



基于深度强化学习的节能工艺路线发现方法

陶鑫钰, 王艳, 纪志成

引用本文:

陶鑫钰,王艳,纪志成. 基于深度强化学习的节能工艺路线发现方法[J]. 智能系统学报, 2023, 18(1): 23–35.

TAO Xinyu,WANG Yan,JI Zhicheng. Energy-saving process route discovery method based on deep reinforcement learning[J]. *CAA Transactions on Intelligent Systems*, 2023, 18(1): 23–35.

在线阅读 View online: <https://dx.doi.org/10.11992/tis.202112030>

您可能感兴趣的其他文章

基于深度强化学习的动态装配算法

Dynamic assembly algorithm based on deep reinforcement learning

智能系统学报. 2023, 18(1): 2–11 <https://dx.doi.org/10.11992/tis.202201006>

基于强化学习的海洋移动观测网络观测路径规划方法

Path planning for mobile ocean observation network based on reinforcement learning

智能系统学报. 2022, 17(1): 192–200 <https://dx.doi.org/10.11992/tis.202106004>

模糊粒子群优化算法的第四方物流运输时间优化

Convergence fuzzy particle swarm optimization based transportation time optimization of 4PL

智能系统学报. 2021, 16(3): 474–483 <https://dx.doi.org/10.11992/tis.202004032>

多智能体分层强化学习综述

A survey on multi-agent hierarchical reinforcement learning

智能系统学报. 2020, 15(4): 646–655 <https://dx.doi.org/10.11992/tis.201909027>

强化学习的地-空异构多智能体协作覆盖研究

Air-ground heterogeneous coordination for multi-agent coverage based on reinforced learning

智能系统学报. 2018, 13(2): 202–207 <https://dx.doi.org/10.11992/tis.201609017>

DOI: 10.11992/tis.202112030

基于深度强化学习的节能工艺路线发现方法

陶鑫钰^{1,2}, 王艳^{1,2}, 纪志成^{1,2}

(1. 江南大学 轻工过程先进控制教育部重点实验室, 江苏 无锡 214122; 2. 江南大学 物联网工程学院, 江苏 无锡 214122)

摘要: 由于传统基于固定加工环境的工艺路线制定规则, 无法快速响应加工环境的动态变化制定节能工艺路线。因此提出了基于深度 Q 网络 (deep Q network, DQN) 的节能工艺路线发现方法。基于马尔可夫决策过程, 定义状态向量、动作空间、奖励函数, 建立节能工艺路线模型, 并将加工环境动态变化的节能工艺路线规划问题, 转化为 DQN 智能体决策问题, 利用决策经验的可复用性和可扩展性, 进行求解, 同时为了提高 DQN 的收敛速度和解的质量, 提出了基于 S 函数探索机制和加权经验池, 并使用了双 Q 网络。仿真结果表明, 相比较改进前, 改进后的算法在动态加工环境中能够更快更好地发现节能工艺路线; 与遗传算法、模拟退火算法以及粒子群算法相比, 改进后的算法不仅能够以最快的速度发现节能工艺路线, 而且能得到相同甚至更高精度的解。

关键词: 深度强化学习; 深度 Q 网络; 动态加工环境; 工艺路线; 马尔可夫决策过程; 智能体决策; 双 Q 网络; 启发式算法

中图分类号: TP273 文献标志码: A 文章编号: 1673-4785(2023)01-0023-13

中文引用格式: 陶鑫钰, 王艳, 纪志成. 基于深度强化学习的节能工艺路线发现方法 [J]. 智能系统学报, 2023, 18(1): 23-35.

英文引用格式: TAO Xinyu, WANG Yan, JI Zhicheng. Energy-saving process route discovery method based on deep reinforcement learning[J]. CAAI transactions on intelligent systems, 2023, 18(1): 23-35.

Energy-saving process route discovery method based on deep reinforcement learning

TAO Xinyu^{1,2}, WANG Yan^{1,2}, JI Zhicheng^{1,2}

(1. China Key Laboratory of Advanced Process Control for Light Industry Ministry of Education, Jiangnan University, Wuxi 214122, China; 2. School of the Internet of Things Engineering, Jiangnan University, Wuxi 214122, China)

Abstract: Due to the traditional process route formulation rules based on the fixed processing environment, it is unable to quickly respond to the dynamic changes of the processing environment to formulate energy-saving process routes. Therefore, an energy-saving process route discovery method based on deep Q network (DQN) is proposed in this paper. Based on the Markov decision process, we define the state vector, action space, and reward function, establish an energy-saving process route model, and transform the energy-saving process route planning problem with dynamic changes in the processing environment into a DQN agent decision-making problem, which uses the reusable and extensible decision-making experience to solve the problem. At the same time, an exploration mechanism based on the S function, a weighted experience pool, and a double-Q network are used to improve the convergence speed and solution quality of DQN. The simulation results show that compared with that before improvement, the improved algorithm can find energy-saving process routes faster and better in the dynamic processing environment; and compared with genetic algorithm, simulated annealing algorithm, as well as particle swarm algorithm, the improved algorithm can not only discover energy-saving process routes at the fastest speed, but also obtain the same or even higher precision solutions.

Keywords: deep reinforcement learning; deep Q network; dynamic machining environment; process planning; Markov decision process; agent decision making; double Q network; heuristic algorithm

工艺路线表述了零件在加工过程中所使用的

加工资源以及特征的加工顺序等。针对某一零件, 如何在满足工艺约束等条件下, 寻找目标工艺路线, 即是工艺路线规划所需要解决的问题。目前对工艺路线规划的研究主要分为两类: 知识驱动的工

收稿日期: 2021-12-14.

基金项目: 国家重点研发计划项目 (2018YFB1701903).

通信作者: 王艳. E-mail: wangyan88@jiangnan.edu.cn.

©《智能系统学报》编辑部版权所有

艺路线规划, 基于启发式算法的艺路线规划。

基于知识驱动的艺路线规划方法, 利用知识专家系统来推理出合理的艺路线, 虽然提高了艺路线规划的可行性, 但是无法保证获得的解能逼近全局最优, 于是众多专家学者将目光转向启发式算法。

启发式算法能够很好地逼近全局最优解, 主要算法包括: 遗传算法^[1-3]、模拟退火算法^[4-5]、粒子群算法^[6-7]等。其中遗传算法广泛应用于艺路线规划, Vafadar 等^[8]利用遗传算法来寻找机器的最优工艺参数和刀具配置, 进行艺路线规划。Wu 等^[9]将和谐搜索和遗传算法相结合, 利用遗传算法来确定机器的分配和操作顺序, 从而解决艺路线规划问题。Ma 等^[10]借助模拟退火算法来解决多约束艺路线规划问题, 该算法考虑了各零件特征之间的约束关系, 实验证明, 模拟退火算法的全局搜索能力强于遗传算法。启发式算法的解虽然能够逼近全局最优解, 但是其先前获得的解对加工环境发生变化的艺路线搜索没有帮助, 需重新规划, 无法在加工环境发生变化时快速响应。因此, 针对加工环境动态变化的场景, 需要一种具有基于学习的、动态决策的艺路线规划方法, 能够以响应式的处理方式进行艺路线规划。

