工时间和 AGV 能耗为目标函数,进行更贴合现实生产的 MK01~MK10 算例实验。并以 MK01 为例计算 AGV 最优数量。本文实验使用 Python 3.10 在个人 PC 上运行,配置为 CPU Intel Core i7-12650H。主要的实验参数为探索率 ϵ (取 0.8~1)、学习率 α (取 0.01~0.03)、折扣因子 γ (取 0.8~1),对三种参数进行多数值组合进行 5 次实验,并对最优解 Best 和平均解 Avg 进行对比,结果如表 5 所示,由对比结果获得表 6 中的实验参数。

表 5 不同参数的实验结果对比

Table 5 Comparison of experimental results with different parameters

$\epsilon, \gamma/\alpha$	0.01		0.02		0.03	
	Best	Avg	Best	Avg	Best	Avg
0.8,0.8	46	47	47	48.4	47	49.0
0.8,0.9	43	43.6	46	47.2	45	46.6
0.8,1	45	45.8	46	47.0	47	47.8
0.9,0.8	42	43.0	45	46.2	46	47.2
0.9,0.9	42	42.6	43	44.6	45	46.8
0.9,1	44	44.8	45	45.4	47	48.4
1,0.8	45	45.6	46	47.2	48	49.6
1,0.9	44	45.0	44	44.8	47	48.8
1,1	45	46.2	44	45.2	47	49.2

4.1 单目标柔性作业车间

本次实验使用基准算例为 Brandimarte^[18] 提出的 MK01~MK10 算例,Kacem 等^[19] 提出的 01~05 算例。这些算例涵盖了各种不同的加工机器数量、工件数和工序

表 6 实验参数

Table 6 Experimental parameters

训练参数	
探索率 ϵ	0.9
学习率 α	0.01
折扣因子 γ	0.9
迭代次数	500

数,可以在一定程度上验证算法的性能。为了验证算法的有效性,在算例上得到数据后与其他各种算法进行对比,包括 MK 算例对比基于规则调度(SPT、EST、EF)、基于组合规则调度(SPT+SRPT、SPT+EF、SRPT+EF)、基于启发式算法调度(grey wolf optimization,GWO)、基于强化学习算法调度(DDQN、DQN),Kacem 算例下对比遗传算法、粒子群、改进近邻人工蜂群算法(improved algorithm of near-neighbor artificial bee colony,INBAC)^[20]。对比结果如表 7 所示。

图 2 为算法 Multi-Step D4QN 在 MK01 算例上的仿真结果。因为 10 个算例的参数差异大,导致每个算例所得数据大小差异也过大。为了更好的表现出算法的稳定性与优越性,提出最优差比率 $rate$ 公式,将实验所得数据与理论最优解进行差值对比。公式由式(29)表示。

$$rate = \frac{Ms - C_{best}}{C_{best}} \tag{29}$$

将各算法的实验数据进行比率计算,并得出比率对比表,如表 8 所示。

由表 8 可知,复合调度规则 SPT+SRPT、SPT+EF、SRPT+EF 的平均 $rate$ 分别为 2.18、1.55、1.14,启发式算法 PSO、GWO 的平均 $rate$ 分别为 0.32、0.54。而本文算法的平均 $rate$ 为 0.19。为了方便观察结果,将数据绘制成

表 7 多种算法 MK 算例实验数据对比结果

Table 7 Experimental data comparison results of MK examples of various algorithms

算例	最优解范围	SPT+SRPT ^[16]	SPT+EF ^[16]	SRPT+EF ^[16]	PSO ^[17]	GWO	本文算法
MK01	(36,42)	73	81	71	44	56	42
MK02	(24,32)	65	85	60	33	43	29
MK03	(204,211)	579	465	374	204	238	204
MK04	(48,81)	205	112	120	69	81	60
MK05	(168,186)	265	250	236	177	186	173
MK06	(33,86)	264	161	126	78	86	58
MK07	(133,157)	446	325	278	148	173	139
MK08	(523)	777	717	643	523	556	523
MK09	(299,369)	716	563	535	341	370	311
MK10	(165,296)	517	509	373	252	312	225

