

文章编号: 1006-2467(2023)10-1378-11

DOI: 10.16183/j.cnki.jsjtu.2022.242

基于改进多种群候鸟迁徙算法的混合流水车间调度

张素君¹, 杨文强¹, 顾幸生²

(1. 河南科技学院 机电学院, 河南 新乡 453003;

2. 华东理工大学 能源化工过程智能制造教育部重点实验室, 上海 200237)

摘 要: 针对带顺序依赖准备时间的混合流水车间调度(HFS-SDST)问题,以最小化总最大作业完成时间为调度目标,提出一种改进多种群候鸟迁徙优化(IMMBO)算法.算法中个体基于工件加工顺序进行编码,用改进的NEH(MNEH)算法产生初始种群,并按照适应度值分配到各子种群.子种群中领飞鸟和跟飞鸟分别利用串行和并行邻域策略产生邻域个体,如果跟飞鸟优于领飞鸟,二者互换,完成种群内部个体的信息交互;在IMMBO算法中嵌入离散鲸鱼优化策略对各子种群的领飞鸟进行优化,实现子种群之间信息交互;为提高算法的局部搜索(LS)能力,对种群中最优个体执行LS,同时,为了避免算法早熟收敛,针对每个种群的领飞鸟设计了种群多样化控制策略.最后,在实验法调整算法参数的基础上,对IMMBO的4个变体进行了仿真实验,通过测试Ta自适应算例验证IMMBO算法各部分的作用;将IMMBO算法与现有3个算法测试Ta自适应算例,进行实验结果比较,证明了IMMBO算法求解混合车间调度问题的有效性.

关键词: 混合流水车间调度;改进多种群候鸟迁徙优化;子种群信息交互;串行邻域;并行邻域

中图分类号: TP 278

文献标志码: A

An Improved Multi-Swarm Migrating Birds Optimization Algorithm for Hybrid Flow Shop Scheduling

ZHANG Sujun¹, YANG Wenqiang¹, GU Xingsheng²

(1. School of Mechanical and Electrical Engineering, Henan Institute of Science and Technology, Xinxiang 453003, Henan, China; 2. Key Laboratory of Advanced Control and Optimization for

Chemical Process of the Ministry of Education, East China University of Science and Technology, Shanghai 200237, China)

Abstract: An improved multi-swarm migrating birds optimization (IMMBO) algorithm is proposed for hybrid flow shop scheduling with sequence-dependent setup times (HFS-SDST), to minimize the total maximum completion time (i. e., makespan). Permutation-based encoding is adopted to substitute the individual. The modified Nawaz-Enscore-Ham (MNEH) algorithm is employed to generate initial population which are assigned to each sub-swarm according to the makespan. For each sub-swarm, the neighborhood individuals of the leader and followers are generated respectively by performing serial and parallel neighborhood strategies. If the follower is better than the leader according to their makespan, they are exchanged to ensure the information interaction of individuals within the sub-swarm. Moreover, the

收稿日期: 2022-06-27 修回日期: 2022-07-24 录用日期: 2022-07-27

基金项目: 国家自然科学基金(61973120, 61973209)资助项目, 河南省科技攻关(202102110282, 222102110095)项目

作者简介: 张素君(1978-), 讲师, 从事生产调度与智能优化算法研究.

通信作者: 顾幸生, 教授, 博士生导师; E-mail: xsgu@ecust.edu.cn.

discrete whale optimization strategy is embedded in IMMBO to optimize the leaders of all sub-swarms to enhance the interaction among them. Furthermore, the local search is designed for the optimal individual to further improve the local search ability of the algorithm. Meanwhile, to avoid algorithm premature convergence, the control strategy for population diversification is designed to the leader of each sub-swarm. Finally, based on adjusting the algorithm parameters experimentally, simulation experiments are conducted on four variants of IMMBO to verify the function of each part by testing an adaptation dataset of Ta. Moreover, the IMMBO is compared with three existing algorithms by testing an adaptation dataset of Ta, and the experimental results demonstrate the effectiveness of the IMMBO algorithm to solve the hybrid flow shop scheduling problem.

Key words: hybrid flow shop scheduling (HFS) problem; improved multi-swarm migrating birds optimization (IMMBO); information interaction among multi-swarm; parallel neighborhood; serial neighborhood

带顺序依赖准备时间的混合流水车间调度^[1] (HFS-SDST)问题是混合流水车间调度^[2] (HFS)问题的一个重要分支. HFS-SDST 广泛存在于钢铁、电子、化工、流程工业及离散智能制造等调度问题中^[2], 70%的实际工业过程需要考虑 SDST, 若不能正确处理该问题, 将使用超过 20%的机器容量^[1], 而且考虑 SDST 时, 92%的订单会达到交货期^[3]. 因此, 虽然在 HFS 问题中考虑 SDST 更贴近生产实际, 但建模难度会增加, 设计调度算法时考虑的约束更多, 寻找全局最优解更加困难^[4].

HFS-SDST 问题的调度涉及工件的排序和机器分配, 是典型的组合优化问题, 已经证明 HFS-SDST 是非确定性多项式难(NP-hard)的问题^[5]. 因此, 利用精确算法理论上可以求解, 但随着问题规模的增大出现组合爆炸问题, 几乎不可能在合理的时间内找到最优调度解, 难以满足实际生产需要. 启发式算法依据某些规则对调度解进行构造, 大大缩短了寻找调度解的时间, Moccellini 等^[6]提出基于最短处理时间(SPT)和最长处理时间(LPT)规则的启发式算法, 求解针对机器阻塞、顺序依赖、顺序独立准备时间的 HFS 问题, 结果证明了对于 HFS-SDST 问题, 最长处理和准备时间(LPS)规则优于其他规则. 然而, 对于较大规模的复杂调度问题, 启发式规则构造的调度解质量不高, 因此, 采用智能优化算法为快速求解组合优化问题提供了可能; 文献[7]中提出基于超启发式学习增强的迭代贪心算法求解 HFS-SDST 问题, 采用该算法对一个实际生产实例的设备数据进行求解可以达到较好的效果; 周炳海等^[8]提出蚁群算法求解双目标 HFS 问题, 但该算法仅限于求解较小规模的问题; 文献[9]中提出基于代理的改进遗传算法, 融合了多智能体和遗传算法的优点求解 HFS-SDST 问题, 并通过实验得到一些算

例的下限值; Pan 等^[10]列出 6 个局部搜索算法和 3 个群智能优化算法, 其中 3 个群智能算法为改进的果蝇优化算法(FOA)、改进的候鸟迁徙优化(IMBO)和离散人工蜂群(DABC)算法, 并给出一种协同优化策略, 通过仿真测试对比可知, 对于大部分算例, DABC 优于其他 8 个算法; Tian 等^[11]提出基于 pareto 的自适应变邻域搜索算法, 证明了变邻域搜索求解 HFS-SDST 问题的有效性; Khare 等^[12]提出了混合松鼠搜索、反向鲸鱼及离散灰狼 3 个群智能算法, 并且为了提高算法的性能, 在松鼠算法中引入变邻域搜索和混合局部搜索, 在鲸鱼算法中引入反向学习策略, 优化目标为总提早和拖期. 文献[10, 12]中为群智能算法求解 HFS-SDST 问题的有效性提供了依据. 虽然针对 HFS-SDST 问题已有一些研究, 但算法方面还未形成系统理论成果, 因此, 研究更高性能的调度算法具有深刻的理论和实际应用意义.

