

文章编号:1006-2467(2019)07-0773-07

DOI: 10.16183/j.cnki.jsjtu.2019.07.002

# 考虑能耗和准时的混合流水线多目标调度

周炳海, 刘文龙

(同济大学机械与能源工程学院, 上海 201804)

**摘要:** 针对加工时间和交货期模糊的混合流水线, 考虑阶段内并行机互不相关、换模时间与工件加工次序相关等约束, 面向降低生产能源消耗和保证准时交货的双目标调度问题, 提出一种改进型差分进化算法。首先, 以最小化系统能耗和准时交货惩罚为优化目标建立双目标优化数学模型; 在此基础上, 使用 NEH(Nawaz, Ensore, Ham)方法获得优质初始解, 并利用优质解挑战机制进行有效的邻域挖掘; 同时, 引入混沌搜索策略以保证算法的全局搜索能力; 最后, 将数值实验与有代表性的算法的计算结果进行对比, 以验证所提算法的可行性与有效性。

**关键词:** 混合流水线调度; 差分进化算法; 模糊; 不相关并行机; 换模时间

中图分类号: TP 391

文献标志码: A

## Multi-Objective Hybrid Flow-Shop Scheduling Problem Considering Energy Consumption and On-Time Delivery

ZHOU Binghai, LIU Wenlong

(School of Mechanical Engineering, Tongji University, Shanghai 201804, China)

**Abstract:** To guarantee on-time delivery of the hybrid flow-shop system and reduce energy consumption at the meantime, a modified differential evolution algorithm is proposed for the multi-objective hybrid flow-shop scheduling problem with fuzzy processing time and due date, considering in-stage unrelated parallel machines and sequence-dependent setup time. First, a bi-objective mathematical model is established to minimize on-time delivery penalty and energy consumption. Then, a modified algorithm is developed which efficiently generates high-quality initial solutions with NEH (Nanaz, Ensore, Ham)-based heuristic method, thoroughly exploits neighborhoods with the elite individual challenging mechanism. The modified algorithm highly improves the exploration ability with chaotic search strategy. Finally, the results of the comparison with existing typical algorithms and numerical experiment demonstrate that the proposed algorithm is feasible and effective.

**Key words:** hybrid flow-shop scheduling; differential evolution algorithm; fuzzy; unrelated parallel machine; setup time

符号说明:

$C^D$ —单位时间延迟交货成本

$C^I$ —单位时间库存成本

$E_{ijk}$ —工件  $i$  在第  $j$  个加工阶段机器  $k$  上的能耗

$\varepsilon_{ijk}^{\text{pro}}$ —工件  $i$  在第  $j$  个加工阶段机器  $k$  上的模糊加工能耗

收稿日期: 2018-01-02

基金项目: 国家自然科学基金资助项目(71471135)

作者简介: 周炳海(1965-), 男, 浙江省浦江县人, 教授, 研究方向为离散制造系统的调度、建模与仿真. E-mail: bhzhou@tongji.edu.cn.

$E_{jk}$ —第  $j$  个加工阶段第  $k$  台机器的能耗

$p_i$ —工件  $i$  的总体加权交货惩罚

$P_{jk}$ —第  $j$  个加工阶段第  $k$  台机器的功率

$P_{ijk}$ —工件  $i$  所在第  $j$  个加工阶段第  $k$  台机器的功率

$t_{ijk}^B$ —工件  $i$  在第  $j$  个加工阶段第  $k$  台机器上的模糊加工开始时间

$t_i^C$ —工件  $i$  的模糊加工完成时间

$t_i^d$ —工件  $i$  的梯形模糊交货时间,  $t_i^d = (t_i^{d1}, t_i^{d2}, t_i^{d3}, t_i^{d4})$

$t_{jk}$ —第  $j$  个加工阶段第  $k$  台机器耗时

$t_{ijk}$ —工件  $i$  在第  $j$  个加工阶段第  $k$  台机器上加工的三角模糊加工时间,  $t_{ijk} = (t_{ijk}^1, t_{ijk}^2, t_{ijk}^3)$

$t_{iljk}$ —第  $j$  个加工阶段第  $k$  台机器上工件  $l$  正好排在工件  $i$  后的加工时间

混合流水线调度问题(HFSP)是传统流水线调度问题的一种推广,具有很强的工程背景,广泛存在于化工、汽车、纺织、半导体等领域。HFSP 同时具有流水线作业和并行机两者的综合特征:工艺流程复杂、不确定性大、求解调度问题难度高,在数学领域属于非确定性多项式(NP-hard)问题。因此,研究 HFSP 具有重要的学术意义和应用价值<sup>[1]</sup>。

