

基于强化学习的多策略 HHO 求解分布式混合流水车间调度^①



李晓鹏, 陶孟欣, 周艳平

(青岛科技大学 信息科学技术学院, 青岛 266061)

通信作者: 周艳平, E-mail: ypzhou@qust.edu.cn

摘要: 针对分布式混合流水车间调度问题, 围绕最小化最大完工时间和延迟时间的优化目标, 构建了一种基于强化学习的多策略哈里斯鹰优化算法 (RLMHHO). 算法使用分组混沌初始化策略, 提升初始搜索的随机性与多样性; 引入探索、开发、均衡与精英这 4 组鹰群管理机制, 实现全局搜索与局部开发的协同; 基于深度 Q 网络的强化学习协调器, 依据 14 维状态空间动态选择最优搜索策略. 仿真实验验证了所提算法求解该类调度问题具有更优的解质量和更强的搜索能力.

关键词: 分布式混合流水车间调度; 哈里斯鹰优化; 强化学习; 混沌映射

引用格式: 李晓鹏, 陶孟欣, 周艳平. 基于强化学习的多策略 HHO 求解分布式混合流水车间调度. 计算机系统应用. <http://www.c-s-a.org.cn/1003-3254/10101.html>

Reinforcement-learning-based Multi-strategy Harris Hawks Optimization for Distributed Hybrid Flow-shop Scheduling

LI Xiao-Peng, TAO Meng-Xin, ZHOU Yan-Ping

(School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266061, China)

Abstract: A reinforcement-learning-based multi-strategy Harris hawks optimization (RLMHHO) is developed to address the scheduling problem of distributed hybrid flow-shop, with the optimization objectives of minimizing the maximum completion time and delay time. The algorithm uses a grouping chaos initialization strategy to improve the randomness and diversity of the initial search. A four-group eagle management mechanism of exploration, development, balance, and elite is introduced to achieve synergy between global search and local development. A reinforcement learning coordinator based on deep Q-networks dynamically selects the optimal search strategy based on a 14-dimensional state space. Simulation experiments have verified that the proposed algorithm offers better solution quality and stronger search capability for solving this type of scheduling problem.

Key words: distributed hybrid flow-shop scheduling; Harris hawks optimization (HHO); reinforcement learning; chaotic mapping

随着工业 4.0 和“中国制造 2025”战略的不断推进, 制造系统向分布式、异构与多目标方向演化. 多目标分布式混合流水车间调度问题 (multi-objective distributed hybrid flow-shop scheduling problem, MO-DHFSP) 因分

布式处理、多目标冲突, 呈现高度组合复杂性, 难以通过精确算法高效求解^[1], 该问题属于典型的 NP-hard 问题, 随着问题规模的快速增长, 依赖精确算法往往难以在可接受时间内获得高质量解, 其有效求解不仅对于

① 收稿时间: 2025-09-03; 修改时间: 2025-09-23; 采用时间: 2025-10-14; csa 在线出版时间: 2026-01-15

提升交付能力、缩短生产周期和降低运营成本具有重要意义,同时在智能优化方法研究方面也具有重要的理论价值。

针对多目标分布式混合流水车间调度问题,许多学者提供了解决方法。例如夏保丽等人^[2]提出基于DQN与调度规则结合的优化算法,将分布式流水车间调度建模为马尔可夫决策过程,通过强化学习与9种规则协同更新解状态,提升调度质量与收敛性能。王镜捷等人^[3]提出多目标社会工程粒子群算法,融合启发式初始化、参考解驱动更新与多邻域搜索策略,提升在分布式混合流水车间中对最大完工时间、延迟工件数和非工艺等待时间的优化效果。轩华等人^[4]提出基于改进灰狼优化算法,将机器故障和运输时间约束下的分布式混合流水线重调度问题构建为整数规划模型,通过该改进算法对最大完工时间、总能耗和总延期进行多目标协同优化;邱荣根^[5]提出基于改进非支配排序遗传算法(INS-GAII),针对多目标混合流水车间分批调度问题构建混合整数线性规划模型,通过协同初始化、混合交叉及模拟退火局部搜索优化算法性能,实现碳排放量和拖期的最小化。Cai等人^[6]针对分布式两阶段混合流水车间,提出协作式变邻域搜索方法,通过多种邻域结构与协同机制改进搜索过程,在兼顾最大完工时间与总拖期的优化中表现出更高的稳定性与解优质量。Wang等人^[7]针对能耗优化混合流水车间调度问题,提出基于模糊相关熵的NSGA-II算法。该方法结合模糊数学与信息熵理论构建新的拥挤度测度,并引入反向学习与关键路径节能策略,在保证完工时间的同时有效降低能耗,提升了算法的搜索效率与解集质量。

哈里斯鹰优化算法(Harris hawks optimization, HHO)是一种由Heidari等人^[8]于2019年提出的元启发式优化方法。该算法因其结构简洁、参数较少且求解效率较高,已广泛应用于数值优化、工程设计、电力系统优化等多个工程领域^[9,10]。然而,目前该算法在生产调度方向的研究仍相对有限,现有工作主要集中于作业车间调度问题^[11,12]。原始的哈里斯鹰优化算法虽然在连续单目标优化问题上表现出色,但在处理多目标分布式混合流水车间调度问题时面临难以处理复杂约束、搜索效率受限、缺乏自适应调控机制的问题,导致解的多样性和稳定性难以保证。

