控制与决策
Control and Decision

基于深度强化学习求解作业车间机器与 AGV联合调度问题

孙爱红, 雷琦, 宋豫川, 杨云帆

引用本文:
孙爱红,雷琦,宋豫川,杨云帆. 基于深度强化学习求解作业车间机器与 AGV联合调度问题[J]. 控制与决策, 2024, 39(1): 253-262.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2022.1821

基于深度强化学习求解作业车间机器与AGV联合调度问题

孙爱红, 雷 琦†, 宋豫川, 杨云帆

(重庆大学 机械传动国家重点实验室，重庆 400044)

摘 要: 针对作业车间中自动引导运输车(automated guided vehicle, AGV)与机器联合调度问题,以完工时间最小化为目标,提出一种基于卷积神经网络和深度强化学习的集成算法框架. 首先,对含AGV的作业车间调度析取图进行分析,将问题转化为一个序列决策问题,并将其表述为马尔可夫决策过程. 接着,针对问题的求解特点,设计一种基于析取图的空间状态与5个直接状态特征;在动作空间的设置上,设计包含工序选择和AGV指派的二维动作空间;根据作业车间中加工时间与有效运输时间为定值这一特点,构造奖励函数来引导智能体进行学习. 最后,设计针对二维动作空间的2D-PPO算法进行训练和学习,以快速响应AGV与机器的联合调度决策. 通过实例验证,基于2D-PPO算法的调度算法具有较好的学习性能和可扩展性效果.

关键词: 作业车间调度；自动引导运输车；深度强化学习；马尔可夫决策过程；近端策略优化；联合调度

中图分类号: TP8 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2022.1821

引用格式: 孙爱红,雷琦,宋豫川,等. 基于深度强化学习求解作业车间机器与AGV联合调度问题[J]. 控制与决策, 2024, 39(1): 253-262.

Deep reinforcement learning for solving the joint scheduling problem of machines and AGVs in job shop

SUN Ai-hong, LEI Qi†, SONG Yu-chuan, YANG Yun-fan

(State Key Laboratory of Mechanical Transmission，Chongqing University，Chongqing 400044，China)

Abstract: Aiming at the joint scheduling problem of automated guided vehicle (AGV) and machines in the job shop, an integrated algorithm framework based on convolutional neural network and deep reinforcement learning is proposed with the goal of minimizing the completion time. Firstly, the job shop scheduling disjunction graph containing an AGV is analyzed, and the problem is transformed into a sequential decision problem, which is expressed as the Markov decision process. Then, according to the solving characteristics of the problem, a spatial state and five direct state features based on the disjunctive graph are designed. In the setting of the action space, a two-dimensional action space including process selection and AGV assignment is designed. According to the characteristics of fixed value of processing time and effective transportation time in the work workshop, a reward function is constructed to guide the agent to learn. Finally, a 2D-PPO algorithm for two-dimensional action space is designed for training and learning to quickly respond to the joint scheduling decision of the AGV and machine. Through case verification, the scheduling algorithm based on the 2D-PPO algorithm has good learning performance and scalability effect.

Keywords: job shop scheduling；automated guided vehicle；deep reinforcement learning；Markov decision process；proximal policy optimization；joint scheduling

0 引 言

作业车间自动引导运输车(automated guided vehicle, AGV)和机器联合调度问题可视为作业车间调度问题和AGV调度问题的集成调度问题. 传统作业车间调度问题中,常假设AGV是时刻准备好且随时可用,因此调度过程中常忽略物料搬运时间,主要决策为工序在机器上的排序问题,这一假设对于那些能保证无限运输能力的作业车间是成立的. 然而,

收稿日期: 2022-10-20；**录用日期:** 2023-02-15.
基金项目: 国家自然科学基金项目(51205429).
责任编委: 王凌.
†通讯作者. E-mail: leiqi@cqu.edu.cn.

在实际生产中,运输能力往往有限,导致AGV运输对车间整体性能有影响,机器和AGV之间存在不同时空的相互依赖,因而,作业车间AGV和机器联合调度问题是在传统作业车间调度问题的基础上还考虑了AGV指派对调度的影响. 经典的作业车间调度模型求解已属NP-hard问题,AGV的指派增加了更多的约束条件和决策变量,问题的解空间显著增大,具有更大的复杂性和难度.

针对上述问题,Bilge等[1]率先进行了研究,建立了非线性规划优化模型,提出了基于时间窗的启发式算法来求解该问题. Xie等[2]对近年来的相关研究进行了总结,关于AGV和机器联合调度的研究主要集中于以下4类方法:在多智能体系统层面,Erol等[3]设计了基于机器和AGV的多智体系统,通过不同智能体之间的协商和投标机制实现联合调度;在智能优化算法层面,耿凯峰等[4]和Ren等[5]针对AGV和机器的联合调度问题,分别采用Memetic算法和带遗传操作的粒子群优化算法进行求解;在仿真方法层面,Zhang等[6]和Guo等[7]结合工业物联网和网络物理系统构建生产物流仿真系统框架,使作业车间中的生产物流设备具有自组织和自适应能力;在精确算法层面,Ham[8]采用约束规划重建了作业车间AGV与机器联合调度问题,并采用CP求解器进行求解. 以上4类研究方法长期以来被认为是处理该问题较为合适的方法,但因受到以下限制而被广泛讨论:1) 要找到接近最优解,算法需要进行大量的迭代来进行种群更新或迭代搜索;2) 一旦问题稍有变化,比如工件数量、工序数量、加工时间等发生改变,就需要重新执行算法;3) 当遇到新的调度问题或类似问题的新实例时,需对算法进行改进才能得到好的结果,这就是所谓的无免费午餐定理[9];4) 以上方法肯定也可以得到较高的性能,但是需对具体的问题进行具体的算法设计.