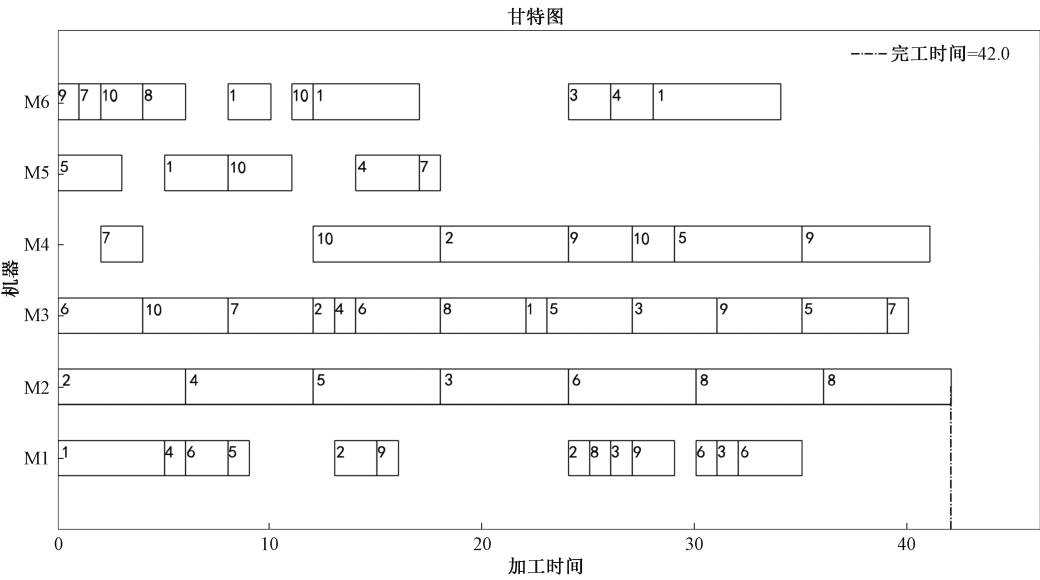


图 2 MK01 甘特图

Fig. 2 Gantt chart of MK01

表 8 多种算法的最优差比率对比

Table 8 Comparison of optimal difference ratios of multiple algorithms

算例	SPT+SRPT ^[16]	SPT+EF ^[16]	SRPT+EF ^[16]	PSO ^[17]	GWO	本文算法
01	1.03	1.25	0.97	0.22	0.55	0.17
02	1.71	2.54	1.5	0.38	0.79	0.21
03	1.84	1.28	0.83	0	0.17	0
04	3.27	1.33	1.5	0.44	0.69	0.25
05	0.58	0.49	0.41	0.05	0.11	0.03
06	7	3.88	2.82	1.36	1.61	0.76
07	2.35	1.44	1.09	0.11	0.30	0.05
08	0.49	0.37	0.23	0	0.06	0
09	1.39	0.88	0.79	0.14	0.24	0.04
10	2.13	2.08	1.26	0.53	0.89	0.36

箱线图,如图 3 所示。

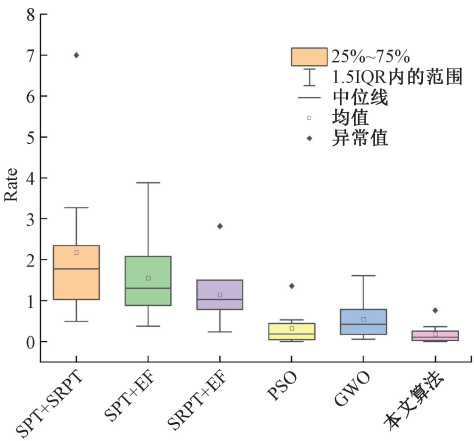


图 3 各算法最优差比率箱线图对比

Fig. 3 Comparison of the rate box plot of each algorithm

通过以上图表,可以明显看出本文算法相较于规则调度和元启发式算法有着较大的优势,尤其是通过箱线图,本文算法的结果更贴近最优解,且更为稳定。

除了上述两种调度方式的对比,本文还与单一优化的DQN算法及文献[21]提出的深度5Q网络(deep 5Q-network,D5QN)算法进行对比,字体加粗部分为各算法间最优解。实验数据如表9所示。