针对原始HHO算法在离散多目标调度问题中的局限、传统智能优化算法对搜索参数依赖较强、探索与开发能力难以平衡以及易陷入局部最优等问题;单纯

强化学习算法在大规模复杂调度场景下训练开销显著增加和收敛效率下降,使算法在复杂问题上的效果和可靠性受限等问题,本文提出一种基于强化学习的多策略哈里斯鹰优化(reinforcement-learning-based multi-strategy Harris hawks optimization, RLMHHO)算法,在保持HHO全局探索能力的基础上,通过强化学习动态调节,实现了对复杂调度问题的高效求解和稳定优化,该算法通过14维状态空间对搜索过程的进度、解的质量及种群多样性进行刻画,以支持策略的自适应选择。在此基础上,设定7类候选策略,由强化学习动态匹配搜索策略,用于调控探索、开发、均衡与精英这4类群体的比例,实现群体层面的自适应资源分配。同时,各类群体内部结合专属混沌映射驱动,使搜索过程在保持方向性的同时具有一定随机性与扰动性。奖励函数综合考虑规模、质量与多样性,并兼顾多目标调度的复杂需求。通过在不同规模算例上的对比实验,验证算法在多目标调度中的适应性与鲁棒性。

1 分布式混合流水车间调度模型

1.1 问题描述

在MO-DHFSP中,系统由 F 个独立的工厂组成,每个工厂包含 m 个顺序加工阶段,各阶段的并行机器数量存在差异,共有 n 个工件需分配至工厂 $f \in \{1, 2, \dots, F\}$ 完成加工,每个工件遵循相同加工顺序,即每道工序需在某工厂对应阶段的一台机器上加工,且在同一阶段内可以选择不同机器并行执行。

1.2 模型描述

本文涉及的参数符号表见表1。

表1 模型参数表

参数	参数说明
n	工件数量
m	阶段数量
F	工厂数量
f	工厂索引
i, j	工件索引
g	阶段索引
S_g	阶段 g 的并行机器数量
k	加工阶段 g 中机器编号
$M_{fg,k}$	工厂 f 内第 g 个阶段的第 k 台并行机
$P_{i,fg,k}$	工件 i 在机器 $M_{fg,k}$ 上的加工时间
$Z_{i,j,fg,k}$	工件 j 在工厂 f 、阶段 g 、机器 k 上优先于工件 i 加工时为1, 反之为0
d_i	工件 i 的交货期
$B_{i,g}$	工件 i 在第 g 阶段的加工开始时间

表 1 模型参数表(续)

参数	参数说明
$C_{i,g}$	工件 i 在第 g 阶段的加工结束时间
C_i	工件 i 的完工时间
T_i	工件 i 的延迟时间
$C_{\max}$	所有工件的最大完工时间
$T_{\max}$	所有工件的最大延迟时间
H	一个充分大的正数
$X_{i,f}$	当工件 i 被分配至工厂 f 时为1, 反之则为0
$Y_{i,f,g,k}$	工件 i 在工厂 f 、阶段 g 、机器 k 上加工时为1, 反之为0

基于以上参数和决策变量, MO-DHFSP 模型与约束条件如下:

$$\min f_1 = C_{\max} = \max_{i \in \{1, \dots, n\}} C_i \quad (1)$$

$$\min f_2 = T_{\text{total}} = \sum_{i=1}^n \max(0, C_i - d_i) \quad (2)$$

$$\sum_{f=1}^F X_{i,f} = 1, \forall i = 1, \dots, n \quad (3)$$

$$\sum_{k=1}^{S_g} Y_{i,f,g,k} = 1, \forall i = 1, \dots, n, \forall g = 1, \dots, m \quad (4)$$

$$B_{j,g} \geq C_{i,g} - H \cdot (1 - Z_{i,j,f,g,k}), \forall i \neq j \quad (5)$$

$$B_{i,g} \geq C_{j,g} - H \cdot Z_{i,j,f,g,k}, \forall i \neq j \quad (6)$$

$$B_{i,g} \geq C_{j,g} - H \cdot Z_{i,j,f,g,k}, \forall i \neq j \quad (7)$$

$$C_{i,g} = B_{i,g} + \sum_{f=1}^F \sum_{k=1}^{S_{f,g}} p_{i,f,g,k} \cdot Y_{i,f,g,k}, \forall i, g \quad (8)$$

$$C_i = C_{i,m}, \forall i = 1, 2, \dots, n \quad (9)$$

$$C_{\max} \geq C_i, \forall i = 1, 2, \dots, n \quad (10)$$

$$T_{\max} \geq \max\{C_i - d_i, 0\}, i = 1, 2, \dots, n \quad (11)$$

$$X_{i,f} \in \{0, 1\} \quad (12)$$

$$Y_{i,f,g,k} \in \{0, 1\} \quad (13)$$

$$Z_{i,j,f,g,k} \in \{0, 1\} \quad (14)$$

$$B_{i,g}, C_{i,g}, C_i, T_i \geq 0 \quad (15)$$

式 (1) 表示计算工序的最大完工时间; 式 (2) 表示计算所有工件的总延迟时间; 式 (3) 表示每个工件必须唯一分配给一个工厂; 式 (4) 表示每个工件在每个加工

阶段只能由一台机器加工^[13]; 式 (5) 表示工件的后一道工序的起始加工时间不早于前一道工序的结束时间; 式 (6) 和式 (7) 表示在同一台机器上, 任意两个工件的加工时间不能重叠; 式 (8) 表示某阶段上的完工时间为其加工开始时间与在被分配机器上的加工时间之和^[14]; 式 (9) 表示工件的总完工时间等于其最后阶段的完工时间; 式 (10) 表示最大完工时间必须不小于任意工件的完工时间; 式 (11) 表示总延迟时间的计算方式为各工件完工时间与交货期差值的非负部分; 式 (12)–式 (14) 定义决策变量的取值范围^[15]; 式 (15) 表示所有时间变量的取值必须大于等于 0。

