

# 基于多目标差分进化算法的机加工柔性作业车间调度

程 强<sup>1,2</sup>, 高元杰<sup>1,2</sup>, 初红艳<sup>1,2</sup>, 张彩霞<sup>1,2</sup>, 刘志峰<sup>1,3</sup>

(1. 北京工业大学先进制造与智能技术研究所, 北京 100124;

2. 北京工业大学先进制造技术北京市重点实验室, 北京 100124;

3. 北京工业大学机械工业重型机床数字化设计与测试技术重点实验室, 北京 100124)

**摘要:** 柔性作业车间的多品种、多件数导致调度难度大、耗费时间较长和成本较高, 为此, 以最大完工时间、能耗和刀具损耗数量为优化目标, 考虑返工、次序的准备时间和批量调度约束, 建立了多目标机加工柔性作业车间调度模型. 提出一种引入综合考虑能耗和加工时间的轮盘赌初始化策略. 针对传统差分进化算法交叉得到的子代机器部分质量较差, 提出一种机器选择的策略, 以此对差分进化算法进行了改进. 将改进后的差分进化算法应用于机加工柔性作业车间调度, 并与传统差分进化算法在机加工柔性作业车间调度进行多组实验对比. 结果表明: 改进差分进化在机加工柔性作业车间调度较传统差分进化算法具有收敛速度较快、鲁棒性较好的优点, 优化后各机器负载更为均衡, 可有效解决多目标机加工柔性作业车间调度问题, 为多品种、多件数类排产任务提供了一种良好的指导方案.

**关键词:** 柔性作业车间; 差分进化算法; 多目标优化; 机加工; 批量调度; 返工

中图分类号: TH 117. 1; TH 132. 1

文献标志码: A

文章编号: 0254 - 0037(2023)03 - 0335 - 11

doi: 10. 11936/bjtxb2021010018

## Flexible Job Shop Scheduling of Machining Based on Multi-objective Differential Evolution Algorithm

CHENG Qiang<sup>1,2</sup>, GAO Yuanjie<sup>1,2</sup>, CHU Hongyan<sup>1,2</sup>, ZHANG Caixia<sup>1,2</sup>, LIU Zhifeng<sup>1,3</sup>

(1. Institute of Advanced Manufacturing and Intelligent Technology, Beijing University of Technology, Beijing 100124, China;

2. Beijing Key Laboratory of Advanced Manufacturing Technology, Beijing University of Technology, Beijing 100124, China;

3. Machinery Industry Key Laboratory of Heavy Machine Tool Digital Design and Testing Technology, Beijing University of Technology, Beijing 100124, China)

**Abstract:** Flexible job shop scheduling is difficult, time-consuming and cost-effective due to its variety and number of pieces. In this paper, a multi-objective flexible job shop scheduling model for machining was established with the optimization objectives of maximum completion time, energy consumption and tool loss, considering rework, preparation time of sequence and batch scheduling constraints. A multi-objective flexible job shop scheduling model considering energy consumption and tool loss was proposed, and a machine selection strategy was proposed to improve the differential evolution algorithm. The improved differential evolution algorithm was applied to machining flexible job shop scheduling. Compared with the traditional differential evolution algorithm, the improved differential evolution algorithm has the advantages of faster convergence speed and better robustness, and the optimized machine load is more balanced, which can effectively solve the multi-objective machining flexible job

收稿日期: 2021-01-06; 修回日期: 2021-05-20

基金项目: 北京市科技计划资助项目(Z201100006720008)

作者简介: 程 强(1979—), 男, 教授, 博士生导师, 主要从事智能产线规划与调度、机床精度设计、产品设计方法学方面的研究, E-mail: chengqiang@bjut.edu.cn

shop scheduling problem, and provide a good guidance for multi variety and multi piece scheduling tasks plan.

**Key words:** flexible job shop; differential evolution algorithm; multi-objective optimization; machining; batch scheduling; rework

调度研究的问题简单来说就是如何将稀缺资源分配,即将资源给在一定时间内的不同任务.调度在大多数制造和生产系统及信息处理环境中扮演着重要的角色<sup>[1]</sup>.目前关于调度的研究主要涉及模型建立和算法设计,其中问题建模主要研究调度模型、调度规则、目标函数等,而调度算法设计主要研究算法复杂性、算法收敛性、算法质量等<sup>[1-6]</sup>.

目前,关于柔性作业车间的模型调度建立国内外都有相关的研究. Willian<sup>[7]</sup>作为调度理论的奠基人,提出了车间调度,开创了调度问题的理论研究工作.针对多品种、多件数的工件调度难的问题, Low<sup>[8]</sup>和 Chinyao等<sup>[9]</sup>在柔性作业车间模型的基础上考虑了分批调度,建立了柔性作业车间分批处理模型,基于其数学模型的研究表明:在作业车间中,通过分批处理可减少机床闲置时间和工件流动时间. Jeong等<sup>[10]</sup>在分批调度的基础上进一步建立单工艺路线生产车间动态批量分批作业计划,将生产辅助时间(setup times)和加工时间(process times)进行了区分.与此同时,一种考虑返工的数学模型也开始被提出.陈建国等<sup>[11]</sup>针对传统Job-Shop数学模型忽略返工及重加工的因素,构建了考虑该情形下的Job-Shop调度数学模型及相应的求解算法.廖怡娜等<sup>[12]</sup>建立了考虑作业返工的资源受限项目调度问题的数学模型,针对该模型,设计了面向3种不同情况的修复算法. Ceylan等<sup>[13]</sup>针对多阶段供应链网络的特点,将多阶段供应约束考虑到模型中,建立了双目标混合整数线性规划模型,提出了一个新的协调调度.目前调度模型研究主要围绕着设置约束条件和优化目标等进行研究,主要包括返工、批量、资源受限等约束条件和能耗、最大完工时间和最大延长时间等优化目标,但是大部分研究关于约束条件的设定较为简单,譬如文献[11]将返工率视为固定值,文献[10]的准备时间不受工序次序影响等.而且在机加工中,刀具的成本往往也占很大的比重,且刀具也属于易耗品,但是关于以刀具作为调度优化目标的研究还较少,因此本文在前人研究的基础上,将机加工的常见约束如批量约束、基于贝叶斯条件概率求返工概率和受次序影响的准备时间等考虑至模型中,并结合机加工中的刀具损耗特点进

行建模,提出一种新的机加工柔性车间调度批量模型,使模型更加符合实际生产需求,最终提出一种以最大完工时间、能耗和刀具损耗数量为优化目标,考虑返工、序列的准备时间和以批量作为约束的多目标机加工柔性作业车间调度模型,并采取差分进化算法进行求解.