表 9 多种深度学习算法 MK 算例实验数据对比
Table 9 Comparison of experimental data of MK examples of various deep learning algorithms

算例	DQN	DDQN	D3QN	PER DDQN	D5QN ^[21]	本文算法
01	42	42	42	42	42	42
02	33	31	33	32	31	29
03	204	204	204	204	204	204
04	74	70	72	72	69	60
05	190	185	182	182	180	173
06	86	75	75	75	68	58
07	206	196	196	195	153	139
08	523	523	523	523	523	523
09	335	329	320	324	315	311
10	243	240	240	242	230	225

将各深度学习算法的数据进行最优比率计算,得出的结果进行对比,如表10所示。

表 10 多种深度学习算法的最优差比率对比
Table 10 Comparison of optimal difference ratios of multiple deep learning algorithms

算例	DQN	DDQN	D3QN	PER DDQN	D5QN ^[21]	本文算法
01	0.17	0.17	0.17	0.17	0.17	0.17
02	0.38	0.29	0.38	0.33	0.29	0.21
03	0	0	0	0	0	0
04	0.54	0.46	0.5	0.5	0.44	0.25
05	0.13	0.1	0.08	0.08	0.07	0.03
06	1.61	1.27	1.27	1.27	1.06	0.76
07	0.55	0.47	0.47	0.46	0.15	0.05
08	0	0	0	0	0	0
09	0.12	0.1	0.07	0.08	0.05	0.04
10	0.47	0.45	0.45	0.46	0.39	0.36

同样的,由表10可知,深度学习算法DQN、DDQN、D3QN、PER-DQN、D5QN的平均rate分别为0.4、0.33、0.34、0.33、0.26,而本文算法的平均rate为0.19。将上表的数据绘制成箱线图,如图4表示。

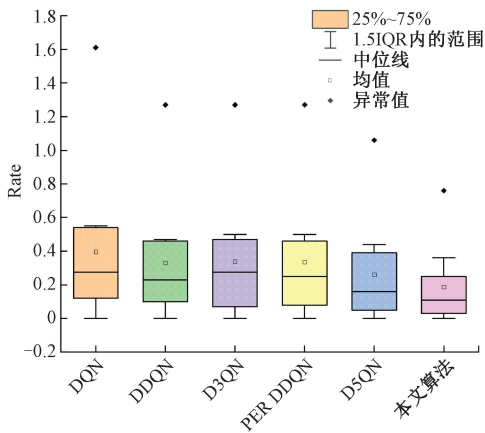


图 4 各深度学习算法最优差比率箱线图对比

Fig. 4 Comparison of the rate box plot of various dql algorithms

除了BR-data数据中的MK算例,本文采用了Kacem提出的01~05算例进行对比,实验数据如表11所示。

表 11 多种算法 Kacem 算例实验数据对比结果
Table 11 Experimental data comparison results of Kacem examples of multiple algorithms

算例	GA	PSO ^[17]	INBAC ^[20]	本文算法
01(4×5)	11	11	11	11
02(8×8)	16	15	14	14
03(10×7)	12	11	11	11
04(10×10)	7	7	7	7
05(15×10)	*	12	12	11

根据表11的实验数据对比可知,本文算法在Kacem这种易获取近似于最优解的算例中也能够获得更为优越的数据结果。

由MK算例与3种算法对比结果的两个箱线图以及Kacem算例对比数据可知,本文算法相比于单一优化DQN或其他类型调度算法有着显著的提升,充分证明了本文算法的有效性和稳定性。

4.2 多目标柔性作业车间

AGV作为智能化生产中的重要组成部分,承担着车间物料搬运、零部件配送、产品转运等多项任务。而常见的车间作业研究中往往会忽略AGV这一重要环节,因此为了更贴合实际生产流程,多目标柔性作业车间调度需要考虑AGV运输工件的调度。关于AGV运输的研究,需要考虑AGV的各种约束,包括电量约束、能耗约束、数量约束。实验的参数与训练参数如表12所示。

