

基于强化学习改进遗传算法求解柔性作业车间调度问题

刘 玥¹ 韩栋梁¹ 康圆圆¹
LIU Yue HAN Dongliang KANG Yuanyuan

摘 要

针对传统遗传算法在解决柔性作业车间调度问题(FJSP)时存在局部搜索能力差,过早陷入最优解等问题,文章建立了以最大完工时间最小为优化目标的模型,提出了一种基于强化学习的改进遗传算法进行求解。首先设计了 Logistic 混沌映射的初始化策略来提高种群多样性;其次通过强化学习动态调整遗传算法关键参数;最后通过变邻域搜索提高算法局部搜索能力。为验证算法有效性,将所提的 QGA-LV 算法与其他方法进行对比,通过 MATLAB 运行标准算例进行分析,实验结果表明,所提算法能获取更好的寻优效果。

关键词

强化学习;混沌映射;改进遗传算法;变邻域搜索

doi: 10.3969/j.issn.1672-9528.2026.04.003

0 引言

经济全球化的深入发展持续刺激着市场对制造业产品的需求。为了应对这一增长的需求,制造企业正面临着前所未有的巨大挑战。柔性作业车间调度问题(flexible job-shop scheduling problem, FJSP),作为制造企业生产管理中的核心环节,近几十年来备受学术界与工业界关注^[1-2]。FJSP 已经被证明是一种 NP-hard 问题^[3],在调度过程中具有复杂多变的特点,因此其在求解过程中难度较高,同时也需要兼顾实际生产环境中高效性与经济性的特点。

近年来,众多学者在求解 FJSP 问题上做出了诸多贡献。基于遗传算法解决车间调度研究方面,唐艺军等^[4]为解决机器和工人双资源约束问题,采用了基于块结构邻域搜索的改进遗传算法进行求解。赵欣越等^[5]提出一种改进双种群遗传算法,运用全局、局部和随机搜索初始化方法增加种群多样性。田梦蝶等^[6]引入混沌理论改进遗传算法,提高初始种群的多样性和种群质量。

基于强化学习解决车间调度问题方面,张国辉等^[7]将改进的禁忌搜索策略和 Q-learning 相结合解决多目标优化问题。杜利珍等^[8]设计了一种改进的奖励函数作为环境,将强化学习与蜂群算法相结合求解置换流水车间调度问题。李敏硕等^[9]利用 Q 学习机制动态选择合适的扰动程度,克服了不合理的扰动导致的搜索性能下降问题。

综上所述,遗传算法的改进和强化学习的改进方法在求

解车间调度问题上均存在不足,将二者结合起来可以更好地解决 FJSP,因此,本文将 Q-learning 与 GA 结合,提出了 QGA-LV 算法,即在二者算法结合的基础上增加混沌映射提高种群的多样性,引入了变邻域搜索策略平衡遗传算法提前早熟的缺陷,最后将所提的 QGA-LV 算法与其他算法做对比,验证其在解决 FJSP 的优越性。

1 柔性作业车间调度模型

1.1 问题描述

柔性作业车间调度问题(FJSP)是在经典作业车间调度问题(JSP)上的进一步扩展,具有更高的复杂性和现实意义。该问题涉及 n 个工件 $\{J_1, J_2, \dots, J_n\}$ 在 m 台机器上 $\{M_1, M_2, \dots, M_m\}$ 加工的过程,每个工件有若干道工序 $\{O_{11}, O_{12}, \dots, O_{ij}\}$, FJSP 的核心是“柔性”,即每道工序可以在多个机器上加工,且每台机器的加工时间可能不同,使问题可以分解为工序排序和机器选择两个子问题,前者用来确定每台机器上各工序的加工顺序,后者用来确定每道工序的加工设备,两者共同决定了最终的生产效率。