1997年,差分进化(differential evolution, DE)<sup>[14]</sup>算法由 Storn 提出,其初衷是解决切比雪夫多项式问题.作为一种基于群体导向的随机搜索技术,DE算法包括初始化、变异、交叉以及选择等操作<sup>[15]</sup>.由于算法原理较为简单、参数较少、鲁棒性较好并且易于实现,因此已有部分研究应用于车间调度问题<sup>[16-17]</sup>中.传统差分进化算法在解决本文提出的多目标机加工柔性作业车间调度模型时,由于随机初始化导致算法求解时间较长,在短期内不能得到较好的调度结果,同时由于生产条件的复杂性,机器差分运算得到的差分向量较差,因此需要进行一定的改进.目前针对DE算法的理论研究主要集中在如何提高算法的寻优能力、收敛速度以及克服启发式算法常见的早熟收敛以及搜索停滞等缺陷方面<sup>[15]</sup>,本文从模型特点出发,提出一种初始化策略和机器选择策略分别对差分进化算法的收敛速度和解的质量进行一定程度上的改进.

## 1 多目标机加工柔性作业车间调度模型

多目标机加工柔性作业车间调度可定义为:有 $N$ 类工件在 $M$ 台机器加工,每类工件包含若干道工序,每道工序再分成若干个任意大小的批次,工序可在多台不同性能的机器上进行作业操作,由于机器性能的不同,工序在不同机器上的加工时间是不同的,同时,能耗和刀具磨损也是不同的.因此工序在不同的机器上加工,调度指标除了最大加工时间外,还有能耗指标和刀具寿命指标.多目标机加工柔性作业车间调度所要解决的问题是:

- 1) 确定各个工件划分批次数量及各批次大小.
- 2) 确定各子批在哪台机器设备上加工.
- 3) 确定最优的工序加工顺序.
- 4) 在多约束条件下求得较优多目标调度方案.

## 1.1 变量定义

为了方便后续模型和算法描述,定义了下列相关变量用于描述数学模型。

$N$ :工件类别数。

$M$ :机器总数。

$H$ :工序总数。

$\Omega$ :总的机器集。

$i$ :工件类别号, $i=1,2,\dots,N$ 。

$k$ :机器序号, $k=1,2,\dots,M$ 。

$j$ :工序序号, $j=1,2,\dots,H$ 。

$n_i$ :每类工件的件数, $i \in \{1,2,\dots,N\}$ 。

$B_i$ :每类工件的批数, $i \in \{1,2,\dots,N\}$ 。

$l_{i,b}$ :每类工件的子批的工件数, $b \in \{1,2,\dots,B_i\}$ 。

$t_{i,l_i,b,j,S}$ : $i$ 类工件的 $l_{i,b}$ 子批 $j$ 工序起始加工时间。

$t_{i,l_i,b,j,E}$ : $i$ 类工件的 $l_{i,b}$ 子批 $j$ 工序终止加工时间。

$t_{i,j,k,P}$ : $i$ 类工件 $j$ 工序在机器 $k$ 上的加工时间。

$t_{i,j,k,R}$ :考虑返工的 $i$ 类工件的 $j$ 工序在机器 $k$ 上的加工时间。

$r_{i,j,k}$ : $i$ 类工件的 $j$ 工序在机器 $k$ 上的加工不合格返工概率。

$X_{i,l_i,b,j,k}$ :如果 $i$ 类工件的 $l_{i,b}$ 子批的 $j$ 工序在机器 $k$ 加工则为1,否则为0。

$t_{k,C}$ :机器 $k$ 上最后一个子批工序的完工时间, $k=1,2,\dots,M$ 。

$t_{Cmax}$ :最大完工时间。

## 1.2 数学模型

模型以优化最大完工时间、能耗和刀具损坏数为优化目标,首先建立批量约束。

每类工件分批后,该类工件的每个子批的工件数量之和为该类工件的总件数。

$$\sum_{b=1}^{B_i} l_{i,b} = n_i \quad (1)$$

根据 $X_{i,l_i,b,j,k}$ 的含义可知,每类工件的子批的每道工序同一时刻只能在一台设备上加工。

$$\sum_{k=1}^M X_{i,l_i,b,j,k} = 1 \quad (2)$$

各子批在加工时存在工序先后顺序约束, $t_{setup}$ 为加工前准备时间,取决于上一工件和目前工件的类别。

$$t_{i,l_i,b,j+1,S} \geq t_{i,l_i,b,j,E} + t_{i,j,k,R} \times l_{i,b} \quad (3)$$

考虑到加工过程中不合格返工现象常有发生,将返工这一动态问题转换为静态问题进行处理,在

已知 $t_{i,j,k,P}$ 的基础上为返工预留加工时间

$$t_{i,j,k,R} = t_{i,j,k,P} + r_{i,j,k} \times \omega_p \quad (4)$$

式中 $\omega_p$ 是由MES系统中的历史数据所得,表示考虑不同概率下加权的返工修复时长。

通过统计可以常见地求得加工条件不合理的先验概率: $P(X=A)$ 、 $P(X=B)$ 、 $P(X=C)$ 。其中 $X$ 表示加工条件不合理的随机事件, $A$ 、 $B$ 、 $C$ 分别表示人操作不规范、毛坯不合格和机床加工性能不达标事件。

进一步地,经过统计可以获得在加工条件不合理时,工件不合格的条件概率: $P(Y=不合格|X=A)$ 、 $P(Y=不合格|X=B)$ 和 $P(Y=不合格|X=C)$ 。最后根据统计的先验概率和条件概率,计算当工件出现不合格时,各种加工条件不合理情况发生的后验概率,并基于后验概率和各种加工条件不合理的情况下的平均修复时长,通过概率加权设计预留的返工时长,其中 $t_{R,person}$ 、 $t_{R,workblank}$ 、 $t_{R,machine}$ 表示统计所得在各种不合理条件下进行修复的平均时长。

$$\begin{aligned} \omega_p = & P(X=A|Y=不合格) \times t_{R,person} + \\ & P(X=B|Y=不合格) \times t_{R,workblank} + \\ & P(X=C|Y=不合格) \times t_{R,machine} \end{aligned} \quad (5)$$

最后让 $t_{i,j,k,R}$ 与子批 $l_{i,b}$ 工件数量相乘,表示 $i$ 类工件的子批 $l_{i,b}$ 的 $j$ 工序在机器 $k$ 上的加工时长。