深度强化学习 (deep reinforcement learning, DRL) 由强化学习和神经网络组成, 已经在多个领域取得了优异的成绩。在飞行领域, DRL 实现了多机协同^[11-12]; 在交通领域, DRL 提高车辆在交通路口的通过效率^[13-15]; 在机器人领域, DRL 提高准被动双足机器人斜坡步行稳定性^[16], 加快了机器人数据学习速率^[17]; 在游戏领域, 由 DRL 实现的 AlphaGO^[18]、AlphaStar^[19]、OpenAIFive^[20] 分别在游戏围棋、星际争霸 II、Dota 2 中战胜了代表人类最高水平的玩家。

深度 Q 网络 (deep Q-network, DQN) 作为 DRL 的一员, 由 Q-Learning 和神经网络组成^[21], 具有当加工环境动态变化时艺路线动态决策的潜力, 具体理由如下: 1) DQN 是一种基于奖励的直面对象的决策方法, 在处理具有约束的艺路线时, 能够在离散可行域中获得高质量的解; 2) DQN 通过计算状态值或动作价值函数, 以网络参数的形式存储和显示顺序决策的结构特征, 能够在处理具有相同决策结构特征的艺路线规划时, 利用先前存储的策略知识, 不仅提高 DQN 的决策速度, 而且使得其具有可扩展性和可重用性; 3) DQN 基于马尔可夫决策过程 (Markov de-

cision process, MDP) 建立模型, 并不需要复杂的数学推导。

因此, 针对基于知识驱动的艺路线规划无法逼近全局最优, 启发式算法无法快速响应加工环境动态变化的问题, 考虑到 DQN 在解决加工环境动态变化时艺路线规划的潜力, 因此本文将 DQN 应用于艺路线规划, 提出了基于 DQN 的节能艺路线规划方法。

1 基于 MDP 节能艺路线建模

本节基于 MDP 建立节能艺路线模型。首先介绍了艺路线特征、艺路线-能耗关联模型, 接着结合 MDP 在艺路线规划下的含义, 分别定义状态向量、动作空间以及奖励函数。

1.1 艺路线特征描述

在制定零件的艺路线时, 需要考虑零件特征之间的工序约束, 例如先面后孔。先粗后细、先基准后其他、夹具要求等。

针对零件特征间的工序约束, 本文设计了特征约束矩阵 \mathbf{P}_{RE} , 来处理零件特征的约束问题。零件 P 的 \mathbf{P}_{RE} 定义为

$$\mathbf{P}_{\text{RE}}(P) = [p_{\text{re}}(p, q)]_{n \times n}$$

式中: n 表示加工零件的特征数, $p_{\text{re}}(p, q)$ 表示约束矩阵的第 p 行、第 q 列的值。 $p_{\text{re}}(p, q) = 1$ 表示特征 F_{p+1} 先于特征 F_{q+1} 加工, $p_{\text{re}}(p, q) = 0$ 表示特征 F_{p+1} 、 F_{q+1} 没有约束关系。

在解决了零件特征之间的约束后, 将从现有的加工资源下, 为特征选取具体的操作。操作由机床、刀具、进刀方向组成, 同一操作由于机床、刀具、进刀方向的不同, 可组合成多个选项, 而这些选项则构成了操作集合, C_i 表示为操作 O_i 的操作集合, 定义为

$$C_i = \{c_{i1}, c_{i2}, \dots, c_{ij}, \dots, c_{ip}\}$$

式中 c_{ij} 表示 O_{pi} 的第 j 个选项。在艺路线规划中, O_i 有且仅能从 C_i 中选取一个 c 。 c_{ij} 定义为

$$c_{ij} = \{m_{ij}, t_{ij}, s_{ij}\}$$

式中: m_{ij} 、 t_{ij} 、 s_{ij} 分别表示 C_i 的第 j 个选项中的机床、刀具以及进刀方向。

1.2 艺路线-能耗关联模型

艺路线总能耗 T_{EC} 定义为

$$T_{\text{EC}} = D_{\text{EC}} + T_{\text{CEC}}$$

式中: D_{EC} 表示设备能耗, T_{CEC} 表示总切换能耗。

1) 设备能耗

D_{EC} 由加工零件 P 时所用设备的总能耗组成, 定义为

$$D_{\text{EC}} = M_C + T_C$$

式中: M_C 、 T_C 分别表示加工零件 P 时, 机床、刀具消耗的总能量。

M_C 、 T_C 定义分别为

$$M_C = \sum_{i=1}^t M_{C_i}$$

$$T_C = \sum_{i=1}^t T_{C_i}$$

式中: t 表示零件 P 的操作个数; M_{C_i} 、 T_{C_i} 分别表示执行第 i 个操作时, 机床、刀具消耗的能量。

2) 总切换能耗

在执行当前操作 b 前, 由于切换了加工资源, 与上一操作 a 所使用的加工资源不同, 进而产生了切换能耗 C_{EC} , 定义为

$$C_{EC_{ab}} = M_{CE_{ab}} + T_{CE_{ab}} + D_{CE_{ab}}$$

式中: $M_{CE_{ab}}$ 表示机床切换能耗, $T_{CE_{ab}}$ 表示刀具切换能耗, $D_{CE_{ab}}$ 表示进刀方向切换能耗。

机床切换能耗定义为

$$M_{CE_{ab}} = \Omega_1(m_a, m_b) \cdot E^{mc}$$

式中: E^{mc} 表示切换机床时所消耗的能量, m_a 、 m_b 分别表示操作 a 、 b 所选择的机床, $\Omega_1(X, Y) = \begin{cases} 1, & X \neq Y \\ 0, & X = Y \end{cases}$

刀具切换能耗计算公式为

$$T_{CE_{ab}} = \Omega_1(t_a, t_b) \cdot \Omega_2(m_a, m_b) \cdot E^{tc}$$

式中: E^{tc} 表示切换刀具所消耗的能量, t_a 、 t_b 分别表示操作 a 、 b 所选择的刀具, $\Omega_2(X, Y) = \begin{cases} 1, & X = Y \\ 0, & X \neq Y \end{cases}$

进刀方向切换能耗为

$$D_{CE_{ab}} = \Omega_1(d_a, d_b) \cdot \Omega_2(m_a, m_b) \cdot E^{dc}$$

式中: E^{dc} 表示进刀方向改变时所消耗的能量, d_a 、 d_b 分别表示操作 a 、 b 所选择的进刀方向。

总切换能耗由每次执行操作时, 而产生的切换能耗累加而成, 定义为

$$T_{CEC} = \sum_{i=1}^{t-1} C_{EC(i+1)}$$

式中 t 表示加工零件 P 的操作个数。

1.3 工艺路线-MDP

MDP 作为 DQN 的数学模型, 由四元组 $\langle S, A, P, R \rangle$ 构成。结合节能工艺路线定义, 给出状态向量 S 、动作空间 A 和奖励函数 R 的定义。

1.3.1 状态向量

为了使智能体能够从状态向量中得知零件的加工状况, 本文将零件加工状况映射到 MDP 的状态向量, 状态向量定义为 $S = [O_1 O_2 \cdots O_i \cdots]$, 状态向量中特征 O_i 的数量由加工零件的操作个数决定。同时考虑到零件特征之间的约束、每个操作执行且仅执行一次, 因此本文状态向量中的特

征 O_i 有 3 种状况:

状况 0 在当前状态下, 该操作可以被执行;

状况 1 该操作违反了零件特征间的约束, 所以在当前状态下该操作无法被执行;