在现实环境之中,由于人为因素和市场波动, HFSP 的生产参数往往是不确定的,可以通过模糊集理论进行描述<sup>[2]</sup>。Hong 等<sup>[3-4]</sup>首次将离散模糊概念从流水线系统推广应用于研究 HFSP 问题,并在此后继续推广到了连续模糊领域。Pinedo 等<sup>[5]</sup>提出与工件加工次序相关的换模时间可能会占据生产中机器 20% 的有效工作时间。

近年来,受到全球变暖和环保要求等因素的影响<sup>[6]</sup>,耗能占全国可消费能源总量 72.8%<sup>[7]</sup>的制造业开始集中关注如何在提升生产效率与减少能源消耗之间寻求一个平衡点,以达到绿色生产的目标。相关研究人员开始将研究内容聚焦于具有节能意识的调度目标上<sup>[8]</sup>。Dai 等<sup>[9]</sup>针对 HFSP 提出一个节能模型,可以同时优化最大完工时间和能源消耗。Luo 等<sup>[10]</sup>在此研究基础上提出电价变动模型,基于高低电价优化电力成本。然而,目前对于考虑能耗的混合流水线在生产中的不确定性及与工件加工次序相关的换模时间约束问题的相关研究较少,缺乏可以参考的研究依据。

因此,本文以最小化系统能耗和准时交货惩罚作为双优化目标,使用模糊数描述加工时间和交货期。同时,考虑换模时间与工件加工次序相关及并行机不相关两种约束,对 HFSP 开展研究。最后,通过理论分析和数值实验验证改进型双目标差分进化算法的可行性和有效性。

$\pi_{jk}$ —第  $j$  个加工阶段第  $k$  台机器上被指派的加工任务集合,

$\pi_{jk} = \{ \pi_{jk}(1), \pi_{jk}(2), \dots, \pi_{jk}(n_{jk}) \}$

$\sigma_{\pi_{jk}(r-1)\pi_{jk}(r)}$ —当  $\sigma_{\pi_{jk}(r-1)\pi_{jk}(r)} = 1$  时,表示第  $j$  个加工阶段第  $k$  台机器上被指派的第  $(r-1)$  和第  $r$  个加工任务中执行 1 次开关机

$\varphi_{jkir}$ —当  $\varphi_{jkir} = 1$  时,表示工件  $i$  在第  $j$  个加工阶段第  $k$  台机器上被指派的加工任务集合中排第  $r$  位

上标

idel—闲时

off—关机

on—开机

pro—加工

set—换模

## 1 问题描述

同时考虑换模时间与工件次序相关及并行机不相关两种约束的 HFSP 可描述如下:  $n$  个工件在流水线上进行  $s$  个阶段的加工,每个阶段至少有 1 台机器且至少有 1 个阶段存在并行机;同一阶段的各机器加工同一工件所需的加工时间不同且互不相关,所需的换模时间与加工的上一个工件种类相关;在每一阶段,各工件均要在其中任意 1 台机器上完成 1 道工序。已知工件各道工序在各机器上的处理时间,要求确定所有工件的排序以及在每个阶段机器的分配情况,使得某种调度指标最小<sup>[1]</sup>。HFSP 示意图如图 1 所示,其中  $m_s$  表示阶段  $s$  并行机数量。为了更加有效地描述 HFSP,现做如下假设:① 工件的加工一旦开始不可中断;② 1 台机器同一时刻只能加工 1 个工件;③ 1 个工件同一时刻只能在 1 台机器上加工;④ 工件可在每阶段的任意机器上加工。

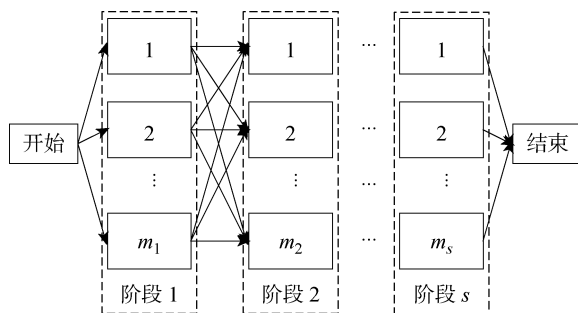


图 1 HFSP 示意图

Fig. 1 Illustration of HFSP

根据问题描述和问题假设对系统问题的建模如下:

$$\min F = (f_1, f_2) \quad (1)$$

$$f_1 = \min \sum_{i=1}^n p_i \quad (2)$$

$$f_2 = \min \sum_{j=1}^s \sum_{k=1}^{m_j} \sum_{r=1}^{n_{jk}} E_{\pi_{jk}(r)jk} \quad (3)$$

