



中国科技期刊卓越行动计划项目入选期刊

控制与决策

CONTROL AND DECISION

考虑批处理机的绿色模糊混合流水车间调度问题

王静, 雷德明

引用本文:

王静,雷德明. 考虑批处理机的绿色模糊混合流水车间调度问题[J]. 控制与决策, 2024, 39(10): 3413–3421.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2023.0914>

您可能感兴趣的其他文章

Articles you may be interested in

[基于多班教学优化的多目标分布式混合流水车间调度](#)

Multi-class teaching-learning-based optimization for multi-objective distributed hybrid flow shop scheduling

控制与决策. 2021, 36(2): 303–313 <https://doi.org/10.13195/j.kzyjc.2020.0549>

[基于改进蛙跳算法的分布式两阶段混合流水车间调度](#)

An improved shuffled frog leaping algorithm for the distributed two-stage hybrid flow shop scheduling

控制与决策. 2021, 36(1): 241–248 <https://doi.org/10.13195/j.kzyjc.2019.0472>

[超启发式交叉熵算法求解模糊分布式流水线绿色调度问题](#)

Hyper-heuristic cross-entropy algorithm for green distributed permutation flow-shop scheduling problem with fuzzy processing time

控制与决策. 2021, 36(6): 1387–1396 <https://doi.org/10.13195/j.kzyjc.2019.1681>

[区间数可重入混合流水车间调度与预维护协同优化](#)

Collaborative optimization of interval number reentrant hybrid flow shop scheduling and preventive maintenance

控制与决策. 2021, 36(11): 2599–2608 <https://doi.org/10.13195/j.kzyjc.2020.0973>

[基于双种群模糊引力搜索算法的舰载机甲板作业调度](#)

Flight deck operations scheduling based on dual population fuzzy gravitational search algorithm

控制与决策. 2021, 36(11): 2751–2759 <https://doi.org/10.13195/j.kzyjc.2020.0523>

考虑批处理机的绿色模糊混合流水车间调度问题

王 静, 雷德明[†]

(武汉理工大学 自动化学院, 武汉 430070)

摘要: 针对考虑批处理机(batch processing machines, BPM)和部分阶段无优先关系的绿色模糊混合流水车间调度问题(energy-efficient fuzzy hybrid flow shop scheduling problem, EFHFSP),提出一种动态人工蜂群(dynamical artificial bee colony, DABC)算法以同时最小化最大模糊完成时间和模糊总能耗.给出基于种群评估的种群裁定方法以在每一代动态决定雇佣蜂种群和跟随蜂种群,并应用了动态雇佣蜂阶段、基于自适应交流的跟随蜂阶段、多样性强化策略以及自适应侦察蜂阶段.最后进行仿真实验,实验结果表明DABC在求解考虑BPM和部分阶段无优先关系的EFHFSP方面具有较强的优势.

关键词: 绿色混合流水车间; 模糊调度; 人工蜂群算法; 批处理机; 优先关系

中图分类号: TP18

文献标志码: A

DOI: 10.13195/j.kzyjc.2023.0914

引用格式: 王静,雷德明.考虑批处理机的绿色模糊混合流水车间调度问题[J].控制与决策,2024,39(10):3413-3421.

Research on energy-efficient fuzzy hybrid flow shop scheduling with batch processing machines

WANG Jing, LEI De-ming[†]

(College of Automation, Wuhan University of Technology, Wuhan 430070, China)

Abstract: A energy-efficient fuzzy hybrid flow shop scheduling problem(EFHFSP) with batch processing machines (BPM) and no precedence between some stages is studied, and a dynamical artificial bee colony (DABC) is proposed to minimize the fuzzy makespan and total fuzzy energy consumption simultaneously. To generate high quality solutions, a swarm decision method is established by population evaluation and used to dynamically determine employed bee swarm and onlooker bee swarm on each generation. The dynamical employed bee phase, the onlooker bee phase with adaptive communication, the diversity reinforcement strategy and the adaptive scout phase are also applied. Extensive experiments are conducted. The computational results show that the DABC has promising advantages on the considered EFHFSP.

Keywords: energy-efficient hybrid flow shop; fuzzy scheduling; artificial bee colony; batch processing machines; precedence

0 引 言

绿色制造是一种降低生产对环境影响的制造模式,制造商必须寻求高效的方法来平衡经济和环境效应^[1].通常,采用节能设备的升级或更换以及绿色运作方法和优化策略这两种有效的措施来实现绿色制造^[2].然而,升级或更换节能设备的成本过高,大多数制造商往往倾向于采用第2种措施来提高制造系统的可持续生产水平.近些年,绿色生产调度作为第2种措施的核心已得到了广泛的应用^[3-13].

绿色混合流水车间调度问题(energy-efficient hybrid flow shop scheduling problem, EHFSP)是一种

典型的绿色生产调度问题,其研究已取得了较大进展.针对目标为最小化最大完成时间和总能耗的EHFSP,研究者已成功应用多级优化方法^[5]和教学优化算法^[6]等算法进行求解.其他多目标如总延误时间和总能耗的EHFSP也受到了关注^[7-8].

EHFSP的已有研究考虑了工人柔性^[9]和阻塞^[10]等实际工况.近年来,考虑批处理机(BPM)的EHFSP也开始受到关注,但大多考虑含一个BPM阶段的EHFSP.针对最后阶段为BPM的问题,Lu等^[11]提出了一种高效的自适应遗传算法;Wu等^[12-13]分别针对考虑连续型BPM和周期型BPM的可重入EHFSP,设

收稿日期: 2023-06-29; 录用日期: 2023-10-09.

基金项目: 国家自然科学基金项目(61573264).

责任编辑: 唐加福.

[†]通讯作者. E-mail: deminglei11@163.com.

计了改进的基于分解的多目标进化算法。

通常,在具有 H 个阶段的混合流水车间中,每个工件按照阶段 $1, 2, \dots, H$ 的顺序进行加工,即任意两个相邻阶段之间存在优先关系约束. 然而,在一些实际的制造车间如热冷铸造车间,制芯和模具组装无优先关系,也就是说,工件在成型阶段结束后,可先加工模具组装阶段,再加工制芯阶段,反之亦然. 考虑部分阶段无优先关系的EHFSP研究进展很少.

另一方面,不确定性始终存在于实际制造过程中,是不可避免的特性. 不确定性的处理方法一般有随机优化、鲁棒优化和模糊理论等^[14]. 随机优化采样基数大、时间成本高且计算量大,难以在实际中广泛运用;鲁棒优化是半无限优化问题,求解困难且存在决策过于保守的问题,难以兼顾风险与成本,与实际不符;模糊理论在缺少变量或信息不全时,能较为准确地描述不确定性,更接近于真实情况. 因此,模糊理论相较于随机优化等,具有较大优势.

在过去的10年中,模糊理论经常被用来描述不确定性,模糊调度研究已取得了较大进展^[15-17]. 为求解绿色模糊混合流水车间调度问题(EFHSP),Wang等^[18]提出了一种包含遗传算子、逆向学习和基于历史信息的扩展NSGA-II.