因为工件在机器的准备时间取决于上一工件,因此考虑带有工件次序的准备时间, $t_{Matrix}$ 为考虑序列准备时间矩阵,其中行与列表示工件和工件之间如果在机器上相邻时,切换工作环境的准备时间。

$$t_{Matrix} = \begin{bmatrix} 0 & t_{Matrix1,2} & \cdots & t_{Matrix1,n-1} & t_{Matrix1,n} \\ t_{Matrix2,1} & 0 & & t_{Matrix2,n-1} & t_{Matrix2,n} \\ \vdots & & \ddots & & \vdots \\ t_{Matrixn-1,1} & t_{Matrixn-1,2} & & 0 & t_{Matrixn-1,n} \\ t_{Matrixn,1} & t_{Matrixn,2} & \cdots & t_{Matrixn,n-1} & 0 \end{bmatrix} \quad (6)$$

最大完工时间取决于机器最后一个子批完工时间的最大值,即

$$t_{Cmax} = \max(t_1, t_2, \dots, t_M) \quad (7)$$

以最后一个子批完工时间为优化目标: $\min(t_{max})$ ,同时为考虑机床机加工过程中的能耗,建立了模型<sup>[18]</sup>

$$E_k = P_{k,f} t_{k,f} + P_{k,air} t_{k,air} + P_{k,v} t_{k,v} \quad (8)$$

式中: $P_{k,f}$ 、 $P_{k,air}$ 和 $P_{k,v}$ 分别表示机床 $k$ 待机功率、空

载功率以及切削功率;而 $t_f$ 、 $t_{air}$ 和 $t_v$ 为其对应的时间。

此机加工能耗模型,主要由3个部分构成,分别是待机能耗、空载能耗以及切削能耗。

$P_{k,air}$ 与主轴转速相关,且

$$P_{k,air} = An^2 + Bn + C \quad (9)$$

式中 $A$ 、 $B$ 、 $C$ 取决于机床特性和主轴转速。类似地, $P_{k,t}$ 和 $P_{k,v}$ 也取决于机床特性,可通过实验观测求得。

因此,可以得到

$$E_{total} = \sum_{k=1}^M E_k \quad (10)$$

以能耗作为优化目标: $\min(E_{total})$ ,同时应注意,在切削速度增大的同时,依据文献[18]给出的模型,可以推测得出切削速度增大尽管能缩短加工时长,但是却会一定程度上增加能耗的结论。

由于在机加工过程中,机床在给定的工艺参数下,刀具寿命是固定的,以铣削的方式进行加工的刀具寿命<sup>[19]</sup>

$$T_{i,j,k} = C \times V_c^{b1} \times f_z^{b2} \times a_p^{b3} \times a_e^{b4} \quad (11)$$

式中 $T_{i,j,k}$ 表示 $i$ 类工件的 $j$ 工序在机器 $k$ 以确定的工艺参数加工下刀具的寿命。工艺参数中 $V_c$ 为切削速度, $f_z$ 为每齿进给量, $a_p$ 为轴向切深, $a_e$ 为径向切深。

在文献[19]中,所求得的 $b1$ 、 $b2$ 、 $b3$ 、 $b4$ 分别为 $-2.9298$ 、 $-1.9128$ 、 $-1.5119$ 和 $-1.1413$ ,这表明切削速度对刀具寿命影响非常显著,实际经验表明切削速度越大,切削区温度越高,刀具磨损程度越大,刀具寿命越短,这与所求得的公式是一致的。

考虑到刀具也有成本,通过计算在此工艺下子批的累计加工时长,让累计加工时长除以该工艺下的刀具寿命,可大概估算出预计耗费刀数

$$N_{i,j,k,tool} = \sum_{b=1}^{B_i} t_{i,j,k,R} \times l_{i,b} \times X_{i,l_{i,b},j,k} / T_{i,j,k} \quad (12)$$

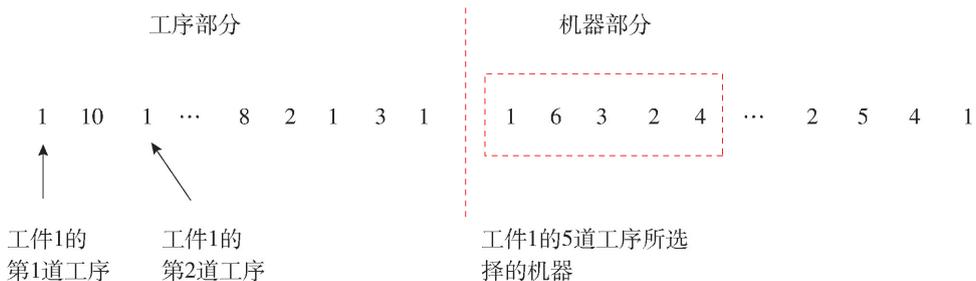


图1 编码图

Fig. 1 Coding diagram

式中 $N_{i,j,k,tool}$ 为所有 $i$ 类工件的子批的 $j$ 工序在机器 $k$ 上加工累计耗费的刀具数,通过对每个设备实际加工时间进行累加求和,然后将累加求和的时间除以刀具预估寿命即可求得。

因此考虑所有子批的刀具磨损情况,以刀具损坏数量为优化目标

$$\min \left( \sum_{i=1}^N \sum_{j=1}^H \sum_{k=1}^M N_{i,j,k,tool} \right)$$

其余假设条件如下。

- 1) 已知每类工件总批量数,划分子批后每批次的工件在一起加工。
- 2) 同种工件每个子批的每道工序可以选择不同设备进行加工。
- 3) 每台设备在同一时刻只能加工某类工件某批次的一道工序。
- 4) 某工序一旦于设备上开始加工,不允许中断。

## 2 差分进化算法设计

考虑到传统差分进化算法在本文调度模型的应用上效果不佳,为得到较优解,在传统差分进化算法的基础上进行一定的改进,主要对初始化策略和差分操作进行改进。

### 2.1 编码方式与解码方式

本文采取以工序为基准的编码方式,以10个工件5道工序6个机器为例,总工序为 $10 \times 5 = 50$ 道工序。为50道工序分配机器,因此染色体中还需要为50个工序预留50个机器位置,染色体总长度为100,为方便解码,设定染色体前半部分代表的是工序,后半部分代表的是加工该工序对应的机器,其中染色体机器位置部分第1~5个位置表示工件1的第1道工序到第5道工序所选择的机器,第6~10个位置表示工件2的第1道工序到第5道工序所选择的机器,以此类推,如图1所示。

工件的个数计算方式为

$$N_{\text{workpiece}} = \sum_{i=1}^N l_{i,b} \quad (13)$$

表示将每个子批作为一个工件视为调度,因此确定好如何分批和每批的批数问题也是本文关注点之一. 考虑到当使用等分批不能使结果更加优化时,才考虑用其他分批策略<sup>[20]</sup>,因此采用等分批进行分批,为确定批数,根据经验所得划分的批次判定公式<sup>[21]</sup>进行批数确定.

$$T_z = \sum_{j=1}^H t_j + K_z(l-1)t_L \quad (14)$$

式中:  $T_z$  为考虑分批的时间参数,即最小加工时间;  $H$  为工序数;  $l$  为批数;  $t_j$  为某工件第  $j$  工序的加工时间;  $t_L$  为最长工序的加工时间;  $K_z$  为子批中包含的工件数.