Duman 等^[13]在 2012 年提出的候鸟迁徙优化(MBO)算法, 因调整参数少、结构简单等优点, 得到了广泛应用, 但该算法容易早熟收敛且不能直接应用于组合优化问题, 因此, Pan 等^[14]和 Han 等^[15]分别提出改进的离散 MBO 算法求解 HFS 问题, 且在此基础上, Pan 等^[10]对文献[14]中提出的 IMBO 进行了改进. Sioud 等^[3]针对带 SDST 的置换流水车间调度(PFSP)问题提出增强的候鸟迁徙优化(EMBO)算法, 利用禁忌表控制混合邻域策略产生邻域解的多样性; Meng 等^[16]和汤洪涛等^[17]分别提出采用和声搜索(HS)策略以及变邻域搜索策略产生邻域解的离散 MBO, 求解批量流车间调度问题. 然而, MBO 算法是一种邻域搜索算法, 因此邻域的选择对算法的优化效果影响较大, 而变邻域搜索是一种可以系统地利用邻域变化思想的算法; Báez 等^[18]提出

结合贪心随机自适应力 (GRASP) 和变邻域搜索的混合算法求解带 SDST 的平行机调度问题. 为了改善 MBO 的早熟收敛问题, 文献[3]中采用重置机制提高算法的全局搜索能力, 此外, 采用多种群智能优化算法为提高该能力提供了另一种思路; Han 等^[19]提出改进的多种群离散差分进化算法, 求解多用途批处理设备调度问题; Gao 等^[20]提出一种动态洗牌的多微种群候鸟迁徙优化 (SM²-MBO) 算法, 利用多微子种群并行搜索, 求解多资源约束柔性作业车间调度问题, 但该算法不能在子种群信息交互时产生更好新个体; Tongur 等^[21]提出了基于粒子群优化 (PSO) 算法的改进多种群 MBO (IMFMBO) 算法, 在多种群 MBO 框架下, 通过另一种算法促进子种群之间的信息交互, 提高算法的全局搜索能力. 因此, 本文结合候鸟迁徙优化、子种群交互策略和多种群并行搜索思想, 提出 IMMBO 算法求解 HFS-SDST 问题.

1 HFS-SDST 问题

HFS-SDST 问题描述如下: n 个工件在 m 个阶段的流水线上加工, 每个工件 j ($j = 1, 2, \dots, n$) 依次按相同的顺序通过阶段 k ($k = 1, 2, \dots, m$), m 个阶段至少有一个阶段的机器数量 $M_k > 1$, 每个工件可以在阶段 k 的任意一台机器上加工. 加工过程满足假设: ①任意时刻每个工件只能在一台机器上加工; ②每台机器同一时间只能加工一个工件; ③工件在某阶段机器上一旦开始加工不允许中断. 优化目标为通过确定工件的加工顺序以及每阶段工件在机器上的分配情况, 使工件的最大加工完成时间 (makespan) 即 C_{\max} 最小.

考虑 SDST, n 个工件在第 1 阶段机器上的加工顺序一旦确定, 依据工件在可用机器上的加工完成时间最短的标准, 工件在后续阶段机器上的加工顺序会相应确定. 若工件在第 1 阶段的加工顺序为 π , $\pi = (\pi_1, \pi_2, \dots, \pi_n)$, 则给出如下 HFS-SDST 问题的数学模型.

若第 1 阶段和阶段 k ($k = 2, 3, \dots, m$) 的第 i ($i = 1, 2, \dots, M_k$) 台机器没有加工过其他工件, 则工件 π_j 的完成时间 $C_{k, \pi_j, i}$ 分别由下式给出:

$$C_{k, \pi_j, i} = S_{k, \pi_j, \pi_j} + p_{k, \pi_j}, \quad k = 1 \quad (1)$$

$$C_{k, \pi_j, i} = \max\{S_{k, \pi_j, \pi_j}, C_{k-1, \pi_j}\} + p_{k, \pi_j}, \quad k = 2, \dots, m \quad (2)$$

式中: $C_{k, \pi_j, i}$ 和 p_{k, π_j} 分别为工件 π_j 在阶段 k 的第 i 台机器上的作业完成时间和作业时间; S_{k, π_j, π_j} 为工件 π_j 在 k 阶段的准备时间.

如果阶段 k 的第 i 台机器加工 π_j 前已经加工过其他工件, 则工件 π_j 的加工完成时间为

$$C_{k, \pi_j, i} = \max\{C_{k, \pi_{\tau_{k, i}}}, S_{k, \pi_{\tau_{k, i}}, \pi_j}, C_{k-1, \pi_j}\} + p_{k, \pi_j} \quad (3)$$

式中: $C_{k, \pi_{\tau_{k, i}}}$ 为阶段 k 的第 i 台机器在加工 π_j 的前 1 个工件 $\pi_{\tau_{k, i}}$ 的加工完成时间; $S_{k, \pi_{\tau_{k, i}}, \pi_j}$ 为 π_j 在阶段 k 的准备时间. 工件 π_j 在阶段 k 的机器选择依据 π_j 在该阶段的作业完成时间最短的原则, 即 π_j 在阶段 k 加工完成时间最短的机器为

$$i^* = \arg \left(\min_{i=1, 2, \dots, M_k} C_{k, \pi_j, i} \right)$$

其作业完成时间为

$$C_{k, \pi_j} = \min C_{k, \pi_j, i} \quad (4)$$

最后一个工件在最后阶段机器上的加工完成时间的最大值为

$$C_{\max} = \max_{j=1, 2, \dots, n} C_{m, \pi_j} \quad (5)$$

优化调度的目标为找到 C_{\max} 最小的 n 个工件在第 1 阶段机器上的加工顺序, 即 $\pi^* = (\pi_1^*, \pi_2^*, \dots, \pi_n^*)$ 是所有调度可行解集合 Π 中的一个调度解.