式中:

$$p_i = \begin{cases} \mu_i(t_i^C)C^1(t_i^{d1} - t_i^C), & t_i^C \leq t_i^{d1} \\ \mu_i(t_i^C)C^1(t_i^C - t_i^{d1}), & t_i^{d1} \leq t_i^C \leq t_i^{d2} \\ 0, & t_i^{d2} \leq t_i^C \leq t_i^{d3} \\ \mu_i(t_i^C)C^D(t_i^C - t_i^{d3}), & t_i^{d3} \leq t_i^C \leq t_i^{d4} \\ \mu_i(t_i^C)C^D(t_i^C - t_i^{d4}), & t_i^{d4} \leq t_i^C \end{cases} \quad (4)$$

$$\mu_i(t_i^C) = \begin{cases} \lfloor (t_i^{d2} - t_i^C)/(t_i^{d2} - t_i^{d1}) \rfloor, & t_i^C \leq t_i^{d1} \\ 1 - (t_i^C - t_i^{d1})/(t_i^{d2} - t_i^{d1}), & t_i^{d1} \leq t_i^C \leq t_i^{d2} \\ 1, & t_i^{d2} \leq t_i^C \leq t_i^{d3} \\ (t_i^C - t_i^{d3})/(t_i^{d4} - t_i^{d3}), & t_i^{d3} \leq t_i^C \leq t_i^{d4} \\ \lfloor (t_i^C - t_i^{d3})/(t_i^{d4} - t_i^{d3}) \rfloor, & t_i^{d4} \leq t_i^C \end{cases} \quad (5)$$

$$\sum_{k=1}^{m_j} \sum_{r=1}^{n_{jk}} \varphi_{jkir} = 1 \quad (6)$$

$$\pi_{jk}(r) = \sum_{i=1}^n i \varphi_{jkir} \quad (7)$$

$$E_{\pi_{jk}(r)jk}^{\text{set}} = P_{\pi_{jk}(r-1)\pi_{jk}(r)jk}^{\text{set}} t_{\pi_{jk}(r-1)\pi_{jk}(r)jk}^{\text{set}} \quad (8)$$

$$E_{jk}^{\text{onoff}} = P_{jk}^{\text{on}} t_{jk}^{\text{on}} + P_{jk}^{\text{off}} t_{jk}^{\text{off}} \quad (9)$$

$$\sigma_{\pi_{jk}(r-1)\pi_{jk}(r)} = \begin{cases} 1, & E_{jk}^{\text{onoff}} + E_{\pi_{jk}(r)jk}^{\text{set}} \geq P_{jk}^{\text{idle}} t_{\pi_{jk}(r-1)\pi_{jk}(r)jk}^{\text{idle}} \\ 0, & E_{jk}^{\text{onoff}} + E_{\pi_{jk}(r)jk}^{\text{set}} < P_{jk}^{\text{idle}} t_{\pi_{jk}(r-1)\pi_{jk}(r)jk}^{\text{idle}} \end{cases} \quad (10)$$

$$E_{\pi_{jk}(r)jk}^{\text{idle}} = \begin{cases} P_{jk}^{\text{idle}} t_{\pi_{jk}(r-1)\pi_{jk}(r)jk}^{\text{idle}}, & \sigma_{\pi_{jk}(r-1)\pi_{jk}(r)} = 1 \\ E_{jk}^{\text{onoff}}, & \sigma_{\pi_{jk}(r-1)\pi_{jk}(r)} = 0 \end{cases} \quad (11)$$

$$\mathcal{E}_{\pi_{jk}(r)jk}^{\text{pro}} = P_{\pi_{jk}(r)jk}^{\text{pro}} t_{\pi_{jk}(r)jk} \quad (12)$$

$$E_{\pi_{jk}(r)jk} = E_{\pi_{jk}(r)jk}^{\text{idle}} + E_{\pi_{jk}(r)jk}^{\text{set}} + \mathcal{E}_{\pi_{jk}(r)jk}^{\text{pro}} \quad (13)$$

s. t.

$$\sum_{i=1}^n x_{il} = 1 \quad (14)$$

$$\sum_{k=1}^{m_j} y_{ijk} = 1 \quad (15)$$

$$t_{(j+1)k}^B + \sum_{l=0}^n \sum_{k=1}^{m_j} (t_{iljk}^{\text{set}} z_{lijk}) + t_{ijk} \leq t_{(j+1)k'}^B \quad (16)$$

$$t_{ijk}^B + \sum_{l=0}^n \sum_{k=1}^{m_j} (t_{iljk}^{\text{set}} z_{lijk}) + t_{ijk} \leq t_i^C \quad (17)$$

$$\sum_{i=1}^n x_{il} t_{ijk}^B \leq \sum_{i=1}^n x_{i(l+1)} t_{ijk'}^B \quad (18)$$