1.2 数学模型

为解决柔性作业车间调度问题,本文进行以下问题的假设。

- (1) 各机器在同一时刻加工时只能处理一道工序;
- (2) 工件各工序可以在不同的机器上加工,且加工时间已知;
- (3) 同一工件各工序加工时需要满足一定的先后顺序;
- (4) 在零时刻,所有工件均可以开始加工,所有机器都可以使用;

1. 太原学院 山西太原 030000

[基金项目] 太原学院青年项目(24TYQN23); 山西省科技战略研究专项(202304031401110)

(5) 忽略工件的运输时间;

(6) 一道工序在加工过程中不能被中断, 即不考虑机器故障、紧急插单等情况。

根据问题的假设, 建立了 FJSP 的数学模型, 相关的参数及含义如表 1 所示。

表 1 模型相关参数及含义

符号	含义
n	工件索引
m	机器索引
e	工序索引
i	工件编号, $i \in \{1, 2, \dots, n\}$
j	工序编号, $j \in \{1, 2, \dots, e\}$
k	机器编号, $k \in \{1, 2, \dots, m\}$
W	一个足够大的正数
C_i	工件 i 最后一道工序的完工时间
C_{ij}	工件 i 的第 j 道工序完工时间
$C_{\max}$	所有工件的最大完工时间
T_{ijk}	工件 i 第 j 道工序在机器 k 上的加工时间
S_{ij}	工件 i 的第 j 道工序开始加工时间
x_{ijpqk}	机器 k 上, 工序 O_{ij} 在工序 O_{pq} 的前面加工为 1, 否则为 0
y_{ijk}	工序 O_{ij} 在机器 k 上加工为 1, 否则为 0

(1) 各工序加工顺序的约束, 只有前一道工序加工结束下一道工序才能开始加工。

$$S_{ij} + y_{ijk}T_{ijk} \leq C_{ij} \quad (1)$$

$$C_{ij} \leq S_{ij+1} \quad (2)$$

(2) 最终的完工时间为各工件最后一道工序完工时间的最大值。

$$C_{\max} = \max(C_i) \quad (3)$$

(3) 各工件每道工序在某一时刻只能在一台机器上加工。

$$\sum_{k=1}^m y_{ijk} = 1 \quad (4)$$

(4) 任一工件的同一道工序中在所有机器上只有一个紧前工序。

$$\sum_{k=1}^m \sum_{i=1}^n \sum_{j=1}^e x_{ijpqk} = 1 \quad (5)$$

(5) 任一时刻所有机器只能加工一道工序。

$$C_{i(j-1)} \leq S_{ij} + W(1 - x_{ijpqk}) \quad (6)$$

(6) 各工序的加工时间大于 0。

$$T_{ijk} > 0 \quad (7)$$

目标函数为最小化最大完工时间 $C_{\max}$, 用公式表示为:

$$f = \min C_{\max} = \min \{ \max(C_i | i \in n) \} \quad (8)$$

2 改进遗传算法求解 FJSP 问题

2.1 遗传操作

2.1.1 编码

本文编码时采取两段式编码结构, 分别是工序编码段和机器编码段, 具体编码结构示意图如图 1 所示。左侧虚线框代表工序编码, 第一列的每一个数字编码都与第二列的工序一一对应, 相同数字重复的次数代表工序的数量, 例如第一次出现的 2 代表工件 2 的第一道工序 O_{21} , 第二次出现的 2 代表工件 2 的第二道工序 O_{22} , 以此类推, 第三列代表各工序从属于哪个工件, 各工件的工序数有所不同; 右侧虚线框代表机器编码, 每一道工序都有与之相对应的机器, 编码的数字代表从可选设备集里选择第几个机器进行加工, 例如 2 代表从可选设备中选择第 2 个机器进行加工, 以此类推。