如上所述,EHFSP已有研究考虑了阻塞和BPM等实际工况,而考虑BPM的EHFSP研究进展不大;此外,在考虑BPM的EHFSP中往往忽略不确定性,也未将一些实际约束(如部分阶段无优先关系)考虑进来. BPM、部分阶段无优先关系和不确定性往往同时存在于实际制造过程中,通过对这3个因素的建模和优化,并采用模糊理论来描述不确定性,能获得具有更高应用价值的优化结果. 因此,研究考虑BPM和部分阶段无优先关系的EFHSP具有现实意义.

人工蜂群算法(artificial bee colony, ABC)是一种受蜜蜂群体觅食行为启发而提出的算法,具有结构简单、易于实现等特点. 作为生产调度的主要方法,ABC已成功应用于EHFSP^[8]、并行BPM调度^[15-16]和模糊调度问题^[17]中,显示出较强的搜索能力和优势. 由于考虑BPM和部分阶段无优先关系的EFHSP可看成EHFSP、并行BPM调度和模糊调度的集成,ABC在这3个方面的成功应用表明它在求解所研究的EFHSP方面具有潜在的优势.

本文针对考虑BPM和部分阶段无优先关系的EFHSP,提出一种动态人工蜂群(DABC)算法,以同时最小化最大模糊完成时间和模糊总能耗. 在DABC中,基于种群评估的种群裁定方法动态确定每

一代的雇佣蜂种群和跟随蜂种群,同时包含动态雇佣蜂阶段、基于自适应交流的跟随蜂阶段、多样性强化策略以及自适应侦察蜂阶段. 最后通过实验验证了DABC在求解所研究的EFHSP方面的优势.

1 问题描述

考虑BPM和部分阶段无优先关系的EFHSP描述如下:存在 n 个工件和一个 H 个阶段的混合流水车间. 阶段 $s \neq \sigma$ 有 m_s 台并行机 $M_{s,1}, M_{s,2}, \dots, M_{s,m_s}$, 每台机器上进行单件加工;阶段 σ 由并行BPM $M_{\sigma,1}, M_{\sigma,2}, \dots, M_{\sigma,m_\sigma}$ 构成,每台机器上进行批加工.

表1给出了相关符号及其描述. 其中: A_v 中所有工件都有相同的重量、材料和模糊加工时间,即对于任意工件 $J_i \in A_v$,有 $w_i = \bar{w}_v, r_i = \bar{r}_v, \tilde{p}_{isk} = \tilde{p}_{vsk}$; 一个批的模糊加工时间等于该批中所有工件的最大模糊加工时间; $\tilde{p}_{isk}, \tilde{p}_{vsk}$ 和 \tilde{t}_s 都是三角模糊数^[19],单位是min.

表1 符号和描述

符号	描述
G	工件的总类型
A_v	第 v 类工件的集合
\bar{w}_v	第 v 类工件的重量
\bar{r}_v	第 v 类工件的材料
\tilde{p}_{vsk}	第 v 类工件的模糊加工时间
w_i	工件 J_i 的重量
r_i	工件 J_i 的材料
Q_k	机器 $M_{\sigma,k}$ 的容量
\tilde{p}_{isk}	工件 J_i 的模糊加工时间
\tilde{t}_s	工件由AGV运输到阶段 $s > 1$ 的模糊运输时间
q_{sk}	机器 $M_{s,k}$ 每小时单位能耗 (kw/h)
E_{AGV}	AGV 每小时单位能耗 (kw/h)
η	总批数
B_b	批 b
X_{isk}	如果工件 J_i 由机器 $M_{s,k}$ 加工,则为1,否则为0
λ_{ji}	阶段 $s \neq \sigma$ 时,在机器 $M_{s,k}$ 上加工完工件 J_j 后立即加工工件 J_i ,则为1,否则为0
μ_i	工件在阶段 $\sigma - 2, \sigma - 1$ 的顺序是 $\sigma - 2, \sigma - 1$, 则为1;否则,顺序是 $\sigma - 1, \sigma - 2$, 则为0
Y_{bk}	如果 B_b 分配到机器 $M_{\sigma,k}$,则为1,否则为0
τ_{ib}	如果工件 J_i 被分配到 B_b 中,则为1,否则为0
\tilde{p}_{bk}	B_b 在 $M_{\sigma,k}$ 上的模糊加工时间
\tilde{C}_i	工件 J_i 的模糊完成时间
\tilde{ST}_{isk}	工件 J_i 在机器 $M_{s,k}$ 上的模糊开始时间, $s \neq \sigma$
\tilde{ET}_{isk}	工件 J_i 在机器 $M_{s,k}$ 上的模糊完成时间, $s \neq \sigma$
\tilde{SBT}_{bk}	B_b 在机器 $M_{\sigma,k}$ 上的模糊开始时间
\tilde{EBT}_{bk}	B_b 在机器 $M_{\sigma,k}$ 上的模糊完成时间
ζ_{db}	在机器 $M_{\sigma,k}$ 上加工完 B_d 后立即加工 B_b ,则为1,否则为0
V	足够大的正数
C_{max}	最大模糊完成时间
ECM	模糊加工总能耗
ECT	模糊运输总能耗
TEC	模糊总能耗

阶段 $\sigma - 2$ 与 $\sigma - 1$ 之间无阶段优先关系约束, 即任意工件 J_i 在阶段 $\sigma - 3$ 完成之后, 可先加工阶段 $\sigma - 1$, 再加工 $\sigma - 2$, 反之亦然, 其中 $2 < \sigma < S$. 存在其他约束, 包括所有工件和机器在 0 时可用, 不得中断或插入等.

该问题包含机器分配、调度、组批和阶段确定 4 个子问题. 问题的目标是同时最小化最大模糊完成时间和模糊总能耗.

$$C_{\max} = \max_{1 \leq i \leq n} \{\tilde{C}_i\}, \quad (1)$$

$$\text{TEC} = \text{ECM} + \text{ECT}. \quad (2)$$

其中

$$\begin{aligned} \text{ECM} = & \sum_{i=1}^n \sum_{s=1 \& s \neq \sigma}^H \sum_{k=1}^{m_s} \tilde{p}_{isk} \times q_{sk} \times X_{isk} + \\ & \sum_{b=1}^{\eta} \sum_{k=1}^{m_{\sigma}} \tilde{b}p_{bk} \times q_{\sigma k} \times Y_{bk}, \end{aligned} \quad (3)$$

$$\text{ECT} = \sum_{i=1}^n \sum_{s=2}^H \sum_{k=1}^{m_s} \tilde{t}p_s \times X_{isk} \times E_{AGV}. \quad (4)$$

机器分配子问题为每个工件或批选择合适的机器, 约束如下:

$$\sum_{k=1}^{m_s} X_{isk} = 1, \forall i, s; \quad (5)$$

$$\sum_{k=1}^{m_{\sigma}} Y_{bk} = 1, \forall b. \quad (6)$$

其中: 式(5)确保每个工件在每个阶段只能被安排到一个机器上加工, 式(6)确保每个批只能被安排到一个机器上加工.

