doi:10.3772/j.issn.1002-0470.2015.01.010

可重入混合流水车间负荷平衡排产优化问题研究①

韩忠华②*** 董晓婷③* 史海波**

(*沈阳建筑大学信息与控制工程学院 沈阳 110168)

(**中国科学院沈阳自动化研究所 沈阳 110016)

摘 要 为了解决可重入混合流水车间(RHFS)负荷平衡调度问题,建立了RHFS负荷平衡优化问题数学规划模型,将工位加工时间负荷平衡代价和总工位等待时间加权求和后作为负荷平衡综合评价指标;设计了基于工件加工流程的编码方法并结合时间窗约束与最大剩余时间规则进行解码,采用动态自适应差分进化(DSADE)算法进行全局优化。DSADE算法根据个体间汉明距离判断个体相似度,动态更新具有高相似性的个体,以增加种群多样性,并引入随停止代数自适应调整进化参数的策略,以增强跃出局部极值,持续进化的能力。基于客车制造中涂装车间多遍彩条工序段的实例数据将DSADE算法与已有遗传算法(GA)、差分进化(DE)算法、自适应差分进化(SADE)算法进行仿真比较,比较结果表明,DSADE算法的负荷平衡评价指标平均降低幅度超过20%。

关键词 可重入混合流水车间(RHFS),负荷平衡,差分进化(DE)算法,个体相似度,涂装车间

0 引 言

可重入混合流水车间(reentrant hybrid flowshop, RHFS)调度问题是以半导体封装、客车制造为代表的一类生产调度问题。可重入混合流水车间(RHFS)生产与传统混合流水车间(hybrid flowshop, HFS)生产一样有着固定的工艺流程,其显著特点是工件的不同工序可能重复访问同一设备或设备组,从而造成产品加工流程中出现大量的重入加工流[1,2]。HFS具有多任务、多工序、多并行机特点,RHFS多存在复杂的串并混合型可重入问题。RHFS受可重入工序的制约,加工完的工件需多次回到排产任务队列等待工位和加工顺序的再次分配,使得生产系统的不稳定性与不确定性增大,各并行工位加工任务的统计也愈加困难,再加上HFS并行机加工能力与不同加工任务工艺差异的影响,常出现各并行机之间负荷严重不平衡现象,即部分设备空闲或过载情况频发,造成设备利用率降低,生产周期延长。因此,研究可重入混合流水车间的负荷平衡调度问题具有重要的学术意义与工程应用价值。

自1983年Graves提出RHFS调度问题以来[3],RHFS调度研究已取得较大进展。Yalaoui等针对具有多批次带可重入流的HFSP,提出了一种基于模糊逻辑控制器的粒子群算法优化最小拖期工件数[4]。Hekmatfar研究了只含两道工序,而且第一道工序是具有可重入性质的HFSP,提出了一种以最小makespan为优化目标的混合遗传算法(genetic algorithm, GA)[5]。刘小华等利用粒子群算法和遗传算法的互补性,设计了一种两者结合的混合算法优化

① 国家重大科技专项(2011ZX02601-005)和辽宁省教育厅(L2013237)资助项目。

② 男,1977年生,博士,教授;研究方向:生产与运作管理,企业自动化系统集成技术,车间排产与生产调度算法的工程应用研究;E-mail:xiaozhonghua1977@163.cn

③ 通讯作者,E-mail:dxt199211@163.com

(收稿日期:2014-09-10)

可重入生产调度系统[6]。从已有 RHFS 调度研究文献看,目前 RHFS 调度研究普遍集中在最小 makespan 与交货期问题上,研究方法主要采用群体智能算法。因此,本研究依托具有典型可重入混合流水车间特点的客车涂装车间多遍彩条工序段问题,建立同时考虑工位加工时间和工位等待时间的负荷平衡综合评价指标,并将其作为适应度函数,而且采用具有较好全局搜索能力的动态自适应差分进化(dynamic self-adaptive differential evolution, DSADE)算法作为全局优化方法深入研究可重入混合流水车间的负荷平衡调度问题。

1 RHFS 负荷平衡调度问题描述与数学模型

在混合流水车间中 n 个工件加工队列进行 m 个工序的加工,工件依照加工流程中工序顺序进行排产,工件的加工流程中至少有一个工序是可重入工序,m 个工序中至少有一个工序包含多个并行工位,工件在并行工位上的加工时间可以不同,但工件在并行工位上至少要选择一个工位进行加工,排产结果要确定工件的工位分配、加工顺序以及在其加工流程上每个工序的开工时间、完工时间。

1.1 模型参数

考虑到客车生产加工工时由车型和加工流程共同决定,增加了对工件类型的描述。模型参数包括:

n,表示加工的工件总数;

J_i,表示第 i 个工件,$i \in \{1,\cdots,n\}$;

m,表示排产的工序总数;