2 RLMHHO 算法设计

2.1 编码与解码策略

基于问题特征设计了一种双层编码方式。该编码方式将问题决策变量划分为工厂分配层与工件排序层, 结合实际调度语义与算法操作需求, 采用整数与序列联合编码, 确保染色体表示的可解析性与可操作性。

在编码层面, 每个个体由两部分构成。第 1 部分为工厂分配向量, 记作:

$$X_F = [x_1, x_2, \dots, x_n], x_i \in \{0, 1, \dots, F-1\} \quad (16)$$

其中, n 表示工件总数, F 为工厂总数, x_i 表示工件被分配至第 x_i 号工厂。

第 2 部分为工厂内部工件排序结构, 记作:

$$S_f = [S_0, S_1, \dots, S_{n_f-1}] \quad (17)$$

其中, S_f 表示工厂 f 内被分配工件的调度顺序, 是一组不重复的工件索引构成的排列序列, 用于控制该工厂内工件的加工先后顺序。 n_f 表示分配到工厂 f 的工件数量。

解码过程根据工厂分配向量 X_F 将工件划分至各自所属工厂; 在每个工厂内部, 按照调度序列 S_f 的顺序依次安排工件进入排产队列, 调度系统为每道工序采用“先可用机 (first available machine, FAM)”的规则来确定机器的选择和机器上的加工顺序, 并结合问题实例中给定的加工时间矩阵, 计算各工序的开始与完成时刻。

2.2 种群分组与混沌初始化

将规模为 N 的种群划分为 4 个功能组, 各组的职责和初始化策略如下。

(1) 探索组。负责全局搜索, 发现新的搜索区域, 初始化策略采用随机生成且均匀分布在整个搜索空间中。

$$\begin{cases} x_{i,j}^{(0)} = l_j + U(0,1) \cdot (u_j - l_j) \\ i \in G_e, j \in \{1, 2, \dots, d\} \end{cases} \quad (18)$$

Logistic 映射公式为:

$$x_{t+1} = \mu x_t (1 - x_t), \mu = 4 \quad (19)$$

当参数 $\mu=4$ 时,该系统呈现强混沌状态,轨迹完全不可预测,初值敏感性最强,适合进行大范围全局初始化。

(2) 开发组. 负责局部搜索,精细化已发现的优质区域,初始化策略采用在当前最优解附近进行高斯扰动生成个体:

$$\begin{cases} x_{i,j}^{(0)} = x_{\text{best},j} + \mathcal{N}(0, \sigma_{\text{init}}^2) \cdot (u_j - l_j) \\ i \in G_o, \sigma_{\text{init}} = 0.1 \end{cases} \quad (20)$$

Tent 映射公式为:

$$x_{t+1} = \begin{cases} \mu x_t, & 0 \leq x_t \leq 0.5 \\ \mu(1 - x_t), & 0.5 < x_t \leq 1 \end{cases} \quad (21)$$

当 $\mu=2$ 时,该系统在 $[0, 1]$ 区间内形成锯齿形周期轨迹,适合进行密集区域的等间距扰动。

(3) 均衡组. 负责维持探索与开发的平衡,初始化策略采用拉丁超立方采样生成个体。

$$x_{i,j}^{(0)} = l_j + \frac{\pi_{i,j} + U(0,1)}{|G_b|} \cdot (u_j - l_j), \quad i \in G_b \quad (22)$$

Sine 映射公式为:

$$x_{t+1} = \sin(\pi x_t) \quad (23)$$

该映射函数生成的轨迹围绕 $[0, 1]$ 区间做周期性振荡,具备在局部与全局之间跳转的能力,适合用于动态行为切换频繁的均衡组。

(4) 精英组. 负责保持优质解并引导搜索方向,所有个体先随机初始化,然后根据质量评估动态分配到精英组。

Chebyshev 映射公式为:

$$x_{t+1} = \cos(t \cdot \arccos(x_t)), \quad t = 4 \quad (24)$$

当 $t=4$ 时,该映射可产生复杂而稳定的高阶轨迹,用于在精英个体的邻域内执行高频率扰动。

2.3 哈里斯鹰群优化

2.3.1 多因子能量计算模型

在 RLMHHO 中,通过模拟鹰群在搜索空间中发现猎物、围捕猎物、突袭猎物的自然行为,动态调整其在解空间中的位置,逐步逼近最优解。整个行为建模过程由猎物能量函数 E 所驱动,其动态变化控制着搜索

行为的转换^[8,16]。该能量函数定义如下:

$$E = 2 \cdot \left(1 - \left(\frac{t}{T}\right)^2\right) \cdot QF_t \cdot SF_t \cdot \left(1 + 0.1 \cdot \sin\left(\frac{2\pi t}{20}\right)\right) \quad (25)$$

$$QF_t = \begin{cases} 0.8, & \text{if } s < 5 \\ 1.2, & \text{if } s \geq 5 \end{cases} \quad (26)$$

$$SF_t = \begin{cases} 1.0, & \text{if } s < 15 \\ 1 + 0.3 \cdot e^{\frac{s-15}{10}}, & \text{if } s \geq 15 \end{cases} \quad (27)$$