调度子问题确定每台机器上各个工件或批的加工顺序, 约束如下:

$$\tilde{E}T_{isk} \geq \tilde{S}T_{isk} + \tilde{p}_{isk} + V(X_{isk} - 1), \forall i, k, s \neq \sigma; \quad (7)$$

$$\tilde{S}T_{isk} \geq \tilde{E}T_{j sk} + V(\lambda_{ji} - 1), \forall i, j, k, s \neq \sigma; \quad (8)$$

$$\tilde{E}B T_{bk} \geq \tilde{S}B T_{bk} + \tilde{b}p_{bk} + V(Y_{bk} - 1), \forall b, k; \quad (9)$$

$$\tilde{S}B T_{bk} \geq \tilde{E}B T_{dk} + V(\zeta_{db} - 1), \forall d, b, k. \quad (10)$$

其中: 式(7)表示 $s \neq \sigma$ 时, 工件 J_i 在机器 $M_{s,k}$ 上的模糊完成时间大于等于模糊开始时间与模糊加工时间之和; 式(8)表示 $s \neq \sigma$ 时, 机器 $M_{s,k}$ 上相邻两工件之间的时间关系; 式(9)表示 B_b 在机器 $M_{\sigma,k}$ 上的模糊完成时间大于等于模糊开始时间与模糊加工时间之和; 式(10)表示机器 $M_{\sigma,k}$ 上相邻两批之间的时间关系.

组批子问题则将每个工件分到合适的批中, 约束如下:

$$\sum_{b=1}^{\eta} \tau_{ib} = 1, \forall i; \quad (11)$$

$$V(2 - \tau_{ib} - \tau_{jb}) \geq r_i - r_e, \forall i, j, b, i \neq j; \quad (12)$$

$$V(2 - \tau_{ib} - \tau_{jb}) \leq r_i - r_e, \forall i, j, b, i \neq j; \quad (13)$$

$$\sum_{k=1}^{m_{\sigma}} Y_{bk} \times Q_k \geq \sum_{i=1}^n \tau_{ib} \times w_i, \forall b. \quad (14)$$

其中: 式(11)确保每个工件在阶段 σ 只能被安排到一个批内, 式(12)和(13)确保只有材料相同的工件才能被分配到同一个批中, 式(14)确保机器 $M_{\sigma,k}$ 上加工的单批总重量不能超过其容量 Q_k .

阶段确定子问题确定访问每个工件在阶段 $\sigma - 2$ 与 $\sigma - 1$ 的顺序, 约束如下:

$$\begin{aligned} \tilde{S}T_{i(\sigma-1)g} \geq & \tilde{E}T_{i(\sigma-2)k} + \tilde{t}p_{\sigma-1} + V(\mu_i + \\ & X_{i(\sigma-2)k} + X_{i(\sigma-1)g} - 3), \forall i, k, g. \end{aligned} \quad (15)$$

其中式(15)确保工件 J_i 在阶段 $\sigma - 2$ 与 $\sigma - 1$ 间的模糊完成时间与模糊开始时间的关系.

2 应用 DABC 求解带 BPM 的 EFHFSP

为求解考虑 BPM 和部分阶段无优先关系的 EFHFSP, 设计 DABC 算法. 该算法由种群裁定方法、动态雇佣蜂阶段、基于自适应交流的跟随蜂阶段、自适应侦察蜂阶段等组成.

2.1 编解码与初始化

采用 3 个串的编码方式描述问题的解, 分别为调度串 $[\pi_1, \pi_2, \dots, \pi_n]$ 、机器分配串 $[\theta_1, \theta_2, \dots, \theta_n]$ 和阶段确定串 $[\varphi_1, \varphi_2, \dots, \varphi_n]$. 其中: $\pi_i \in \{1, 2, \dots, n\}$. θ_i 用于工件 J_i 选择机器 $M_{s,k}$, $s \neq \sigma$, $\theta_i \in \{1, 2, 3\}$. 对于工件 J_i , 如果 $\theta_i = 1$, 则选择最小模糊可用时间的机器; 如果 $\theta_i = 2$, 则选择最小模糊完成时间的机器; 如果 $\theta_i = 3$, 则选择最小模糊能耗的机器. $\varphi_i \in \{\sigma - 2, \sigma - 1\}$, 如果 $\varphi_i = \sigma - 1$, 则工件 J_i 的访问顺序是 $\sigma - 1, \sigma - 2$; 否则为 $\sigma - 2, \sigma - 1$.

从阶段 1 开始, 依次对各阶段解码, 步骤如下.

step 1: 关于阶段 $1 \leq s \leq \sigma - 3$: 如果 $s = 1$, 则从调度串的第 1 个工件开始, 依次针对每个工件 J_{π_i} , 根据 θ_{π_i} 选择机器 $M_{1,k}$, 将 J_{π_i} 安排到 $M_{1,k}$ 上; 否则, 对 n 个工件根据它们在阶段 $s - 1$ 的模糊完成时间进行升序排序, 得到阶段 s 的排列 $\pi_1^s, \pi_2^s, \dots, \pi_n^s$, 再依次针对每个工件 $J_{\pi_i^s}$, 根据 $\theta_{\pi_i^s}$ 选择机器 $M_{s,k}$, 将 $J_{\pi_i^s}$ 安排到 $M_{s,k}$ 上.

step 2: 关于阶段 $s = \sigma - 2, \sigma - 1$: 先对 n 个工件根据它们在阶段 $\sigma - 3$ 的模糊完成时间进行升序排序, 得到阶段 $\sigma - 2$ 和 $\sigma - 1$ 的排列 $\pi_1^s, \pi_2^s, \dots, \pi_n^s$. 依次针

对每个工件 $J_{\pi_i^s}$, 先确定访问顺序 $\varphi_{\pi_i^s}$, 再依次将工件 $J_{\pi_i^s}$ 安排到由 $\theta_{\pi_i^s}$ 选择的机器 $M_{\varphi_{\pi_i^s}, k}$ 和 $M_{2\sigma-3-\varphi_{\pi_i^s}, g}$ 上.

step 3: 关于阶段 σ : 对 n 个工件根据它们在 $2\sigma - 3 - \varphi_{\pi_i^s}$ 阶段的模糊完成时间进行升序排序, 得到阶段 σ 的排列 $\pi_1^\sigma, \pi_2^\sigma, \dots, \pi_n^\sigma$. 组批过程与文献[20]类似.

step 4: 关于阶段 $\sigma + 1 \leq s \leq H$: 步骤同 step 1 中的后一种情况.

组批步骤如下: 根据 step 3 中得到的排列, 重复下列操作直至所有工件都完成组批. 令 $b = 0$, 依次为每个工件 $J_{\pi_i^\sigma}$, 找到当前所有与工件 $J_{\pi_i^\sigma}$ 材料相同且加入 $J_{\pi_i^\sigma}$ 不超出其容量的批, 存在以下 3 种情况: 不存在这样的批, 则建立新的批, $b = b + 1, B_b = B_b \cup J_{\pi_i^\sigma}$, 随机选择机器 $M_{\sigma, k}$, 将 B_b 分配到 $M_{\sigma, k}$; 只存在一个批, 则直接将工件 $J_{\pi_i^\sigma}$ 加入该批; 存在不止一个批, 则优先选择剩余容量最小的批, 随后直接将工件 $J_{\pi_i^\sigma}$ 加入该批.