$$\sum_{i=1}^n x_{il} (t_{ijk}^B + \sum_{l=0}^n \sum_{k=1}^{m_j} (t_{iljk}^{\text{set}} z_{lijk}) + t_{ijk}) \leq$$

$$\sum_{i=1}^n x_{i(l+1)} t_{ijk'}^B \quad (19)$$

$$l = 1, 2, \dots, n$$

$$i = 1, 2, \dots, n$$

$$j = 1, 2, \dots, s$$

$$x_{il} = \begin{cases} 1, & \text{工件 } i \text{ 被安排在第 } l \text{ 个位置} \\ 0, & \text{其他} \end{cases}$$

$$y_{ijk} = \begin{cases} 1, & \text{工件 } i \text{ 的工序 } j \text{ 在机器 } k \text{ 上} \\ 0, & \text{其他} \end{cases}$$

$$z_{lijk} = \begin{cases} 1, & \text{工序 } j \text{ 在机器 } k \text{ 上工件 } l \text{ 先于 } i \text{ 加工} \\ 0, & \text{其他} \end{cases}$$

当  $r = 1$  时, 设

$$P_{\pi_{jk}(r-1)\pi_{jk}(r)jk}^{\text{set}} = 0$$

$$t_{\pi_{jk}(r-1)\pi_{jk}(r)jk}^{\text{set}} = 0$$

$$t_{\pi_{jk}(r-1)\pi_{jk}(r)jk}^{\text{idle}} = 0$$

模型中, 式(1)为模型的目标; 式(2)和(3)为目标函数; 约束(14)表示每个优先级位置只能对应 1 个工件; 约束(15)表示任一阶段每个工件只能由 1 台机器加工; 约束(16)表示每一工件在进行下一道工序前必须完成当前工序; 约束(17)表示每个工件在任一阶段上的加工完成时间小于其完工时间; 约束(18)表示同一阶段调度排列中排位越靠前的工件开始处理的时间越早; 约束(19)表示同一阶段分配在同一机器上的排位靠后的工件必须等排位靠前的工件加工完后才可进行加工。

## 2 改进型双目标差分进化算法

采用传统的解析方法并不能求解中大规模的调度问题. 目前, 国内外研究人员均基于启发式算法对此类调度问题进行研究. Storn 等<sup>[11]</sup>于 1997 年提出的一种非常有效的优化算法——差分进化(DE)算法, 已经被广泛地应用于各工程领域. 本文基于 DE 算法提出一种改进型双目标差分进化(MODE)算法, 使用非支配排序以及拥挤距离评价策略比较与排序可行解. 同时, 引入优质解挑战机制<sup>[12]</sup>以及混沌搜索策略以保证算法的全局搜索能力. 提前设置以下初始参数: 种群规模( $Q$ ); 解的寿命( $L$ ); 解的年龄( $A$ ); 最大循环次数( $Y$ ). MODE 算法的具体步骤如下.

**步骤 1 编码.** 根据模型特点, 采用双染色体形式表示可行解的编码. 编码的每一个元素包含一个工件在 1 个阶段中的机器指派情况和待加工次序. 例如, 一个 HFSP 中有 5 个待加工工件和 3 个阶段, 各阶段中的机器数分别为 2、1、3, 一个可行解的编码可写成:

$$\begin{bmatrix} (1,1) & (1,2) & (3,1) \\ (2,4) & (1,1) & (1,5) \\ (1,3) & (1,3) & (2,3) \\ (2,2) & (1,4) & (1,4) \\ (2,5) & (1,5) & (3,2) \end{bmatrix}_{5 \times 3}$$

编码首行的 3 个元素表示工件 1 分别在 3 个阶段的 1,1,3 号机器上加工,且在各机器上的排序分别是 1,2,1.

**步骤 2** 初始化阶段. 引入了 NEH (Nanaz, Enscore, Ham) 启发式方法<sup>[13]</sup>初始化 MODE 算法, 建立基于 NEH 和随机生成的混合方法产生初始种群. 首先, 利用 NEH 启发式方法构建一个基础可行解, 再通过成对交换方法生成  $Q/2$  个可行解, 剩余  $Q/2$  个可行解随机生成.

**步骤 3** 基于反向的搜索. 反向学习 (OBL) 是由 Tizhoosh<sup>[14]</sup>为机器智能的种群初始化提出的一种方法. 对初始种群  $Q_0$  反向学习后可以得到反向解集, 将该解集与  $Q_0$  合并后共同评估选择得到新解集. OBL 的应用能够有效地提高群体的多样性, 避免算法陷入局部最优.