状况 2 该操作已经执行了, 不能再次选择。

1.3.2 动作空间

动作是连接 MDP 中状态转换的桥梁。本文将零件加工步骤映射到动作空间。动作空间定义为: $A = \{1, 2, \cdots\}$ 。动作空间的动作个数取决于加工零件的操作数, 当选取的动作为 i 时, 表示执行操作 O_i ; 当选取动作表示的操作处于 1、2 状况时, 此动作默认无效, 不会引起状态的变化。

执行操作 O_i 时, 需要从集合 C_i 中选取一个 c , 其选取原则为: 机床、刀具、进刀方向应与前一个操作保持一致, 如果无法保持一致, 则选择加工资源中能耗最小的。

1.3.3 奖励函数

奖励作为强化学习的重要组成部分, 智能体可以通过奖励来判断动作的好坏, 从而进行有效的学习。本文将奖励设计为状态的函数。奖励函数定义为

$$R_{t+1} = \begin{cases} -(D_{EC_t} + C_{EC_{(t-1)Y}}), & s_t \neq s_{t+1} \\ -X, & s_t = s_{t+1} \end{cases}$$

式中: R_{t+1} 表示在执行动作 a_t 后, 智能体所获得的奖励; D_{EC_t} 、 $C_{EC_{(t-1)Y}}$ 表示执行动作 a_t 时, 设备能耗以及切换能耗; $s_t \neq s_{t+1}$, 表示执行动作 a_t 后, 智能体的状态发生变化, 即此动作有效; 而 $s_t = s_{t+1}$ 则表示此动作无效, 显然这样的情况应该避免发生, 所以当出现这种情况时, 对智能体进行惩罚; X 是惩罚系数, $X > \max(D_{EC_t} + C_{EC_{(t-1)Y}})$ 。

2 基于 DQN 的节能工艺路线问题求解

本节将节能工艺路线规划问题转化为 DQN 智能体决策问题, 利用决策经验的可复用性和可扩展性, 进行求解。同时为了提高 DQN 的收敛速度和解的质量, 提出了基于 S 函数探索机制和加权经验池技术, 并使用双 Q 网络改进 DQN。

2.1 节能工艺路线规划问题求解

结合 1.3 节 MDP 的含义, 本文将节能工艺路线规划问题转化为 DQN 智能体决策问题, 利用决策经验的可复用性和可扩展性进行求解。如图 1 所示, 智能体在加工零件状况为 s_t 下, 通过决策从动作空间 $A = \{1, 2, \cdots, k\}$ 中, 选取动作 (操作) a_t , 零件进入新的状态 s_{t+1} , 同时, 零件根据动作的优劣反馈给智能体奖励 R_{t+1} 。DQN 根据以上产生的决策经验 $\langle s_t, a_t, R_{t+1}, s_{t+1} \rangle$ 进行学习, 使其获得

的奖励最大,从而获得节能工艺路线。

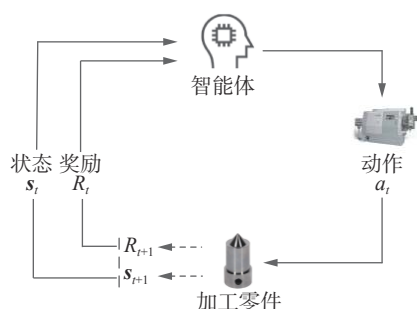


图 1 DQN 模型
Fig. 1 DQN model

2.2 改进 DQN

2.2.1 基于 S 函数的探索机制

在强化学习中,“利用”表示执行当前能获得奖励最大的动作,“探索”是指牺牲眼前最大奖励,去探索可能存在更大奖励的区域。“利用”和“探索”存在博弈,如果只使用“利用”,则有可能陷入局部最优解;如果只使用“探索”,算法则可能无法收敛。

为了解决这个问题,DQN 算法采用 ε -greedy 策略 G ,策略 G 定义为

$$G = \begin{cases} 1, & \varepsilon > K \\ 0, & \varepsilon \leq K \end{cases} \quad (1)$$

式中:1 表示“探索”;0 表示“利用”; $K \in [0, 1]$; $\varepsilon \in [0, 1]$,与随机值 K 不同的是,贪婪值 ε 会随着迭代次数改变而改变。

常规 DQN 算法贪婪值 ε 会随着迭代次数线性减小,虽然避免了陷入局部最优解,但可能会导致智能体在迭代中后期仍出现“探索”,使经验池出现意想不到的经验,导致神经网络学习次数增加甚至出现已经收敛的神经网络发散的现象,神经网络的收敛速度变慢,从而使智能体学习时间变长。

为了加快 DQN 算法的收敛速度,同时避免算法陷入局部最优解,本文提出了基于 S 函数的探索机制,使得在算法运行前期,智能体尽可能地使用“探索”,增加经验池中样本的多样性,避免陷入局部最优解,在算法运行中后期,智能体尽可能地使用“利用”,避免样本池中出现意想不到的样本,加快算法的收敛速度,缩短学习时间。

为了达到预期目标,定义 S 函数:

$$\varepsilon = -\frac{1}{1 + e^{-\left(\frac{15\rho}{v} - s\right)}} + 1 \quad (2)$$

式中: ε 是贪婪值, v 是算法迭代总次数, ρ 是当前迭代次数,切换权值 s 用于控制智能体使用“探索”和“利用”的权重。当 $v = 10, s = 6$ 时, ε 的取值情况如图 2 所示,图中红点 K 值表示随机值 K 在

0 到 1 之间随机取值的情况。由式 (1) 可知,在某次迭代中,贪婪值大于红点 K 值智能体使用“探索”,否则使用“探索”,所以红点位于曲线之下则表示“探索”,位于曲线之上或在曲线上表示“利用”。从图中可以看出,在迭代前期即前 3 次迭代,红点均位于 S 型探索曲线下方,反观线性探索曲线,在迭代一开始,就有一个红点位于线性探索曲线上方,即算法执行“利用”,显然这样极易导致算法陷入局部最优解。在迭代中后期,即 4~10 次迭代,所有的红点均在 S 型探索曲线上方,而在线性探索曲线中,依旧出现了“探索”与“利用”交替的情况,即红点交叉位于曲线上下。在迭代前期,S 型探索探索率为 100%,线性探索探索率为 66.7%;在迭代中后期,S 型探索利用率为 100%,线性探索利用率为 71.43%。显然在迭代前期,S 型探索的探索率大于线性探索,在迭代中后期,S 型探索的利用率大于线性探索,导致这样的结果是因为在迭代前期,位于 S 型探索曲线以下的面积大于线性探索,即紫色区域,在迭代中后期,位于 S 型探索曲线以上的面积大于线性探索,即绿色区域。可以通过式 (2) 中的切换权值 s 来调整紫色区域和绿色区域的大小,从而调整“探索”与“利用”在 DQN 算法中的权重,提高解的确定性,加快算法收敛速度。

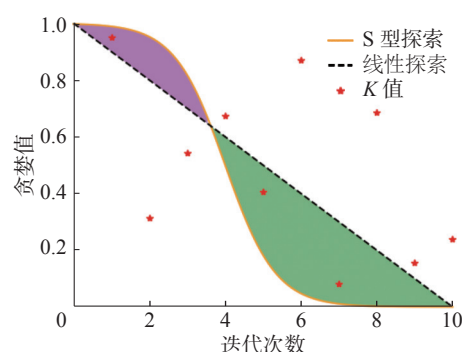


图 2 ε 取值情况
Fig. 2 Value of ε

2.2.2 加权经验池

DQN 算法为了保证神经网络收敛使用经验回放技术。该技术设置了容量为 D 的经验池,用于保存智能体与环境交互过程中生成的经验,每一个经验由四元组 $\langle s_t, a_t, R_{t+1}, s_{t+1} \rangle$ 构成。在神经网络学习的过程中,将随机从经验池中选取一定数量 M 的经验,从而打破经验间的高相关度。