深度学习(deep reinforcement learning, DL)和强化学习(reinforcement learning, RL)相结合产生的深度强化学习(deep reinforcement learning, DRL)在自适应和自学习方面显示出强大的数据处理能力和环境交互能力,为复杂车间制造系统提供了新的解决方案. 与上诉经典方法不同,基于DRL的方法将耗时的训练过程离线转移并将训练好的模型直接用于在线决策中,可以无需迭代快速地获得较为满意的调度方案;其次,基于DRL训练获得的模型具有强大的概括能力,无需对每个新实例进行再训练就可以解决类似问题,这使得它很容易在实际应用中部署. 受DRL的启发以及现实制造环境的迫切需求,王凌等[10]设计了一种基于DRL与迭代贪婪算法的框架来求解流水车间调度问题;Shi等[11]针对自动化生产线调度问题,提出了一种基于强化学习的智能调度算法;朱家政等[12]针对具有模糊加工时间和模糊交货期的作业车间调度问题,提出了LSTM-PPO强化学习框架进行求解.

本文探索基于DRL算法求解作业车间AGV与机器联合问题的可能性. 本文的动机源于最近提出的几个基于DRL的车间调度问题解决方案. Palombarini等[13]利用CNN直接提取彩色甘特图图像作为调度特征,以弥补手工设计的不足;Zhang等[14]和Park等[15]都对作业车间调度(JSP)问题的析取图进行深入研究,提出了基于图神经网络(GNN)的方案来嵌入求解过程中遇到的状态,被证明具有较强的扩展性. 本文结合CNN特征提取和析取图对车间调度问题的表达两方面的优点,基于析取图转换将作业车间中AGV和机器联合调度问题转换为马尔科夫决策过程,通过CNN对析取图信息提取和人工特征结合的方式对环境状态进行完整表达. 此外,针对本文问题包含的工序排序和AGV指派两个子问题,设计考虑两个问题的二维动作空间,并提出一种可用于二维动作空间的2D-PPO算法.

1 问题建模

1.1 问题描述

作业车间AGV和机器联合调度问题可以描述为:N个工件需要在m台机器上加工,每个工件包含n_j道工序,且必须在指定机器上加工,工件在机器之间的转移通过有限个AGV来实现. 对任一工件的转运,AGV动作的时间可分为取件时间和交付时间. 调度目标为如何安排合适的工序顺序,并将工件指派给合适的AGV进行运输,使得总完工时间最小.

为简化问题,本文给出如下假设:

假设1 每台机器旁有充足的出入缓冲区空间;

假设2 AGV一次只搬运一个工件,并沿着预定的最短路径移动,假定不会因拥塞而造成延迟;

假设3 每台机器M_j一次只能加工一个工件J_i,并且每个工件在同一时间只能由一台机器进行处理;

假设4 工件的各工序加工时间已知;

假设5 不同的工件具有相同的加工优先级;

假设6 AGV数量已知,且在速度和承载特性上完全相同;

假设7 初始状态AGV和工件都在装卸站,当工

件加工完所有工序且返回装卸站则加工结束.

1.2 数学模型

参数和索引: J表示工件集; g表示工件数; n_j为工件j的工序数; n表示总工序数, $n=\sum_{j\in J} n_j$; I表示工序索引, $I=\{1,2,\ldots,n\}$; I_j表示工件j的工序索引, $I_j=\{N_j+1,\ldots,N_j+n_j\}$, 其中$N_j=\sum_{i=1}^{j-1} n_i$; $\overline{I}_i$表示同一工件j的工序i之后的工序; $\underline{I}_i$表示同一工件j的工序i之前的工序; K表示AGV数; p_i表示工序i的加工时间; t_i表示工序i的AGV负载时间; τ_{rs}表示从工序r所在机器到工序s所在机器的AGV运输时间.

决策变量: $C_{\max}$为最大完工时间; c_i为工序i的完工时间; T_i为工序i到达加工机器的时间; q_{rs}为二进制变量, 工序r在工序s前加工则为1, 否则为0; x_{hi}为二进制变量, 当AGV被指派给工序h到工序i做运输时为1, 否则为0; x_{oi}为二进制变量, 当AGV的第1次运输指派给从装卸站到工序i的运输时为1, 否则为0; x_{ho}为二进制变量, 当AGV的最后1次运输指派给从装卸站到工序i时为1, 否则为0; D_{jih}为辅助变量, 表示工件j的工序i到工序h之间的总运输时间; S_{jh}为辅助变量, 表示从AGV开始运输工件j的第1道工序到工件j的第h到工序的时间; s_{ti}为辅助变量, 表示工序i的开始加工时间.

作业车间AGV与机器联合调度问题可建模为如下线性规划模型:

$$\min\ C_{\max}. \tag{1}$$

$$C_{\max}\geqslant C_{N_j+n_j},\ \forall\, j\in J; \tag{2}$$

$$c_i-c_{i-1}\geqslant p_t+t_i,\ \forall\, i,i-1\in I_j,j\in J; \tag{3}$$

$$C_{N_j+1}\geqslant p_{N_j+1}+t_{N_j+1},\ \forall\, j\in J; \tag{4}$$

$$\begin{cases}(1+H\tau_{rs})c_r\geqslant c_s+p_s-Hq_{rs},\\(1+H\tau_{rs})c_r\geqslant c_r+p_s-H(1-q_{rs}),\end{cases}$$
$$\forall\, r\in I_j,\ s\in I_k,\ j,k\in J,\ j\neq k; \tag{5}$$

$$x_{oi}+\sum_{h\in\underline{I}_i}x_{hi}=1,\ \forall\, i\in I; \tag{6}$$

$$x_{ho}+\sum_{i\in\underline{I}_h}x_{hi}=1,\ \forall\, h\in I; \tag{7}$$

$$\sum_{i\in I}x_{oi}\leqslant K; \tag{8}$$

$$\sum_{i\in I}x_{oi}-\sum_{h\in I}x_{ho}=0; \tag{9}$$

$$T_i-t_i\leqslant c_i-p_i,\ \forall\, i\in I; \tag{10}$$

$$T_i-t_i\geqslant c_{i-1},\ \forall\, i,i-1\in I_j,j\in J; \tag{11}$$

$$D_{jih}=T_h+\tau_{h,i-1},\ \text{if}\ x_{hi}=1,$$
$$\forall\, i,i-1\in I_j,\ h\in\overline{I}_i,\ j\in J; \tag{12}$$

$$D_{jih}=0,\ \text{if}\ x_{hi}=0,$$
$$\forall\, i,i-1\in I_j,\ h\in\overline{I}_i,\ j\in J; \tag{13}$$

$$T_i-t_i\geqslant x_{oi}\tau_{o,i-1}+\sum_{h\in\overline{I}_i}D_{jih},$$
$$\forall\, i,i-1\in I_j,\ j\in J; \tag{14}$$

$$S_{jh}=T_h+\tau_{ho},\ \text{if}\ x_{h,N_j+1}=1,$$
$$\forall h\in\overline{I}_{N_j+1},\ j\in J; \tag{15}$$

$$S_{jh}=0,\ \text{if}\ x_{h,N_j+1}=0,\ \forall h\in\overline{I}_{N_j+1},\ j\in J; \tag{16}$$