定义反向点与基于反向的优化: 设  $D$  维空间中任意一个点  $X = (x_1, x_2, \dots, x_D)$ , 其中  $x_d \in [a_d, b_d]$ . 若  $X = (x_1, x_2, \dots, x_D)$  是最小化问题的一个可行解, 则其反向点  $\tilde{X} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_D)$  同样可行, 其中  $\tilde{x}_d = a_d + b_d - x_d$ . 设  $f(\cdot)$  为适应值函数, 若  $f(\tilde{X}) \leq f(X)$ , 则用  $\tilde{X}$  代替  $X$ . 该过程即为基于反向的优化.

初始种群  $Q_0$  在经过反向学习后得到其反向种群  $\tilde{Q}_0$ , 将其与  $Q_0$  合并后利用非支配排序和拥挤距离计算得到新种群及作为领导解集的最优 Pareto 前沿, 即所有非支配的 Pareto 解的集合. 其中, 以拥挤距离最小的解作为精英解.

**步骤 4** 差分变异. 使用 DE/current-to-best/1 策略以精英解针对新解集的每一个解  $x_i$  生成其变异解  $v_i$ . 变异解经过与原解的交叉操作得到试验解, 通过比较适应值决定是否替代原解. 差分变异过程完成后得到的解集将再次经过反向学习得到新解集, 与原解集合并后利用非支配排序和拥挤距离计算得到新种群及作为领导解集的最优 Pareto 前沿. 其中, 以拥挤距离最小的解作为精英解.

**步骤 5** 挑战机制. 为避免陷入局部最优解, 1 次循环内领导解集中的每个优质解都需要比较当前年龄值 (未更新迭代数) 和寿命, 判断其是否需要接受挑战. 一个解每经过 1 次差分变异或混沌搜索, 若

被更新, 则其对应年龄值置 0, 寿命重置为初始年龄; 否则, 其年龄值加 1. 一旦达到寿命限制, 启动对该解的挑战机制. 为了生成更具多样性和竞争力的挑战解, 每次该算法都将从以下两种邻域搜索策略中随机选取一种用于生成挑战解, 生成的挑战解分别为 GPBX- $\alpha$ <sup>[12]</sup> 和下式:

$$x_{cha,j} = \begin{cases} x_j^{\min} + \text{rand}(0,1)(x_j^{\max} - x_j^{\min}), & \text{rand}(0,1) \leq \text{CR} \\ x_{lea,j}, & \text{rand}(0,1) > \text{CR} \end{cases} \quad (20)$$

式中: CR 为差分进化的交叉率;  $x_{cha,j}$  为挑战解的第  $j$  个编码;  $x_{lea,j}$  为领导解的第  $j$  个编码;  $x_j^{\max}$ 、 $x_j^{\min}$  分别为当前解集第  $j$  个编码的最大值和最小值;  $\text{rand}(0,1)$  表示 0~1 之间的随机数.

与经过多代进化的“年老”优质解相比, 挑战解难以立即具备足够的竞争力, 故较难在挑战机制中获胜. 因此, 对失败的挑战解再进行一次差分进化操作给予第 2 次挑战机会. 挑战解一旦通过比较适应值战胜对应优质解, 则取代优质解并为挑战解设置初始年龄值和寿命; 若 2 次挑战均失败, 则优质解的年龄值减 1, 并允许其进入下一次循环.

**步骤 6** 混沌搜索. 在每次 MODE 算法循环的末尾引入混沌搜索, 以彻底探索解空间. 由于其独特的周期规律、固有的随机属性等特点, 混沌搜索能够有效地更改粒子的位置, 从而有助于提高算法的多样性.

利用下式将解向量连续化后计算每个维度对应的混沌数以构建混沌序列:

$$x_j^c(1) = \frac{x_j - x_j^{\min}}{x_j^{\max} - x_j^{\min}} \quad (21)$$

式中:  $x_j^c(1)$  表示所有解向量第  $j$  个编码的第 1 代混沌变量. 混沌数的迭代如下式:

$$x_j^c(t+1) = 4x_j^c(t)[1 - x_j^c(t)]$$

$$x_j^c(t) \in [0,1]$$

其中:  $t$  为混沌迭代数,  $t=1,2,\dots,10$ . 同样地, 混沌解若战胜对应原解, 则取代原解并为混沌解设置初始年龄值和寿命.

**步骤 7** 算法终止判断. 若算法迭代次数达到规定上限  $Y$ , 则输出当前的最优 Pareto 前沿及其适应值; 否则, 进入步骤 4.

### 3 数值实验与分析

所有算法都在 MATLAB(2016a) 环境下编程实现, 并在主频为 3.2 GHz, 内存为 8 GB、Intel(R) Core(TM) i5-4570 CPU 的 PC 机上进行数值实验, 实验分析和结果如下.