随着零件特征数的增加,特征间的约束愈加复杂,智能体可能会花费大量时间才会获得一条合理的工艺路线,且导致经验池中存在大量违反零件加工约束的经验,这些错误的经验很容易将经验池“撑爆”,导致经验溢出,在 DQN 算法中,

当发生经验溢出时, 常将经验池中的第一条经验删除, 但这种不加思考的删除, 极易容易将关键经验删除, 导致智能体学习速率变慢。同时在学习开始, 由于经验池中仅存在少量且合理的经验, 导致智能体很难获得有效经验, 使得智能体将花费大量精力来获取有效经验, 极大地降低了智能体的学习速率。

为了避免删除关键经验和提高智能体对少量且合理经验的利用率, 本文提出加权经验池技术, 对经验池中的每一条经验赋予权重。文献[23]认为神经网络预测值与真实值误差越大, 越值得智能体学习, 但其没有考虑到经验的合理性, 如果两条经验神经网络的预测值和误差值相同, 但前者违反了零件之间的特征约束, 而后者没有, 显然, 智能体应该优先学习没有违反零件特征约束的经验, 提高智能体规避零件特征约束的能力, 加快智能体学习速率。因此, 针对经验的合理性, 定义了优势权值 g , 当经验中的奖励大于负的惩罚系数 X 时, 即经验中的操作没有违反零件操作约束, 智能体应该学习, 增大其权值, 加上优势权值 g , 提高智能体对该经验的利用率。

经验池中第 i 条经验 E_i 的权重为 w_i :

$$w_i = \begin{cases} |\text{loss}_i| + l + g, & R_i > -X \\ |\text{loss}_i| + l, & R_i \leq -X \end{cases} \quad (3)$$

式中: $|\text{loss}_i|$ 表示神经网络的输出值与目标值差值的绝对值; $l \in (0, 0.001]$, 作用是防止 $|\text{loss}_i|$ 为 0 减小经验的多样性; R_i 为经验 E_i 中奖励大小; X 表示惩罚系数; g 表示优势权值, $g \in (0, 1)$ 。

在经验池中, 经验 E_i 被采样选中的概率为

$$P_i = \frac{w_i^o}{\sum_{j=1}^D w_j^o} \quad (4)$$

式中: w_i 表示经验 E_i 的权值; D 表示经验池总容量; 优先权值 o , 表示经验采样的优先情况, $o \in [0, 1]$ 。当 $o = 0$ 时, 优先经验采样就变为了随机经验采样。

2.2.3 双 Q 网络

Hasselt 等^[24]指出 DQN 算法的目标值计算公式往往会使智能体过高地估计动作价值, 导致智能体有时学习到不切实际的动作价值, 且随着动作数的增加过估计现象愈加严重, 为了解决此问题, 提出了双 Q 网络技术。

双 Q 网络技术主要将 DQN 算法中动作的选择和 Q 值的评估分别使用估计网络和目标网络来完成, 从而实现选择与评估的解耦, 其中估计网络与目标网络结构完全相同, 为了减小因策略不收敛而引起的误差。相比较 DQN 算法, 双 Q 网络技术仅将目标值计算公式进行改写:

$$y = R_{t+1} + \gamma Q_t(s_{t+1}, \arg\max_{a' \in A} Q_e(s_{t+1}, a'; \theta); \theta^-)$$

式中: R_{t+1} 表示 t 时刻智能体所获得的奖励; γ 为衰减因子; Q_t 为目标网络; Q_e 为估计网络; θ^- 、 θ 分别为其权值; $\arg\max_{a' \in A} Q_e(s_{t+1}, a'; \theta)$ 表示智能体在状态 s_{t+1} 下通过 Q_e 网络估计行为价值来贪婪地选择动作 a' , 使得 Q_e 值最大, 接着使用 Q_t 网络合理地评估动作价值。

2.3 改进后的 DQN——SWDDQN

综上, 最终获得改进后的 DQN: 基于 S 函数和加权经验池的深度双 Q 网络 (deep double Q network based on S-function and weighted experience pool, SWDDQN), SWDDQN 算法伪代码如下所示:

输入 Q_t 网络权值更新周期 N , 迭代次数 episodes, 经验池容量 D , 经验暂存区容量 M , 切换权值 s , 惩罚系数 X 。

初始化 经验池容量 D , 容量为 M 的经验暂存区 L , 网络权值 θ 、 θ^- 。

1) for $i = 1$: episodes do

2) 根据零件加工状态, 初始化状态向量 s_t

3) while 零件没有加工完成 do

4) 计算 $\varepsilon = 1 - 1 / \left\{ 1 + \exp \left[- \left(\frac{15\rho}{v} - s \right) \right] \right\}$, 根据 ε 值选取随机动作 a_t 或 $a_t = \arg\max_a Q_e(s_t, a; \theta)$

5) 执行动作 a_t , 获取零件新状况 s_{t+1} 以及零件反馈的奖励 R_{t+1}

6) 若经验池容量已满, 则选取经验 $E_i \sim P(i)_{\min} = w_i^o / \sum_j w_j^o$, 并删除该经验

7) 经验 $\langle s_t, a_t, R_{t+1}, s_{t+1} \rangle$ 存入经验池 D

8) 更新经验池中经验的权重与被采样概率

9) 更新状态向量 $s_t \leftarrow s_{t+1}$

10) 从经验池中获取数量为 M 的经验放入 L

11) 根据 L 中的经验计算目标值:

$$y_i = R_{t+1} + \gamma Q_t(s_{t+1}, \arg\max_{a' \in A} Q_e(s_{t+1}, a'; \theta); \theta^-)$$

12) 根据函数: $J(\theta) = E_\pi[(y_i - Q_e(s_i, a_i; \theta))^2]$, 使用梯度下降更新权值 θ

13) 每隔 N 步, 更新 Q_t 网络权重: $\theta^- \leftarrow \theta$

14) end while

15) end for

将 SWDDQN 应用在节能工艺路线规划中, 其机理如图 3 所示, 主要分为决策模块和学习模块。在决策模块中, 智能体根据加工零件初始化状态向量 s_t , 并根据 S 函数探索机制决定探索或者利用, 从而选择动作 a_t , 加工零件根据动作的优劣反馈奖励 R_{t+1} , 同时更新自身状态进入新状态 s_{t+1} , 状态向量更新 $s_t = s_{t+1}$, 此时, 经验 E_i 生成, 经验 E_i 根据式 (3) 获得权重 w_i , 并存放中加权经验池中。在学习模块中, 从加权经验池中, 根据式

(4) 选取一批经验, 使用双 Q 网络计算目标值, 使用梯度下降更新估计网络 Q_e 的权值 θ , 并每隔 N 步, 将估计网络 Q_e 权值 θ 赋值目标网络 Q_t 权值 θ , 更新目标网络 Q_t 。智能体根据学习模块的

估计网络 Q_e 更新自己, 接着再根据加工零件的新状态 s_{t+1} 结合 S 函数探索机制选取动作, 重复以上步骤, 直到零件加工完成。智能体选取的动作组成了加工零件的工艺路线。