其中, t 表示当前迭代次数, T 为最大迭代次数, s 为未改进次数。

2.3.2 分组位置更新策略

以下沿用式(25)的全局能量 E 以及逃逸概率 r ,阐述各条件下4组鹰群的具体行为。

当猎物能量 $E \geq 1$ 时,进入全局探索阶段,鹰群行为如表2所示,探索策略如下:

$$X_{\text{exploration}}^{t+1} = \begin{cases} \text{RandomFactory}() & \text{if } \xi_i > 0.6 \\ X_{\text{best}} & \text{if } 0.3 < \xi_i \leq 0.6 \\ X_{\text{exploration}}^t & \text{if } \xi_i \leq 0.3 \end{cases} \quad (28)$$

表2 全局探索阶段鹰群行为表

组别	位置更新与操作
探索组	探索后采用Logistic混沌值 ξ_i 触发更新: 当 $\xi_i > 0.6$ 时进行随机工厂重分配; $0.3 < \xi_i \leq 0.6$ 时跟随最优解; 否则保持原位
开发组	探索后立即调用Tent映射的交换、插入与局部工厂重分配3个算子,对新位置做精细局部开发
均衡组	仅执行探索策略
精英组	在探索策略后进行一次轻量Chebyshev扰动

当猎物能量 $0.5 \leq E < 1$ 且 $r \geq 0.5$ 进入软包围,鹰群行为如表3所示,更新公式为:

$$X^{t+1} = X_r^t - E_t |2(1-r)X_r^t - X^t| \quad (29)$$

表3 软包围阶段鹰群行为表

组别	位置更新与操作
探索组	执行软包围后,仅在工厂内部做小幅随机扰动
开发组	在软包围结果的基础上,优先触发Tent映射的插入与交换算子,对执行序列进行精细排序
均衡组	先完成软包围; 随后读取一次Sine混沌值 ϕ ,若 $\phi > 0.5$,维持软包围; 若 $\phi \leq 0.5$,立即切换至硬包围(式(30))
精英组	在软包围结果附近做Chebyshev轻微扰动

当猎物能量 $0.5 \leq E < 1$ 且 $r < 0.5$ 进入硬包围,鹰群行为如表4所示,更新公式为:

$$X^{t+1} = X_r^t - E_t |X_r^t - X^t| \quad (30)$$

权重融合公式:

$$X_s^{t+1} = (1 - \lambda)X_s^t + \lambda X_r^t, \lambda \in [0, 0.3] \quad (31)$$

当猎物能量 $E < 0.5$ 进入极限开发阶段, 极限开发鹰群行为如表 5 所示。

表 4 硬包围鹰群行为表

组别	位置更新与操作
探索组	仅执行一次硬包围后停止深度操作
开发组	以较高阈值使用 Tent 混沌映射驱动 3 个算子协同
均衡组	重复硬包围
精英组	根据式(31)对硬包围结果与现有精英解做权重融合

表 5 极限开发鹰群行为表

组别	位置更新与操作
探索组	保留极小幅随机扰动
开发组	连续执行 Tent 映射三算子, 直至最近 3 轮无任何改进, 形成准局部最优
均衡组	以 50% 触发率跟随开发组
精英组	采用 Chebyshev 映射进行微小扰动, 若微扰无法改进, 则直接保持上一代位置

2.4 强化学习策略控制

引入强化学习策略控制机制, 取代传统依赖启发规则的行为调度方式。该机制通过构建具有区分性与反馈性的状态向量, 引导策略网络动态选择最优行为, 联动种群分组执行对应策略, 在行为调度与资源配置之间建立闭环, 引导搜索过程持续向有效区域演化。

2.4.1 多维状态空间构建

为提高搜索过程中的演化特征与策略效果, 控制器在第 t 次迭代中构建 14 维状态向量:

$$S_t = [r_t, i_t, s_t, n_t, q_t, b_t, m_{1:5}^{(E)}, m_{1:3}^{(D)}] \quad (32)$$

其中, 向量包含进度率 r_t 、改进率 i_t 、停滞率 s_t 、帕累托规模比 n_t 、质量评分 q_t 、负载均衡度 b_t 、探索组性能 $m_{1:5}^{(E)}$ 和开发组性能 $m_{1:3}^{(D)}$ 。

策略选择采用深度 Q 网络进行建模, 其行为策略函数为:

$$A = \{a_1, a_2, \dots, a_7\} \quad (33)$$

$$a_t = \pi(S_t) = \arg \max_{a \in A} Q(S_t, a) \quad (34)$$

其中, 每个动作 a_i 表征一种搜索行为及其资源重构: a_1 表示强化全局探索; a_2 表示强化局部开发; a_3 表示平衡搜索; a_4 表示多样性救援; a_5 表示精英强化; a_6 表示部分重启; a_7 表示资源重分配。

2.4.2 多目标奖励函数设计

为学习最优策略, 控制器根据行为效果构造奖励

函数:

$$r_t = 0.4\Delta N_t + 0.3\Delta Q_t + 0.2D_t + 0.1R_t \quad (35)$$

其中, ΔN_t 反映帕累托前沿规模的相对增长; ΔQ_t 为两个目标的瞬时改进幅度; D_t 表示用帕累托点的标准差衡量多样性; R_t 为规模奖励上限归一化项。

该函数综合奖励解集扩展性、均匀性改进与精英边界活跃度, 对解集过度聚集予以惩罚。策略网络依据该奖励通过标准 DQN 更新规则进行训练:

$$Q(S_t, a_t) \leftarrow Q(S_t, a_t) + \eta \left[r_t + \gamma \cdot \max_{a'} Q(S_{t+1}, a') - Q(S_t, a_t) \right] \quad (36)$$