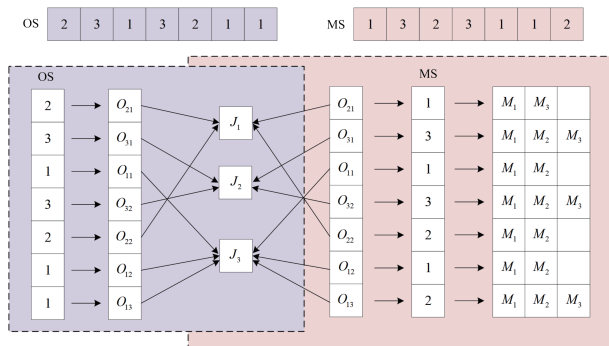


图 1 编码结构示意图

2.1.2 初始化改进

初始种群的质量对搜索效率和最终成果有着深远影响。遗传算法初始解通常都是随机产生的, 难以保留种群的多样性, 而混沌运动随机性强, 对初始值敏感, 能够快速跳出局部最优解, 提高全局搜索能力, 使得初始解方案的个体在解空间中分布更加均匀。在众多混沌映射方法中, Logistic 映射是较为典型的混沌系统, 可以避免出现种群分布不均匀的现象, 因此本文采用 Logistic 混沌映射来提高种群多样性, 用公式表示为:

$$x_{t+1} = \mu x_t(1 - x_t) \quad (9)$$

式中: $x_t \in [0, 1]$; μ 为一个可调整的参数, 当 $\mu \in [0, 4]$ 时, 系统处于混沌状态。

2.1.3 选择操作

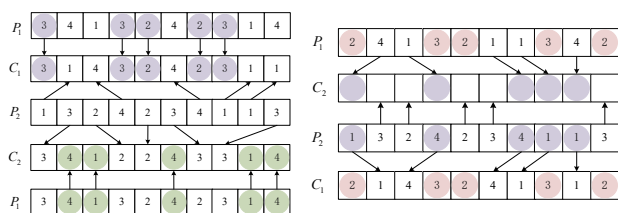
在选择操作中, 根据精英保留策略, 按一定比例从当前种群中保留适应度较高的部分染色体, 确保优秀个体能够进入下一代, 再通过锦标赛选择方式补充剩余的染色体, 重复上述操作过程, 直至达到种群规模。

2.1.4 交叉操作

交叉操作负责扩大搜索范围, 增强种群的多样性。本文采用了双策略混合交叉操作的方法, 工序部分采用 POX 交叉,

将工件集划分为两个非空集合 $J_1 = \{2, 3\}$ 和 $J_2 = \{1, 4\}$ ，复制 P_1 中属于 J_1 的部分到 C_1 ，属于 J_2 的部分到 C_2 ， P_2 中属于 J_1 的部分保留到 C_2 ，属于 J_2 的部分保留到 C_1 。

机器部分采用 OX 交叉，将机器集划分两个非空集合 $M_1 = \{2, 3\}$ 和 $M_2 = \{1, 4\}$ ，复制 P_1 中属于 M_1 的部分到 C_1 ，再将 P_2 中属于 M_2 的部分依次保留到 C_1 ， C_2 过程与上述类似。交叉操作示意图如图 2 所示。



(a) POX 交叉示意图 (b) OX 交叉示意图
图 2 交叉操作示意图

2.1.5 变异操作

变异操作负责局部开发，增加寻优的随机性。本文工件部分采用互换变异法，即随机选取两个位置进行基因点互换；机器部分采用负荷最小化单点变异法，即随机选取一个基因位，在可选设备集中选择加工时间最短的机器进行加工。变异操作示意图如图 3 所示。

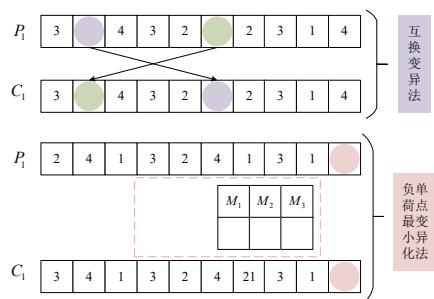


图 3 变异操作示意图

2.2 变邻域搜索

为避免算法过早收敛，提升算法的局部搜索能力，引入了变邻域搜索 (variable neighborhood search, VNS)，通过采取多个不同的邻域结构动态调整搜索范围。针对柔性作业车间调度问题的工序编码和机器编码设计了 3 种邻域结构，具体如下：