3.1 评价指标

对于多目标优化问题,仅将获得的最优 Pareto 前沿及算法运行时间(CPU)作为评价指标不足以评判算法的性能优劣.因此,为了更为有效地判断得到的最优 Pareto 解的性能,还将选用 Pareto 解的个数(NS)、世代距离(GD)、解间距(SP)作为比较各算法的性能指标.其中,GD 越小,表明算法收敛性越好;SP 越小,表明算法多样性越佳.

3.2 参数设置

测试算例中,有  $n=20,40,60$  共 3 个水平.每个水平各自对应 2 个阶段数和机器数,共有 12 个组合.由于研究考虑的是确定性的调度问题,所以每次计算的初始阶段都会随机生成完整的测试数据:工件加工时间的模糊数最可能值服从 $[10,120]$ 的均匀分布;模糊上下限取最可能值的 75%与 125%;加工功率服从 $[3,7]$ 的均匀分布;工件的换模时间服从 $[10,20]$ 的均匀分布;换模功率服从 $[2,6]$ 的均匀分布;机器的开机时间服从 $[5,10]$ 的均匀分布;关机时间服从 $[2,7]$ 的均匀分布;开机功率服从 $[2,6]$ 的均匀分布;关机功率服从 $[1,5]$ 的均匀分布;闲时功率服从 $[2,6]$ 的均匀分布;每个工件交货期的可能值表示为

$$t_i^{d_1} = \sum_{j=1}^s \frac{1}{m_j} \sum_{k=1}^{m_j} t_{ijk}^1$$
$$t_i^{d_2} = \frac{t_i^{d_1}}{2} + \frac{1}{2} \sum_{j=1}^s \frac{1}{m_j} \sum_{k=1}^{m_j} t_{ijk}^2$$
$$t_i^{d_3} = \frac{t_i^{d_1}}{2} + \frac{1}{2} \sum_{j=1}^s \frac{1}{m_j} \sum_{k=1}^{m_j} t_{ijk}^2$$
$$t_i^{d_4} = \sum_{j=1}^s \frac{1}{m_j} \sum_{k=1}^{m_j} t_{ijk}^3$$

将  $n=20, s=3, m=5$  的算例组合记为  $n20s3m5$ .数值实验中,将  $n=20,40,60$  的 3 类问题分别记为小、中、大规模问题.

3.3 数值实验分析

将 MODE 算法与带精英策略的非支配选择遗传算法(NSGA-II)、强度帕累托进化算法(SPEA2)

进行对比实验,以测试问题的求解性能.这两种经典算法在解决多目标问题上都具有较好的求解性能.由于此问题的真实 Pareto 前沿难以得到,故采用以下方法获得近似前沿:各算法独立运行多次并记录每次获得的 Pareto 解集,从 Pareto 解合集中集结新的 Pareto 解集作为近似前沿.

MODE 算法的参数设置对于其表现和效率有着重要的影响,实验采用田口方法确定各参数值,以保证算法的高效性.首先进行正交组合,然后测试各组参数以确定最优组合.据测试,当交叉率  $CR=0.7$ ,第 1 个权重因子  $W_1=0.6$ ,第 2 个权重因子  $W_2=0.4$ ,可行解的初始寿命  $L=8$  时,MODE 算法能以较高的质量对问题进行求解.MODE 算法以及 NSGA-II、SPEA2 算法的参数设置如表 1 所示.

由于算法的单次运行结果具有一定的随机性,所以分别用上述 3 种算法对各算例进行 20 次独立数值实验,结果如表 2 所示.

由表 2 可知,当问题规模较小时,SPEA2 与 NSGA-II 的解的个数、世代距离以及解间距性能指标都十分接近,略劣于 MODE 的相应性能指标.但是,在运算时间上,NSGA-II 以及 SPEA2 所需要的运行时间略优于 MODE;当问题规模逐渐变大时,MODE 相对于 NSGA-II、SPEA2 在解个数、世代距离和解间距方面的优势逐渐扩大,其相对较大的运算时间也在可接受范围内.其中,MODE 在 Pareto 解个数上的优势意味着有更多备选决策可供从业人员选择.