分批算法的基本流程如下.

- 1) 初始化参数,令  $x = 1, n = 1, x$  表示工件类别,  $n$  表示批数.
- 2) 将  $x$  分成  $n$  批,按式(14)求得  $T_z$ ,即此类工件分批时间参数.
- 3) 若  $n < n_x$  ( $x$  类工件数量),  $n++$ .
- 4) 计算批数为  $1, 2, \dots, n_x$  批的时间参数  $T_z$ .
- 5) 从 4) 计算得到的一组  $T_z$  中找到最小的  $T_z$ , 存储并输出此时的批数  $n$ .
- 6) 若  $x < N, N$  为工件种类数,则  $x++$ ,  $n = 1$ , 转 2), 否则算法结束.

解码时将染色体中的工序部分编码和机器部分编码分别与加工信息表一一对应,如表 1 所示,可获取每类工件的每道工序的加工时长和能耗信息,并将其代入各个适应度函数进行求解,进行解码.

表 1 5 种类-4 工序-6 机器加工信息

Table 1 5 types-4 processes-6 machining information

min-kW·h

工件	工序	机器1	机器 2	机器 3	机器 4	机器 5	机器 6
种类 1	1	52-10. 13	43-17. 29		42-15. 34	36-17. 53	
	2	13-3. 53		7-6. 037	11-4. 63		16-2. 96
	3		21-8. 45	17-12. 80		18-8. 76	21-4. 96
	4	8-1. 43	6-3. 06	4-3. 95	10-1. 04		15-0. 81
种类 2	1	54-10. 13		36-26. 32		39-16. 59	
	2	16-2. 73	8-7. 29		9-5. 34	7-9. 53	11-3. 14
	3		22-8. 86	18-13. 55	21-7. 69		24-5. 21
	4	9-1. 67		10-1. 32	7-2. 42		11-1. 14
种类 3	1		48-19. 29		46-16. 81	39-16. 91	47-15. 44
	2	14-2. 73	11-5. 86	15-2. 32		9-7. 53	
	3	29-15. 63		19-20. 01	23-18. 45	31-14. 53	27-16. 74
	4	9-3. 13	13-1. 29			16-1. 03	11-2. 14
种类 4	1		42-28. 29	39-36. 32	52-13. 24	41-30. 53	
	2	17-1. 73		15-2. 32		10-7. 13	13-4. 24
	3	33-19. 69	43-17. 39			36-18. 77	40-18. 48
	4	8-1. 39		4-3. 81	10-1. 12	14-0. 92	
种类 5	1		45-19. 18	36-27. 38		34-29. 18	43-18. 29
	2			15-2. 58	19-1. 53	9-6. 74	11-3. 57
	3	30-19. 69	32-18. 29	-	42-15. 54	46-11. 99	
	4	9-1. 49		13-1. 09	15-0. 84		10-1. 24

## 2.2 多目标适应度函数设计

在实际生产中,调度常常需要满足多目标的任务需求,为此,需要将多个目标进行综合考虑,本文考虑各目标的单位成本生成多目标适应度函数

$$F_{\text{total}} = C_{\text{tool}} \times N_{\text{tool}} + C_{\text{energy}} \times E_{\text{total}} + C_{\text{time}} \times t_{\text{Cmax}} \quad (15)$$

式中  $C_{\text{tool}}$ 、 $C_{\text{energy}}$  和  $C_{\text{time}}$  分别为对应目标的单位成本,可根据企业生产的实际情况获取,而  $N_{\text{tool}}$ 、 $E_{\text{total}}$  和  $t_{\text{Cmax}}$  可分别根据式(12)(10)和(7)求得。

## 2.3 初始化种群

传统差分进化算法中,使用随机初始化对种群进行初始化,种群的全局性虽然能够得到保证,但是个体质量较差,将导致迭代次数过多才能获得较优个体,为保证全局性,同时一定程度上提高初始种群的质量,本节在编码的工序部分采用随机初始化,而机器部分考虑到模型的多目标特点,提出一种综合考虑能耗和加工时间的轮盘赌初始化策略,这种初始化策略缩短了差分进化算法的运行时间,一定程度上改进了差分进化算法。

首先确定好种群个体的数量,之后,对每个个体进行以下初始化操作,初始化流程如下。

### 1) 工序初始化

根据工件分批的批数,确定工件数量,根据每工件的工序数量确定工序总数,例如工件共有 15 件,工序 4 道,生成 4 个 1、4 个 2……4 个 15,然后将生成 60 道工序乱序处理,其中第 1 个 1 表示工件 1 的第 1 道工序,第 2 个 1 表示工件 1 第 2 道工序,以此类推,其余工件类似。

### 2) 机器初始化

#### ① 开始循环,令 $i = 1$ 。

② 循环读取已生成的工序部分,获取每工序的所有可用设备的加工时长和能耗,基于轮盘赌规则生成相应的数据矩阵。

③ 生成随机数,根据随机数在该工序对应染色体的机器位置查轮盘赌数据矩阵所对应的机器部分,为该位置赋值找到的机器。例如,某工件的工序 1 有 3 个机器可以选择,机器 1、机器 2、机器 3 对应的加工时间分别为 6、3、2 h,而能耗分别为 12、5、2 kW·h,此时基于自定义的轮盘赌规则,考虑模型中的优化目标能耗和加工时长,假设以 0.75 表示考虑能耗的权重,0.25 表示考虑时长的权重,则该工序选择机器 1 进行加工的概率区间为

$$P_1 = [0 \sim (6 \div 11 \times 0.75 + 12 \div 19 \times 0.25)]$$

类似可得到关于机器 2、机器 3 的轮盘赌概率

区间  $[0.567, 0.838]$ 、 $[0.838, 1.000]$ ,可以得到轮盘赌矩阵  $(0.567, 1)$ 、 $(0.838, 2)$ 、 $(1, 3)$ 。如若生成随机数 0.80,根据矩阵,随机数在 0.567 和 0.838 之间,根据轮盘赌矩阵,此时使用机器 2 加工该工序。