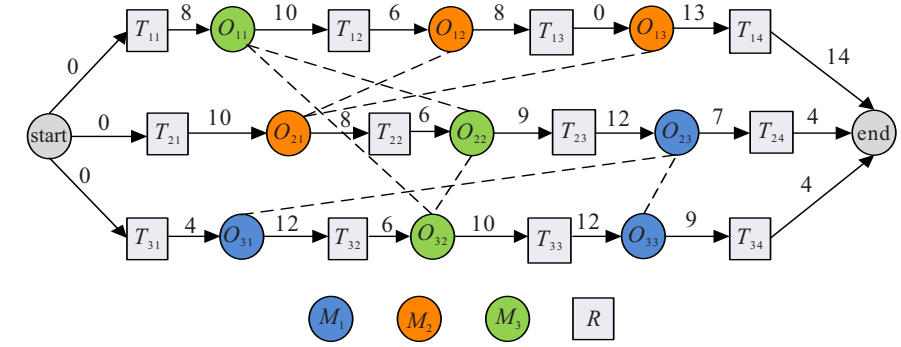
$$T_{N_j+1}-t_{N_j+1}\geqslant\sum_{h\in\overline{I}_{N_j+1}}S_{jh},\ \forall\, j\in J; \tag{17}$$

$$x_{oi}T_i=x_{oi}\tau_{oi}. \tag{18}$$

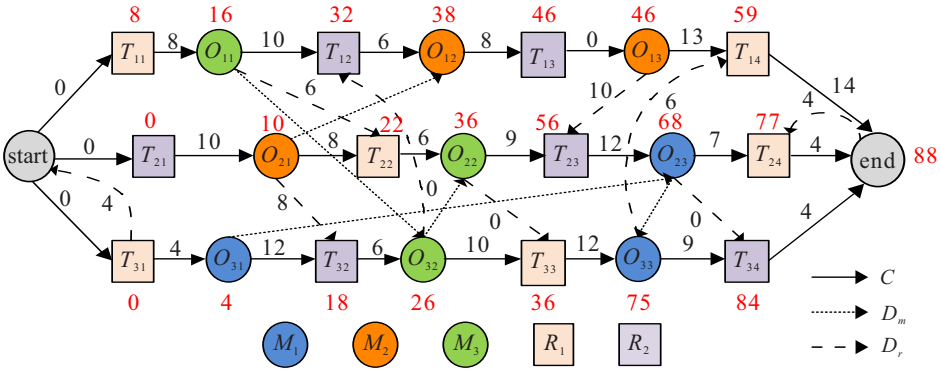
其中: 式(1)为本文的目标函数; 式(2)～(5)可视为传统作业车间调度问题的约束, 不同的是本问题模型加入了运输时间t_i用以表达工件在机器间转移的运输时间; 式(6)～(9)构成了典型AGV调度问题的约束; 式(10)～(18)表达了两个问题的交互约束, 式(10)确保工件到达加工机器方能加工, 式(11)确保了工件加工完成才能开始运输, 式(12)～(17)用以确定工序间的时间约束, 式(18)用以确保初始状态AGV和工件都在装卸站, 并在加工完成返回装卸站.

1.3 析取图模型

本文采用析取图对作业车间AGV和机器联合调度问题进行表达, 问题的可行解可表达为一个有向图$G=(V_O\bigcup V_T, C\bigcup D_m\bigcup D_r)$. 图中包含$V_O$和$V_T$两类顶点以及两个虚拟顶点start和end, 顶点V_O表示调度系统中包含所有机加工工序, 顶点V_T表示调度系统中包含所有运输交付工序; 图G还包括3类有向边集合C、D_m和D_r, C表示工序间优先约束, D_m表示工序在机器上的加工顺序, D_r表示工序的指派情况. 从V_O^i到V_T^i的有向弧权重为机加工时间, 从V_T^i到V_O^i的有向弧权重为运输时间, 有向弧D_r需要在将工序指派给AGV后才能表达. 图1(a)为一个简单算例在未求解前的无向析取图. 要解决作业车间AGV和机器联合调度问题, 需要将所有无向弧D_m转换为有向弧, 并添加有向弧D_r为每个运输工序分配AGV, 其中D_r的权重为取件时间, 于是该问题的一个可行解析取图如图1(b)所示.



(a) 无向析取



(b) 有向析取

图 1 简单实例析取图

2 模型求解

在本节中,首先将作业车间AGV和机器联合调度问题转换为马尔可夫决策过程(Markov decision process, MDP);然后给出状态、动作、奖励函数的定义方式;最后,给出求解该问题的DRL总体框架.

2.1 问题转换

用析取图建模的作业车间AGV和机器联合调度问题很难直接看作是一个序列决策问题. 为直观挖掘析取图的信息,通过拓扑排序将机器工序和转移工序分开,并根据工序加工时间顺序从左到右依次排序. 其中:顶点V_O被转换为线性序列,顶点V_T根据AGV的数量排列成多个线性序列. 为了匹配机器工序和运输工序,在每个作业中添加虚拟工序使机器工序与运输工序的数量相同,拓扑排序的最终调度结果与有向析取图的调度结果是一致的. 以图1(b)为例,对于有向析取图中的3个工件,添加了3个虚拟进程O_{14}、O_{24}和O_{34},这样图1(b)中的可行调度解可转化为图2所示的拓扑排序. 从图2可以看出,通过析取图的转换,该问题的决策过程可视为与工序排序线性相关的序列决策问题,对问题的求解只需$n+g$个连续决策的时间步长. 在每个时间步,DRL智能体选择下一加工工序,并选择一个合适的AGV负责该工序的运输,因此该环境可以转化为MDP. 在本问题中,动作包含工序选择动作集AO和AGV选择动作集AR,为二维动作集,这种耦合多动作空间的MDP也称为MMDP[16].