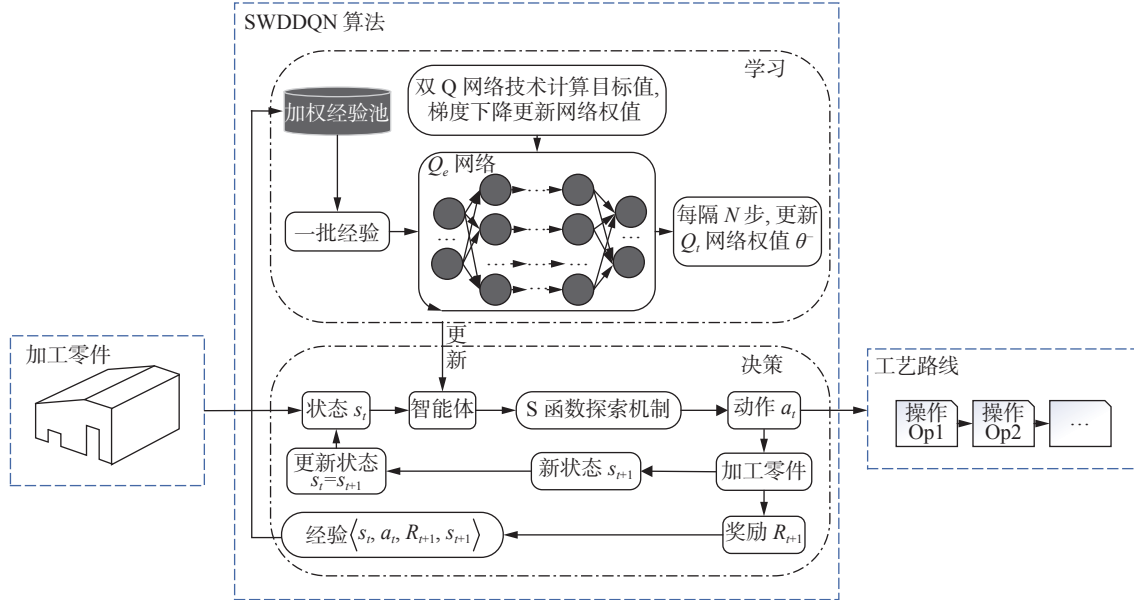


图 3 SWDDQN 算法机理图

Fig. 3 SWDDQN algorithm mechanism diagram

3 实验与分析

为了验证本文改进的 SWDDQN 在工艺路线规划中的有效性和优越性, 进行计算机仿真实验。模拟复杂生产系统加工环境动态变化, 将 SWDDQN 与 DQN 对比, 验证算法在加工环境变动时工艺路线规划的能力, 同时与遗传算法 (genetic Algorithm, GA)、模拟退火算法 (simulated Annealing, SA)、蚁群算法 (ant colony system, ACO) 进行对比实验, 验证 SWDDQN 搜索节能工艺路线的速度和精度。本文实验均运行在 CPU 为 Intel(R) Core(TM) i7-8 700 CPU @3.20GHZ、GPU 为 NVIDIA GeForce GTX 960M、内存为 16 GB 的计算机上运行。

3.1 实验设计

在复杂生产系统中, 加工环境时常动态变化, 因此, 在本文仿真实验中, 随机选取 0~1 台机床、0~1 个刀具, 使其发生故障无法使用, 来模拟复杂生产系统加工环境动态变化, 使用 DQN、SWDDQN 算法对目标零件进行工艺路线规划, 寻找节能工艺路线。算法执行流程如图 4 所示, DQN 或 SWDDQN 根据加工零件的操作执行状态, 选择动作 a_t , 零件加工, 同时更新操作的执行状态, 重复以上步骤, 直到零件加工完成, 选取的动作构成了零件的工艺路线。

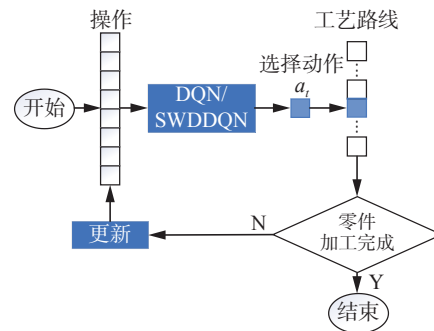


图 4 算法执行流程图

Fig. 4 Algorithm execution flow chart

本文以文献 [25] 给出的零件 P_2 为目标工件, P_2 如图 5, 特征详情如表 1 所示, P_2 共 19 个特征, 其中特征 F_7 、 F_9 需要多个操作配合完成, P_2 的操作约束如表 2 所示, 加工资源及成本如表 3 所示, 其中 $E^{mc} = 300 \text{ kJ}$ 、 $E^{tc} = 10 \text{ kJ}$ 、 $E^{dc} = 90 \text{ kJ}$ 。

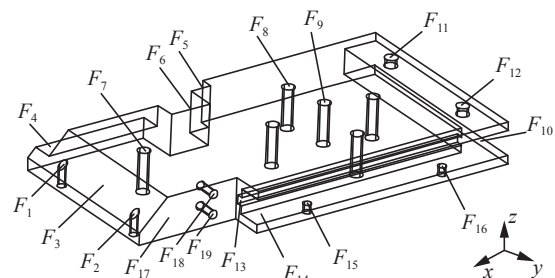


图 5 零件 P_2

Fig. 5 Part P_2

表 1 零件 P_2 特征详情
Table 1 Part P_2 feature details

特征	操作类型	机床编号	刀具编号	进刀方向
F_1	钻孔(O_1)	M_1, M_2, M_3, M_7	T_1, T_9	$+z, -z$
F_2	钻孔(O_2)	M_1, M_2, M_3, M_7	T_1, T_9	$+z, -z$
F_3	铣削(O_3)	M_2, M_3, M_6	T_7	$+z, -z$
F_4	铣削(O_4)	M_2, M_3, M_5, M_6	T_5, T_6	$+y, -z$
F_5	铣削(O_5)	M_2, M_3	T_5, T_6	$+y$
F_6	铣削(O_6)	M_2, M_3	T_5, T_6	$+y$
	钻孔(O_7)	M_1, M_2, M_3, M_7	T_2, T_9	$+z, -z$
F_7	铰孔(O_8)	M_1, M_2, M_3	T_3	$+z, -z$
	钻孔(O_9)	M_3, M_4, M_7	T_4, T_9	$+z, -z$
F_8	钻孔(O_{10})	M_1, M_2, M_3, M_7	T_1, T_9	$-z$
	钻孔(O_{11})	M_1, M_2, M_3, M_7	T_2, T_9	$+z, -z$
F_9	铰孔(O_{12})	M_1, M_2, M_3	T_3	$+z, -z$
	钻孔(O_{13})	M_3, M_4, M_7	T_4, T_9	$+z, -z$
F_{10}	铣削(O_{14})	M_2, M_3, M_5, M_6	T_5, T_6	$+x$
F_{11}	钻孔(O_{15})	M_1, M_2, M_3, M_7	T_1, T_9	$-z$
F_{12}	钻孔(O_{16})	M_1, M_2, M_3, M_7	T_1, T_9	$-z$
F_{13}	铣削(O_{17})	M_2, M_3	T_5, T_8	$-y, -z$
F_{14}	铣削(O_{18})	M_2, M_3, M_5, M_6	T_5, T_6	$-y, -z$
F_{15}	钻孔(O_{19})	M_1, M_2, M_3, M_7	T_1, T_9	$+z, -z$
F_{16}	钻孔(O_{20})	M_1, M_2, M_3, M_7	T_1, T_9	$+z, -z$
F_{17}	铣削(O_{21})	M_2, M_3, M_5, M_6	T_5, T_6	$-y$
F_{18}	钻孔(O_{22})	M_1, M_2, M_3, M_7	T_1, T_9	$-y$
F_{19}	钻孔(O_{23})	M_1, M_2, M_3, M_7	T_1, T_9	$-y$