随机产生 N 个初始解, 构成初始种群 P . 初始时, 两个子种群 $P_1 = P_2 = \emptyset$, 种群划分的步骤描述如下: 随机选择 P 中 $N/2$ 个解, 将它们加入 P_1 中, P_2 由剩下的 $N/2$ 个解组成.

2.2 搜索算子

设计 6 种搜索算子 $SO_1 \sim SO_6$, SO_1 和 SO_2 是 2 个全局搜索算子, $SO_3 \sim SO_6$ 是 4 个局部搜索算子.

SO_1 的描述如下: 对 x 和 y 的调度串执行基于位置的交叉^[21]生成新解 z . 如果 $z \succ x$, 则用 z 代替 x , 并用 z 更新 Ω ; 否则, 针对 x 和 y 的机器分配串执行两点交叉生成新解 z' . 若 $z' \succ x$, 则用 z' 代替 x , 并用 z' 更新 Ω ; 否则, 继续针对 x 和 y 的阶段确定串执行单点交叉生成新解 z'' . 若 $z'' \succ x$, 则用 z'' 代替 x , 并用 z'' 更新 Ω ; 若 z'' 和 x 没有支配关系, 则用 z'' 更新 Λ .

$x \succ y$ 表示解 x 支配 y . 如果 $C_{\max}(x) < C_{\max}(y)$ 和 $TEC(x) \leq TEC(y)$ 或者 $C_{\max}(x) \leq C_{\max}(y)$ 和 $TEC(x) < TEC(y)$, 则 $x \succ y$. 当 $x \succ y$ 和 $y \succ x$ 都不满足时, 则 x 和 y 彼此非受支配, 其中应用三角模糊数的比较操作^[19]判断 $C_{\max}(TEC)(x)$ 和 $C_{\max}(TEC)(y)$ 的大小.

外部档案 Ω 用于存储 DABC 产生的非劣解. Ω 的更新方式是将新解添加到 Ω 中, 进行基于 Pareto 支配的比较, 保留非劣解并删除掉 Ω 中所有受支配的解.

记忆解集 Λ 用于存储不能添加到 Ω 的中间数据. Λ 的更新方式与 Ω 一样.

SO_2 的过程同 SO_1 , 只是依次针对调度串、机器分配串和阶段确定串执行顺序、单点和两点交叉.

采用 6 个邻域结构 $\mathcal{N}_1 \sim \mathcal{N}_6$ 组成局部搜索算子. $\mathcal{N}_1 \sim \mathcal{N}_4$ 作用于调度串. \mathcal{N}_1 用来互换随机选择的工件 π_i 和 π_j ; \mathcal{N}_2 针对随机选择工件 π_i 和 π_j , 将 π_i 插入到 π_j 的位置上; $\mathcal{N}_3, \mathcal{N}_4$ 分别与 $\mathcal{N}_1, \mathcal{N}_2$ 类似, 只是在 \mathcal{N}_3 和 \mathcal{N}_4 中, J_{π_i} 为最大模糊完成时间的工件. \mathcal{N}_5 用来逆转机器分配串中位于随机选择的两点 θ_i 与 θ_j 之间的基因, $i < j$. \mathcal{N}_6 针对阶段确定串, 先确定负荷最大的阶段 $h, h \in \{\sigma - 2, \sigma - 1\}$, 然后随机选择值为 h 的基因 φ_i , 令 $\varphi_i = 2\sigma - 3 - h$.

$SO_3 \sim SO_6$ 由 $\mathcal{N}_1 \sim \mathcal{N}_6$ 构成. $SO_l (l > 2)$ 的描述如下: 确定排列 ξ_1, ξ_2, ξ_3 , 令 $d = 1$, 重复下列操作直至 $d > 3$. 生成新解 $z \in \mathcal{N}_{\xi_d}(x_i)$, 一旦 $z \succ x$, 则用 z 替代 x 并用 z 更新 Ω , $d = 4$, 否则 $d = d + 1$. 其中: $\xi_d \in \{l - 2, 5, 6\}$, $\mathcal{N}_g(x_i)$ 是对 x_i 执行 \mathcal{N}_g 获得的邻域解的集合. 为减少计算时间, 只有当 $d = 3$, 且 z 和 x 没有支配关系时, 才用 z 更新 Λ .

2.3 种群裁定方法

EB 和 OB 分别为雇佣蜂种群和跟随蜂种群, 现有 ABC 中 EB 和 OB 往往都是提前确定的^[15-17]. 本研究中, 应用一种基于种群评估的种群裁定方法以动态确定 EB 和 OB.

种群裁定方法的具体步骤如下:

step 1: 如果 $gen = 1$, 则随机从 P_1 和 P_2 中选择一个作为 EB, 另一个即为 OB, γ 随机从 $\{0, 1\}$ 选择.

step 2: 如果 $|EI_l^{gen} - EI_{3-l}^{gen}| \geq \alpha$, 则 P_l 是 EB, P_{3-l} 是 OB, $\gamma = 0, \beta = 0$; 否则, 执行 step 1 中的随机选择确定 EB 和 OB, $\gamma = 1, \beta = \beta + 1$. 其中: γ 是二进制数, β 是整数, α 是阈值, EI_l^{gen} 是 P_l 在第 gen 代的种群评估, 假设 $EI_l^{gen} \geq EI_{3-l}^{gen}$.

$$EI_l^{gen} = U_l^{gen} + V_l^{gen} + Con_l^{gen}. \quad (16)$$

U_l^{gen}, V_l^{gen} 以及 Con_l^{gen-1} 分别定义如下:

$$U_l^{gen} = \sum_{x_i \in P_l} \text{new}_i^{gen-1} / (N/2), \quad (17)$$

$$V_l^{gen} = 1 - \sum_{x_i \in P_l} \text{rank}_i^{gen} / \sum_{x_i \in P} \text{rank}_i^{gen}, \quad (18)$$

$$Con_l^{gen} = |\{x_i \in Z^{gen} | x_i \in P_l^{gen}\}| / |Z^{gen}|, \quad (19)$$

其中 $l \in \{1, 2\}$. 如果 x_i 在第 $gen - 1$ 代被支配从而得到了更新, 则 $\text{new}_i^{gen-1} = 1$; 否则, $\text{new}_i^{gen-1} = 0$. rank_i^{gen} 表示种群 P 在第 gen 代后进行快速非支配排序后解 x_i 的等级值^[22]. $Z^{gen} = \{x_i \in P | \text{rank}_i^{gen} = 1\}$.

2.4 DABC 的 3 个阶段

动态雇佣蜂阶段的具体步骤如下: 依次针对每个解 $x_i \in EB$, 根据搜索策略选择获得 So_i , 对 x_i 执行 So_i , 其中 So_i 表示解 x_i 的搜索策略.

搜索策略选择描述如下: 如果 $i = 1$, 则 So_i 随机从 $\{SO_3, SO_4, \dots, SO_6\}$ 中进行选择; 否则, 若 x_{i-1} 执行 So_{i-1} 后得到更新, 则继续使用 So_{i-1} , $So_i = So_{i-1}$; 否则, 随机从 $\{SO_3, \dots, SO_6\} \setminus So_{i-1}$ 中选择 So_i .