2 IMMBO

IMMBO 算法把初始化种群分成多个子种群, 各个子种群利用 MBO 机制平行搜索, 子种群之间利用离散鲸鱼优化策略对各子种群的领飞鸟优化完成信息交互; 同时设计局部搜索增强和种群多样化控制机制提高算法的探索和开发能力, IMMBO 算法框架如下.

步骤 1 设置算法相关参数. 利用 MNEH 算法初始化种群, 并把最好的 1/3 分到每个子种群作为其领飞鸟, 剩下 2/3 个体随机分配到各子种群, 每个子种群包括两个跟飞鸟.

步骤 2 保存本代最优个体.

步骤 3 子种群领飞鸟利用串行变邻域搜索策略产生邻域个体, 若邻域个体中最好的优于领飞鸟, 替换领飞鸟.

步骤 4 子种群跟飞鸟利用并行变邻域搜索策略产生邻域个体, 若邻域个体中最好的优于该跟飞鸟, 替换跟飞鸟, 若跟飞鸟优于本群的领飞鸟, 二者交换.

步骤 5 检查是否达到巡回次数, 若未达到, 转到步骤 3, 否则转到步骤 6.

步骤 6 各子种群领飞鸟利用离散鲸鱼优化策略进行交叉, 完成子种群之间的信息交互.

步骤 7 对种群中的最优个体执行局部搜

索(LS).

步骤 8 更新各子种群领飞鸟的年龄变量,若某子种群的领飞鸟年龄达到设定年龄限制,启动多样化控制机制,重新产生一个个体放入种群.

步骤 9 判断算法迭代是否达到最大迭代次数,若未达到,转到步骤 2,否则,算法结束,输出最优个体及其 C_{\max} 值.

2.1 编码和解码

连续的优化算法不能直接应用于组合优化问题 HFS-SDST,故需要设计离散形式的优化算法,而设计离散优化算法需要对算法中个体进行离散编码和解码.针对 HFS 问题,常用的编码和解码方法有矩阵和向量两种,本文在考虑 SDST 的同时采用向量编码,即基于所有工件的一个调度顺序对个体编码^[12],然后,将工件按照排列的顺序分配到第 1 阶段的空闲机器上加工.解码方法为:考虑准备时间,工件在第 1 阶段按照编码顺序加工,第 $k(k = 2, \cdots, m)$ 阶段的工件加工顺序则依据在该阶段加工完成时间最短的原则,重新排列得到工件在阶段 k 的加工顺序.

2.2 初始化种群

NEH 启发式规则是依据工件在所有阶段机器上的总加工时间越短,优先级别越高的原则得到的工件排序,适用于调度目标为最大完成时间的流水车间调度问题,若考虑带 SDST 的车间调度问题,不仅需要考虑总加工时间,同时需要考虑 SDST.针对带 SDST 的平行机调度,对于未开始工作的机器,工件 j 的加工完成时间 $\phi_{1,j} = s_{1,j,j} + p_{1,j}$,其中 $s_{1,j,j}$ 和 $p_{1,j}$ 分别为工件 j 在平行机上的准备时间和加工时间,求出每个工件的完成时间后,按照由小到大的顺序进行加工可以使加工完成时间最短.但对于多阶段的 HFS-SDST 问题,需要考虑工件 j 在所有阶段的加工完成时间和 ϕ_j ,令 $\phi_j = \sum_{k=1}^m \phi_{k,j}$,其中 $\phi_{k,j} = s_{k,j,j} + p_{k,j}$.采用 MNEH^[10]产生初始种群,描述如下:

- (1) 计算每个工件的 ϕ_j ,对 ϕ_j 升序排列,得到工件在第 1 阶段的排序 $\pi = (\pi_1, \pi_2, \cdots, \pi_n)$.
- (2) 依据第 1 阶段的机器数 M_1 ,把 π 分成两个子序列 $\pi^1 = (\pi_1, \pi_2, \cdots, \pi_{M_1})$ 和 $\pi^2 = (\pi_{M_1+1}, \pi_{M_1+2}, \cdots, \pi_n)$, $\pi = \pi^1 \cup \pi^2$.
- (3) 随机调换 π^2 中工件的顺序得到 $\pi^{2'}$,与 π^1 合并得到一个新的第 1 阶段工件的排序 $\pi' = \pi^1 \cup \pi^{2'}$.
- (4) 重复(3),得到与种群规模相同的工件在第 1 阶段机器上的加工顺序,并分别求出其 C_{\max} .

针对初始种群,按种群中个体的 C_{\max} 值升序排列,其中前 1/3 分配到各个子种群,作为领飞鸟;其余 2/3 随机分配到各子种群,作为跟飞鸟.子种群的规模为 3,即 1 个领飞鸟和 2 个跟飞鸟.

2.3 子种群领飞鸟进化

领飞鸟是一个子种群中最好的个体,领飞鸟的质量决定子种群的搜索方向.各子种群中的领飞鸟通过其邻域个体更新.文献[22]中已经证明插入、交换及迭代贪婪(IG)算法中的破坏重建(DC)操作在流水车间调度问题中是较好的邻域解产生策略,因此,领飞鸟的邻域个体由插入、交换以及 DC 串联起来的串行变邻域搜索策略产生,即依次执行插入、交换和 DC,只要采用某一策略能够产生更好的邻域个体就继续使用,反之,则执行下一个策略,直到结束.

串行变邻域策略伪代码如下:

输入 π ; 输出 π
 $k=1$
while $k \leq 3$
 switch k
 case 1 随机选择位置 r_1 上的工件, $1 \leq r_1 \leq n$, 插入到 π 的所有可能位置,并依据指标 C_{\max} , 找到最好的 π'
 如果 $C_{\max}(\pi') < C_{\max}(\pi)$
 $\pi = \pi', k=1$
 否则
 $k=k+1$
 case 2 随机选择位置 r_1 上的工件, $1 \leq r_1 \leq n$, 与 π 的其他所有位置上工件交换,并依据指标 C_{\max} , 找到最好的 π'
 如果 $C_{\max}(\pi') < C_{\max}(\pi)$
 $\pi = \pi', k=2$
 否则
 $k=k+1$
 case 3 对 π 执行破坏重建(DC),得到 π'
 如果 $C_{\max}(\pi') < C_{\max}(\pi)$
 $\pi = \pi', k=3$
 否则
 $k=k+1$
 End switch
End while

2.4 子种群跟飞鸟进化

产生跟飞鸟邻域个体的策略,要兼具邻域个体

的质量和多样性,因此子种群中的跟飞鸟领域个体采用并行变邻域搜索产生,即按照一定概率选择插入或交换两种邻域之一产生跟飞鸟邻域个体.在生产调度问题中,插入更易产生较好的邻域个体,因此,设置选择插入邻域结构的概率设为 $P_m=0.6$,即产生一个随机数 $r, r \in [0, 1]$,如果 $r < P_m$,执行插入操作直到该跟飞鸟不能更新为止,否则执行交换.对邻域个体采用贪婪选择,只要产生的邻域个体优于跟飞鸟,则跟飞鸟被替换;同时,在跟飞鸟被替换后,若跟飞鸟优于其领飞鸟,二者交换,完成子种群中领飞鸟和跟飞鸟的信息交互.