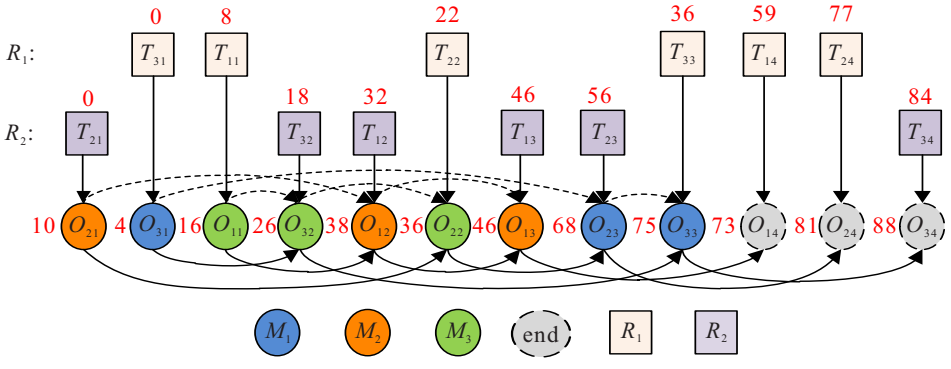


图 2 加工工序和运输工序的拓扑排序

2.1.1 状态表示

状态代表了DRL智能体感知到的环境信息以及自身行为所带来的变化,因此状态的设计需要在深入理解任务逻辑的基础上进行. 为此,本文使用析取图的抽象信息和特定的车间特征来构建状态,析取图的抽象信息可以表达为一个四维矩阵 $\mathrm{St}=\{\mathrm{PT}_t, \mathrm{ST}_t, \mathrm{ET}_t, D_t\}$.

PT_t 用于表示析取图中的析取弧 C. 在析取图模型中,析取弧 C 包含两种信息,即交互时间和加工时间,因此 PT_t 可以表示如下形式的矩阵:

$$\mathrm{PT}_t=\begin{bmatrix} t_1 & p_1 & \cdots & t_{n_1} & p_{n_1} \\ t_{N_2+1} & p_{N_2+1} & \cdots & t_{N_2+n_2} & p_{N_2+n_2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ t_{N_g+1} & p_{N_g+1} & \cdots & t_{N_g+n_g} & p_{N_g+n_g} \end{bmatrix}. \tag{19}$$

ST_t 和 ET_t 两个矩阵分别用于表示析取图中的析取弧的开始时间和结束时间. 每个工序对应的位置与 PT_t 矩阵完全一致,在 t 步时,对于给定的动作 a, ST_t 和 ET_t 在 t 时刻的状态转移通过给在 t 步时已完成的运输工序、加工工序的开工时间和完工时间在 ST_t 和 ET_t 中对应位置进行赋值的方式来实现.

D_t 用于表示析取图中每个工序指派的赋值. 在该问题中,由于机器分配是固定的,在 D_t 中主要考虑AGV指派. D_t 参数由机器索引和AGV索引组成. 在 t 步时,对于给定的动作 a, D_t 在 t 时刻的状态转移通过对已完成的运输工序所在位置赋值给AGV的索引的方式来实现.

析取图映射变换得到的4个矩阵相互关联,可看作是具有空间结构的多通道特征. 本文利用CNN来对具有空间结构的状态特征进行提取,以缓解网络表示能力不足与DRL算法学习效率低下的矛盾. 将状态信息视为4通道图像,使用浅CNN对析取图的空间信息进行挖掘. 此外,还设计了5个人工特征来进一步补充状态,具体描述如下:

1) $\mathrm{PT}_{\mathrm{art}}$ 表示所有工件的平均剩余加工时间:

$$\mathrm{PT}_{\mathrm{art}}=\sum_{i=1}^{g} t_{irt}\Big/ n_t. \tag{20}$$

其中: t_{irt} 表示在 t 步时工件 i 的剩余加工时间, n_t 表示当前剩余待加工工序数.

2) pRT_t^r 表示AGV $k(K)$ 当前总运输时间与当前完工时间之比:

$$\mathrm{pRT}_t^k=\sum_{T_i\in L_{kt}} t_{T_i}\Big/ C_{kt}. \tag{21}$$

其中: C_{kt} 表示AGV r 的在 t 步时的完工时间, L_{kt} 表示在AGV k 上已完成的运输工序集合.

3) pRRT_t 表示剩余的平均有效运输时间,即AGV将工件从当前位置转移到下一个加工位置所用时间:

$$\mathrm{pRRT}_t=\sum_{T_i\in \mathrm{UL}_t} t_{T_i}\Big/ K, \tag{22}$$

其中 UL_t 表示在 t 步时未完成的运输工件集.

4) $\mathrm{pML}_t^{M_i}$ 表示机器 M_i 的当前负载与其当前完成时间之比:

$$\mathrm{pML}_t^{M_i}=\sum_{O_i\in F_{M_it}} p_{O_i}\Big/ C_{M_it}. \tag{23}$$

其中: F_{M_it} 表示在 t 步时机器 M_i 上加工过的全部工序集, C_{M_it} 表示在 t 步时机器 M_i 上的完成时间.

5) pML_t 表示机器总负荷与机器完成时间之比:

$$\mathrm{pML}_t=\sum_{O_i\in F_t} p_{O_i}\Big/ (m\times C_{M_it}). \tag{24}$$

其中: F_t 表示 t 步时所有加工工序的集合, C_t 表示 t 时的系统完成时间, m 表示总机器数.