## 2.4 差分进化

为得到较优解,通过差分进化算法对初始化的种群进行更新,差分进化包括变异、交叉及选择,算法流程如下。

**步骤 1** 算法初始化,生成最大迭代次数  $\max_{\text{gen}}$ 、变异因子  $F$  和交叉因子  $C_r$ 。

**步骤 2** 开始循环,令  $i = 1$ 。

**步骤 3** 随机选择 3 个个体  $X_1$ 、 $X_2$ 、 $X_3$ ,并生成随机数  $\text{rand}$ ,对  $X_1$ 、 $X_2$  进行差分操作:首先生成随机数,如果随机数小于变异因子,则进行变异操作,子代向量为变异操作后的向量个体,否则子代向量为父代个体  $X_1$ 。

$$V = F \otimes G(X_1, X_2) = \begin{cases} G(X_1, X_2), & \text{rand} < F \\ X_1, & \text{其他} \end{cases} \quad (16)$$

式中  $G(X_1, X_2)$  表示父代个体彼此之间进行交叉,由于涉及工序和机器 2 个子问题,该差分操作较为复杂,具体如下。

1) 关于编码中的工序部分交叉采用优先交叉 (POX),如图 2 所示: $P_1$  和  $P_2$  为父代的 2 个个体, $C_1$  和  $C_2$  为子代 2 个个体, $P_1$  和  $P_2$  将若干工件置入一个集合中,之后  $C_1$  和  $C_2$  继承集合中的工件在父代的位置和值,并且将父代  $P_1$  的剩余部分按照原顺序置入  $C_2$ ,将父代  $P_2$  的剩余部分按照原顺序置入  $C_1$ 。

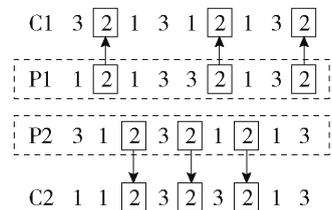


图2 POX交叉过程

Fig.2 Pox crossover process

2) 关于编码中的机器部分交叉如果采用 POX 由于部分机器无法加工某道工序(见表 1)可能导致出现不可行解,传统差分进化算法虽然在这方面设计了一种交叉方式,但是这种交叉方式得到的子代机器部分的信息继承有着一定的缺陷.因此本文在传统差分进化算法中的机器交叉的基础上进行一定的改进,操作流程如下。

以常见的一种传统交叉进化算法中机器部分交叉方法为例, 首先遍历每个机器位置, 例如遍历到机器位置 1, 生成随机数, 如果随机数小于等于  $F$ , 将  $X_1$  的机器位置 1 的值赋到子代机器位置 1 上, 否则, 将  $X_2$  的机器位置 1 的值赋到子代机器位置 1 上, 对子代的每个机器位置操作都类似. 考虑到  $X_1$  和  $X_2$  的机器部分的编码都是合理的, 因此不会产生不可行解, 但是这样的交叉方式存在一定的问题, 子代基因的继承是从  $X_1$  和  $X_2$  轮番按位继承, 优秀段的信息可能无法继承, 除非连续产生的随机数都小于交叉因子, 否则子代继承的基因很可能就是  $X_1$  的小片段, 然后是  $X_2$  的小片段, 之后又是  $X_1$  的小片段, 这样的处理仅能增加新个体的多样性, 子代的质量却无从保证.

因此本节在传统差分进化算法机器交叉后得到的子代的基础上再添加一个处理方案, 即对子代的机器位置再进行一次循环遍历, 设定机器位置的变异因子, 为每个机器位置设计一种机器选择策略, 提高了差分得到的向量的质量, 一定程度上改进了差分进化算法, 具体操作流程如下.

① 设定机器变异因子  $M_r$ , 循环遍历子代个体的机器编码.

② 遍历机器位置  $h$  时,  $h \geq 2$ , 生成随机数, 如果随机数小于  $M_r$ , 对该位置进行变异, 变异时统计之前各位置机器  $k$  出现的数量  $M_k$ , 该位置变异取决于之前位置各机器的累计数量. 例如遍历到某位置, 此时生成随机数小于变异因子  $M_r$ , 则变异时选择该位置为某机器  $k$  的概率为

$$P_k = 1 - \frac{M_k}{\sum_{k=1}^{h-1} M_k} \quad (17)$$

式中:  $M_k$  该机器在之前位置出现的次数;  $\sum_{k=1}^{h-1} M_k$  表示之前位置的所有机器数, 根据该式, 如果某机器在之前位置出现次数过多, 变异时在该位置再出现它的次数将降低. 至此,  $G(X_1, X_2)$  完成, 得到差分操作后的变异向量  $\gamma$ .

**步骤 4** 将变异个体和父代个体  $X_3$ , 进行交叉

$$U = C_r \otimes G(\gamma, X_3) = \begin{cases} G(\gamma, X_3), & \text{rand} < F \\ \gamma, & \text{其他} \end{cases} \quad (18)$$

差分操作同步步骤 3, 差分操作后得到试验个体  $U$ .

**步骤 5** 将试验个体  $U$  与当前种群中某个体进行比较, 如果该试验个体的适应度较好则将该个体暂存

至临时种群中, 待此次迭代完成后, 采用轮盘赌方法从最终种群挑取和最初种群个体数量一致的个体进入下一轮迭代<sup>[4]</sup>.

**步骤 6**  $i++$ , 判断  $i$  是否满足最大迭代次数要求, 否跳到步骤 3, 是则退出程序, 最优个体从此时的种群选出.

### 2.5 小结

本节提出了一种结合模型多目标特点的初始化方法, 包括工序初始化方法和机器初始化方法, 其中机器初始化方面结合模型多目标特点, 设定多权值综合考虑多目标进行机器初始化. 并且提出一种机器选择策略用于改进差分进化算法, 即在满足设定的机器变异条件时, 如何对传统差分进化算法得到的子代个体的机器部分进行一次优化. 下面就改进差分进化算法与传统差分进化算法进行算例验证.

## 3 算例验证分析

为验证算法的实用性, 对某加工厂的 5 类工件进行批量调度, 5 类工件的数量分别为 33、45、36、60、26, 根据式 (15), 求得等分批的最好方案为类 1 工件等分 3 批, 类 2 工件等分 3 批, 类 3 工件等分 3 批, 类 4 工件等分 4 批, 类 5 工件等分 2 批, 将每个子批视为一个工件进行调度, 因此共有  $3+3+3+4+2=15$  个工件, 每工件 4 道工序, 共 60 道工序. 每类工件的单件加工信息见表 1, 表中 52-10.13 表示加工时长和能耗, 其中 52 表示加工时长为 52 min, 能耗为 10.13 kW·h. 考虑序列的准备时间表如表 2 所示.