图 2 以  $n20s3m5, n60s7m15$  两个算例为例,表明了在小、大两种规模下各算法之间的 Pareto 前沿比较情况.由图 2(a)可知,当问题规模较小时,NSGA-II、SPEA2 求得的解集互有支配,并且都被 MODE 求得的解集所支配,但在部分位置差距较小;而随着问题规模的扩大,如图 2(b)所示,MODE 的 Pareto 解集基本支配了 NSGA-II 与 SPEA2 的 Pareto 解集,且分布更为均匀.数值实验对比结果证明了 MODE 算法的有效性.而 MODE 算法的优

表 1 不同算法的参数设置表  
Tab. 1 Parameter settings of different algorithms

算法	Q		Y		CR		变异率		$\omega_1/\omega_2$	
	$n=20$	$n=40,60$	$n=20$	$n=40,60$	$n=20$	$n=40,60$	$n=20$	$n=40,60$	$n=20$	$n=40,60$
NSGA-II	50	100	200	400	0.80	0.85	0.05	0.20	—	—
SPEA2	50	100	200	400	1.00	1.00	0.01	0.01	—	—
MODE	50	100	200	400	0.70	0.70	—	—	0.6/0.4	0.6/0.4

表 2 不同规模调度问题的数值实验结果  
Tab. 2 Experimental results for different instances

算例	SPEA2				NSGA-II				MODE			
	NS	GD	SP	CPU/s	NS	GD	SP	CPU/s	NS	GD	SP	CPU/s
<i>n</i> 20 <i>s</i> 3 <i>m</i> 5	10	280.15	25 106.31	1.48	11	278.26	17 316.12	1.50	12	239.74	15 134.91	2.26
<i>n</i> 20 <i>s</i> 3 <i>m</i> 10	12	286.03	23 379.99	1.51	9	252.37	16 105.22	1.49	16	243.92	16 005.92	2.26
<i>n</i> 20 <i>s</i> 4 <i>m</i> 5	8	233.78	28 113.68	2.92	12	237.49	21 011.43	1.92	14	268.54	19 091.24	4.22
<i>n</i> 20 <i>s</i> 4 <i>m</i> 10	10	236.43	26 171.82	3.06	11	254.64	19 913.73	1.75	17	255.60	18 993.27	3.99
<i>n</i> 40 <i>s</i> 5 <i>m</i> 10	13	269.21	39 805.21	16.63	14	284.06	37 901.62	14.12	16	238.84	35 901.62	36.15
<i>n</i> 40 <i>s</i> 5 <i>m</i> 15	14	208.79	38 704.43	17.15	12	247.07	38 544.51	14.30	18	218.37	36 544.51	38.60
<i>n</i> 40 <i>s</i> 6 <i>m</i> 10	13	253.23	42 017.35	18.01	13	296.95	40 051.45	15.14	17	227.18	37 951.54	40.17
<i>n</i> 40 <i>s</i> 6 <i>m</i> 15	12	211.34	39 567.39	17.93	15	279.16	39 132.52	15.12	18	234.93	36 023.26	41.52
<i>n</i> 60 <i>s</i> 7 <i>m</i> 15	16	297.11	55 424.34	30.19	19	252.46	53 633.87	26.01	24	222.51	48 633.87	86.14
<i>n</i> 60 <i>s</i> 7 <i>m</i> 20	19	304.25	53 663.87	31.04	18	371.62	52 713.75	25.90	22	215.66	48 512.21	85.84
<i>n</i> 60 <i>s</i> 8 <i>m</i> 15	17	260.87	58 975.26	33.28	22	282.52	56 645.23	26.93	28	200.64	49 945.89	89.37
<i>n</i> 60 <i>s</i> 8 <i>m</i> 20	20	232.92	57 174.19	32.96	19	296.37	55 364.11	27.05	26	190.87	50 414.19	90.26
均值	14	256.18	40 675.32	17.18	15	277.75	37 361.13	14.27	17	229.73	34 429.37	43.40

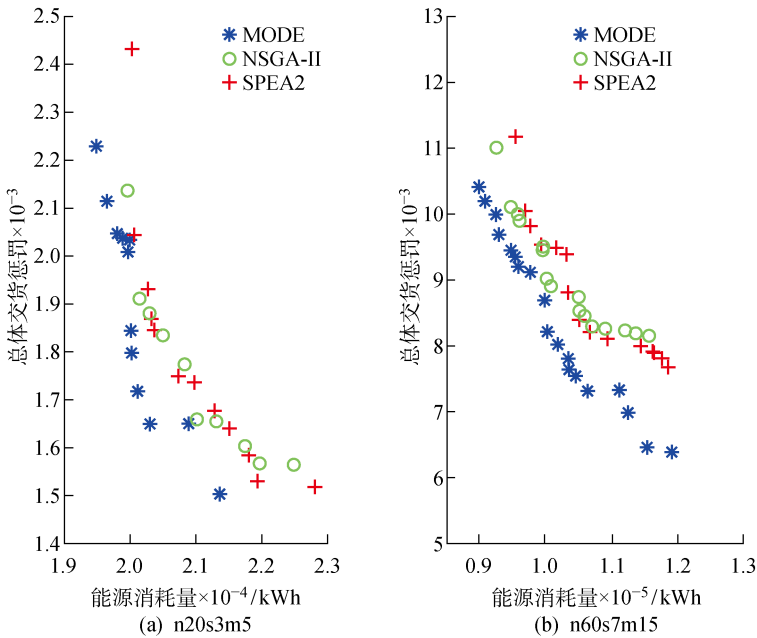


图 2 两个算例下的 Pareto 解集  
Fig. 2 Pareto solutions for two instances

秀求解能力应该归功于算法基于 DE 算法的多个改进之处,其中包括基于 NEH 算法得到优质初始解、优质差分进化策略带来更好的邻域挖掘能力、挑战优质解更易于跳出局部最优解、反向学习和混沌搜索策略保证算法的全局搜索能力。