其中, η 为学习率, γ 为未来折扣因子, a_t 为当前时间选择的动作, a' 为下一个状态中所有可能的动作。

为提高学习效率, 采用优先级经验回放进行经验采样概率计算:

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha} \quad (37)$$

其中, p_i 表示第 i 个经验在经验回放缓冲区中的优先级:

$$p_i = |\delta_i| + \epsilon \quad (38)$$

$$\delta_i = r_i + \gamma \max_{a'} Q(S_{i+1}, a') - Q(S_i, a_i) \quad (39)$$

其中, $|\delta_i|$ 为第 i 个经验的 TD 误差的绝对值, ϵ 为很小的正数, 确保优先级不为 0。

2.4.3 ϵ -贪婪策略与自适应衰减

动作选择遵循 ϵ -贪婪策略, 为在探索与开发之间实现平滑过渡, ϵ 取值如下:

$$\begin{cases} \epsilon_t = \max(\epsilon_{\min}, \epsilon \times \epsilon_{\text{decay}}) \\ \epsilon_0 = 0.9, \epsilon_{\min} = 0.01, \epsilon_{\text{decay}} = 0.993 \end{cases} \quad (40)$$

当随机数 $\xi \sim \mathcal{U}(0, 1)$ 满足 $\xi < \epsilon_t$ 时, 强化学习协调器随机采样 7 组动作空间; 否则选择对应状态的最大 Q 值动作。协调器伪代码如算法 1。

算法 1. 强化学习协调器

输入: 当前状态 state, 动作空间 ActionSpace。

输出: 选定的策略动作 action。

```

1. Begin
2. Function SelectAction(state)
3.   If random() <  $\epsilon$  then
4.     action  $\leftarrow$  RandomChoice(ActionSpace);
5.   Else
6.     /*前向传播*/

```

```

7.   Q_values ← DQN.Forward(state);
8.   /*选择最优动作*/
9.   action ← ArgMax(Q_values);
10.  EndIf
11.  Return action;
12. EndFunction
13. Function UpdateNetwork(state, action, reward)
14.   experience ← (state, action, reward, next_state);
15.   ReplayBuffer.Store(experience);
16.   If ReplayBuffer.Size() ≥ batch_size then
17.     batch ← ReplayBuffer.Sample(batch_size);
18.     /*计算 TD 损失、反向传播更新*/
19.     loss ← ComputeTDLoss(batch);
20.     DQN.BackwardUpdate(loss);
21.   EndIf
22.   If t % target_update_freq = 0 then
23.     /*更新目标网络*/
24.     TargetDQN ← DQN;
25.   EndIf
26.   ε ← max(ε_min, ε × ε_decay);
27. EndFunction
28. End

```

2.5 结构资源调度机制

行为策略的执行往往需匹配特定资源配置,因此 RLMHHO 将策略动作映射至种群结构调整机制,在策略动作为结构型策略时,系统根据结构比例向量对子群体进行调整。

$$R^{*(t+1)} = [r_E^*, r_D^*, r_B^*, r_H^*], \quad \sum_i r_i^* = 1 \quad (41)$$

$$\Delta N_i = (r_i^* - r_i^{(t)}) \cdot N \quad (42)$$

其中, r_E 、 r_D 、 r_B 、 r_H 分别表示探索组、开发组、均衡组和精英组的目标比例、 N 为总个体数, ΔN_i 表示第 i 个子群体需增加(正)或减少(负)的个体数。

计算各子群体的目标个体数,并与当前比例 $R^{(t)}$ 进行比较,通过式(42)确定第 i 个子群体需增加或减少的个体数。

2.6 RLMHHO 算法流程

RLMHHO 算法实现流程如下。

Step 1: 初始化参数,包括种群规模、工件数 n 、机器数 m 、工厂数 f 、最大迭代次数 T 、强化学习参数。

Step 2: 采用 4 种混沌映射初始化种群,并利用 4 层鹰群分组管理器将种群分为探索组、开发组、均衡组、精英组。

Step 3: 计算所有个体的双目标适应度,评估各组内个体的能量状态和搜索性能。

Step 4: 强化学习协调器基于 DQN 网络(状态空间 14 维、动作空间 7 维)选择搜索策略。

Step 5: 各组使用专属混沌映射更新位置,根据能量状态 E 动态调整搜索策略,执行哈里斯鹰搜索模式,同时根据强化学习决策动态调整 4 层鹰群的比例结构。

Step 6: 更新帕累托档案,利用拥挤距离和 ε -贪婪策略进行环境选择,维护解集的多样性和收敛性。

Step 7: 强化学习协调器根据多目标奖励函数计算收益,更新 Q 值表,采用优先级经验回放机制提升学习效率,实现策略的持续优化。

Step 8: 判断是否达到最大迭代次数 T 。若满足则输出帕累托前沿解集;否则返回 Step 3 继续迭代优化,实现算法的自适应优化。

3 仿真实验及结果分析

目前 MO-DHFSP 尚没有标准实例集,本文根据文献[15,17,18]随机生成多个规模实例:工件数 $n \in \{20, 50, 70, 100, 200\}$, 阶段数 $m \in \{3, 4, 5, 6\}$, 工厂数 $f \in \{2, 3, 4, 5\}$, 并行机数量 $p \in \{2, 3, 4, 5\}$ 。交货期通过以下方式确定:计算作业的总处理时间作为基准值,依据 3 类紧急度(紧急、常规、宽松)随机选取紧急度系数,将其与基准处理时间相乘,得到最终交货期。为评估算法在多目标优化中的表现,引入生成距离(generational distance, GD)、反向生成距离(inverted generational distance, IGD)、超体积(hypervolume, HV)以及帕累托最优解比率(RA)这 5 项指标,分别用于评估帕累托前沿解集的逼近程度、覆盖能力及解集分布的广度和均匀性[1,18]。