(1) 邻域结构 1：随机连续选择工序编码的多个位置，将选出的序列进行反转排序；

(2) 邻域结构 2：在工序编码中随机挑选至少三个位置，提取出各位置对应的工序编码，将其进行全排列，随机选择一种排列结果按顺序放入原位置；

(3) 邻域结构 3：在机器编码中随机选择一个位置，将其替换成完工时间最短的机器编码。各邻域结构示意图如图 4 所示。

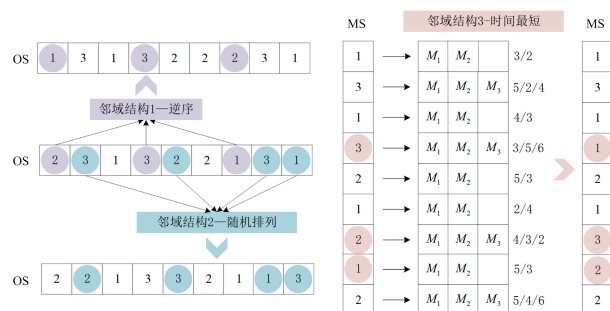


图 4 邻域结构示意图

3 基于强化学习的参数自适应策略

传统遗传算法中的交叉概率和变异概率是两个预先设定好的固定值，是影响算法性能的重要因素。若交叉概率 PC 过大会导致种群多样性下降，陷入局部最优，过小会导致新的染色体生成效率低；若变异概率 PM 过大会降低算法的收敛性，过小会使种群陷入早熟收敛，如何动态调整两个参数的值是优化过程中的一个关键问题。因此，本文设计了基于强化学习的参数自适应策略，根据种群状态动态调整交叉概率和变异概率。

3.1 强化学习简介

强化学习 (reinforcement learning, RL) 近年来受到诸多研究者的广泛关注。在强化学习过程中，智能体跟环境一直在交互。智能体会利用在环境里获取到的状态 s_t ，根据某种策略输出相应的动作 a_t ，环境会根据智能体采取的动作，输出下一个状态 s_{t+1} 以及当前的动作得到的奖励 R_{t+1} ，如果是正值，代表再次做出该动作的概率会增加，反之会减少。智能体会根据得到的奖励值大小调整策略，反复执行上述步骤，直到从环境中获取最大的奖励。

Q-learning 算法是强化学习中通过动作价值来选取特定策略的算法，在迭代中通过更新 Q 表，智能体选择与当前状态相符合的动作使其最终达到收敛状态。 Q 表更新公式为：

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha [R_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (10)$$

式中： R_{t+1} 表示第 t 次迭代的状态 s_t 执行动作 a_t 后的奖励值； s_{t+1} 表示新状态； γ 表示衰减率， $\gamma \in (0, 1)$ ； α 表示学习率， $\alpha \in (0, 1)$ 。

3.2 状态空间

状态空间由 9 个状态组成，是种群的平均适应度指标 F_t 和多样性指标 D_t 共同决定的，用公式表示为：

$$F_t = \frac{\sum_{b=1}^M H_t^b}{M} \quad (11)$$

$$D_t = \frac{\sum_{b=1}^M |H_t^b - F_t|}{M} \quad (12)$$

式中: H_t^b 代表第 t 次迭代的第 b 个个体的适应度。状态空间如表 2 所示。

表 2 状态空间

状态	定义	状态	定义
S_0	$F_t < F_0, D_t > D_{t-1}$	S_5	$F_t = F_0, D_t > D_{t-1}$
S_1	$F_t < F_0, D_t < D_{t-1}$	S_6	$F_t < F_0, D_t = D_{t-1}$
S_2	$F_t > F_0, D_t > D_{t-1}$	S_7	$F_t > F_0, D_t = D_{t-1}$
S_3	$F_t > F_0, D_t < D_{t-1}$	S_8	$F_t = F_0, D_t = D_{t-1}$
S_4	$F_t = F_0, D_t > D_{t-1}$		