4 结论

(1) 针对考虑并行机不相关、换模时间与工件加工次序相关两种约束的混合流水线系统,以保证准时交货和降低生产能耗为双调度目标,使用模糊数描述加工时间和交货期以处理系统的不确定性。

(2) 以差分进化算法为基础,引入 NEH 启发式方法构建高质量初始解,建立优质解挑战机制避免种群陷入局部最优,通过反向学习和混沌搜索探索全局解空间,提出改进双目标差分进化算法。

(3) 仿真实验验证 MODE 算法的有效性和较好的求解性能,丰富解决此类调度问题的理论方法。

(4) 未来的研究可在本文基础上考虑有限缓冲区,增加研究的复杂性和现实意义。

参考文献:

[1] 王凌,周刚,许烨,等. 求解不相关并行机混合流水

- 线调度问题的人工蜂群算法[J]. **控制理论与应用**, 2012, 29(12): 1551-1557.
- WANG Ling, ZHOU Gang, XU Ye, *et al.* An artificial bee colony algorithm for solving hybrid flow-shop scheduling problem with unrelated parallel machines [J]. **Control Theory & Applications**, 2012, 29(12): 1551-1557.
- [2] ABDULLAH S, ABDOLRAZZAGH-NEZHAD M. Fuzzy job-shop scheduling problems: A review[J]. **Information Sciences**, 2014, 278: 380-407.
- [3] HONG T P, WANG T T. A heuristic palmer-based fuzzy flexible flow-shop scheduling algorithm[C]// **Proceedings of 8th International Fuzzy Systems Conference**. Seoul, South Korea: IEEE, 1999: 1493-1497.
- [4] HONG T P, WANG T T, WANG S L. A palmer-based continuous fuzzy flexible flow-shop scheduling algorithm[J]. **Soft Computing**, 2001, 5(6): 426-433.
- [5] PINEDO M, HADAVI K. Operations research proceedings 1991[M]. Berlin: Springer, 1992: 35-42.
- [6] ZHOU B H, SHEN C Y. Multi-objective optimization of material delivery for mixed model assembly lines with energy consideration[J]. **Journal of Cleaner Production**, 2018, 192: 293-305.
- [7] 周炳海, 苏谊. 基于可变缓冲区存储量的串行生产线节能分析[J]. **哈尔滨工程大学学报**, 2016, 37(6): 832-836.
- ZHOU Binghai, SU Yi. Energy-saving analysis of serial production lines based on the changeable buffers' storage[J]. **Journal of Harbin Engineering University**, 2016, 37(6): 832-836.
- [8] ZHANG R, CHIONG R. Solving the energy-efficient job shop scheduling problem: A multi-objective genetic algorithm with enhanced local search for minimizing the total weighted tardiness and total energy consumption [J]. **Journal of Cleaner Production**, 2016, 112: 3361-3375.
- [9] DAI M, TANG D B, GIRET A, *et al.* Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm[J]. **Robotics and Computer-Integrated Manufacturing**, 2013, 29(5): 418-429.
- [10] LUO H, DU B, HUANG G Q, *et al.* Hybrid flow shop scheduling considering machine electricity consumption cost[J]. **International Journal of Production Economics**, 2013, 146(2): 423-439.
- [11] STORN R, PRICE K. Differential evolution——A simple and efficient heuristic for global optimization over continuous spaces[J]. **Journal of Global Optimization**, 1997, 11(4): 341-359.
- [12] FU C M, JIANG C, CHEN G S, *et al.* An adaptive differential evolution algorithm with an aging leader and challengers mechanism[J]. **Applied Soft Computing**, 2017, 57: 60-73.
- [13] NAWAZ M, ENSCORE E E, HAM I. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem[J]. **Omega**, 1983, 11(1): 91-95.
- [14] TIZHOOSH H R. Opposition-based learning: A new scheme for machine intelligence [C]// **Computational Intelligence for Modelling, Control and Automation/International Conference on Intelligent Agents, Web Technologies and International Commerce**. Vienna, Austria: IEEE, 2005: 695-701.

(本文编辑:石易文)