2.5 巡回

多个子种群领飞鸟和跟飞鸟依据 2.3 节和 2.4 节循环 G 次,利用其邻域解,完成领飞鸟和跟飞鸟的充分开发.

2.6 子种群信息交互

IMMBO 算法采用多种群并行搜索,但随着算法的执行,子种群内个体失去多样性,因此需要通过子种群之间的信息交互,产生更好的个体.鲸鱼优化算法(WOA)^[23]是最优个体引领搜索方向的算法,这是利用鲸鱼可以通过螺旋上升过程中泡泡网围攻以及随机搜索猎物完成捕食过程.因此,设计离散鲸鱼优化算法(DWOA)完成子种群之间的信息交互.最优个体存在于各子种群的领飞鸟中,因此,DWOA 只对它们优化,同时利用 3 种交叉策略模拟鲸鱼的捕食过程.鲸鱼围攻和螺旋上升过程分别利用当前个体和最优个体进行次序交叉和两段交叉模拟,分别以 50% 的概率完成局部搜索,产生一个随机数 $p, p \in [0, 1]$,若 $p < 0.5$,执行次序交叉,否则执行两段交叉;随机搜索猎物过程则利用当前个体和种群中随机个体进行随机交叉模拟完成全局搜索.3 种交叉的具体操作如图 1~3 所示,其中 P1 和 P2 为两个父代个体,C1 和 C2 为两个子代交叉个体.

(1) 次序交叉(OX).

步骤 1 随机选择两个交叉点 r_1 和 r_2 ,且 $1 \leq$

		r_1 r_2					
P1	3	5	7	2	1	8	4 6
C1	4	7	6	2	1	8	5 3
C2	3	7	1	8	2	5	4 6
P2	4	7	6	8	2	5	3 1

图 1 次序交叉操作
Fig. 1 Order crossover operation

$r_1 < r_2 \leq n$.
步骤 2 把 P1 和 P2 中 $r_1 \sim r_2$ 的工件分别复制到 C1 和 C2 中的 $r_1 \sim r_2$ 位置.

步骤 3 把 P1 不包含在 C2 中的工件依次复制到 C2;把 P2 不包含在 C1 中的工件依次复制到 C1.

步骤 4 将 C1 和 C2 中 C_{\max} 值较小的作为交叉结果.

(2) 随机交叉(JBX).

步骤 1 构造两个子序列 S1,S2.

步骤 2 把 P1 中对应 S1 的元素复制到 C1;把 P2 中对应 S2 的元素复制到 P2.

步骤 3 把 P2 中对应 S2 的元素复制到 C1;把 P1 中对应的 S1 的元素复制到 C2.

步骤 4 将 C1 和 C2 中 C_{\max} 值较小的作为交叉结果.

(3) 两段交叉(TSX).

步骤 1 随机选择两个交叉点 r_1 和 r_2 ,且 $1 \leq r_1 < r_2 \leq n$.

步骤 2 把 P1 中 $r_1 \sim r_2$ 的工件复制到子序列 sub1;从 P2 中去除 sub1 中工件,剩余的工件作为子序列 sub2.

步骤 3 把子序列 sub1 和 sub2 复制到 C1,把子序列 sub2 和 sub1 复制到 C2.

步骤 4 如果 $r < 0.5$,C1 作为交叉结束,否则 C2 作为交叉结果.

		S1={1,2,6,7}; S2={3,4,5,8}					
P1	3	5	7	2	1	8	4 6
C1	4	8	7	2	1	5	3 6
C2	4	7	2	8	1	5	3 6
P2	4	7	6	8	2	5	3 1

图 2 随机交叉操作
Fig. 2 Job-based crossover operation

		r_1 r_2					
P1	3	5	7	2	1	8	4 6
C1	2	1	8	4	7	6	5 3
C2	4	7	6	5	3	2	1 8
P2	4	7	6	8	2	5	3 1

图 3 两段交叉操作
Fig. 3 Two-segment crossover operation

在 DWOA 中,LS 和全局搜索平衡至关重要. 为达到更好的优化效果,通过一个选择概率 P^b 完成算法探索和开发的切换, P^b 采用非线性的正弦函数替代 WOA^[23] 中的线性函数,由下式给出:

$$P^b = P^b_{\max} - (P^b_{\max} - P^b_{\min})\sin \frac{t\pi}{2t_{\max}} \quad (6)$$

式中: P^b_{\max} 和 P^b_{\min} 分别为 P^b 的最大值和最小值,即 1 和 0; t 和 t_{\max} 分别为当前进化代数和最大进化代数. 如果随机数 $r > P^b$ 执行 LS, 否则执行全局搜索. DWOA 伪代码如下:

```
For 每个个体
    如果  $p < 0.5$ 
        如果  $r > P^b$ 
            种群最优个体与当前个体执行次序交叉
```

```
(OX)
否则
    当前个体与随机选择个体进行随机交叉 (JBX)
End if
否则
    种群最优个体与当前个体执行两段交叉 (TSX)
End if
End for
```

2.7 LS 能力增强

为了进一步提高全局最优个体的质量,对其进行 LS. 但最优个体是经过多次进化所得,直接进行 LS 可能进入循环搜索,因此,首先对最优个体进行干扰,然后再执行 LS. LS 算法流程图如图 4 所示.