2.1.2 动作定义

在作业车间AGV和机器联合调度问题中,DRL智能体的动作表示在决策点的调度行为,动作由向量 $a_t=(\mathrm{AO}_t, \mathrm{AR}_t)$ 表示. 其中: AO_t 表示决策点 t 步时的工序选择规则, AR_t 表示决策点 t 步时的AGV分配. AR由AGV的下标组成, $\mathrm{AR}=1,2,\ldots,K$, AR动作个数等于AGV数. 考虑到工件数量较大,且随着加工的进行,工件的数量逐渐减少,直接使用工件下标的方式将造成动作空间过大并产生无效动作. AO动作空间由7条启发式规则组成,即 $\mathrm{AO}=\{\mathrm{FCFS}, \mathrm{SOPT}, \mathrm{SJPT}, \mathrm{SRW}, \mathrm{PDJT}, \mathrm{PDRW}, \mathrm{PMJT}\}$. 各规则的具体描述如下:

1) FCFS表示选择到达时间最早的工件;

2) SOPT表示选择工序加工时间最短的工件;

3) SJPT表示选择工件加工时间最短的工件;

4) SRW表示选择剩余工时最短的工件;

5) PDJT表示选择工序加工时间与工件总加工时间和之比最小的工件;

6) PDRW表示选择工序加工时间与剩余工时之比最小的工件;

7) PMJT表示选择将工序加工时间乘以工件总加工时间获得的最小值对应的工件.

2.1.3 奖励定义

奖励对于引导DRL智能体实现最优策略具有重

要作用,它直接决定了调度的质量,设计的奖励函数应该紧密地以原始目标为基础. 本文考虑了作业车间AGV和机器联合调度问题的特点,即在没有机器灵活性的情况下,不考虑AGV的异质性,机器加工时间和交付时间的总和是一个固定值D,计算方式为

$$D=\sum_{i=1}^{n}(p_i+t_i). \tag{25}$$

在此基础上,在t步时,将DRL智能体的奖励$r(t)$定义为

$$D(t)=\sum_{i=1}^{F_i}(p_i+t_i), \tag{26}$$

$$U(t)=D(t)/(m+r)\times C_t, \tag{27}$$

$$r(t)=U(t)-U(t-1). \tag{28}$$

其中:$K(t)$表示机器和AGV在t步时完成的总工作量;$U(t)$表示机器和AGV在t步时的平均利用率,此处令$U(0)=0$. 回报R的计算如下:

$$\begin{aligned}&R=\\&\sum_{t=1}^{n}r(t)=\sum_{t=1}^{n}(U(t)-U(t-1))=\\&U(n)-U(0)=U(n)=\\&D/(m+r)\times C_{\max}.\end{aligned} \tag{29}$$

由于D是一个固定值,很容易看出,总完工时间与回报R成反比,回报越大,$C_{\max}$越小,实现了系统优化的目标.

2.2 强化学习算法

为了将强化学习算法应用于本文问题的二维离散动作空间环境,常见的作法是将二维动作进行组合,即动作数为动作组合数,这一做法对于动作数较少的环境是合理的. 然而,随着AGV数量的增加,AR动作维的动作数也随之增加,组合数也会依次增多,使得动作数量较大,从而导致训练慢、收敛不稳定等现状. 在强化学习算法中,近端策略优化算法[17](proximal policy optimization algorithms, PPO)是目前适用性最广的算法,该算法基于AC的框架,可以很好地解决连续动作空间问题和离散动作空间问题. 然而,PPO算法不能直接用于处理多维动作空间的环境,因为它使用随机策略网络输出描述动作的单个分布,对分布进行一次采样得到单个动作. 因此,针对二维离散动作空间,本文在PPO的基础上提出一种2D-PPO算法(2-demension proximal policy optimization algorithms, 2D-PPO),该算法通过修改PPO的Actor网络架构,使输出端输出两组动作概率值,通过对两个动作概率分别采样得到关于工序选择和AGV指派两个动作,从而实现对二维离散动作空间的处理. 对PPO算法的主要改动如下:1)对于t时刻的环境信息s_t,Actor策略网络θ得到两组动作概率值$p_{\theta^1}(a_{1t}|s_t)$和$p_{\theta^2}(a_{2t}|s_t)$,其中θ^1和θ^2除输出端使用不同的全连接层外,其他隐藏层都一样,可以看作是θ的两个子网络,且$p_{\theta^1}(a_{1t}|s_t)$和$p_{\theta^2}(a_{2t}|s_t)$独立同分布,即

$$\begin{aligned}&\log\pi(a_t|s_t)=\\&\log p_{\theta^1}(a_{1t}|s_t)p_{\theta^2}(a_{2t}|s_t)=\\&\log p_{\theta^1}(a_{1t}|s_t)+\log p_{\theta^2}(a_{2t}|s_t).\end{aligned} \tag{30}$$

于是,工序选择动作和AGV指派动作通过对这两组概率值进行采样得到.

2)在反向传播的计算中,由于动作为二维动作,无法直接计算损失函数,为确保策略梯度有稳定的学习性能,本文对$\ell(\theta')$的计算进行修改,原式为

$$\ell(\theta')=\frac{\log\pi'_\theta(a_t|s_t)}{\log\pi_\theta(a_t|s_t)}, \tag{31}$$

其中π'_θ和$\pi_\theta(a_t|s_t)$分别表示新策略和旧策略. 由于输出端两个网络服从独立同分布,对二维动作空间的2D-PPO在计算$\ell(\theta')$时转换为下式进行:

$$\ell(\theta')=\frac{\log\pi'_{\theta_1}(a_1t|s_t)+\log\pi'_{\theta_1}(a_2t|s_t)}{\log\pi_{\theta_1}(a_1t|s_t)+\log\pi_{\theta_1}(a_2t|s_t)}. \tag{32}$$

更新策略时,通过将$\ell(\theta')$的计算替换为本文方式,采用PPO-Clip版本的目标函数计算如下:

$$\begin{aligned}&L^{\text{PPO-Clip}}(\pi'_\theta)=\\&E_{\pi_\theta}[\min(\ell(\theta'))A^{\pi_\theta}(S_t,A_t),\\&\text{clip}(\ell(\theta'),1-\epsilon,1+\epsilon)A^{\pi_\theta}(S_t,A_t)].\end{aligned} \tag{33}$$

其中:$A^{\pi_\theta}(S_t,A_t)$表示优势函数,$\text{clip}(x,1-\epsilon,1+\epsilon)$表示将$x$截断在$(1-\epsilon,1+\epsilon)$内.

3 数值实验

为了验证基于2D-PPO调度方法的性能,本文进行3个实验:

1)使用作业车间AGV和机器联合调度问题的随机算例来测试基于2D-PPO算法的训练效果和收敛性能.