其中, F_t 和 F_0 分别表示第 t 次迭代种群的平均适应度和初始种群的平均适应度; D_t 和 D_{t-1} 分别表示第 t 次迭代种群和上一代迭代种群的多样性指标; S_0 表示最佳状态。

3.3 动作空间

对于每一代种群, 智能体都会通过不同的动作来动态调整交叉概率 P_c 和变异概率 P_m , 二者都分为增加、减少和不变三种情况, 具体如下:

$$P_c^{t+1} = \begin{cases} P_c' + \Delta_c & \text{if 增加} \\ P_c' & \text{if 不变} \\ P_c' - \Delta_c & \text{if 减少} \end{cases} \quad (13)$$

$$P_m^{t+1} = \begin{cases} P_m' + \Delta_m & \text{if 增加} \\ P_m' & \text{if 不变} \\ P_m' - \Delta_m & \text{if 减少} \end{cases} \quad (14)$$

式中: Δ_c 和 Δ_m 为步长, $\Delta_c=0.05$, $\Delta_m=0.01$, P_c 取值范围为 $P_c \in [0.5, 0.9]$, P_m 的取值范围为 $P_m \in [0.01, 0.20]$ 。

3.4 奖励机制

交叉和变异操作完成后, 会根据下一个状态获得对应的奖励, 即种群的平均适应度和多样性二者加权值相较之前若有所提升, 则给予正奖励, 表明该操作有利于种群进化; 反之则给予负奖励, 抑制此类操作的发生。具体公式为:

$$r = x_1 \frac{F_t}{F_1} + x_2 \frac{D_t}{D_1} \quad (15)$$

$$R = \begin{cases} 1, & r > 1 \\ -1, & r \leq 1 \end{cases} \quad (16)$$

式中: x_1 和 x_2 取值均为 0.5; r 为种群平均适应度和种群多样性加权后的值。

3.5 改进算法流程

具体步骤为: ①输入工件、机器参数, 算法的基本参数和初始化 Q 表的相关参数; ②采用 Logistic 混沌映射对种群进行初始化; ③计算每条染色体的适应度值, 进行遗传算法的选择、交叉、变异操作; ④根据染色体适应度和种群平均适应度获取强化学习的状态和动作; ⑤动态调整交叉、变异算子; ⑥计算奖励值, 更新 Q 表; ⑦若迭代次数达到最高迭代次数, 则输出最优解, 否则重复进行③以后的步骤。流程图如图 5 所示, 图中 GA 代表遗传算法, LV 代表