表 2 考虑上一道工序的准备时间

Table 2 Information of setup time considering the previous process min

工件种类	种类 1	种类 2	种类 3	种类 4	种类 5
种类 1	0	12	16	12	9
种类 2	14	0	11	13	12
种类 3	13	12	0	17	11
种类 4	12	13	14	0	10
种类 5	10	12	11	16	0

为验证算法的性能, 在 CPU 型号为 Intel(R) Core(TM) i5-9300H CPU @ 2.40 GHz、计算机内存为 8 GB 的笔记本电脑上以 3.7.0 版本的 Python 语言进行算法测试. 在参数选择上, 考虑到变异因子  $F$  过大会导致算法的收敛速度变慢, 而过小会导致

种群的多样性降低,出现早熟,因此选择变异因子  $F=0.5$ . 交叉因子  $C_r$  越大交叉概率就越大,一般选择  $C_r=0.1$ . 机器变异因子不宜太大,否则减缓收敛速度,本文中  $M_r=0.05$ . 对比迭代次数为 500、700 和 1 000 次时,种群最优个体的适应度函数值收敛情况,如图 3 所示,分别代表迭代次数为 500、700 和

1 000 次时,最优个体的适应度函数值的收敛情况. 通过观察可以发现迭代次数为 500 时,3 张图收敛效果较好,超过 500 次后,700 次迭代图和 1 000 次迭代图的继续收敛的程度不明显,因此取迭代次数为 500 进行算法对比.

综上所述,本文以最大迭代次数 500 次,变异因

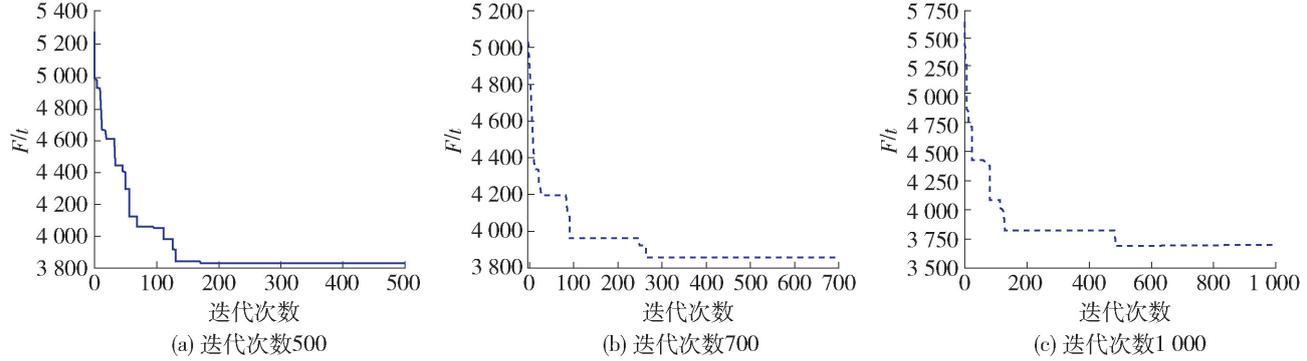


图 3 500、700、1 000 次迭代最优个体适应度值变化曲线

Fig. 3 Variation curve of optimal individual fitness value for 500, 700 and 1 000 iterations

子  $F=0.5$ , 交叉因子  $C_r=0.1$ , 机器变异因子  $M_r=0.05$ , 种群个体数量为 50 作为算法参数. 将改进差分进化算法对算例的调度结果和传统差分进化算法对算例的调度结果进行对比. 表 3 为 5 组实验下,传统差分进化算法调度求解的各目标的调度结果,表 4 为 5 组实验下,改进差分进化算法调度求解的各目标的调度结果. 从整体效果上来看,改进差分

进化算法求解耗时平均为 81.01 s, 超过传统差分进化算法 6 s, 6 s 的运行时间增加主要来自于初始化策略和机器选择策略的引入. 但是优化后差分进化算法调度求解所得的最优个体最大完工时间、能耗、刀具磨损件数较传统算法分别优化了 11.64%、14.93% 和 3 件,极大地优化了最大完工时长、能耗和刀具磨损这 3 个目标.

表 3 传统差分进化算法调度结果

Table 3 Scheduling results of traditional differential evolution algorithm

	实验 1	实验 2	实验 3	实验 4	实验 5	均值
最大完工时长/min	4 154	4 085	4 237	4 212	4 321	4 201
能耗/(kW·h)	10 243	10 355	11 931	10 912	10 836	10 855
刀具磨损数/件	33	34	37	35	34	35
运行时间/s	75.62	76.33	71.25	73.36	78.53	75.01

表 4 改进差分进化算法调度结果

Table 4 Scheduling results of improved differential evolution algorithm

	实验 1	实验 2	实验 3	实验 4	实验 5	均值
最大完工时长/min	3 726	3 805	3 800	3 815	3 511	3 732
能耗/(kW·h)	8 805	9 341	9 172	10 212	8 640	9 234
刀具磨损数/件	31	32	32	33	30	32
运行时间/s	79.83	86.48	80.16	78.52	80.06	81.01

图 4(a) 为传统差分进化算法 500 次迭代过程最优个体最大完工时间,图 4(b) 为改进差分进化算

法 500 次迭代过程最优个体最大完工时间. 通过对传统差分进化算法和改进的差分进化算法的 500

次迭代种群中最优个体的情况,可发现在初始迭代时,改进的差分进化算法所得种群中最优个体的适应度值较低. 5组实验中,改进差分进化算法初始化时所得最优个体的最大完工时间5组实验均值较传统差分进化算法初始化时所得最优个体的最大完工时间5组实验均值低100 min左右,且达到500次迭代次数时,改进差分进化算法所得最优个体最大完工时间的收敛也低于传统差分进化算法所得最优个体300 min. 这表明初始化策略在算法收敛上有较为明显的作用.