3.1 算法参数校正

为优化 RLMHHO 算法性能,采用四因子七水平正交实验设计对关键参数进行校正。选取 4 个核心因子:学习率 α 、衰减率 ε 、鹰群分组比例与折扣因子 γ , 基于 $L_{49}(7^4)$ 正交表构建实验矩阵,以覆盖各因子的典型取值组合,参数配置见表 6、表 7。

实验规模采用 $100 \times 5 \times 3$ 随机数据集进行测试,每组实验独立运行 10 次取平均值,每组实验运行后,采用 5:3:2 加权的多目标评价体系,融合帕累托最优解比率(RA)、IGD 与 GD 计算综合得分,并以田口方法中的信噪比(SNR)作为效应值衡量算法稳健性。正交实验结果在表 8 中,仅展示排名前 10 组数据,实验结果

显示在学习率 0.000 1、衰减率 0.993、鹰群分组比例 [0.7, 0.15, 0.1, 0.05]、折扣因子 0.98 时, 获得最高 SNR 值-17.29, 该配置在探索与开发间取得良好平衡, 在多目标优化任务中实现更好的收敛性和多样性平衡。

表 6 算法参数水平表

水平	学习率	衰减率	分组策略 (鹰群分组比例)	折扣因子
1	0.000 05	0.988	探索主导1	0.80
2	0.000 1	0.990	探索主导2	0.85
3	0.000 2	0.993	探索主导3	0.90
4	0.000 5	0.995	基准均衡	0.95
5	0.001	0.997	开发主导1	0.98
6	0.002	0.999	开发主导2	0.99
7	0.005	0.999 5	开发主导3	0.995

表 7 鹰群分组比例表

水平	探索组	开发组	均衡组	精英组	分组策略
1	0.70	0.15	0.10	0.05	探索主导1
2	0.60	0.20	0.15	0.05	探索主导2
3	0.50	0.30	0.15	0.05	探索主导3
4	0.45	0.25	0.20	0.1	基准均衡
5	0.35	0.40	0.20	0.05	开发主导1
6	0.25	0.45	0.20	0.10	开发主导2
7	0.20	0.50	0.20	0.01	开发主导3

表 8 参数实验结果

学习率	衰减率	鹰群分组比例	折扣因子	SNR值 (dB)
0.000 1	0.993	[0.7, 0.15, 0.1, 0.05]	0.98	-17.29
0.001	0.999 5	[0.6, 0.2, 0.15, 0.05]	0.99	-17.89
0.001	0.99	[0.5, 0.3, 0.15, 0.05]	0.8	-17.95
0.003	0.99	[0.45, 0.25, 0.2, 0.1]	0.9	-18.32
0.000 5	0.999 5	[0.35, 0.4, 0.2, 0.05]	0.95	-18.40
0.000 5	0.995	[0.25, 0.45, 0.2, 0.1]	0.8	-18.41
0.000 05	0.988	[0.2, 0.5, 0.2, 0.1]	0.8	-18.65
0.000 05	0.995	[0.6, 0.2, 0.15, 0.05]	0.95	-18.86
0.003	0.993	[0.5, 0.3, 0.15, 0.05]	0.95	-18.8
0.005	0.997	[0.45, 0.25, 0.2, 0.1]	0.8	-19.00

3.2 对比分析

为验证 RLMHHO 算法的有效性, 选择 MOPSO、MODE、改进 NSGA-II (I-NSGA-II)^[19]、DQN^[2]以及改进 Q-learning 人工蜂群算法 (QL-ABC)^[20]这 5 种算法进行对比实验, 每种算法均独立运行 10 次, 并对实验结果进行统计分析, 结果见表 9-表 12, 加粗为最优结果, 实例编号含义为“工件数_阶段数_工场数”。

从 IGD 指标来看, RLMHHO 在多个实例上的 IGD 值保持在 0.000-0.073 范围内, 均显著低于其他算法, 其中多个实例达到 IGD=0.000 的完美表现。相比之下, 其他算法如 MODE 虽然在个别小规模实例上表现较好, 但整体稳定性不足; 而 I-NSGA-II、MOPSO、DQN

和 QL-ABC 等算法在多个实例上 IGD 值普遍较高。GD 指标进一步验证了 RLMHHO 算法的收敛精度优势, RLMHHO 在多个实例上的 GD 值保持在 0.02-0.143 范围内, 即使在 200 作业的大规模实例上仍能保持 GD 为 0.047 的优异表现。HV 指标显示了算法解集的覆盖范围和多样性, RLMHHO 在多个实例上的 HV 值保持在 0.871-1.000 范围, 其中小规模实例达到 HV=1.000 的最优水平, 显著高于其他对比算法, 说明其解集在多目标空间中分布均匀且接近最优前沿。RA 指标反映了算法获得的解集质量, RLMHHO 在 8 个实例上获得最高 RA 值, 其在多个实例上的 RA 值保持在 0.279-0.811 范围内, 均显著高于其他对比算法。