基于自适应交流的跟随蜂阶段具体步骤如下:

step 1: 如果 $\gamma = 0$, 则对每个解 $x_i \in OB$, 随机选择 $x_j \in EB$, 对 x_i 和 x_j 执行 SO_1 生成新解 z . 如果 z 不支配 x_i , 则将 SO_2 作用于 x_i 和 x_j 生成新解 z' . 如果 z' 仍不支配 x_i , 则继续对 x_i 执行 So_j .

step 2: 如果 $\gamma = 1$, 则对每个解 $x_i \in OB$, 随机选择 $x_j \in EB$, 若 $x_j \succ x_i$, 则执行 step 1 中 x_i 和 x_j 的操作; 若 $x_i \succ x_j$, 则对 x_i 执行 So_i , 且对 x_j 和 x_i 执行 SO_1 生成解 z . 如果 z 不支配 x_j , 则将 SO_2 作用于 x_j 和 x_i ; 若 x_i 和 x_j 互不支配, 则随机选择 $x_v \in \Omega$, 对 x_i 和 x_v 执行 SO_1 生成新解 z ; 如果 z 不支配 x_i , 则将 SO_2 作用于 x_i 和 x_v .

当 $\gamma = 0$ 时, EB 、 OB 间差异性可能较大, 依次执行 SO_1 、 SO_2 和 So_j , 实现 $x_i \in OB$ 向 $x_j \in EB$ 学习的单向交流; 当 $\gamma = 1$ 时, EB 、 OB 间差异性可能较小, 此时 x_i 和 x_j 间存在 3 种情况. 情况 1 同 step 1 中操作; 情况 2 时, $x_j \in EB$ 也可能得到更新, 这是一种双向交流.

初始 $trial_i = 0$. 若解 $x_i \in P$ 在一代搜索过程中得到更新, 则 $trial_i = 0$; 否则, $trial_i = trial_i + 1$.

自适应侦察蜂阶段描述如下: 对 $trial_i \geq Limit^{gen}$ 的每个解 x_i , 随机选择解 $x_j \in \Lambda$, 用 x_j 替换 x_i , 随机从 $\{SO_3, SO_4, \dots, SO_6\}$ 中选择 So_i , 并对 x_i 执行 So_i .

$Limit^{gen}$ 表示第 gen 代的阈值, 如果 $gen = 1$, 则 $Limit^{gen} = \lambda$; 否则, $Limit^{gen} = \lfloor e^{-t_{gen}/T_{max}} \times \lambda \rfloor$. 其中: λ 为实数, T_{max} 为总的 CPU 运行时间, t_{gen} 为第 gen 代的 CPU 时间, $\lfloor u \rfloor (\leq u)$ 为最接近 u 的整数.

2.5 算法描述

DABC 的具体步骤描述如下:

step 1: 随机产生初始种群 P , 将 P 划分为 P_1 和 P_2 , $\Lambda = \emptyset, \Omega = \{x_i \in P | rank_i^1 = 1\}$, $gen = 1$.

step 2: 应用种群裁定方法动态确定 EB 和 OB .

step 3: 执行动态雇佣蜂阶段.

step 4: 实行基于自适应交流的跟随蜂阶段.

step 5: 进行多样性强化策略.

step 6: 实施自适应侦察蜂阶段.

step 7: $gen = gen + 1$, 终止条件成立, 则结束搜索; 否则, 返回 step 2.

step 8: 输出外部档案 Ω .

多样性强化策略的具体步骤如下:

step 1: 如果 $\beta = \beta_{min}$, 则对 P 进行快速非支配排序^[22], 得到每个解 x_i 的拥挤度距离 $dist_i$, 定义集合 $\Phi_1 = \{x_i \in P | dist_i = +\infty\}$, 依次针对每个解 $x_i \in \Phi_1$, 随机选择 $x_j \in \Lambda$, 用 x_j 替换 x_i .

step 2: 如果 $\beta = \beta_{max}$, 则对 P 进行快速非支配排序^[22], 获得每个解 x_i 的等级值 $rank_i^{gen}$, 定义集合 $\Phi_2 = \{x_i \in P | rank_i^{gen} = 1\}$, 依次针对每个解 $x_i \in \Phi_2$, 随机选择 $x_j \in \Lambda$, 用 x_j 替换 x_i . 初始 $\beta = 0, \beta_{min}$ 和 β_{max} 是整数, $\beta_{min} < \beta_{max}$. $\beta \geq \beta_{min}$ 表明 DABC 至少连续 β 代 $|EI_1^{gen} - EI_2^{gen}| < \alpha$.

DABC 具有如下特点: 根据 EI_1^{gen} 、 EI_2^{gen} 建立种群裁定方法动态确定 EB 和 OB , 给出动态雇佣蜂阶段、基于自适应交流的跟随蜂阶段、多样性强化策略以及自适应侦察蜂阶段. 这些特点使得全局搜索和局部搜索的有效协调成为可能, 从而提高算法的求解质量.

3 实验结果与分析

为测试 DABC 在求解所研究 EFHFSP 上的性能, 进行仿真实验. 实验均由 CodeBlocks 16.01 C 编程实现, 并运行于 8 GB RAM i5-10210 CPU 的计算机上.

3.1 测试实例、对比算法及评价指标

产生 36 个测试实例, 各实例表示为 $n \times H \times G$, $n \in \{8\omega, 13\omega, 18\omega, 23\omega\}$, $\omega \in \{1, 2, 3\}$, $H \in \{6, 8, 10\}$, $G \in \{5, 8, 10\}$. 表 2 和表 3 分别描述了 n 和 G 等相关数据. $\tilde{p}_{usk} = (\delta_1 \times \bar{p}'_{usk}, \bar{p}'_{usk}, \delta_2 \times \bar{p}'_{usk})$, δ_1

表 2 n 、 G 、 $|A_v|$ 、 \bar{w}_v 和 \bar{r}_v 的相关数据

n	G	$ A_v , \bar{w}_v, \bar{r}_v$									
		$v = 1$	$v = 2$	$v = 3$	$v = 4$	$v = 5$	$v = 6$	$v = 7$	$v = 8$	$v = 9$	$v = 10$
8 ω	5	2 ω , 469, 1	1 ω , 634, 3	2 ω , 333, 3	1 ω , 584, 2	2 ω , 284, 1	—	—	—	—	—
13 ω	8	2 ω , 469, 1	2 ω , 634, 3	2 ω , 333, 3	1 ω , 584, 2	2 ω , 284, 1	1 ω , 537, 3	1 ω , 371, 2	2 ω , 460, 2	—	—
18 ω	8	2 ω , 469, 1	2 ω , 634, 3	2 ω , 333, 3	3 ω , 584, 2	2 ω , 284, 1	2 ω , 537, 3	3 ω , 371, 2	2 ω , 460, 2	—	—
23 ω	10	2 ω , 469, 1	3 ω , 634, 3	2 ω , 333, 3	1 ω , 584, 2	4 ω , 284, 1	3 ω , 537, 3	2 ω , 371, 2	2 ω , 460, 2	3 ω , 646, 2	1 ω , 478, 1

表3 H, m_s, Q_k 和 σ 的相关数据

H	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9	m_{10}	Q_1	Q_2	Q_3	σ
6	2	3	3	2	2	3	—	—	—	—	1500	2000	—	5
8	2	3	3	2	2	3	3	2	—	—	1500	2000	—	4
10	2	3	3	2	2	3	3	2	2	3	1500	2000	1800	6

和 δ_2 为实数, $\delta_1 \in [0.5, 0.8], \delta_2 \in [1.2, 1.5]$. 表4给出了 \bar{p}'_{vsk} 和 q_{sk} 的取值区间. $\tilde{tp}_s = (tp_s^1, tp_s^2, tp_s^3), tp_s^1 = [2, 4], tp_s^2 = [5, 7], tp_s^3 = [8, 10]. E_{AVG} = 1.25. \bar{p}'_{vsk}, q_{sk}, tp_s^1, tp_s^2$ 和 tp_s^3 是整数, 它们在上述相关区间内均服从均匀分布.