OP_j,表示第 j 个工序,$j \in \{1,\cdots,m\}$;

M_j,表示工序 OP_j 的最大并行工位数;

$WS_{j,k}$,表示工序 OP_j 的第 k 个工位,$k \in \{1,\cdots,M_j\}$;

$flsn$,表示加工流程的总数;

FL_q,表示工件的加工流程,是生产过程中工件顺序经过加工工序的集合,重复加工的工序会重复出现在加工流程中,$q \in \{1,\cdots,flsn\}$;

om_q,表示加工流程 FL_q 中工件经过加工工序的总数,对重复加工的工序进行累计计数,$om_q \geqslant m$;

t_q,表示加工流程 FL_q 中加工工序的执行顺序的序号,$t_q \in \{1,\cdots,om_q\}$;

$OP_j^{t_q}$,表示工序 OP_j 是在加工流程 FL_q 中第 t_q 个执行的加工工序;

$bmsn$,表示工件类型的总数;

BM_b,表示 HFS 中能够生产的工件类型,在客车制造行业中表示制造车间能够生产的客车车型,$b \in \{1,\cdots,bmsn\}$。

fl_i,表示给工件 J_i 指定的加工流程,$fl_i \subseteq \{FL_1,\cdots,FL_q,\cdots,FL_{flsn}\}$;

bm_i,表示工件 J_i 自身的类型,$bm_i \subseteq \{BM_1,\cdots,BM_b,\cdots,BM_{bmsn}\}$;

om_i,表示工件 J_i 的加工流程 fl_i 中加工工序的总数。

t_i,表示工件 J_i 的加工流程 fl_i 中加工工序的执行顺序的序号,$t_i \in \{1,\cdots,om_i\}$;

$S_{i,j,k}^{t_i}$,表示工件 J_i,其流程 fl_i 中第 t_i 个执行的工序 OP_j 在工位 $WS_{j,k}$ 上的加工开始时间;

$C_{i,j,k}^{t_i}$,表示工件 J_i,其流程 fl_i 中第 t_i 个执行的工序 OP_j 在工位 $WS_{j,k}$ 上的加工结束时间;

$Tw_{i,j,k}^{t_i}$,表示 bm_i 类型的工件 J_i,其流程 fl_i 中第 t_i 个执行的工序 OP_j 中,在工位 $WS_{j,k}$ 上的加工时间;

$n_{j,k}$,表示工序 OP_j 的工位 $WS_{j,k}$ 上的加工工件个数。

1.2 假设变量和基本约束关系

假设变量 $At_{i,j,k}^{t_i}$ 表示工件是否在其流程 fl_i 第 t_i 个执行的工序 OP_j 中,在工位 $WS_{j,k}$ 上的加工状态。

$$card\{J_i \mid i \in \{1,\cdots,n\}, fl_i \equiv FL_q, q \in \{1,\cdots,flsn\}\} \geqslant 1 \quad (1)$$

$$card\{J_i \mid i \in \{1,\cdots,n\}, bm_i \equiv BM_b, b \in \{1,\cdots,bmsn\}\} \geqslant 1 \quad (2)$$

$$FL_q = \{OP_j^{t_q} \mid j \in \{1,\cdots,m\}, t_q \in \{1,\cdots,OM_q\}\} \quad (3)$$

$$C_{i,j,k}^{t_i} = S_{i,j,k}^{t_i} + Tw_{i,j,k}^{t_i},\ i \in \{1,2,\cdots,n\},\ j \in \{1,2,\cdots,m\},\ t_i \in \{1,\cdots,om_i\} \quad (4)$$

$$C_{i,j_1,k}^{t_i} \leqslant S_{i,j_2,k}^{t_i+1},\ i \in \{1,2,\cdots,n\}\ j_1, j_2 \in \{1,2,\cdots,m\},\ t_i \in \{1,\cdots,om_i - 1\} \quad (5)$$

$$n_{j,k} = \sum_{i=1}^{n} \sum_{t_i=1}^{om_i} At_{i,j,k}^{t_i} \tag{6}$$

式(1)和式(2)说明排产流程和工件类型与工件是一对多的关系,在排产过程包括的工件中,可能会存在多个工件采用同样加工流程或是多个工件具有同样的类型。式(3)说明在加工流程 FL_q 中可以包含多个重复加工的工序(通过加工顺序序号区分);式(4)说明在具有可重入工序情况下,工件在其加工流程中开始加工时间、加工时间和结束加工时间之间的关系;式(5)说明同一工件在其加工流程中连续加工工序的开始加工时间和结束加工时间的之间关系;式(6)说明在具有可重入工序的车间中,统计工位上加工工件的个数,要累计重复加工工件的数量。其它约束条件如下:每个工位在同一时刻只能加工一个工件,每个工件在同一时刻只能在一个工位加工;工件加工过程不允许中断;涂装车间中设有大量缓冲工位,在排产过程中不考虑缓冲工位限制;现场承载客车的滑壳平移车是人工操作的,时间难以统计,所以把转运时间计入加工时间。