2)在现有实例上将训练好的2D-PPO算法与传统方法进行比较,用于论证本文所提的基于DRL的调度方法的效果.

3)在更大规模的车间系统中进行扩展性测试,以验证基于2D-PPO调度方法在大规模场景下是否具有可扩展性.

针对第2个和第3个实验,为保证2D-PPO算法可以直接用于测试,本文采用不同规模的两个数据集进行训练:针对小规模扩展性能的验证,使用20个随机算例(4～8个工件,5个机器和2个AGV)进行训练;针对大规模扩展性能的验证,使用50个随机算例(15～30个工件,8～10个机器和2～7个AGV)进行训练. 实验使用Python 3.7和PyTorch 1.8在个人PC上进行,Intel Core i5-10400 @ 2.90 GHz, 16 GB RAM.

3.1　性能指标

为了验证本文提出的基于2D-PPO调度方法的优化水平,使用以下性能指标进行度量.

1) BRPD值:最佳结果的百分比偏差,其计算方法为

$$\mathrm{BRPD}=\frac{100\times(A_i^*-A^*)}{A^*}. \tag{34}$$

其中:A_i^*表示第i个算法10次运行的最佳结果,A^*表示所有算法10次运行的最佳结果.

2) ARPD值:平均结果的百分比偏差,其计算方法为

$$\mathrm{ARPD}=\frac{100\times(\overline{A}_i^*-\overline{A}^*)}{\overline{A}^*}. \tag{35}$$

其中:$\overline{A}_i^*$表示第i个算法10次运行的平均结果,$\overline{A}^*$表示所有算法10次运行的最佳平均结果.

另外,对于本文提出的基于2D-PPO调度方法,本文将得分最高的方法绘制为绿色的点线.

3.2　DRL算法的训练和验证

为证明所提2D-PPO的学习能力及扩展性能,在状态、奖励函数等设计一致的条件下,通过对比采用将二维动作进行组合的PPO算法的学习和扩展效果来论证本文算法和环境设置的有效性. 算法的参数设置如下:训练次数为5 000,采样步数为1 024,批次尺寸为64,裁剪系数为0.2,学习率为0.000 1. 在本小节中,在含不同AGV的数量下训练和验证两种算法的收敛性能和学习能力. 训练过程中,在不同AGV数量场景下所有DRL智能体在相同算例集上进行训练,训练曲线如图3所示.

图3绘制了在不同AGV数量下各DRL算法的平均回报变化曲线,图中阴影部分为实际训练过程的取值,实线为实际取值的平滑值. 经过多次训练后,两种算法都实现了收敛,由图可见,随着AGV数量的增加,基于PPO算法的动作数量显著增加,导致学习能力下降,然而本文所提算法相较PPO受到AGV数量增加的影响较小,表现出更好的学习性能.

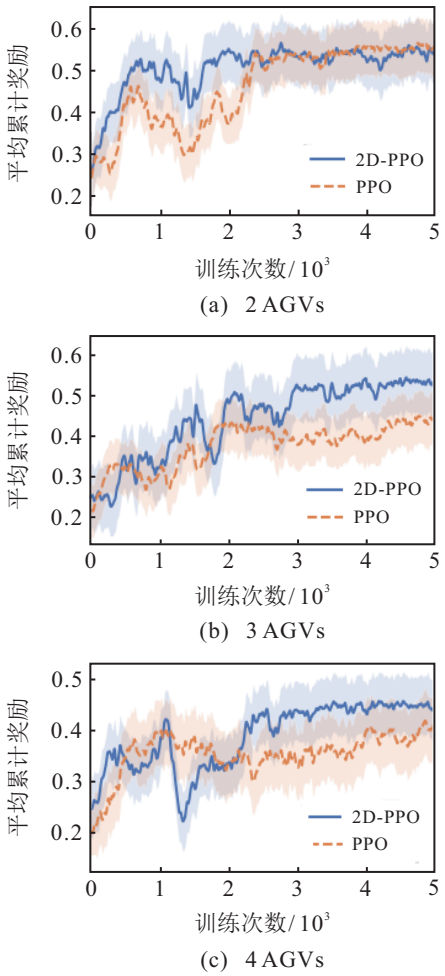


图 3　DRL算法在随机算例上训练的平均累计奖励变化曲线

3.3　小规模扩展性能

本小节中,在一个小规模场景中测试本文提出的2D-PPO调度方法的扩展性能,该场景采用了Bilge等[1]提出的40个算例,这些算例由10个工件集组成,每个工件集在4个不同车间布局中进行加工,车间系统均包含4台机器和2台AGV,其中EX11表示第1组工件集在第1种车间布局中加工.

表1为本文算法与其他方法的对比结果,表中best代表该方法下最好完工时间. MAS表示为Erol等[3]提出的基于多智体系统的调度方法,MIP为Huang[18]在网络优化系统上运行MIP模型得到的结果. 可以看出,与同样基于智能体的MAS相比,本文方法除EX22、EX102和EX103略差于MAS外,在大多数实例的最优解上都有较大的提升,如EX44;相对于优化求解系统对MIP的求解,本文所提算法在求解质量上有一定的不足,但本文所提算法在部分算例上取得了与MIP方法相同或相近的结果;在求解时间方面,求解器的求解效率极低,且对不同的算例的求解时间差异很大,而本文所提2D-PPO的运行时间基本保持在0.01 s左右,具有较高的效率.