以MK01为例,基于AGV运输的车间调度仿真实验所得结果甘特图如图5所示。

本次实验采用10台AGV对MK01算例中的加工工件进行运输,各个AGV的能耗平均为120 kws,实验迭代过程图由图6表示。

表 12 实验参数与训练参数

Table 12 Experimental parameters and training parameters

参数		训练参数	
算例	MK01	探索率 ϵ	0.9
AGV 数量	10 台	学习率 α	0.01
充电阈值	20%	折扣因子 γ	0.9
充足阈值	90%	迭代次数	200
空载功率	0.5 kW		
负载功率	2 kW		

在柔性作业车间中,生产过程复杂且变动较大,合理配置 AGV 数量可以避免过多的空闲和等待时间。通过优化 AGV 数量和调度策略,能够最大限度地缩短生产周期。AGV 作为运输工具可以完美代替人工搬运,尤其是在大规模生产中,合适的 AGV 数量可以在不影响生产周期的情况下控制运行成本。因此,通过合理配置 AGV 数量,既能确保成本控制,又能保证生产效率。

为了计算最优 AGV 数量,在 MK01 算例上分别进行 2~10 台 AGV 数量的实验,并得到完工时间、AGV 平均利用率、AGV 平均充电次数、AGV 能耗的数据。将这些数