Logistic 映射 (L) 和变邻域搜索 VNS (V), Q 代表强化学习中的 Q-learning 算法。

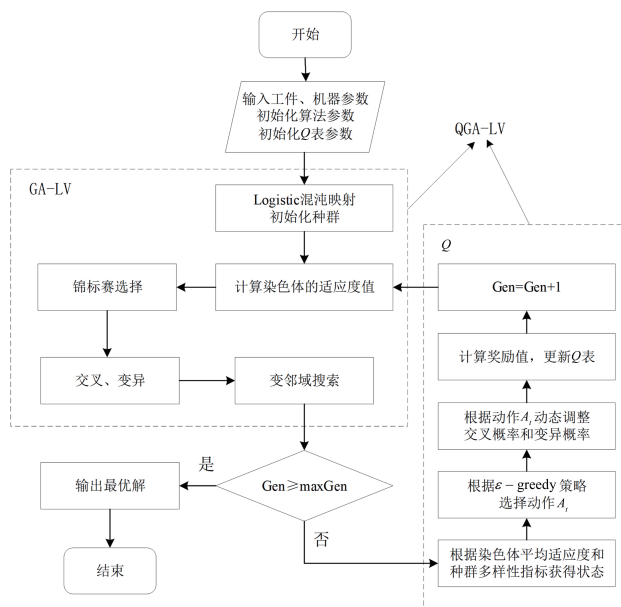


图 5 算法改进流程图

4 算例分析

为了测试本文算法在求解柔性作业车间调度问题的性能, 本文采用标准算例 Brandimarte 算例 MK01~MK10 共 10 个算例。改进算法具体参数设置如下: 种群规模 200, 迭代次数 100, 初始交叉概率 $P_c=0.85$, 初始变异概率 $P_m=0.2$, 学习率 $\alpha=0.8$, 衰减率 $\gamma=0.7$, 贪心算法参数 $\epsilon=0.8$ 。将本文提出的 QGA-LV 算法与未改进前的 GA 算法进行对比, 实验结果如表 3 所示。

表 3 标准算例实验结果对比

标准算例	规模	理论最优解	GA	QGA-LV
MK01	10×6	36	44	41
MK02	10×6	24	36	30
MK03	15×8	204	223	208
MK04	15×8	48	80	71
MK05	15×4	168	191	177
MK06	10×15	33	98	72
MK07	20×5	133	178	151
MK08	20×10	523	542	525
MK09	20×10	299	348	340
MK10	20×15	165	313	280

由此可知, 改进后的 QGA-LV 算法虽然尚未达到理论最优解的值, 但相较于传统的 GA 算法, 在标准算例中运行效率都有明显的提升, 最高提升了 18%。以 MK02 为例, 生成调度方案的甘特图如图 6 所示。图中用每一个矩形来表示一道工序, 不同颜色代表不同工件, 相同颜色代表某一工件的不同工序, 沿 x 轴方向从左往右代表该机器上各工序的加工

顺序,矩形的长度代表每道工序的加工时间,所有工序全部加工完毕的时间是虚线处的 30。

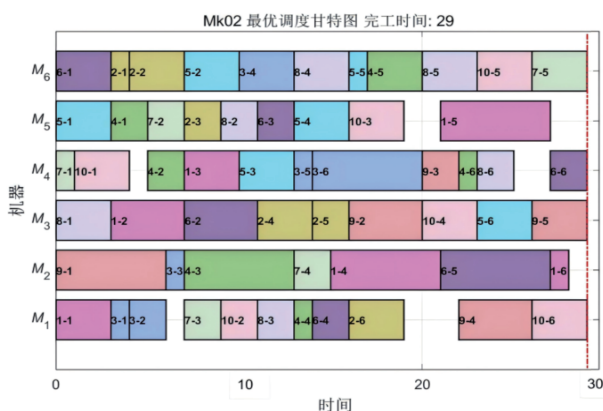


图 6 MK02 甘特图

为进一步验证本文提出算法的准确性和有效性,对上述三种改进方法,Logistic 混沌映射方法、Q-learning 方法和变邻域搜索方法分别单独从 QGA-LV 算法中移除,对应标记为 QGA-V、GA-LV 和 QGA-L,选取 4 种规模的算例进行求解,最终绘制出完工时间的迭代图,如图 7 所示。