表 1 小规模算例的求解结果

EX.	MAS	MIP		2D-PPO	
		best	CPU / s	best	CPU / s
EX11	130	**96**	31	100	**0.008**
EX13	109	**84**	8	87	**0.006**
EX21	143	**100**	731	104	**0.009**
EX23	98	**86**	96	90	**0.011**
EX31	142	**99**	177	113	**0.007**
EX33	103	**86**	7	94	**0.008**
EX41	198	**112**	50 803	132	**0.009**
EX43	155	**89**	3 997	104	**0.011**
EX51	130	**87**	3 119	89	**0.007**
EX53	109	**74**	83	76	**0.007**
EX61	153	**118**	7 927	142	**0.009**
EX63	128	**103**	23	126	**0.009**
EX71	129	**117**	11 235	118	**0.009**
EX73	93	**83**	33 725	90	**0.009**
EX81	196	**151**	15	173	**0.009**
EX83	172	**153**	14	155	**0.011**
EX91	178	**116**	22	133	**0.008**
EX93	119	**105**	10	118	**0.011**
EX101	188	**146**	7 138	170	**0.011**
EX103	158	**137**	291	164	**0.011**
EX12	98	**82**	4	**82**	**0.007**
EX14	168	**103**	28	104	**0.007**
EX22	86	**76**	5	87	**0.008**
EX24	169	**108**	3 699	110	**0.008**
EX32	114	**85**	8	92	**0.008**
EX34	167	**111**	50 803	123	**0.007**
EX42	129	**87**	3 119	107	**0.011**
EX44	242	**121**	22 554	144	**0.011**
EX52	98	**69**	18	77	**0.006**
EX54	168	**96**	176	105	**0.007**
EX62	123	**98**	10	119	**0.011**
EX64	189	**120**	1 760	152	**0.011**
EX72	92	–	–	**85**	**0.008**
EX74	156	–	–	136	**0.009**
EX82	172	**151**	15	**151**	**0.009**
EX84	251	**163**	4 681	193	**0.009**
EX92	123	**102**	10	116	**0.008**
EX94	181	**120**	62	131	**0.008**
EX102	154	**135**	162	161	**0.011**
EX104	246	**157**	79 885	172	**0.011**

3.4 大规模扩展性能

上一节中用2D-PPO对已有算例进行了验证,但算例规模仅为2个AGV和4台机器,最大工件数只有8个,说明性不强. 为了进一步验证2D-PPO调度方法的性能,本文对规模更大、AGV更多的自生成算例(https://github.com/Aihong-Sun/self _GeneIns)进行验证,这些自生成算例涵盖了各种不同的车间布局、AGV数量、工件数和工序数,可以在一定程度上验证算法的性能. 为了比较2D-PPO算法的泛化性能,将其与以下调度规则进行对比: 1) FIFO + FAFS, 先到达机器的工件先加工,优选最先到达的AGV进行运输; 2) LOR + FAFS,剩余工序数量最多的工件优先加工,优选最先到达的AGV进行运输; 3) LRPT + FAFS,剩余加工时间最长的工件优先加工,最先到达的AGV优先选择; 4) FIFO + ST, 先到达机器的工件先加工,优先选择行程时间最短的AGV进行运输; 5) LOR + ST,剩余工序数量最多的工件优先加工,优选行程时间最短的AGV进行运输; 6) LRPT + ST,剩余加工时间最长的工件优先加工,优选行程时间最短的AGV进行运输.

表2显示了不同调度规则和2D-PPO调度方法运行10次得到的最优解,其中“15_8_2”表示车间环境中包含8台机器、2个AGV和15个待加工工件. 图4和图5分别为不同调度方法求解的BPRD值箱型图和APRD值箱型图. 从图4和图5可以看出,无论在最优值上还是平均值上,基于2D-PPO的方法都具有最小偏差,可以表明所提算法的扩展能力. 在求解时间上,所有算例的CPU时间均保持在0.4 s以内,可以看出即使在规模相对大的环境中,基于2D-PPO算法依然能保证较高的求解效率. 但当工件数超过25后,2D-PPO算法的求解能力开始下降. 例如,在算例“25_10_3”上,规则FIFO + FAFS取得的最优完工时间为647,优于本文所提算法求解的最优完工时间648;在算例“25_10_5”和“30_10_7”上,本文算法取得了与规则FIFO + FAFS相同的效果,可见此时基于2D-PPO算法的性能在规模超过一定范围后会出现无法远超规则的现象,保持了与最好调度规则求解质量一致的效果,这与动作空间的第1维动作采用工件选择规则有直接联系,也进一步表明了所提算法的稳健性,防止了更坏求解结果的发生.

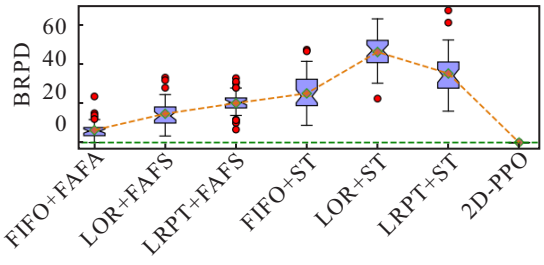


图 4 大规模算例上各调度方法的BRPD箱线图

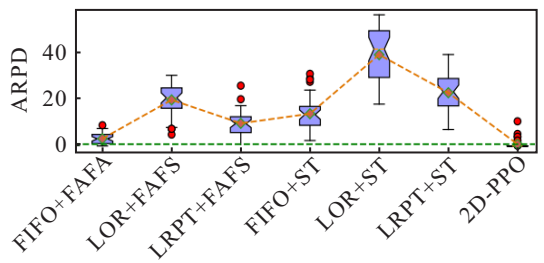


图 5 大规模算例上各调度方法的ARPD箱线图

表 2　大规模算例的求解结果

序号	算例名称	FIFO+FAFS	LOR+FAFS	LRPT+FAFS	FIFO+ST	LOR+ST	LRPT+ST	2D-PPO	CPU/s
1	15_8_2	570	598	593	646	692	720	**531**	**0.124**
2	15_8_3	413	446	458	477	545	518	**383**	**0.142**
3	15_8_4	334	363	394	378	470	456	**318**	**0.107**
4	15_10_2	617	626	662	657	845	754	**551**	**0.156**
5	15_10_3	429	481	510	533	651	608	**399**	**0.123**
6	15_10_4	361	410	421	423	510	520	**346**	**0.157**
7	20_8_2	767	789	813	959	1 024	884	**697**	**0.164**
8	20_8_3	538	560	606	736	783	646	**502**	**0.132**
9	20_8_4	415	430	480	480	611	501	**404**	**0.142**
10	20_10_2	831	822	869	1 036	1 024	962	**775**	**0.192**
11	20_10_3	543	557	601	675	763	750	**505**	**0.191**
12	20_10_4	409	458	519	505	637	568	**396**	**0.183**
13	20_10_5	434	460	488	527	623	560	**410**	**0.163**
14	20_10_6	408	425	461	429	543	472	**386**	**0.177**
15	25_10_2	962	985	1 008	1 205	1 287	1 247	**902**	**0.277**
16	25_10_3	**647**	706	712	870	1 004	861	648	**0.395**
17	25_10_4	516	582	608	686	800	757	**515**	**0.348**
18	25_10_5	**491**	537	569	566	714	615	**491**	**0.289**
19	25_10_6	482	558	572	556	670	622	**472**	**0.252**
20	30_10_2	1 220	1 215	1 254	1 480	1 610	1 590	**1 176**	**0.343**
21	30_10_3	774	793	849	937	1 045	927	**762**	**0.348**
22	30_10_4	**632**	714	773	894	1 014	1 019	**632**	**0.320**
23	30_10_5	621	753	721	805	907	856	**605**	**0.310**
24	30_10_6	592	602	617	713	797	755	**542**	**0.295**
25	30_10_7	580	691	691	598	801	761	**540**	**0.357**