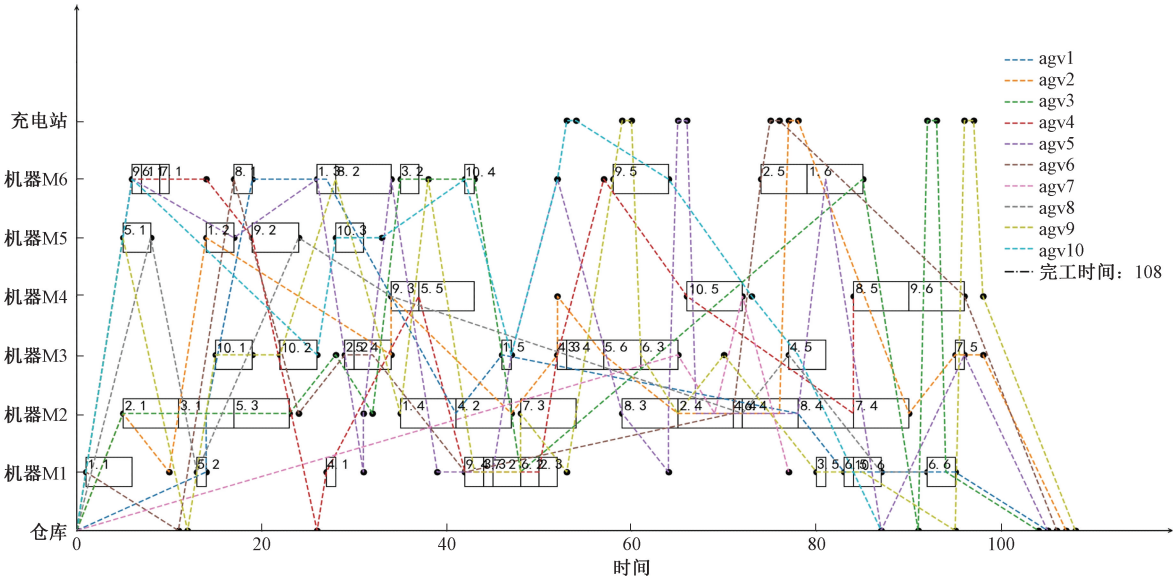


图 5 基于 AGV 运输的 MK01 仿真甘特图

Fig. 5 MK01 simulation Gantt chart based on AGV transportation

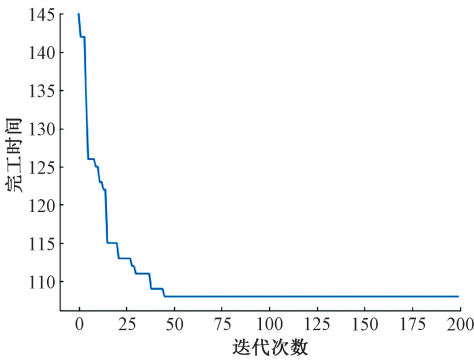


图 6 迭代次数

Fig. 6 The number of iterations

表 13 不同 AGV 数量的实验数据

Table 13 Experimental data of different AGV quantities

AGV 数量	完工时间	AGV 平均利用率	AGV 平均充电数	AGV 能耗
2	300	99.00	5.50	742.5
3	233	98.48	3.67	860.47
4	169	98.22	2.75	829.96
5	149	93.83	2.00	873.79
6	137	92.43	1.67	949.72
7	127	91.45	1.29	1 016.24
8	120	90.87	1.13	1 090.44
9	118	89.34	1.00	1 185.99
10	108	89.25	0.70	1 204.88

据用表 13 表示。

表 13 中表示了不同 AGV 数量的 4 种指标数据,为了更直观且清晰的反映出各个指标与不同 AGV 数量之间的关联,将完工时间、利用率、能耗绘制成折线图,分别为图 7 和图 8。

由图 7 可知,不同的 AGV 数量与完工时间和 AGV 利

用率成反比,即所用数量越多,则完工时间越短,利用率越

低,而当数量处于 9 台和 10 台时,完工时间相同都为 108,因此当数量超过 10 台时,完工时间将没有变化,而利用率会显著降低。由图 8 可知,不同的 AGV 数量与 AGV 能耗成正比,即所用数量越多,则能耗越高。

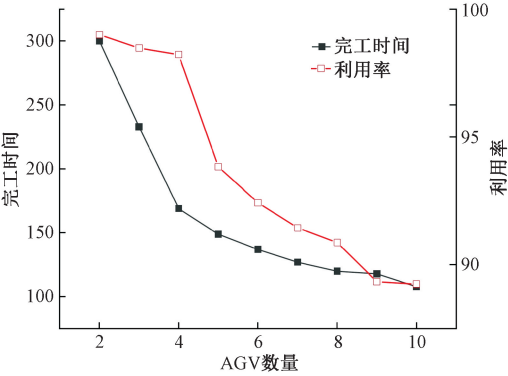


图 7 各 AGV 数量下的完工时间与利用率

Fig. 7 Completion time and utilization rate for each AGV quantity

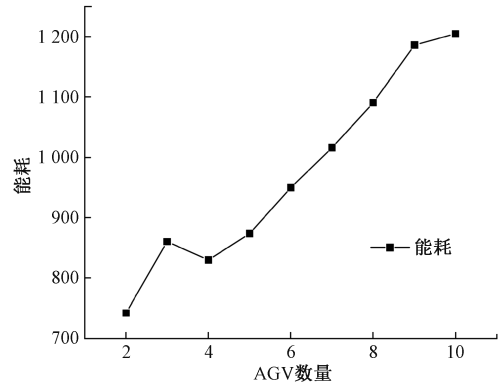


图 8 各 AGV 数量下的总能耗

Fig. 8 Total energy consumption by number of AGVs

多目标柔性作业车间有两个目标函数,因此需要设计一个权重系数 λ 及权重函数 num_{agv} ,使得求得解为最适合当前车间的 AGV 数量,即采用求得的最优 AGV 数量可以使车间的最大完工时间较小的同时保证能耗也较小。权重函数的计算方式由式(30)表示:

$$num_{agv} = \lambda \times \frac{Ms}{maxMs} + (1 - \lambda) \times \frac{E}{maxE} \quad (30)$$

由表 11 可知,2~10 台 AGV 中完工时间相互之间的比值与能耗相互之间比值无法匹配,因此需要添加权重系数 λ 从而使得两个变量实现度量标准化。通过计算, λ 取 0.2,经过标准化处理,各 AGV 数量的标准化数据如表 14 所示。