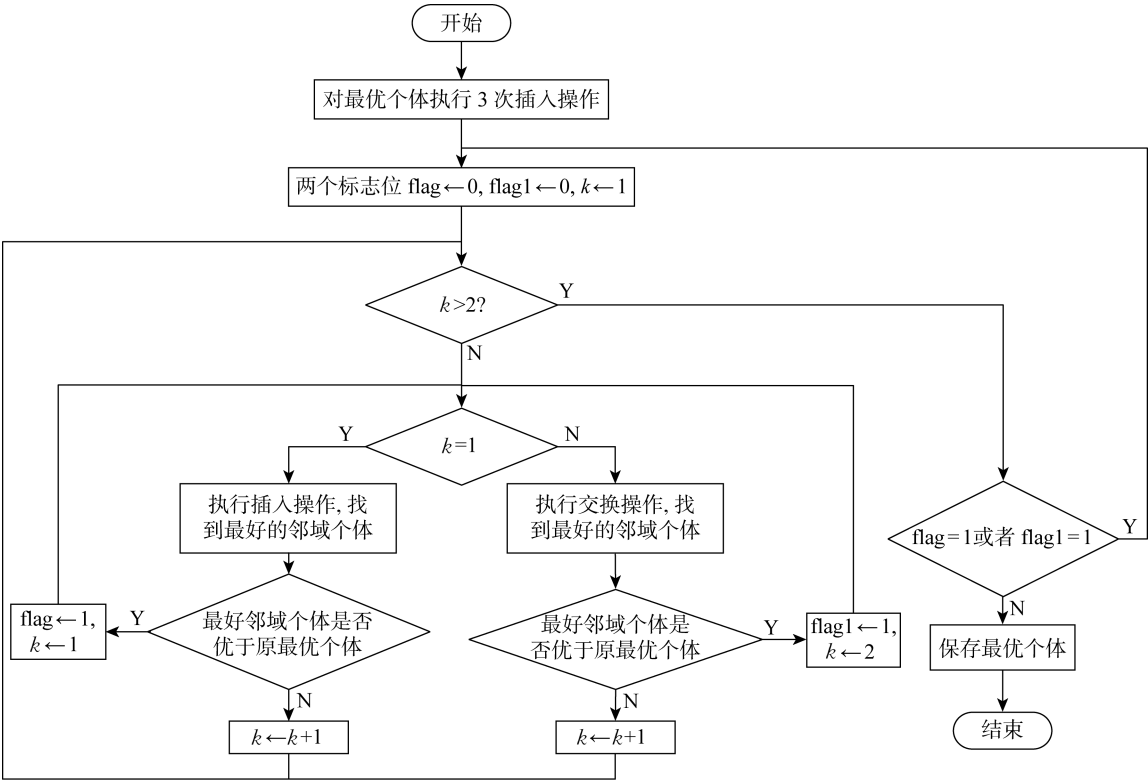


图 4 局部搜索算法流程图
Fig. 4 Flow chart of LS algorithm

2.8 种群多样化控制

随着 IMMBO 算法进化,种群中个体失去多样性,因此,为各子种群的领飞鸟设置一个年龄变量,可记录其更新程度,年龄越大,说明更新能力越差. 年龄变量初始值为 0,如果个体更新,年龄变量清 0, 否则增加 1,并与设置的最大年龄限制,即 a_{limit} 比较,当某个体年龄大于该值时,可能在它的邻域循环

搜索,此时,启动种群多样化控制机制,产生一个新的个体替换该个体. 但随机产生的个体质量可能更差,导致算法收敛速度下降,而且被放弃的个体进化了多代,必然携带较好个体信息,其邻域可能就是更好的可行区域. 为兼顾随机性和质量,对被放弃的个体执行 3 次插入干扰,产生一个邻域个体,但该邻域个体很难保证质量. 为了找到较好的邻域个体,使算

法朝着期望的区域搜索,重复上述操作 τ 次,产生 τ 个邻域个体,将其中最好的一个个体放入种群替换原个体.考虑新个体的质量和算法效率,将 τ 设置为 10.

3 仿真研究

3.1 仿真环境和算例

算法采用 MATLAB R2016b 编写,并在 Intel (R) Core(TM) i5-9600KF/3.7 GHz/ 16.0 GB,系统为 Windows 10 的平台运行.

为了验证实验效果,选择适用于带 SDST 的流水或 HFS 问题的算例,即基于 Ta 的自适应算例^[24],通过测试该算例进行参数调整及算法的有效性验证.

3.2 参数调整

参数设置对于智能优化算法性能影响较大,为了使 IMMBO 算法在最佳状态下对 HFS-SDST 问题求解,需要对其参数进行调整.参数调整通过测试中等规模的 Ta42 衍生的 8 个自适应算例进行,Ta42 算例的工件数 n 为 50,阶段数 m 为 10,每个阶段的机器数有 2 种情况:3 台(P3)或者服从 1~3 台(P13)之间的统一分布,同时准备时间考虑 4 种情况,分别为平均加工时间的 10%(SSD10)、50%(SSD50)、100%(SSD100)、125%(SSD125).根据上述每个阶段机器数和准备时间情况产生的 8 个自适应算例记作:SSD10_P13_50、SSD50_P13_50、SSD100_P13_50、SSD125_P13_50、SSD10_P3_50、SSD50_P3_50、SSD100_P3_50 和 SSD125_P3_50. IMMBO 算法中涉及参数较多,通过前期测试,可知对算法优化效果影响较大的 5 个参数及其大致范围.为了确定参数的值,采用正交实验法对算法中的 5 个参数的 4 个水平进行测试.5 个参数分别为子种群个数 N_p 、巡回次数 G 、最大年龄 a_{limit} 、破坏重建中的破坏程度 d 和最大进化代数 t_{max} .正交表如表 1 所示.

表 1 正交实验参数/水平表

Tab. 1 Orthogonal experiment parameters/levels

水平	N_p	G	a_{limit}	d	t_{max}
1	11	2	20	3	300
2	13	3	30	4	400
3	15	4	40	5	500
4	17	5	50	6	600

如果对 IMMBO 算法的 5 个参数的 4 个水平全部组合进行测试,需要进行 $4^5=1\,024$ 组实验,但采

用正交实验法只需测试 $4^2=16$ 组,即可为算法选出合适的参数值,显然,用正交实验法可以使参数选择效率大幅度提高.16 种参数组合及测试上述 8 个算例的结果如表 2 所示.

表 2 $L_{16}(4^5)$ 正交表和实验结果

Tab. 2 Orthogonal parameter $L_{16}(4^5)$ and results

序号	N_p	G	a_{limit}	d	t_{max}	平均值
1	11(1)	2(1)	20(1)	3(1)	300(1)	3 929.29
2	11(1)	3(2)	30(2)	4(2)	400(2)	3 909.13
3	11(1)	4(3)	40(3)	5(3)	500(3)	3 926.50
4	11(1)	5(4)	50(4)	6(4)	600(4)	3 911.50
5	13(2)	2(1)	30(2)	5(3)	600(4)	3 915.21
6	13(2)	3(2)	20(1)	6(4)	500(3)	3 901.25
7	13(2)	4(3)	50(4)	3(1)	400(2)	3 938.75
8	13(2)	5(4)	40(3)	4(2)	300(1)	3 925.88
9	15(3)	2(1)	40(3)	6(4)	400(2)	3 928.04
10	15(3)	3(2)	50(4)	5(3)	300(1)	3 912.08
11	15(3)	4(3)	20(1)	4(2)	600(4)	3 892.96
12	15(3)	5(4)	30(2)	3(1)	500(3)	3 915.92
13	17(4)	2(1)	50(4)	4(2)	500(3)	3 928.29
14	17(4)	3(2)	40(3)	3(1)	600(4)	3 920.92
15	17(4)	4(3)	30(2)	6(4)	300(1)	3 914.88
16	17(4)	5(4)	20(1)	5(3)	400(2)	3 891.58

注:()内数值表示水平编号.