表4 \bar{p}'_{vsk} 和 q_{sk} 的取值区间

v	$H = 6$		$H = 8$		$H = 10$	
	\bar{p}'_{vsk}	q_{sk}	\bar{p}'_{vsk}	q_{sk}	\bar{p}'_{vsk}	q_{sk}
1	[10, 20]	[1, 5]	[10, 20]	[1, 5]	[5, 20]	[1, 5]
2	[10, 20]	[1, 5]	[20, 30]	[5, 10]	[5, 20]	[2, 5]
3	[20, 30]	[5, 10]	[20, 30]	[5, 10]	[10, 20]	[3, 10]
4	[20, 30]	[20, 30]	[100, 160]	[150, 180]	[20, 30]	[5, 10]
5	[100, 160]	[150, 180]	[100, 130]	[40, 60]	[20, 30]	[5, 10]
6	[100, 130]	[100, 130]	[100, 130]	[60, 80]	[100, 160]	[100, 120]
7	—	—	[100, 130]	[20, 30]	[100, 130]	[40, 60]
8	—	—	[50, 130]	[15, 20]	[100, 130]	[60, 80]
9	—	—	—	—	[100, 130]	[20, 30]
10	—	—	—	—	[50, 100]	[15, 20]

选择扩展的 NSGA-II (extended NSGA-II, ENSGA-H)^[18] 和多目标 ABC (multiobjective ABC, MOABC)^[23] 作为对比算法. ENSGA-II 和 MOABC 在求解双目标不确定混合流水车间方面都有很好的性能, 因此选择这2个算法作为对比算法.

为了应用上述对比算法求解所研究的 EFHFSP, 针对这2个算法作如下修改: 在编码中加入阶段确定串, 解码过程中添加运输和组批; 去掉 ENSGA-II 的速度串和相关算子.

NSGA-II^[22] 作为著名的多目标优化算法, 广泛应用于多目标优化问题的求解, 故选择它作为比较算法. 为求解所研究的问题, NSGA-II 采用从 SO_1 和 SO_2 中随机选择的算子作为交叉, 变异则执行从 $SO_3 \sim SO_6$ 中随机选择的算子.

构建 ABC 显示 DABC 新策略的有效性. 从 DABC 中删除种群裁定方法、搜索策略动态选择、基于自适应交流、多样性强化策略等, 雇佣蜂阶段执行从 SO_1 和 SO_2 中随机选择的算子, 跟随蜂阶段则运行从 $SO_3 \sim SO_6$ 中随机选择的局部搜索, 侦察蜂阶段对满足条件的解采用与雇佣蜂阶段一样的全局搜索.

应用 C ^[24] 和 DI_R ^[25] 这2个指标评价算法的性能, 其中 C 用于衡量两个算法非劣解集之间的支配关系, DI_R 用于评价算法的收敛性能.

3.2 参数设置

通过实验发现, 当 CPU 运行时间达到 0.5n 秒时, DABC 和对比算法关于所有实例都收敛, 因此将 $T_{max} = 0.5n$ 秒设置为所有算法的终止条件.

DABC 的其他5个参数为: $N, \lambda, \alpha, \beta_{min}, \beta_{max}$. 为确定合适的参数, 设置25种组合. 其中: $N = 50, 80, 100, 120, 150; \lambda = 20, 30, 40, 50, 60; \alpha = 0.08, 0.10, 0.12, 0.14, 0.16; \beta_{min} = 5, 10, 15, 20, 25; \beta_{max} = 10, 15, 20, 25, 30$. 应用 Taguchi 方法^[26] 进行参数设置, 发现使得 DABC 性能最好的一组参数为 $N = 100, \lambda = 30, \alpha = 0.01, \beta_{min} = 10, \beta_{max} = 20$, 故采用上述参数设置.

ABC 中参数 N 和终止条件的设置与 DABC 一样, 且侦查蜂阶段的阈值 $limit = \lambda$. 除终止条件外, ENSGA-II 和 MOABC 的参数设置分别选自文献[18]和文献[23]. 实验结果表明, 这些参数设置依旧有效, 能使相应的算法性能最好.

NSGA-II 除终止条件外, 根据实验得到最合适的参数: 种群规模 100, 交叉概率 0.8, 变异概率 0.2.

3.3 结果与分析

表5和表6给出了 DABC 与其他4种算法的计算结果, 5种算法关于每个实例各自独立运行10次. 图1描述了2个实例非劣解的分布图. 表7为配对样本均值 t 检验的结果, t 检测 (A_1, A_2) 表示一次配对, 用来判断算法 A_1 是否给出了优于 A_2 的样本均值. 假设显著水平为 0.05, 若 p 值小于 0.05, 则说明算法 A_1 优于算法 A_2 . D、A、N、E 和 M 分别表示 DABC、ABC、NSGA-II、ENSGA-II 和 MOABC.

如表5和表6所示, 关于全部36个实例, $C(D, A)$ 总是大于 $C(A, D)$, 且 DABC 仅关于实例 $69 \times 10 \times 10$ 产生了比 ABC 更大的 DI_R . 可见, DABC 的搜索性能明显优于 ABC, 表7和图1也揭示了两者的显著性能差异. 上述结论表明, 种群裁定方法、搜索策略动态选择、自适应交流等新策略的引入显著增强了 DABC 的搜索能力, 新策略合理有效.