为了更直观的展现各 AGV 数量的标准化指标,将这些标准化指数绘制成折线图,如图 9 所示。

从折线图可以看出,基于 MK01 算例生产的车间中,采用 4 台 AGV 相比于其他 AGV 数量标准化指数降低 3%~31.8%,因此 4 台 AGV 的调度效率更高。

表 14 各 AGV 数量的标准化指数

Table 14 Standardized index of the number of AGVs

AGV 数量	标准化指数
2	0.69
3	0.73
4	0.66
5	0.68
6	0.72
7	0.76
8	0.81
9	0.87
10	0.87

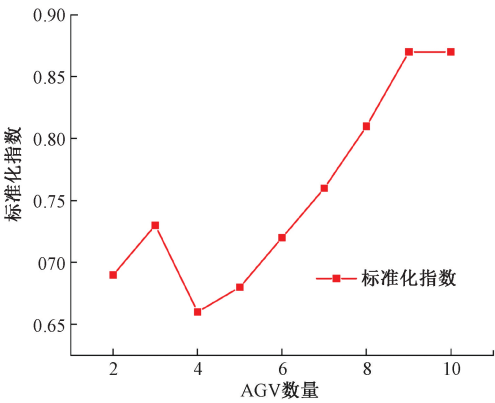


图 9 各 AGV 数量下的标准化指数折线图

Fig. 9 Normalized index line chart for each number of AGVs

5 结 论

对于单目标 FJSP,本文算法在算例 Mk01~Mk10 和 Kacem01~05 上与多种算法进行完工时间和最优差比率两项数据对比。结果表明,本文算法的平均时间比其他深度学习算法缩短了 5.1~17.2 s,平均最优差比率减少了 7.5%~21%,证明了算法的收敛性和稳定性。对于多目标 FJSP,本文算法以最小化完工时间和 AGV 能耗为目标,在 MK01 算例上计算 AGV 最优数量,实验表明 4 台 AGV 相较于其他数量完工时间较短且能耗较低,因此效率更高。本研究考虑了静态生产环境中的 AGV 数量,并未考虑动态生产环境,后续应对紧急任务、设备故障等特殊情况进行研究。

参考文献

[1] LU C, ZHANG B, GAO L, et al. A knowledge-based multiobjective memetic algorithm for green job shop scheduling with variable machining speeds[J]. IEEE Systems Journal, 2021, 16(1): 844-855.

[2] 邓希,胡晓兵,江代渝,等. 基于混合遗传算法的柔性作业车间机器和 AGV 规划[J]. 四川大学学报(自然科学), 2021, 58(2): 73-82.