表 2 最后一列测试结果为每种参数组合测试 8 个算例,对每个算例算法独立测试 5 次,共 40 次实验测得的 C_{max} 平均值.表 3 为 5 个参数,4 种水平的平均值以及标准偏差(SD),据此选择参数的水平.4 行中每一列的最小值所对应的水平值,表示较好的参数水平值.SD 为该参数各个水平的标准偏差,SD 值越大,对算法的优化效果影响越大,表中最后一行为 5 个参数按照 SD 值降序排列.5 个参数在不同水平的平均 C_{max} (AVG)的变化曲线如图 5 所示,其中 AVG 最小的水平即为该参数的最佳设置值.

表 3 参数等级及均值响应

Tab. 3 Parameter rank and mean response values

水平	N_p	G	a_{limit}	d	t_{max}
1	3 919.1	3 925.21	3 903.77	3 926.22	3 920.53
2	3 920.27	3 910.84	3 913.78	3 914.06	3 916.88
3	3 912.25	3 918.27	3 925.33	3 911.34	3 917.99
4	3 913.92	3 911.22	3 922.66	3 913.92	3 910.15
SD	3.38	5.89	8.45	5.78	3.84
排序	5	2	1	3	4

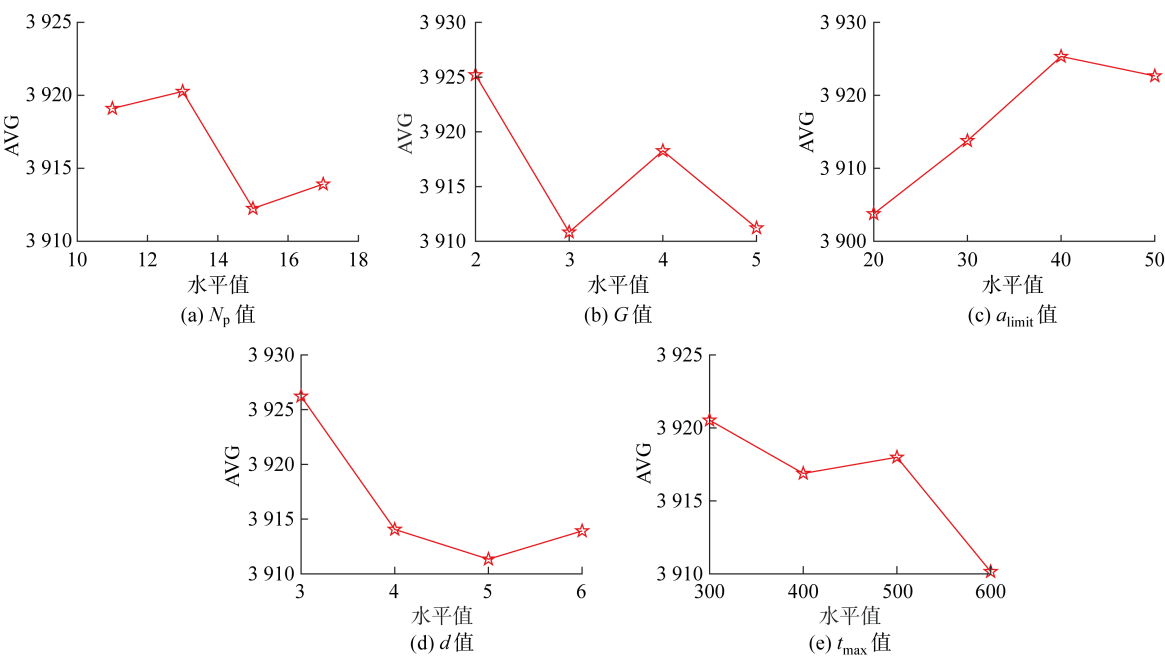


图 5 IMMBO 算法中的 5 个参数在 4 个水平变化趋势图
Fig.5 Trend of 5 parameters in IMMBO algorithm at 4 levels

由图 5 和表 3 可以看出, a_{limit} 对算法的优化效果影响最大, 子种群个数 N_p 的影响最小. 因此, 设置参数 $N_p=15, d=5, a_{\text{limit}}=20, G$ 和 t_{max} 分别设为 3 和 600.

3.3 算法各部分有效性测试

通过测试 IMMBO 的 4 个变体分别说明算法各部分的有效性. 选择 32 个自适应算例进行测试, 即 Ta 算例(Ta32, Ta34, ..., Ta46), 自适应考虑 4 种准备时间且每个阶段 3 台机器情况. IMMBO 的 4 个变体分别为: 去掉领飞鸟邻域搜索策略中的 DC (IMMBO_NDC)、去掉子种群交互机制的 DWOA 算法(IMMBO_NDWOA)、去掉 LS 的算法 (IMMBO_NLS) 和去掉种群多样化控制机制 (IMMBO_NR). 表 4 中所列出的值表示选择不同准备时间时, 每个变体独立运行 5 次测试 8 个算例($8 \times 5 = 40$)的 C_{max} 平均值, 表达式为

表 4 IMMBO 算法 4 个变体测试 P3 情况均值响应表
Tab.4 Mean response of four variants of IMMBO algorithms in P3

算法	SSD10_	SSD50_	SSD100_	SSD125_
	P3_50	P3_50	P3_50	P3_50
IMMBO_NDC	1 511.98	1 994.43	2 584.83	2 871.30
IMMBO_NDWOA	1 502.53	1 973.93	2 553.50	2 848.08
IMMBO_NLS	1 495.90	1 965.15	2 536.13	2 824.58
IMMBO_NR	1 511.20	1 992.33	2 581.05	2 875.90

$$C_{\text{AVG}} = \frac{1}{8} \sum_{h=1}^8 C_{\text{AVG}}^h$$

(7)

式中: C_{AVG}^h 为该算法测试第 h 个算例 5 次得到的 C_{max} 平均值.