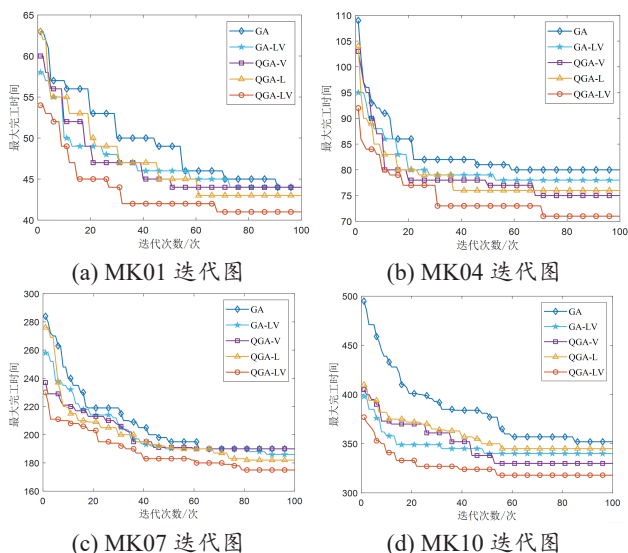


图 7 不同规模迭代图

通过上述迭代曲线可知,虽然减少改进方法的 QGA-V、GA-LV 和 QGA-L 三种策略相较于传统的 GA 算法效率都有提升,但是相较于本文提出的 QGA-LV 算法,在收敛速度和收敛效率方面还是略有逊色,因此,综合运用多种改进方法能够让算法效率得到显著提升。

5 结语

本文在柔性作业车间调度问题的基础上,提出了一种以完工时间为目标的基于强化学习改进遗传算法的方法。该方法采用 Logistic 混沌映射进行初始化,提高了种群的多样性;通过强化学习动态调整交叉概率和变异概率,让种群朝着更加合理的方向进化;最后,使用变邻域搜索提高算法的局部

搜索能力。实验表明,本文提出的 QGA-LV 算法收敛速度相较于未改进前的算法有明显提升,能够快速跳出局部最优,得到更显著的实验效果。然而本文只考虑了单目标问题,在实际生产过程中往往需要建立更多的目标进行求解,如成本、能耗等目标,在之后的研究中可以进一步考虑。

参考文献:

- [1]SHAO W S, SHAO Z S, PI D C. Effective constructive heuristics for distributed no-wait flexible flow shop scheduling problem[J]. Computers & operations research, 2021, 136: 105482.
- [2]LUO Q, DENG Q W, GONG G L, et al. An efficient memetic algorithm for distributed flexible job shop scheduling problem with transfers [J]. Expert systems with applications, 2020,160:113721.
- [3]GAO K Z, YANG F J, ZHOU M C, et al. Flexible job-shop rescheduling for new job insertion by using discrete Jaya algorithm[J]. IEEE transactions on cybernetics, 2019, 49(5): 1944-1955.
- [4]唐艺军,谢志坤.改进遗传算法求解装配式预制构件双资源调度问题[J]. 工程管理学报,2025,39(4):145-151.
- [5]赵欣越,苗鸿宾,张士博,等.基于改进双种群遗传算法的多目标绿色柔性作业车间调度[J/OL]. 工业工程,1-16[2026-03-26].<https://link.cnki.net/urlid/44.1429.TH.20250730.0958.002>.
- [6]田梦蝶,贾世会,迟晓妮,等.基于混沌映射的改进 GA 求解柔性作业车间调度[J]. 组合机床与自动化加工技术, 2025(3): 226-231.
- [7]张国辉,李志霄,张利平,等.基于强化学习协同进化算法求解柔性作业车间节能调度问题[J]. 计算机应用研究, 2025, 42(7): 2039-2047.
- [8]杜利珍,宣自凤,唐家琦,等.改进的 Q-learning 蜂群算法求解置换流水车间调度问题[J]. 组合机床与自动化加工技术, 2024(10):175-180.
- [9]李敏硕,刘翔,王珂瑶,等.一类分布式柔性作业车间调度问题的基于 Q 学习的迭代贪婪算法[J]. 系统科学与数学, 2025, 45(6): 1734-1751.

【作者简介】

刘玥(1996—),女,山西长子人,硕士,助教,研究方向:智能制造、车间调度。

韩栋梁(1979—),男,山西祁县人,博士,高级工程师,研究方向:智能制造。

康圆圆(1992—),男,山西大同人,博士,讲师,研究方向:金属基复合材料界面性能。

(收稿日期:2025-09-01 修回日期:2026-03-31)