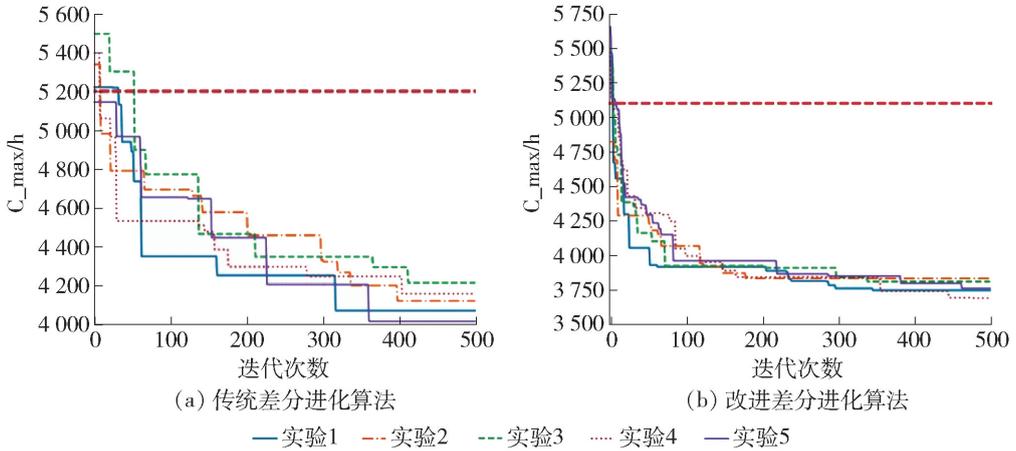


图4 500次迭代过程中最优个体最大完工时间  
Fig. 4 Maximum completion time of the optimal individual in 500 iterations

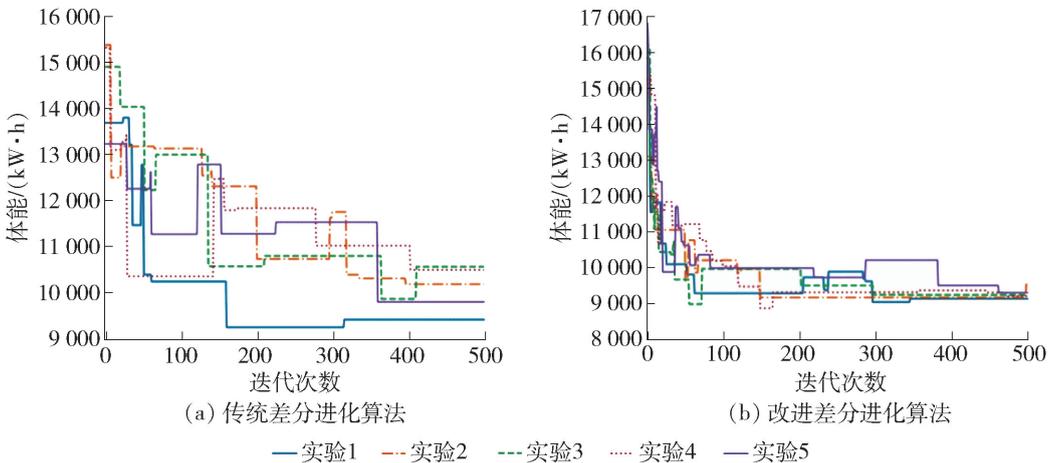


图5 500次迭代过程中最优个体能耗  
Fig. 5 Optimal individual energy consumption in 500 iterations

图6(a)为传统差分进化算法所得最优个体设备累计加工时间,从左到右,设备累计加工时间(单位:min)为3 107、4 102、3 212、4 034、3 783、3 751. 图6(b)为改进差分进化算法所得最优个体设备累计加工时间,从左到右,设备累计加工时间(单位:min)为3 223、3 362、3 342、2 994、3 201、3 402. 同样

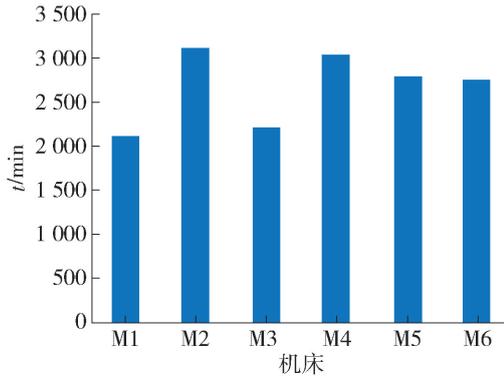
图5是在采取机器选择策略后,改进差分进化算法能耗和机器加工时间,其中图5(a)为传统差分进化算法500次迭代过程最优个体能耗图,图5(b)为改进差分进化算法500次迭代过程最优个体能耗图. 可以发现,未采用机器选择策略的传统差分进化算法其最优个体能耗图各组之间上下限区间较大,震荡性较为明显,而改进差分进化算法收敛较为平滑,各组之间上下限区间较小,达到500次迭代时能耗小于传统差分进化算法,优化值为500 kW·h,一定程度上表明初始化策略在本调度模型的有效性.

可以发现传统差分进化算法进行机器部分编码的交叉时只是单纯地增加机器编码的全局性而未考虑机器编码部分的质量,极易出现负载不均的现象,而考虑机器选择策略的改进差分进化算法的设备工作时间较为均衡. 最终获得一个可用的调度结果染色体:[工序

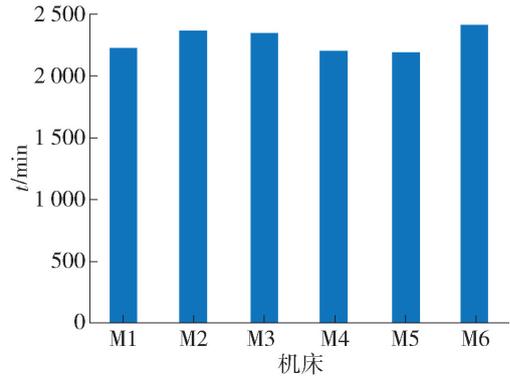
部分:1 5 2 3 13 1 7 14 5 6 11 4 12 15 9 5 7 2 4 14 5  
 1 15 12 8 3 13 1 10 11 12 4 12 3 9 11 10 8 10 3 9 11  
 8 14 13 6 14 15 4 6 7 10 9 8 2 2 7 15 6 13;机器部分:  
 6 4 2 4 5 2 1 4 5 4 1 5 2 3 4 4 2 1 1 6 5 2 4 2 4 4 4 3

2 5 1 3 4 6 3 3 2 5 1 3 6 3 6 5 1 3 6 3 5 2 4 6 3 1 3 6  
 2 5 3 1].

其调度甘特图如图7所示,其最大完工时间为3 511 min,能耗为8 640 kW·h,预估损坏刀数30把.