表 4 的同一组算例中, C_{AVG} 越大, 说明去掉的对应算子对优化结果影响越大. 可知, 邻域搜索中的 DC 和种群多样化控制机制对算法优化效果影响最大, 子种群交互机制对算法的影响次之, LS 对算法的影响最小.

3.4 算法性能测试

为了验证 IMMBO 算法求解 HFS-SDST 问题的效果, 将 IMMBO 和文献[10]中求解同一问题的 ILS_{MRLS}、IMBO 和 DABC 算法通过测试 Ta32 ~ Ta60 的自适应算例进行比较, 准备时间和机器数同 3.2 节算例设计, 总算例数目为 $15 \times 4 \times 2 = 120$, 每个算法独立运行 5 次测试所有算例, 结果如表 5 所示. 表 5 中所列出的值表示每个算法独立运行 5 次测试 15 个算例 ($15 \times 5 = 75$) 的 C_{max} 平均值, 表达式为

$$C_{\text{AVG}} = \frac{1}{15} \sum_{h=1}^{15} C_{\text{AVG}}^h$$

(8)

表 5 中 4 个算法相比, 可知, IMMBO 算法对各种机器配置和准备时间算例都有较好的优化效果. 表 6 为上述 4 个算法分别测试 8 个自适应 Ta 算例的运行时间. 可以看出, 测试相同算例时, 4 个算法的运行时间由小到大依次是 DABC、IMBO、ILS_{MRLS}、IMMBO, 虽然 IMMBO 所用时间稍长于

表 5 IMMBO/IMBO/DABC/ILS_{MRLS} 算法针对 P13 和 P3 算例的测试结果对比
Tab. 5 Comparson of IMMBO/IMBO/DABC/ILS_{MRLS} algorithms in cases P13 and P3

算法	P13				P3			
	SSD10_	SSD50_	SSD100_	SSD125_	SSD10_	SSD50_	SSD100_	SSD125_
	P13_50	P13_50	P13_50	P13_50	P3_50	P3_50	P3_50	P3_50
ILS _{MRLS}	2 860.13	3 887.24	5 058.04	5 605.45	1 163.29	1 616.35	2 125.87	2 389.60
IMBO	2 891.37	3 882.24	5 094.32	5 646.12	1 178.13	1 611.34	2 124.73	2 390.25
DABC	2 857.31	3 981.69	5 082.48	5 627.08	1 173.28	1 607.95	2 113.76	2 354.56
IMMBO	2 834.60	3 856.52	5 000.99	5 526.81	1 110.60	1 498.79	1 955.79	2 170.61

表 6 4 个算法测试 8 个自适应算例的运行时间对比
Tab. 6 Comparison of running time of 4 algorithms when testing 8 adaptive benchmarks

算例	算法	P3			
		SSD10_P3_50	SSD50_P3_50	SSD100_P3_50	SSD125_P3_50
Ta32	ILS _{MRLS}	2 033.51	1 995.49	2 015.03	1 920.50
	IMBO	1 093.25	1 099.30	1 097.90	1 099.67
	DABC	108.93	108.25	108.34	108.14
	IMMBO	3 173.05	3 134.05	3 148.59	3 132.99
Ta34	ILS _{MRLS}	2 102.98	1 911.52	2 009.56	2 056.72
	IMBO	1 086.26	1 097.10	1 095.33	1 089.54
	DABC	109.96	107.98	108.21	107.21
	IMMBO	3 160.43	3 143.98	3 147.07	3 129.43

IMBO 和 ILS_{MRLS}, 但可以得到更好的优化效果. 图 6 和图 7 分别为 4 个算法测试算例 Ta56 的 SSD125_P3_50 和 Ta60 的 SSD50_P13_50 的收敛曲线. 由图可知, IMMBO 算法与其他 3 个算法相比, 无论是进化前期的收敛速度还是后期的收敛精度都是较优的.

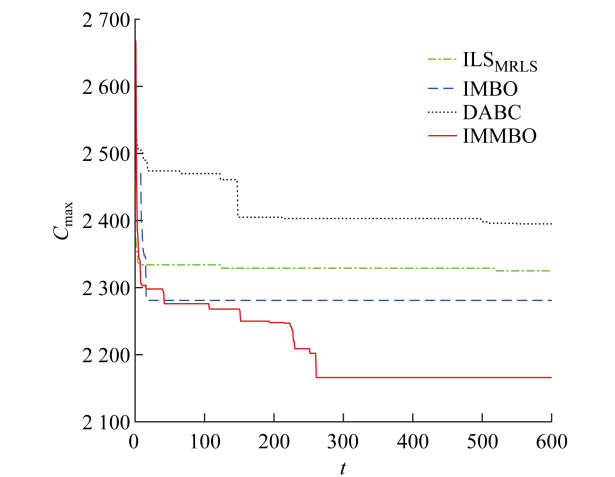


图 6 算例 Ta56 中 SSD125_P3_50 的 4 个算法收敛曲线图
Fig. 6 Convergence of 4 algorithms for case SSD125_P3_50 of Ta56

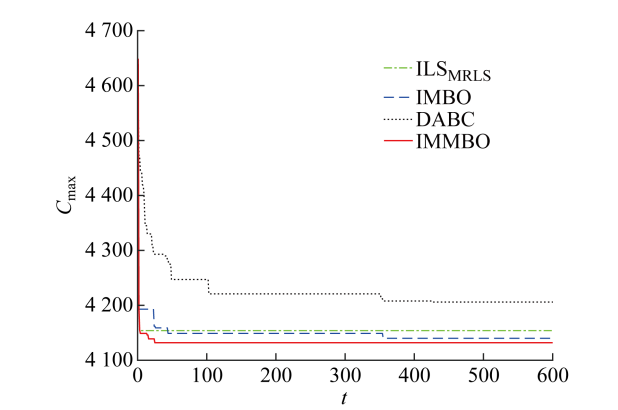


图 7 算例 Ta60 中 SSD50_P13_50 的 4 个算法收敛曲线图
Fig. 7 Convergence of 4 algorithms for case SSD50_P13_50 of Ta60

4 结语

针对有较强工业背景的 HFS-SDST 问题, 提出 IMMBO 算法, 优化调度目标是找到使最大完成时间最短的工件在第 1 阶段机器上的加工顺序, 即使 C_{max} 最小的调度解. 首先, 采用 MNEH 初始化种群, 并将其随机分配到各子种群, 子种群规模为 3, 由 1

个领飞鸟和2个跟飞鸟组成.然后采用多种群候鸟迁徙算法的各子种群独立并行搜索和DWOA算法实现子种群信息交互,同时加入LS和种群多样化控制策略平衡算法的探索 and 开发能力.为了使算法在最佳状态下求解HFS-SDST问题,针对IMMO算法进行参数调整和算法各部分的性能测试,为了验证算法的有效性,与ILS_{MRLS}、IMBO和DABC算法进行比较.结果表明,IMMO的优化效果优于其他算法,且在进化前期所提算法有较快的收敛速度,后期能够跳出局部最优.