表 2 零件 P_2 操作约束
Table 2 Part P_2 operational constraints

操作	优先于	操作	优先于
O_1	O_2, O_3	O_{13}	O_{10}
O_2	O_3	O_{14}	$O_3, O_4, O_{17}, O_{18}, O_{19}, O_{20}, O_{21}, O_{22}, O_{23}$
O_3	无	O_{15}	O_{16}
O_4	O_3	O_{16}	无
O_5	$O_4, O_6, O_{15}, O_{16}, O_{17}, O_{18}, O_{19}, O_{20}, O_{21}, O_{22}, O_{23}$	O_{17}	无
O_6	$O_4, O_{15}, O_{16}, O_{17}, O_{18}, O_{19}, O_{20}, O_{21}, O_{22}, O_{23}$	O_{18}	O_{17}, O_{19}

续表 2

操作	优先于	操作	优先于
O_7	$O_8, O_9, O_{10}, O_{11}, O_{12}, O_{13}$	O_{19}	无
O_8	$O_9, O_{10}, O_{11}, O_{12}, O_{13}$	O_{20}	无
O_9	$O_{10}, O_{11}, O_{12}, O_{13}$	O_{21}	$O_1, O_2, O_3, O_7, O_8, O_9, O_{10}, O_{11}, O_{12}, O_{13}, O_{17}, O_8, O_{19}, O_{22}, O_{23}$
O_{10}	无	O_{22}	无
O_{11}	O_{10}, O_{12}, O_{13}	O_{23}	无
O_{12}	O_{10}, O_{13}		

表 3 加工资源和成本
Table 3 Processing resources and costs kJ

机床编号	类型	能耗	刀具编号	类型	能耗
1	钻床	40	1	钻孔	3
2	立式数控铣削	65	2	立铣	3
3	立铣	35	3	铰刀	8
4	镗床	50	4	镗刀	15
5	刨床	45	5	铣刀1	10
6	磨床	40	6	铣刀2	15
7	车床	45	7	倒角刀	10
			8	起槽刀	10
			9	车刀	14

3.2 实验结果与分析

3.2.1 神经网络层数敏感性实验

在进行 DQN、SWDDQN 算法性能对比前, 为了选取合适的神经网络层数, 开展神经网络层数敏感性实验。在此实验中, 假设加工环境没有发生变动, 使用 DQN 算法对零件 P_2 进行工艺路线规划, 实验全程使用“利用”。实验结果如图 6 所示。

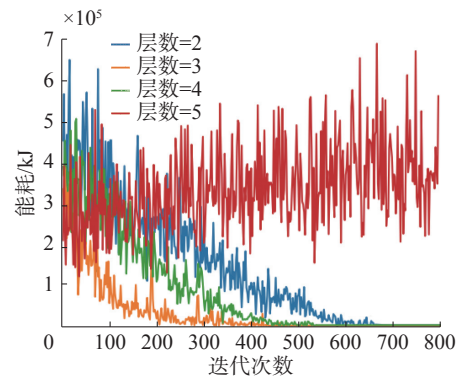


图 6 神经网络层数敏感图

Fig. 6 Neural network layer sensitive map

从图中可以看出, 当神经网络由 2 层增加到 3 层时, 算法的收敛速度加快; 但随着神经网络由

3 层增加到 4 层时,由于神经网络层数的增加,使得学习难度增加,导致算法的收敛速度减缓;当神经网络层数由 4 层增加到 5 层时,因为梯度消失,所以算法不再收敛,曲线发散。因此综合考虑算法的收敛性和收敛速度,神经网络层数选择 3 层比较合理。所以在本文实验中,DQN、SWD-DQN 算法均选择 3 层神经网络。

3.2.2 智能体学习实验

模拟复杂生产系统加工环境变动,随机选取 0~1 台机床、0~1 个刀具,使其故障不可用,使用 DQN、SWDDQN 算法对目标工件 P_2 进行工艺路线规划,寻找节能工艺路线。DQN、SWDDQN 算法模型参数相同,如表 4 所示。两种模型学习后,能耗对比如图 7 所示。

表 4 模型参数
Table 4 Model parameter

参数	参数含义	取值
N	Q 网络权值更新周期	50
episodes	迭代次数	700
D	经验池容量	20000
M	经验暂存区容量	32
s	切换权值	2
X	惩罚系数	600

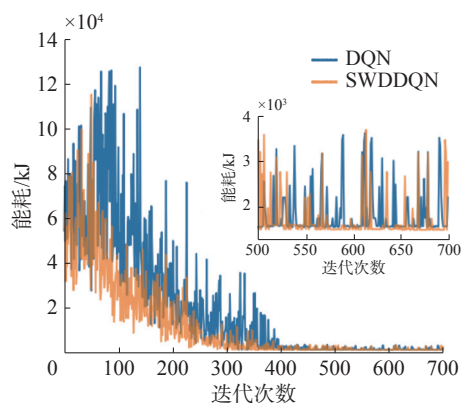


图 7 模型能耗对比

Fig. 7 Model energy comparison

因为 SWDDQN 算法使用了基于 S 型函数探索机制,所以从图 7 中可以看出在迭代中后期,SWDDQN 算法能耗突变相比较 DQN 算法明显减少,如在 220、240 等迭代次数时,DQN 算法能耗发生突变,而 SWDDQN 算法则没有,即 SWDDQN 解的确定性高于 DQN 算法。又因为 SWDDQN 算法使用了双 Q 网络技术,智能体会避免高估动作价值,所以从图 7 中可以看出 SWDDQN 算法获得的能耗小于 DQN 算法。从图 7 的局部放大图中可以看出,由于加工环境发生变动,所以模

型学习过程中,智能体在迭代后期,能耗在一定范围内震荡,但并未出现突变点,即智能体具有规避特征约束的能力。

在学习过程中,DQN、SWDDQN 的智能体运行总步数对比如图 8 所示。因为 DQN、SWDDQN 算法在学习前期几乎都在“探索”,所以两者智能体的运行总步数大致相同。但随着学习的进行,SWDDQN 算法因为使用了加权经验池技术,相比较 DQN 算法能更快速地找到关键经验进行学习,所以 SWDDQN 算法智能体的运行总步数远远小于 DQN 算法。

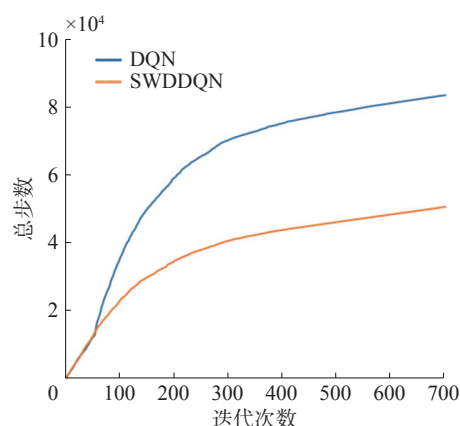


图 8 模型总步数对比

Fig. 8 Comparison of total steps of the model

模拟复杂生产系统加工环境变动,在 DQN、SWDDQN 学习过程中,神经网络损失对比如图 9 所示。由于 DQN、SWDDQN 算法学习前期几乎都在探索,模型还没有开始学习,所以损失为零。但随着学习次数的增加,模型损失开始变化。从图 9 中可以看出,DQN 神经网络大概在迭代 370 次收敛,模型完成学习;SWDDQN 神经网络大概在迭代 330 次收敛,模型完成学习。从图 9 中,可以看出 SWDDQN 神经网络收敛速度快于 DQN 算法。