表5 5种算法关于指标C的计算结果

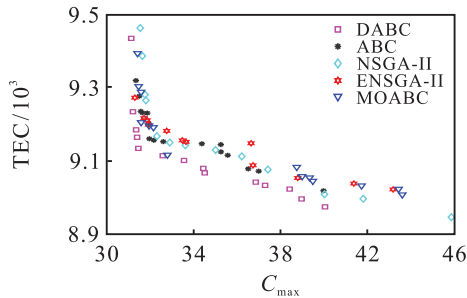
$n \times H \times G$	$C(D, A)$	$C(A, D)$	$C(D, N)$	$C(N, D)$	$C(D, E)$	$C(E, D)$	$C(D, M)$	$C(M, D)$
8×6×5	0.857	0.071	0.786	0.143	1.000	0	1.000	0.071
8×8×5	0.636	0.154	0.667	0.077	0.800	0.038	0.900	0
8×10×5	0.840	0.042	0.818	0.042	0.846	0	0.759	0.125
13×6×8	1.000	0	1.000	0	1.000	0	1.000	0
13×8×8	0.700	0.091	1.000	0	1.000	0	0.909	0.030
13×10×8	0.583	0.083	0.933	0	1.000	0	0.852	0.042
18×6×8	0.487	0.211	0.522	0.263	0.885	0.018	0.744	0.123
18×8×8	0.933	0	0.909	0	0.833	0	0.750	0.111
18×10×8	0.692	0.091	0.813	0.091	0.667	0.182	0.478	0.409
23×6×10	0.964	0.031	0.900	0.031	0.900	0.031	1.000	0
23×8×10	1.000	0	0.929	0	1.000	0	0.882	0.071
23×10×10	0.700	0.267	0	0.200	1.000	0	0.533	0.533
16×6×5	1.000	0.042	1.000	0.021	1.000	0	1.000	0.042
16×8×5	0.842	0.042	0.615	0.083	0.846	0	0.750	0.042
16×10×5	0.960	0	0.909	0	0.846	0.031	1.000	0
26×6×8	0.952	0	0.917	0.065	1.000	0	0.722	0.226
26×8×8	1.000	0	1.000	0	0.333	0.444	0.773	0.056
26×10×8	0.625	0.100	0.706	0.100	1.000	0	0.640	0.200
36×6×8	0.921	0	0.783	0.067	1.000	0	0.913	0.022
36×8×8	0.750	0	0.857	0.200	0.800	0.200	0.833	0
36×10×8	1.000	0	1.000	0	0.833	0	0.933	0.083
46×6×10	0.769	0.074	0.741	0.111	0.969	0	0.805	0.185
46×8×10	0.818	0	0.667	0.136	1.000	0	0.900	0.045
46×10×10	0.882	0	0.500	0.357	0.556	0.143	0.600	0.357
24×6×5	0.931	0.018	0.675	0.268	0.939	0.018	0.958	0
24×8×5	0.947	0	0.875	0.059	0.923	0	0.917	0.059
24×10×5	0.667	0.563	0.909	0	0.833	0.063	0.750	0.188
39×6×8	0.857	0.069	1.000	0	1.000	0	0.938	0
39×8×8	0.733	0.111	0.800	0.111	0.750	0.111	0.917	0.056
39×10×8	0.944	0.167	0.900	0.042	0.923	0	0.600	0.208
54×6×8	0.872	0.067	0.778	0.117	1.000	0	0.791	0.083
54×8×8	1.000	0	1.000	0	0.889	0	1.000	0
54×10×8	1.000	0	0.750	0.182	1.000	0	1.000	0
69×6×10	0.926	0.083	0.579	0.250	1.000	0	0.806	0
69×8×10	0.889	0.083	1.000	0	0.889	0	0.722	0.333
69×10×10	1.000	0	0.800	0.143	0.800	0	0.600	0

表6 5种算法关于指标DI_R的计算结果

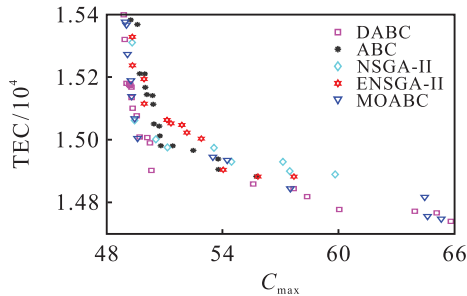
$n \times H \times G$	DABC	ABC	NSGA-II	ENSGA-II	MOABC	$n \times H \times G$	DABC	ABC	NSGA-II	ENSGA-II	MOABC
8×6×5	0.315	4.250	3.848	5.842	6.796	36×6×8	0.710	6.946	3.789	11.383	4.667
8×8×5	1.462	2.664	5.364	5.824	7.220	36×8×8	6.083	30.024	19.798	23.277	9.970
8×10×5	1.499	2.834	7.685	7.401	5.751	36×10×8	1.398	17.966	13.049	15.458	7.451
13×6×8	0.000	13.462	11.766	16.640	12.455	46×6×10	0.492	3.810	3.245	5.285	2.748
13×8×8	0.339	4.362	16.775	14.802	4.850	46×8×10	1.407	10.841	7.935	8.756	6.053
13×10×8	3.300	8.933	8.951	8.558	4.908	46×10×10	6.583	10.311	10.735	15.094	6.505
18×6×8	0.858	4.852	2.812	4.318	2.005	24×6×5	0.264	2.273	2.424	2.982	2.167
18×8×8	1.789	10.026	12.439	16.264	5.708	24×8×5	0.791	9.473	18.696	12.491	6.305
18×10×8	1.672	10.169	6.996	10.074	1.977	24×10×5	4.202	11.254	13.873	20.196	14.360
23×6×10	0.103	4.930	6.873	6.873	5.917	39×6×8	0.812	8.207	13.680	16.008	7.072
23×8×10	3.067	11.218	8.159	13.247	7.427	39×8×8	1.304	7.343	24.518	18.031	13.167
23×10×10	6.198	8.881	13.137	19.106	6.709	39×10×8	1.224	17.535	16.063	7.355	4.139
16×6×5	0.000	25.400	21.300	26.500	26.900	54×6×8	0.490	4.194	5.792	6.506	2.244
16×8×5	1.287	6.531	10.009	7.647	10.289	54×8×8	4.660	22.206	21.039	18.522	19.520
16×10×5	0.293	9.219	10.556	9.278	36.466	54×10×8	0.163	24.317	22.295	19.573	18.441
26×6×8	0.651	9.194	15.026	17.141	4.678	69×6×10	1.520	8.322	4.362	9.939	4.988
26×8×8	4.763	10.419	17.459	21.663	6.925	69×8×10	3.391	17.417	11.843	15.056	7.940
26×10×8	2.279	5.112	6.697	7.460	2.700	69×10×10	26.963	19.142	25.834	17.466	5.974

表7 配对样本t检测

t检测	p-value (C)	p-value (DI _R)
t检测 (DABC, ABC)	0	0
t检测 (DABC, NSGA-II)	0	0
t检测 (DABC, ENSGA-II)	0	0
t检测 (DABC, MOABC)	0	0.001



(a) 23×8×10



(b) 39×10×8

图1 5种算法的非劣解分布

由表5和表6可知,关于35个实例有 $C(D, N) > C(N, D)$,其中关于8个实例有 $C(D, N) = 1$,即NSGA-II关于这些实例产生的所有解均受DABC的非劣解支配. DABC几乎所有的实例均产生了比NSGA-II更小的DI_R.从表7和图1也可以看出,DABC的结果明显优于NSGA-II.

比较DABC与其他两个对比算法的计算结果,可以发现,DABC的搜索优势更大. DABC关于几乎所有实例有 $C(D, E) > C(E, D)$ 和 $C(D, M) > C(M, D)$,且DABC几乎所有的实例都取得了比ENSGA-II和MOABC更优的DI_R.总之,DABC明显优于其所有对比算法.

DABC的优越性能主要来自于种群裁定方法、搜索策略动态选择、自适应交流和多样性强化策略.种群裁定方法基于种群评估动态决定EB和OB,可以有效避免算法陷入局部最优.搜索策略动态选择使得效率较好的搜索策略得到充分利用,可提高算法的进化效率.自适应交流可产生强大的搜索能力.多样性强化策略有利于保持种群多样性,避免算法过早收敛.因此,DABC在求解考虑BPM和部分阶段无优先关系的EFHFSP方面具有较强的搜索优势.