4　结　论

本文针对作业车间AGV和机器的联合调度问题提出了一种基于2D-PPO的强化学习调度方法. 建立了基于析取图的作业车间AGV和机器联合模型,并将其调度过程描述为马尔可夫决策过程;设计了基于析取图的状态特征和自定义特征以反应实际车间环境,并设计了包含工件选择规则和AGV指派的二维动作空间;为实现最小化完工时间,设计了基于问题特征的奖励函数. 通过3个不同的实验,依次证明了2D-PPO算法在求解作业车间AGV与机器联合调度问题上与采用一维动作的PPO相比具有更好的收敛性能和学习效果,在小规模问题上与基于多智体系统的算法相比具有更好的求解质量,在大规模问题上与单一调度规则相比具有更突出的求解性能.

本文目前使用的DRL算法是集中式调度算法,面对大规模制造环境,状态空间的复杂性会呈现爆发式增长的趋势,不利于DRL智能体的学习. 因此,接下来将在当前研究上进一步探索DRL智能体状态和动作的设计,并考虑采用多智能体强化学习的方式进行求解.

参考文献(References)

[1] Bilge Ü, Ulusoy U. A time window approach to simultaneous scheduling of machines and material handling system in an FMS[J]. Operations Research, 1995, 43(6): 1058-1070.

[2] Xie C, Allen T T. Simulation and experimental design methods for job shop scheduling with material handling: A survey[J]. The International Journal of Advanced Manufacturing Technology, 2015, 80(1): 233-243.

[3] Erol R, Sahin C, Baykasoglu A, et al. A multi-agent based approach to dynamic scheduling of machines and automated guided vehicles in manufacturing systems[J]. Applied Soft Computing, 2012, 12(6): 1720-1732.

[4] 耿凯峰, 叶春明. 带工序跳跃的绿色混合流水车间机器与AGV联合调度[J]. 控制与决策, 2022, 37(10): 2723-2732.
(Geng K F, Ye C M. Joint scheduling of machines and AGVs in green hybrid flow shop with missing operations[J]. Control and Decision, 2022, 37(10): 2723-2732.)

[5] Ren W, Yan Y, Hu Y, et al. Joint optimisation for dynamic flexible job-shop scheduling problem with transportation

time and resource constraints[J]. International Journal of Production Research, 2021, 60(18): 5675-5696.

[6] Zhang Y F, Guo Z G, Lv J X, et al. A framework for smart production-logistics systems based on CPS and industrial IoT[J]. IEEE Transactions on Industrial Informatics, 2018, 14(9): 4019-4032.

[7] Guo Z G, Zhang Y F, Zhao X B, et al. CPS-based self-adaptive collaborative control for smart production-logistics systems[J]. IEEE Transactions on Cybernetics, 2021, 51(1): 188-198.

[8] Ham A. Transfer-robot task scheduling in job shop[J]. International Journal of Production Research, 2021, 59(3): 813-823.

[9] Wolpert D H, Macready W G. No free lunch theorems for optimization[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 67-82.

[10] 王凌, 潘子肖. 基于深度强化学习与迭代贪婪的流水车间调度优化[J]. 控制与决策, 2021, 36(11): 2609-2617.
(Wang L, Pan Z X. Scheduling optimization for flow-shop based on deep reinforcement learning and iterative greedy method[J]. Control and Decision, 2021, 36(11): 2609-2617.)

[11] Shi D M, Fan W H, Xiao Y Y, et al. Intelligent scheduling of discrete automated production line via deep reinforcement learning[J]. International Journal of Production Research, 2020, 58(11): 3362-3380.

[12] 朱家政, 张宏立, 王聪, 等. 基于深度强化学习的模糊作业车间调度问题[J]. 控制与决策，DOI: 10.13195/j.kzyjc.2022.1345.
(Zhu J Z, Zhang H L, Wang C, et al. Fuzzy job shop scheduling problem based on reinforcement learning[J]. Control and Decision, DOI: 10.13195/j.kzyjc.2022.1345.)

[13] 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.

[14] Zhang C, Song W, Cao Z G, et al. Learning to dispatch for job shop scheduling via deep reinforcement learning[J/OL]. 2020, arXiv: 2010.12367.

[15] 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, 59(11): 3360-3377.

[16] Wang H, Yu Y. Exploring multi-action relationship in reinforcement learning[C]. Pacific Rim International Conference on Artificial Intelligence. Cham: Springer, 2016: 574-587.

[17] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[J/OL]. 2017, arXiv: 1707.06347.

[18] Huang S Y. Optimization of job shop scheduling with material handling by automated guided vehicle[D]. Ames: Iowa State University, 2018.

作者简介

孙爱红 (1997−), 女, 硕士生, 从事智能优化调度方法的研究, E-mail: sunaihon2021@163.com;

雷琦 (1976−), 女, 副教授, 博士生导师, 从事网络化制造、制造业信息化等研究, E-mail: leiqi@cqu.edu.cn;

宋豫川 (1973−), 男, 教授, 博士生导师, 从事网络化制造、制造业信息化等研究, E-mail: syc@cqu.edu.com;

杨云帆 (1997−), 男, 博士生, 从事智能优化算法、智能制造等研究, E-mail: yyf970816@163.com.