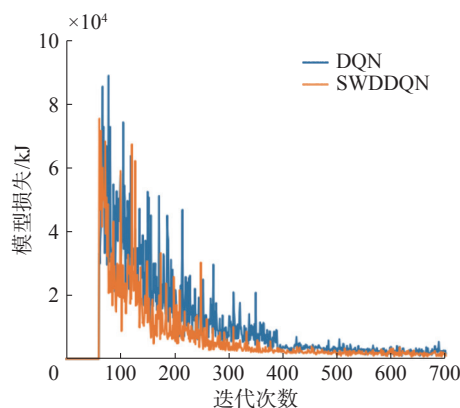


图 9 神经网络损失对比

Fig. 9 Neural network loss comparison

表 7 案例 1: DQN 算法最优工艺路线

Table 7 Case 1: DQN algorithm optimal process route[illegible]

表 8 案例 2: SWDDQN 算法最优工艺路线

Table 8 Case 2: SWDDQN algorithm optimal process route

[illegible]

表 9 案例 2: DON 算法最优工艺路线

Table 9 Case 2: DON algorithm optimal process route

[illegible]

3.3 算法性能对比

为了验证 SWDDQN 搜索节能工艺路线的速度和精度,与启发式算法 GA、SA 以及 ACO 算法进行对比,以案例 1 为例,GA、SA 以及 ACO 算法对零件 P_2 进行工艺路线规划,算法获得最优工艺路线如表 10~12 所示,能耗收敛曲线如图 12 所示,算法收敛时其迭代次数及能耗详情如表 13 所示。从图 12、表 13 中可以看出,在 4 种算法中,SWDDQN 算法收敛速度最快,迭代 33 次收敛;而在 3 种启发式算法中,ACO 算法收敛速度最快,迭代 56 次收敛,GA 算法收敛速度第 2,迭代 78 次收敛,而 SA 算法收敛速度最慢,迭代 95 次收敛。在 4 种算法获得最优解中,SWDDQN 算法

能耗 1412 kJ 最低;在 3 种启发式算法中,ACO 算法收敛速度虽然最快,但收敛的解能耗 1542 kJ 最高,而 GA 算法收敛速度虽然慢于 ACO 算法,但收敛解的能耗 1502 kJ 低于 ACO 算法,SA 算法收敛速度虽然最慢,但收敛的解能耗 1412 kJ 最小,同时也验证了文献 [10] 的结论:SA 算法的全局搜索能力强于 GA 算法。从整体看来,SWD-DQN 模型泛化后不仅能够以最快地速度发现节能工艺路线,而且有相同甚至更高精度的解。综上,SWDDQN 不仅能够处理加工环境动态变化的节能工艺路线规划问题,而且能快速准确地获取与启发式算法精度相同甚至更高的节能工艺路线,体现了 SWDDQN 的优越性。

表 10 GA 最优工艺路线

Table 10 GA optimal process route

[illegible]

表 11 SA 最优工艺路线
Table 11 SA optimal process route

序号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
操作	14	5	6	4	21	18	17	22	23	1	2	19	20	7	8	3	9	11	12	13	10	15	16
机床	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
刀具	5	5	5	5	5	5	5	1	1	1	1	1	1	2	3	7	9	2	3	9	1	1	1
进刀方向	+x	+y	+y	+y	-y	-y	-y	-y	-y	+z	+z	+z	+z	+z	+z	+z	+z	+z	+z	+z	-z	-z	-z

表 12 ACO 算法最优工艺路线
Table 12 ACO algorithm optimal process route

序号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
操作	14	5	6	4	15	16	20	21	18	17	22	23	1	2	19	7	8	3	9	11	12	13	10
机床	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
刀具	6	6	6	6	1	1	1	5	5	5	1	1	1	1	1	2	3	7	9	2	3	9	1
进刀方向	+x	+y	+y	+y	-z	-z	-z	-y	-y	-y	-y	-y	+z	+z	+z	+z	+z	+z	+z	+z	+z	+z	-z

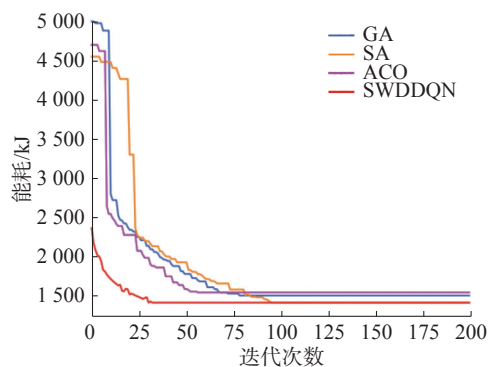


图 12 算法性能对比

Fig. 12 Algorithm performance comparison

表 13 算法性能对比详情

Table 13 Algorithm performance comparison details

	GA	SA	ACO	SWDDQN
迭代次数	78	95	56	33
能耗/kJ	1502	1412	1542	1412

4 结束语

本文将 DQN 算法应用于工艺路线规划问题, 提出了基于 DQN 的节能工艺路线发现方法。基于 MDP 建立节能工艺路线模型, 将节能工艺路线规划问题转化为智能体决策问题进行求解, 为了提高 DQN 的收敛速度和解的质量, 使用基于 S 型函数探索机制、加权经验池、双 Q 网络技术来改进算法。以实际加工零件为例, 模拟加工环境变动, 对目标工件进行工艺路线规划, 同时与多种启发式算法对比, 验证所提算法的优越性。

未来将关注多零件并行加工工艺路线规划问题, 同时考虑加工能耗与加工时间, 将强化学习与图神经网络相结合, 来解决此类问题。

参考文献:

- [1] HALIM A H, ISMAIL I. Combinatorial optimization: comparison of heuristic algorithms in travelling salesman problem[J]. *Archives of computational methods in engineering*, 2019, 26(2): 367–380.
- [2] REZOUQ A, BADER-EL-DEN M, BOUGHACI D. Guided genetic algorithm for the multidimensional knapsack problem[J]. *Memetic computing*, 2018, 10(1): 29–42.
- [3] KIEFFER E, DANOY G, BRUST M R, et al. Tackling large-scale and combinatorial bi-level problems with a genetic programming hyper-heuristic[J]. *IEEE transactions on evolutionary computation*, 2020, 24(1): 44–56.
- [4] 陈科胜, 鲜思东, 郭鹏. 求解旅行商问题的自适应升温模拟退火算法[J]. *控制理论与应用*, 2021, 38(2): 245–254.
CHEN Kesheng, XIAN Sidong, GUO Peng. Adaptive heating simulation annealing algorithm for solving the traveling salesman problem[J]. *Control theory & applications*, 2021, 38(2): 245–254.
- [5] 何庆, 吴意乐, 徐同伟. 改进遗传模拟退火算法在 TSP 优化中的应用[J]. *控制与决策*, 2018, 33(2): 219–225.
HE Qing, WU Yile, XU Tongwei. Improve the application of genetic simulation annealing algorithm in TSP optimization[J]. *Control and decision*, 2018, 33(2):