表 9 各算法 IGD 对比

实例	RLMHHO	I-NSGA-II	MOPSO	MODE	DQN	QL-ABC
20_3_2	0.067	0.347	0.278	0.020	0.270	0.169
20_5_3	0.000	0.083	0.155	0.000	0.177	0.000
50_3_2	0.005	0.293	0.322	0.011	0.234	0.212
50_5_3	0.073	0.221	0.468	0.243	0.413	0.414
70_3_2	0.033	0.309	0.391	0.134	0.410	0.405
70_5_3	0.000	0.300	0.416	0.316	0.348	0.447
100_3_2	0.013	0.166	0.302	0.102	0.345	0.254
100_5_3	0.018	0.362	0.429	0.126	0.649	0.427
200_3_2	0.041	0.260	0.373	0.120	0.464	0.352
200_5_3	0.003	0.211	0.347	0.128	0.592	0.297

表 10 各算法 GD 对比

实例	RLMHHO	I-NSGA-II	MOPSO	MODE	DQN	QL-ABC
20_3_2	0.022	0.442	0.383	0.014	0.686	0.294
20_5_3	0.079	0.203	0.408	0.085	0.595	0.071
50_3_2	0.143	0.307	0.319	0.018	0.210	0.475
50_5_3	0.037	0.190	0.362	0.307	0.465	0.360
70_3_2	0.016	0.262	0.271	0.087	0.580	0.254
70_5_3	0.015	0.388	0.428	0.358	0.488	0.581
100_3_2	0.069	0.092	0.237	0.116	0.634	0.196
100_5_3	0.032	0.214	0.478	0.123	0.755	0.323
200_3_2	0.059	0.194	0.256	0.117	0.624	0.288
200_5_3	0.047	0.259	0.291	0.151	0.680	0.295

表 11 各算法 HV 对比

实例	RLMHHO	I-NSGA-II	MOPSO	MODE	DQN	QL-ABC
20_3_2	0.977	0.614	0.851	1.000	0.762	0.872
20_5_3	1.000	0.981	0.949	1.000	0.943	1.000
50_3_2	0.987	0.676	0.596	0.900	0.728	0.706
50_5_3	0.871	0.579	0.470	0.782	0.506	0.676
70_3_2	1.000	0.767	0.695	0.892	0.687	0.734
70_5_3	0.939	0.580	0.571	0.702	0.654	0.597
100_3_2	0.900	0.851	0.791	0.829	0.745	0.809
100_5_3	0.941	0.784	0.655	0.801	0.551	0.705
200_3_2	0.924	0.711	0.755	0.888	0.503	0.741
200_5_3	0.927	0.769	0.759	0.868	0.566	0.803

结果显示, RLMHHO 算法具有良好的可扩展性。

在小规模实例上,所有算法表现相对较好,但 RLMHHO 能够达到完美性能;在中等规模实例上,算法性能开始分化,RLMHHO 的优势更加明显;在大规模实例上,其他算法性能显著下降,而 RLMHHO 仍能保持优异表现。

表 12 各算法 RA 对比

实例	RLMHHO	I-NSGA-II	MOPSO	MODE	DQN	QL-ABC
20_3_2	0.309	0.013	0.120	0.490	0.000	0.067
20_5_3	0.279	0.115	0.032	0.271	0.012	0.291
50_3_2	0.512	0.040	0.030	0.322	0.083	0.014
50_5_3	0.737	0.146	0.000	0.076	0.000	0.041
70_3_2	0.588	0.058	0.039	0.272	0.000	0.043
70_5_3	0.898	0.035	0.000	0.047	0.021	0.000
100_3_2	0.612	0.089	0.063	0.145	0.000	0.091
100_5_3	0.640	0.077	0.000	0.250	0.000	0.033
200_3_2	0.742	0.025	0.000	0.233	0.000	0.000
200_5_3	0.811	0.039	0.000	0.150	0.000	0.000

图 1 和图 2 为不同规模数据集下,算法在最大完工时间 (makespan) 与最大延迟时间 (total tardiness) 两个目标上的帕累托前沿分布情况。

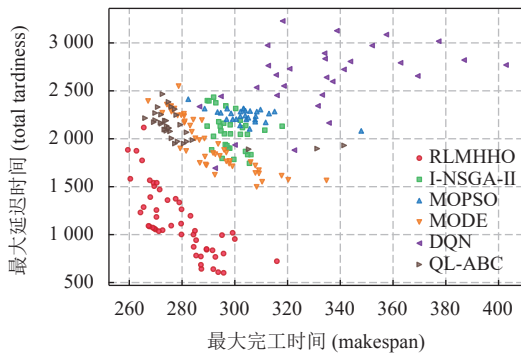


图 1 规模 50_3_2 帕累托前沿效果对比图

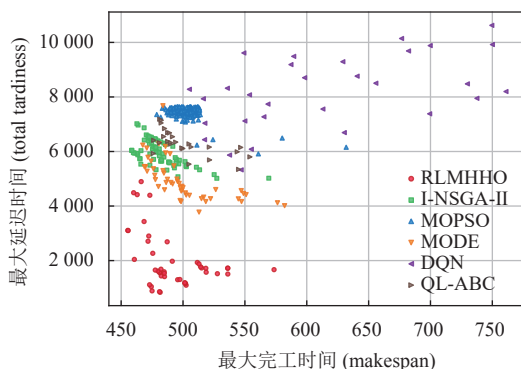


图 2 规模 100_5_3 帕累托前沿效果对比图

4 结语

本文提出的 RLMHHO 算法有效解决了多目标分

布式混合流水车间调度问题,通过融合强化学习协调器、4 组鹰群分组管理和增强混沌映射系统,实现了完工时间和总延迟时间的协同优化。实验结果表明,该算法具有良好的收敛性和解集多样性。然而,算法仍存在改进空间,强化学习协调器的状态空间设计有待进一步优化,可考虑引入更丰富的环境特征;混沌映射自适应机制需要完善,以提高算法对不同问题规模的适应性;未来研究可从深度强化学习、自适应参数控制等方向深入,为复杂生产调度问题提供更高效的解决方案。