4 结论

本文针对考虑BPM和部分阶段无优先关系的EFHFSP,提出了DABC以同时最小化最大模糊完成时间和模糊总能耗.给出了基于EI₁^{gen}、EI₂^{gen}的种群裁定方法以动态决定每一代的EB和OB,应用了动态雇佣蜂阶段,基于自适应交流的跟随蜂阶段,多样性强化策略以及自适应侦察蜂阶段.仿真实验表明,DABC中策略的有效性以及在求解所研究的EFHFSP方面具有较强优势.

EHFSP和模糊调度已得到广泛关注,然而针对实际制造过程中的EFHFSP还未得到充分的研究.后续将继续深入研究这类问题,并尝试通过使用其他元启发算法和一些新的优化机制来求解.另外,分布式模糊绿色调度问题也是未来研究的主要课题.

参考文献(References)

- [1] Bendig D, Kleine-Stegemann L, Gisa K. The green manufacturing framework—A systematic literature review[J]. Cleaner Engineering and Technology, 2023, 13: 100613.
- [2] Jiang Y W, Tang X L, Li K, et al. Approximation algorithms for bi-objective parallel-machine scheduling in green manufacturing[J]. Computers & Industrial Engineering, 2023, 176: 108949.
- [3] 王建华, 潘宇杰, 孙瑞. 自适应Jaya算法求解多目标柔性车间绿色调度问题[J]. 控制与决策, 2021, 36(7): 1714-1722.
(Wang J H, Pan Y J, Sun R. Multi-objective flexible job shop green scheduling problem with self-adaptive Jaya algorithm[J]. Control and Decision, 2021, 36(7): 1714-1722.)
- [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] Yan J H, Li L, Zhao F, et al. A multi-level optimization approach for energy-efficient flexible flow shop scheduling[J]. Journal of Cleaner Production, 2016, 137: 1543-1552.
- [6] Lei D M, Gao L, Zheng Y L. A novel teaching-learning-based optimization algorithm for energy-efficient scheduling in hybrid flow shop[J]. IEEE Transactions on Engineering Management, 2018, 65(2): 330-340.
- [7] Li M, Lei D M, Cai J C. Two-level imperialist competitive algorithm for energy-efficient hybrid flowshop scheduling problem with relative importance of objectives[J]. Swarm and Evolutionary Computation, 2019, 49: 34-43.

- [8] Zuo Y D, Fan Z, Zou T R, et al. A novel multi-population artificial bee colony algorithm for energy-efficient hybrid flow shop scheduling problem[J]. *Symmetry*, 2021, 13(12): 2421.
- [9] Gong G L, Chiong R, Deng Q W, et al. Energy-efficient flexible flow shop scheduling with worker flexibility[J]. *Expert Systems with Applications*, 2019, 141: 112902.
- [10] Qin H X, Han Y Y, Zhang B, et al. An improved iterated greedy algorithm for the energy-efficient blocking hybrid flow shop scheduling problem[J]. *Swarm and Evolutionary Computation*, 2022, 69: 100992.
- [11] Lu H, Qiao F. An efficient adaptive genetic algorithm for energy saving in the hybrid flow shop scheduling problem with batch production at last stage[J]. *Expert Systems*, 2022, 39(2): 1-15.
- [12] Wu X L, Cao Z. An improved multi-objective evolutionary algorithm based on decomposition for solving re-entrant hybrid flow shop scheduling problem with batch processing machines[J]. *Computers & Industrial Engineering*, 2022, 169: 108236.
- [13] 吴秀丽, 曹铮. 带连续式批处理机的可重入混合流水车间调度[J]. *计算机集成制造系统*, 2022, 28(11): 3365-3378.
(Wu X L, Cao Z. Re-entrant hybrid flow shop scheduling problem with continuous batch processing machines[J]. *Computer Integrated Manufacturing Systems*, 2022, 28(11): 3365-3378.)
- [14] Lu X X, Li K P, Wang F, et al. Optimal bidding strategy of DER aggregator considering bilateral uncertainty via information gap decision theory[C]. 2020 IEEE/IAS 56th Industrial and Commercial Power Systems Technical Conference (I&CPS). Las Vegas, 2020: 1-8.
- [15] Wang L, Zhou G, Xu Y, et al. A hybrid artificial bee colony algorithm for the fuzzy flexible job-shop scheduling problem[J]. *International Journal of Production Research*, 2013, 51(12): 3593-3608.
- [16] Gao K Z, Suganthan P N, Pan Q K, et al. An improved artificial bee colony algorithm for flexible job-shop scheduling problem with fuzzy processing time[J]. *Expert Systems with Applications*, 2016, 65: 52-67.
- [17] Jia Z H, Yan J H, Leung J Y T, et al. Ant colony optimization algorithm for scheduling jobs with fuzzy processing time on parallel batch machines with different capacities[J]. *Applied Soft Computing*, 2019, 75: 548-561.
- [18] Wang Y J, Wang G G, Tian F M, et al. Solving energy-efficient fuzzy hybrid flow-shop scheduling problem at a variable machine speed using an extended NSGA-II[J]. *Engineering Applications of Artificial Intelligence*, 2023, 121: 105977.
- [19] Cai J C, Lei D M. A cooperated shuffled frog-leaping algorithm for distributed energy-efficient hybrid flow shop scheduling with fuzzy processing time[J]. *Complex & Intelligent Systems*, 2021, 7(5): 2235-2253.
- [20] Li X X, Guo X, Tang H T, et al. An improved cuckoosearch algorithm for the hybrid flow-shop scheduling problem in sand casting enterprises considering batch processing[J]. *Computers & Industrial Engineering*, 2023, 176: 108921.
- [21] Lei D M, Su B, Li M. Cooperated teaching-learning-based optimisation for distributed two-stage assembly flow shop scheduling[J]. *International Journal of Production Research*, 2021, 59(23): 7232-7245.
- [22] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182-197.
- [23] Fu Y P, Wang H F, Wang J W, et al. Multiobjective modeling and optimization for scheduling a stochastic hybrid flow shop with maximizing processing quality and minimizing total tardiness[J]. *IEEE Systems Journal*, 2021, 15(3): 4696-4707.
- [24] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach[J]. *IEEE Transactions on Evolutionary Computation*, 1999, 3(4): 257-271.
- [25] Pan Q K, Wang L, Qian B. A novel differential evolution algorithm for bi-criteria no-wait flow shop scheduling problems[J]. *Computers & Operations Research*, 2009, 36(8): 2498-2511.
- [26] Taguchi G. *Introduction to quality engineering: Designing quality into products and processes*[J]. Tokyo: Asian Productivity Organization, 1986: 191.

作者简介

王静(1996—),女,博士生,从事制造系统智能优化与调度的研究, E-mail: 307624@whut.edu.cn;

雷德明(1968—),男,教授,博士生导师,从事智能系统优化与控制、基于数据和知识驱动下的智能控制等研究, E-mail: deminglei11@163.com.