- 219–225.
- [6] JOY J, RAJEEV S, ABRAHAM E C. Particle swarm optimization for multi resource constrained project scheduling problem with varying resource levels[J]. *Materials today:proceedings*, 2021, 47: 5125–5129.
- [7] PETROVIĆ M, VUKOVIĆ N, MITIĆ M, et al. Integration of process planning and scheduling using chaotic particle swarm optimization algorithm[J]. *Expert systems with applications*, 2016, 64: 569–588.
- [8] VAFADAR A, HAYWARD K, TOLOUEI-RAD M. Drilling reconfigurable machine tool selection and process parameters optimization as a function of product demand[J]. *Journal of manufacturing systems*, 2017, 45: 58–69.
- [9] WU Xiuli, LI Jing. Two layered approaches integrating harmony search with genetic algorithm for the integrated process planning and scheduling problem[J]. *Computers & industrial engineering*, 2021, 155: 107194.
- [10] MA G H, ZHANG Y F, NEE A Y C. A simulated annealing-based optimization algorithm for process planning[J]. *International journal of production research*, 2000, 38(12): 2671–2687.
- [11] 施伟, 冯旻赫, 程光权, 等. 基于深度强化学习的多机协同空战方法研究 [J]. *自动化学报*, 2021, 47(7): 1610–1623.
- SHI Wei, FENG Yanghe, CHENG Guangquan, et al. Research on multi-aircraft collaborative air combat method based on deep reinforcement learning[J]. *Acta automatica sinica*, 2021, 47(7): 1610–1623.
- [12] ZHOU Wenhong, LIU Zhihong, LI Jie, et al. Multi-target tracking for unmanned aerial vehicle swarms using deep reinforcement learning[J]. *Neurocomputing*, 2021, 466: 285–297.
- [13] 王云鹏, 郭戈. 基于深度强化学习的有轨电车信号优先控制 [J]. *自动化学报*, 2019, 45(12): 2366–2377.
- WANG Yunpeng, GUO Ge. Tram signal priority control based on deep reinforcement learning[J]. *Acta automatica sinica*, 2019, 45(12): 2366–2377.
- [14] GUO Ge, WANG Yunpeng. An integrated MPC and deep reinforcement learning approach to trans-priority active signal control[J]. *Control engineering practice*, 2021, 110(5): 104758.
- [15] PENG Bile, KESKIN M F, Kulcsár B, et al. Connected autonomous vehicles for improving mixed traffic efficiency in unsignalized intersections with deep reinforcement learning[J]. *Communications in transportation research*, 2021, 1: 100017.
- [16] 吴晓光, 刘绍维, 杨磊, 等. 基于深度强化学习的双足机器人斜坡步态控制方法 [J]. *自动化学报*, 2021, 47(8): 1976–1987.
- WU Xiaoguang LIU Shaowei, YANG Lei, et al. A slope gait control method for bipedal robots based on deep reinforcement learning[J]. *Acta automatica sinica*, 2021, 47(8): 1976–1987.
- [17] JIANG Rong, WANG Zhipeng, HE Bin, et al. A data-efficient goal-directed deep reinforcement learning method for robot visuomotor skill[J]. *Neurocomputing*, 2021, 462: 389–401.
- [18] SILVER D, HUANG A, MADDISON C J, et al. Mastering the game of go with deep neural networks and tree search[J]. *Nature*, 2016, 529: 484–489.
- [19] VINYALS O, BABUSCHKIN I, CZARNECKI W M, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning[J]. *Nature*, 2019, 575: 350–354.
- [20] BERNER C, et al. Dota 2 with large scale deep reinforcement learning[EB/OL]. (2019–10–1)[2021–12–14]. <https://arxiv.org/abs/1912.06680>.
- [21] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing atari with deep reinforcement learning[EB/OL]. (2013–12–19) [2021–12–14]. <https://arxiv.org/abs/1312.5602>.
- [22] LUO Shu, ZHANG Linxuan, FAN Yushun. Dynamic multi-objective scheduling for flexible job shop by deep reinforcement learning[J]. *Computers & industrial engineering*, 2021, 159: 107489.
- [23] SCHAU T, QUAN J, ANTONOGLOU I, et al. Prioritized experience replay[EB/OL]. (2016–2–25) [2021–12–14]. <https://arxiv.org/abs/1511.05952>.
- [24] VAN HASSELT H, GUEZ A, SILVER D. Deep reinforcement learning with double Q-learning[EB/OL]. (2015–12–8)[2021–12–14]. <https://arxiv.org/abs/1509.06461>.
- [25] LIU Xiaojun, YI Hong, NI Zhonghua. Application of ant colony optimization algorithm in process planning optimization[J]. *Journal of intelligent manufacturing*, 2013, 24(1): 1–13.

作者简介:



陶鑫钰, 硕士研究生, 主要研究方向为深度强化学习在工艺路线中的应用。



王艳, 教授, 博士生导师, 工业物联网技术集成应用方向技术带头人, 主要研究方向为基于大数据知识自动化的离散制造能耗网络协同优化。承担国家自然科学基金项目 2 项、中国博士后特别资助项目 1 项、江苏省自然科学基金项目 1 项、教育部人文社

科规划基金项目 1 项, 发表学术论文近百篇。



纪志成, 教授, 博士生导师, 主要研究方向为制造物联集成与优化。申请及授权发明专利 40 余项, 登记软件著作权 100 余项, 发表学术论文 200 余篇, 出版学术著作 1 部。

2023 全球人工智能技术创新大赛 (2023 GAIIC) — 算法挑战赛

2023 全球人工智能技术创新大赛 (2023 GAIIC) — 算法挑战赛正在积极筹备中, 拟于 2023 年 2 月至 6 月举办, 全球人工智能技术创新大赛 (GAIIC) — 算法挑战赛由中国人工智能学会和杭州市余杭区人民政府共同创办, 大赛立足国际视野, 聚焦前沿科技与应用创新, 推进人工智能领域的学术交流、人才培养、技术发展以及跨界应用与融合, 打造人工智能的人才交流平台与产业生态圈。

大赛迄今已成功举办了两届, 累计吸引了来自海内外高校、知名企业以及科研机构等 17000 余支队伍报名参赛, 逐渐成长为人工智能领域颇具影响力的活动之一。历届大赛汇集学界大咖, 赛事指导嘉宾由国务院参事、CAAI 理事长、中国工程院院士、清华大学信息学院院长、CAAI Fellow 戴琼海, CAAI 副理事长、中国工程院院士、CAAI Fellow 陈杰, CAAI 副理事长、中国工程院院士、浪潮集团首席科学家、CAAI Fellow 王恩东, CAAI 副理事长、中国工程院院士、国家农业信息化工程技术研究中心主任、CAAI Fellow 赵春江, CAAI 监事、中国工程院院士、湖南大学教授、CAAI Fellow 王耀南, 欧洲科学院外籍院士、清华大学人工智能研究院常务副院长、CAAI Fellow 孙茂松, CAAI 副理事长、欧洲科学院外籍院士、南京大学人工智能学院院长、CAAI Fellow 周志华共同担任。

2023 赛事规划:

报名、组队: 2~3 月

初赛: 3~4 月

复赛: 4~5 月

决赛: 6 月

决赛将于线下举办, 举办地点: 杭州市 未来科技城, 详情请浏览网址: <http://www.caii.cn/index.php?s=/home/article/detail/id/2379.html>。