- DENG X, HU X B, JIANG D Y, et al. Flexible job shop machine and AGV planning based on hybrid genetic algorithm[J]. Journal of Sichuan University (Natural Science), 2021, 58(2): 73-82.
- [3] 罗雄, 钱谦, 伏云发. 无非法解遗传算法求解柔性作业车间调度问题[J]. 电子测量技术, 2020, 43(7): 36-40.
- LUO X, QIAN Q, FU Y F. Non-illegal solution genetic algorithm for flexible job shop scheduling problem[J]. Electronic Measurement Technology, 2020, 43(7): 36-40.
- [4] POPPER J, YFANTIS V, RUSKOWSKI M. Simultaneous production and AGV scheduling using multi-agent deep reinforcement learning[J]. Procedia CIRP, 2021, 104: 1523-1528.
- [5] LI Z, CHEN Y. Minimizing the makespan and carbon emissions in the green flexible job shop scheduling problem with learning effects[J]. Scientific Reports, 2023, 13(1): 6369.
- [6] 盛煜, 朱正伟, 朱晨阳, 等. 基于深度强化学习的多目标边缘任务调度研究[J]. 电子测量技术, 2023, 46(8): 74-81.
- SHENG Y, ZHU ZH W, ZHU CH Y, et al. Research on multi-objective edge task scheduling based on deep reinforcement learning [J]. Electronic Measurement Technology, 2023, 46(8): 74-81.
- [7] 祝正宇, 郭具涛, 吕佑龙, 等. 面向柔性作业车间生产调度的深度强化学习方法[J]. 中国机械工程, 2024, 35(11): 2007-2014, 2034.
- ZHU ZH Y, GUO J T, LYU Y L, et al. Deep reinforcement learning method for flexible job shop scheduling[J]. China Mechanical Engineering, 2024, 35(11): 2007-2014, 2034.
- [8] 曹红倩. 应用改进 Q-learning 算法解决柔性作业车间调度问题 [J]. 国外电子测量技术, 2022, 41(4): 164-169.
- CAO H Q. Application of improved Q-learning algorithm to solve flexible job shop scheduling problem[J]. Foreign Electronic Measurement Technology, 2022, 41(4): 164-169.
- [9] LEI K, GUO P, ZHAO W, et al. A multi-action deep reinforcement learning framework for flexible Job-shop scheduling problem[J]. Expert Systems with Applications, 2022, 205: 117796.
- [10] YUAN E, WANG L, CHENG S, et al. Solving flexible job shop scheduling problems via deep reinforcement learning [J]. Expert Systems with Applications, 2024, 245: 123019.
- [11] CAI A, YU Y, ZHAO M. Deep reinforcement learning solves job-shop scheduling problems [J]. Instrumentation, 2024, 11(1): 88-100.
- [12] BERTEROTTIERE L, DAUZERE-PERES S, YUGMA C. Flexible job-shop scheduling with transportation resources [J]. European Journal of Operational Research, 2024, 312(3): 890-909.
- [13] ZHANG L, FENG Y, XIAO Q, et al. Deep reinforcement learning for dynamic flexible job shop scheduling problem considering variable processing times[J]. Journal of Manufacturing Systems, 2023, 71: 257-273.
- [14] 乐明皓, 李凌. 多策略融合的改进麻雀搜索算法及其 AGV 路径规划应用[J]. 国外电子测量技术, 2024, 43(7): 129-139.
- LE M H, LI L. Improved sparrow search algorithm with multi strategy fusion and application in AGV path planning [J]. Foreign Electronic Measurement Technology, 2024, 43(7): 129-139.
- [15] TREMBLET D, THEVENIN S, DOLGUI A. Makespan estimation in a flexible job-shop scheduling environment using machine learning[J]. International Journal of Production Research, 2024, 62(10): 3654-3670.
- [16] ZHANG Y, ZHU H, TANG D, 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.
- [17] ZARROUK R, BENNOUR I E, JEMAI A. A two-level particle swarm optimization algorithm for the flexible job shop scheduling problem [J]. Swarm Intelligence, 2019, 13: 145-168.
- [18] BRANDIMARTE P. Routing and scheduling in a flexible job shop by tabu search [J]. Annals of Operations research, 1993, 41(3): 157-183.
- [19] KACEM I, HAMMADI S, BORNE P. Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic [J]. Mathematics and Computers in Simulation, 2002, 60(3-5): 245-276.
- [20] 李瑞, 徐华, 杨金峰, 等. 改进近邻人工蜂群算法求解柔性作业车间调度问题[J]. 计算机应用研究, 2024,

41(2):438-443.

LI R, XU H, YANG J F, et al. Improved algorithm of near-neighbor artificial bee colony for flexible Job-Shop scheduling [J]. Application Research of Computers, 2024, 41(2):438-443.

[21] 张凯,毕利,焦小刚.集成强化学习算法的柔性作业车间调度问题研究[J].中国机械工程,2023,34(2):201-207.

ZHANG K, BI L, JIAO X G. Research on flexible job-shop scheduling problems with integrated reinforcement learning algorithm [J]. China

Mechanical Engineering, 2023, 34(2):201-207.

作者简介

彭国峰, 硕士研究生, 主要研究方向为调度算法、深度学习。

E-mail: penggf99@163.com

胡彬, 硕士研究生, 主要研究方向为深度学习、图形识别。

E-mail: binhu@njit.edu.cn

朱晓春(通信作者), 教授, 主要研究方向为柔性制造技术、深度学习。

E-mail: zhuxc@njit.edu.cn