参考文献

- 李颖俐. 基于人工蜂群算法的分布式混合流水车间调度方法研究 [博士学位论文]. 武汉: 华中科技大学, 2021. [doi: 10.27157/d.cnki.ghzku.2021.006559]
- 夏保丽, 马永忠. 基于深度 Q 学习网络的分布式流水车间调度问题优化. 机械设计与研究, 2025, 41(1): 311–319. [doi: 10.13952/j.cnki.jofmdr.2025.0054]
- 王镜捷, 陶翼飞, 崔海, 等. 多目标社会工程粒子群算法求解分布式混合流水车间调度问题. 控制工程, <https://doi.org/10.14107/j.cnki.kzgc.20230325>. (2024-02-27).
- 轩华, 熊梦莹, 曹颖. 基于改进灰狼优化算法的分布式混合流水线重调度. 郑州大学学报 (工学版), 2025, 46(4): 47–54, 99. [doi: 10.13705/j.issn.1671-6833.2025.04.020]
- 邱荣根. 基于改进 NSGA-II 算法的混合流水车间分批调度问题研究. 自动化技术与应用, <https://link.cnki.net/urlid/23.1474.TP.20241203.1407.018>. (2024-12-03).
- Cai JC, Lu SJ, Cheng J, *et al.* Collaborative variable neighborhood search for multi-objective distributed scheduling in two-stage hybrid flow shop with sequence-dependent setup times. Scientific Reports, 2022, 12(1): 15724. [doi: 10.1038/s41598-022-19215-3]
- Wang YJ, Li J, Wang GG. Fuzzy correlation entropy-based NSGA-II for energy-efficient hybrid flow-shop scheduling problem. Knowledge-based Systems, 2023, 277: 110808. [doi: 10.1016/j.knosys.2023.110808]
- Heidari AA, Mirjalili S, Faris H, *et al.* Harris hawks optimization: Algorithm and applications. Future Generation Computer Systems, 2019, 97: 849–872. [doi: 10.1016/j.future.2019.02.028]
- 段宏瑾, 张毅. 基于改进 Harris 鹰算法的微电网优化调度. 计算机与现代化, 2025(6): 34–41. [doi: 10.3969/j.issn.1006-2475.2025.06.006]
- Zhu YT, Zhang WY, Wang HN, *et al.* An improved chaotic quantum multi-objective Harris hawks optimization algorithm for emergency centers site selection decision

- problem. *Computers, Materials & Continua*, 2025, 82(2): 2177–2198. [doi: [10.32604/cmc.2024.057441](https://doi.org/10.32604/cmc.2024.057441)]
- 11 Yin J, Zhang SN, Liu L, *et al.* An improved discrete Harris Hawks optimization algorithm for the no-wait job shop problem to minimize total weighted tardiness. *The Journal of Supercomputing*, 2025, 81(6): 753. [doi: [10.1007/s11227-025-07234-6](https://doi.org/10.1007/s11227-025-07234-6)]
- 12 Lv ZL, Zhao YX, Kang HY, *et al.* An improved Harris hawk optimization algorithm for flexible job shop scheduling problem. *Computers, Materials & Continua*, 2024, 78(2): 2337–2360. doi: [10.32604/cmc.2023.045826](https://doi.org/10.32604/cmc.2023.045826).
- 13 李晨. 分布式流水车间调度的混合粒子群优化研究 [硕士学位论文]. 郑州: 河南工业大学, 2023. [doi: [10.27791/d.cnki.ghegy.2023.000585](https://doi.org/10.27791/d.cnki.ghegy.2023.000585)]
- 14 代礼奇, 彭可, 崔焱, 等. 基于多种群遗传算法与 Plant Simulation 的车间调度优化. *智能制造*, 2022(3): 81–86. [doi: [10.3969/j.issn.1671-8186.2022.03.020](https://doi.org/10.3969/j.issn.1671-8186.2022.03.020)]
- 15 雷德明, 苏斌. 基于多班教学优化的多目标分布式混合流水车间调度. *控制与决策*, 2021, 36(2): 303–313. [doi: [10.13195/j.kzyjc.2020.0549](https://doi.org/10.13195/j.kzyjc.2020.0549)]
- 16 Liu C. An improved Harris hawks optimizer for job-shop scheduling problem. *The Journal of Supercomputing*, 2021, 77(12): 14090–14129. [doi: [10.1007/s11227-021-03834-0](https://doi.org/10.1007/s11227-021-03834-0)]
- 17 李浩然, 高亮, 李新宇. 基于离散人工蜂群算法的多目标分布式异构零等待流水车间调度方法. *机械工程学报*, 2023, 59(2): 291–306. [doi: [10.3901/JME.2023.02.291](https://doi.org/10.3901/JME.2023.02.291)]
- 18 颜雪松, 张峻华, 胡成玉. 面向分布式同构混合流水车间绿色调度的多目标优化方法. *控制理论与应用*, <https://link.cnki.net/urlid/44.1240.TP.20250311.1635.076>. (2025-03-12).
- 19 Chu XX, Yu XJ. Improved crowding distance for NSGA-II. arXiv:1811.12667, 2018.
- 20 杜利珍, 宣自风, 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. *组合机床与自动化加工技术*, 2024(10): 175–180. [doi: [10.13462/j.cnki.mmtamt.2024.10.035](https://doi.org/10.13462/j.cnki.mmtamt.2024.10.035)]

(校对责编: 张重毅)