参考文献:

- [1] ZANDIEH M, FATEMI GHOMI S M T, MOAT-TAR HUSSEINI S M. An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times[J]. **Applied Mathematics and Computation**, 2006, 180(1): 111-127.
- [2] 李颖俐, 李新宇, 高亮. 混合流水车间调度问题研究综述[J]. **中国机械工程**, 2020, 31(23): 2798-2813.
LI Yingli, LI Xinyu, GAO Liang. Review on hybrid flow shop scheduling problems[J]. **China Mechanical Engineering**, 2020, 31(23): 2798-2813.
- [3] SIOUD A, GAGNÉ C. Enhanced migrating birds optimization algorithm for the permutation flow shop problem with sequence dependent setup times[J]. **European Journal of Operational Research**, 2018, 264(1): 66-73.
- [4] 李俊青, 李文涵, 陶昕瑞, 等. 时间约束混合流水车间调度问题综述[J]. **控制理论与应用**, 2020, 37(11): 2273-2290.
LI Junqing, LI Wenhan, TAO Xinrui, *et al.* A survey on time constrained hybrid flow shop scheduling problems[J]. **Control Theory & Applications**, 2020, 37(11): 2273-2290.
- [5] NADERI B, ZANDIEH M, ROSHANAEI V. Scheduling hybrid flowshops with sequence dependent setup times to minimize makespan and maximum tardiness[J]. **The International Journal of Advanced Manufacturing Technology**, 2009, 41(11/12): 1186-1198.
- [6] MOCCELLIN J V, NAGANO M S, PITOMBEIRA NETO A R, *et al.* Heuristic algorithms for scheduling hybrid flow shops with machine blocking and setup times[J]. **Journal of the Brazilian Society of Mechanical Sciences and Engineering**, 2018, 40(2): 1-11.
- [7] BURCIN OZSOYDAN F, SAĞIR M. Iterated greedy algorithms enhanced by hyper-heuristic based learning for hybrid flexible flowshop scheduling problem with sequence dependent setup times: A case study at a manufacturing plant[J]. **Computers & Operations Research**, 2021, 125: 105044.
- [8] 周炳海, 刘文龙. 考虑能耗和准时的混合流水线多目标调度[J]. **上海交通大学学报**, 2019, 53(7): 773-779.
ZHOU Binghai, LIU Wenlong. Multi-objective hybrid flow-shop scheduling problem considering energy consumption and on-time delivery [J]. **Journal of Shanghai Jiao Tong University**, 2019, 53(7): 773-779.
- [9] GÓMEZ-GASQUET P, ANDRÉS C, LARIO F C. An agent-based genetic algorithm for hybrid flowshops with sequence dependent setup times to minimise makespan[J]. **Expert Systems With Applications**, 2012, 39(9): 8095-8107.
- [10] PAN Q K, GAO L, LI X Y, *et al.* Effective metaheuristics for scheduling a hybrid flowshop with sequence-dependent setup times[J]. **Applied Mathematics and Computation**, 2017, 303: 89-112.
- [11] TIAN H X, LI K, LIU W. A Pareto-based adaptive variable neighborhood search for biobjective hybrid flow shop scheduling problem with sequence-dependent setup time[J]. **Mathematical Problems in Engineering**, 2016, 2016: 1257060.
- [12] KHARE A, AGRAWAL S. Scheduling hybrid flowshop with sequence-dependent setup times and due windows to minimize total weighted earliness and tardiness [J]. **Computers & Industrial Engineering**, 2019, 135: 780-792.
- [13] DUMAN E, UVSAL M, ALKAVA A F. Migrating birds optimization; a new metaheuristic approach and its performance on quadratic assignment problem [J]. **Information Sciences**, 2012, 217(24): 65-77.
- [14] PAN Q K, DONG Y. An improved migrating birds optimisation for a hybrid flowshop scheduling with total flowtime minimisation[J]. **Information Sciences**, 2014, 277: 643-655.
- [15] HAN D Y, TANG Q H, ZHANG Z K, *et al.* An improved migrating birds optimization algorithm for a hybrid flow shop scheduling within steel plants [J]. **Mathematics**, 2020, 8(10): 1661.
- [16] MENG T, PAN Q K, LI J Q, *et al.* An improved migrating birds optimization for an integrated lot-streaming flow shop scheduling problem[J]. **Swarm and Evolutionary Computation**, 2018, 38: 64-78.
- [17] 汤洪涛, 王丹南, 邵益平, 等. 基于改进候鸟迁徙优化的多目标批量混合流水车间调度[J]. **上海交通**

大学学报, 2022, 56(2): 201-213.

TANG Hongtao, WANG Dannan, SHAO Yiping, *et al.* A modified migrating birds optimization for multi-objective lot streaming hybrid flowshop scheduling [J]. **Journal of Shanghai Jiao Tong University**, 2022, 56(2): 201-213.

[18] BÁEZ S, ANGEL-BELLO F, ALVAREZ A, *et al.* A hybrid metaheuristic algorithm for a parallel machine scheduling problem with dependent setup times [J]. **Computers & Industrial Engineering**, 2019, 131: 295-305.

[19] HAN Y X, GU X S. Improved multipopulation discrete differential evolution algorithm for the scheduling of multipurpose batch plants [J]. **Industrial & Engineering Chemistry Research**, 2021, 60 (15): 5530-5547.

[20] GAO L, PAN Q K. A shuffled multi-swarm micro-migrating birds optimizer for a multi-resource-constrained flexible job shop scheduling problem [J]. **Information Sciences**, 2016, 372: 655-676.

[21] TONGUR V, ÜLKER E. PSO-based improved multi-flocks migrating birds optimization (IMFMBO) algorithm for solution of discrete problems [J]. **Soft Computing**, 2019, 23(14): 5469-5484.

[22] RUIZ R, STÜTZLE T. A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem [J]. **European Journal of Operational Research**, 2007, 177(3): 2033-2049.

[23] MIRJALILI S, LEWIS A. The whale optimization algorithm [J]. **Advances in Engineering Software**, 2016, 95: 51-67.

[24] VALLADA E, RUIZ R, MAROTO C. Synthetic and real benchmarks for complex flow-shops problems [R]. Valencia, Spain: Universitat Politècnica de València, 2003.

(本文编辑:孙启艳)