(a) 传统差分进化算法



(b) 改进差分进化算法

图6 最优个体设备累计加工时间

Fig. 6 Cumulative processing time of optimal individual equipment

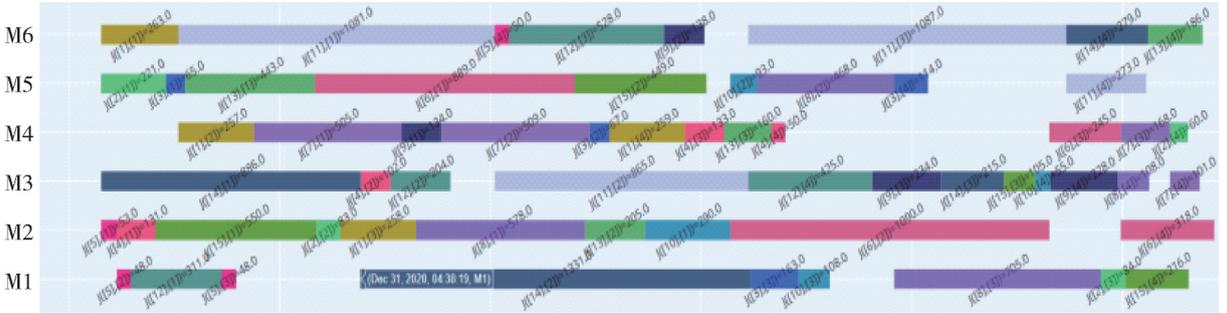


图7 最优调度方案

Fig. 7 Optimal scheduling scheme

### 4 结论

本文针对机加工柔性作业车间工件品种和批量较多、车间环境较为复杂导致的调度成本过高、机器设备负载不均衡的问题进行研究,取得了以下结论.

1) 以最大完工时间、能耗和刀具损耗数量为优化目标,以返工、批量调度和考虑序列的准备时间作为约束建立了多目标机加工柔性作业车间调度模型,该模型能有效反映实际多目标机加工柔性作业车间生产特点.

2) 设计一种改进差分进化算法,通过实际算例测试,分别分析改进差分进化算法和传统差分进化算法对多目标机加工柔性作业车间调度模型求解结果,发现基于改进差分进化算法的多目标机加工柔性作业车间调度结果较优,有一定的实用价值.

### 参考文献:

[1] MICHAEL P. 调度:原理、算法和系统[M]. 张智海,

译. 北京:清华大学出版社,2007.

[2] 王凌. 车间调度及其遗传算法[M]. 北京:清华大学出版社,2003.  
 [3] 雷德明. 现代制造系统智能调度技术及其应用[M]. 北京:中国电力出版社,2011.  
 [4] 王万良,吴启迪. 生产调度智能算法及其应用[M]. 北京:科学出版社,2007.  
 [5] 柳毅. 企业生产调度的智能优化方法[M]. 北京:人民邮电出版社,2008.  
 [6] WANG L, ZHENG D Z. An effective hybrid heuristic for flow shop scheduling [J]. International Journal of Advanced Manufacturing Technology, 2003, 21: 38-44.  
 [7] WILLIAN L. Theory of scheduling [M]. New York: Dover Publications INC, 2003.  
 [8] LOW C. An approach of lot streaming in job shop production system[J]. Journal of the Chinese Institute of Industrial Engineers, 1999, 22(5): 671-680.  
 [9] CHINYAO L, JINN Y Y, HSU J L. Effect evaluation of lot splitting in job shop scheduling [J]. Journal of the

- Chinese Institute of Industrial Engineers, 2004, 24 (9/10): 773-780.
- [10] JEONG H, PARK J, LEACHMAN R C. A batch splitting method for a job shop scheduling problem in an MRP environment [J]. *International Journal of Production Research*, 1999, 37(15): 3583-3598.
- [11] 陈建国, 舒辉, 余平祥. 随机返工及重加工情形下的 Job-Shop 调度问题[J]. *工业工程*, 2015, 18(5): 127-133, 147.  
CHEN J G, SHU H, YU P X. Job shop scheduling problem with random rework and reprocessing [J]. *Industrial Engineering*, 2015, 18(5): 127-133, 147. (in Chinese)
- [12] 廖怡娜, 陆志强. 考虑作业返工的资源受限项目调度问题[J]. *计算机集成制造系统*, 2019, 25(11): 2762-2774.  
LIAO Y N, LU Z Q. Resource constrained project scheduling problem with job rework [J]. *Computer Integrated Manufacturing System*, 2019, 25(11): 2762-2774. (in Chinese)
- [13] CEYLAN Z, TOZHAN H, BULKAN S. A coordinated scheduling problem for the supply chain in a flexible job shop machine environment [J]. *Operational Research*, 2021, 21(2): 1-26.
- [14] STORN R. Differential evolution—a simple and efficient heuristic for global optimization over continuous space [J]. *Journal of Global Optimization*, 1997, 11(4): 341-359.
- [15] 丁青锋, 尹晓宇. 差分进化算法综述[J]. *智能系统学报*, 2017, 12(4): 431-442.  
DING Q F, YIN X Y. Review of differential evolution algorithm [J]. *Journal of Intelligent Systems*, 2017, 12(4): 431-442. (in Chinese)
- [16] 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.
- [17] WANG H Y, ZHAO Y W, XU X L, et al. Scheduling batch and continuous process production based on an improved differential evolution algorithm [J]. *Systems Engineering—Theory & Practice Online*, 2009, 29(11): 157-167.
- [18] 徐立云, 邓伟, 高翔宇, 等. 机加工过程一类能耗估算模型[J]. *同济大学学报(自然科版)*, 2015, 43(9): 1367-1372.  
XU L Y, DENG W, GAO X Y, et al. A class of energy consumption estimation model for machining process [J]. *Journal of Tongji University (Natural Science Edition)*, 2015, 43(9): 1367-1372. (in Chinese)
- [19] 王晓琴, 艾兴, 赵军, 等. 涂层刀具铣削 Ti6Al4V 刀具寿命及切削参数优化[J]. *武汉理工大学学报*, 2008, 189(10): 109-112.  
WANG X Q, AI X, ZHAO J, et al. Tool life and cutting parameter optimization of coated tool milling Ti6Al4V [J]. *Journal of Wuhan University of Technology*, 2008, 189(10): 109-112. (in Chinese)
- [20] 黄刚, 庞志国, 胡振华. 带准备时间的作业车间分批调度算法[J]. *工业工程*, 2007(6): 91-95, 121.  
HUANG G, PANG Z G, HU Z H. Job shop batch scheduling algorithm with preparation time [J]. *Industrial Engineering*, 2007(6): 91-95, 121. (in Chinese)
- [21] 李风光, 初红艳, 李鹏, 等. 运用混合优化算法的作业车间分批调度[J]. *现代制造工程*, 2010(8): 51-54.  
LI F G, CHU H Y, LI P, et al. Job shop batch scheduling using hybrid optimization algorithm [J]. *Modern Manufacturing Engineering*, 2010(8): 51-54.

(责任编辑 杨开英)