1.3 时间窗约束模型

对可重入混合流水车间负荷平衡调度问题模型进行调整,加入时间窗约束,使每个工序在固定的时间节点开始工作,有利于均匀分配加工任务,更加均衡各并行机所承受的负载,提高设备利用率[7]。每个工序 OP_j 的开始加工时间为 Twb_j,Pt_j 表示两个顺序相邻的工序 OP_{j-1} 和工序 OP_j 的开始加工时间的时间间隔,公式如下:

$$Pt_j = Twb_j - Twb_{j-1}, \quad j \in \{2,\cdots,m\} \tag{7}$$

$$Twb_j \leqslant \min\{S_{i,j,k}^{t_i}\} \tag{8}$$

式(8)是工序 OP_j 的开始加工时间 Twb_j 与工件 J_i 的开工时间 Tlb 的约束关系,表示在工序 OP_j 所有工件的开工时间 Tlb 都要大于等于工序 OP_j 开始加工时间 Twb_j。Pt_j 还用下式表示:

$$Pt_j = \begin{cases} \overline{\left(\sum_{i=0}^{n} \sum_{k=1}^{M_j} Tw_{i,j,k}^{t_i} \times \left\lceil \frac{M_j}{M_{j-1}} \right\rceil\right)} & M_j > M_{j-1} \\ \overline{\sum_{i=0}^{n} \sum_{k=1}^{M_j} Tw_{i,j,k}^{t_i}} & M_j \leqslant M_{j-1} \end{cases} \tag{9}$$

如式(9)所示,如果当前工序 OP_{j-1} 的工位数 M_{j-1} 大于等于紧后工序 OP_j 的工位数 M_j,则 Pt_j 等于在当前工序全部加工任务的工时的均值;如果当前工序 OP_{j-1} 的工位数 M_{j-1} 小于等于紧后工序的并行工位数 M_j,则 Pt_j 等于相邻两个工序的工位数比值取整加 1,再乘上当前工序加工时间的均值。

2 RHFS 负荷平衡调度数学模型

各并行工位有效加工时间是否均衡是车间负荷平衡的主要判断依据,同时,工位上加工任务之间的等待时间体现车间运作效率,所以在研究 HFFS 负荷平衡调度问题时,需要综合考虑工位加工时间和工位加工等待时间,两者重要程度通过权值进行体现。

2.1 建立基于工位加工时间的负荷平衡评价指标

将车间中每一个工位 $WS_{j,k}$ 上的总加工时间

$$Ts_{j,k} = \sum_{i=1}^{n} \sum_{t_i=1}^{on_i} (Tw_{i,j,k}^{t_i} \times At_{i,j,k}^{t_i}) \tag{10}$$

与其所处工序的并行工位加工时间平均值 $\overline{Tw_j}$ 的差值进行平方后再求和,建立基于工位加工时间的负荷平衡代价 Nlb。

在式(10)中 $Ts_{j,k}$ 表示在工序 OP_j 的工位 $WS_{j,k}$ 上的加工时间之合。其中 $\sum_{t_i=1}^{on_i} (Tw_{i,j,k}^{t_i} * At_{i,j,k}^{t_i})$ 表示工件 J_i 根据其加工流程多次在工序 OP_j 的工位 $WS_{j,k}$ 上进行加工的加工时间之和。

在工序 OP_j 的 M_j 个并行机上的平均加工工时 $\overline{Tw_j}$ 用下式表示:

$$\overline{Tw_j} = \left(\frac{\sum_{k=1}^{M_j} \sum_{i=1}^{n} \sum_{t_i=1}^{on_i} (Tw_{i,j,k}^{t_i} * At_{i,j,k}^{t_i})}{M_j}\right) = \left(\frac{\sum_{k=1}^{M_j} Ts_{j,k}}{M_j}\right) \tag{11}$$

基于工位加工时间的负荷平衡代价 Nlb 用下式表示:

$$Nlb = \sum_{j=1}^{m} \left(\sqrt{\sum_{k=1}^{M_j} ((Ts_{j,k} - \overline{Tw_j})^2)}\right) \tag{12}$$

Nlb 越小说明并行工位负荷分配越平衡。

2.2 总工位加工等待时间

通过对工位在加工过程中等待时间

$$Tms_{j,k} = \begin{cases} (\max\{C^{t_i}_{i,j,k} \times At^{t_i}_{i,j,k}\} - \min\{S^{t_i}_{i,j,k} \times At^{t_i}_{i,j,k}\}) \\ \quad - \sum_{i=1}^{n} \sum_{t_i=1}^{on_i} (Tw^{t_i}_{i,j,k} \times At^{t_i}_{i,j,k}), & n_{j,k} \geqslant 2 \\ 0, & n_{j,k} < 2 \end{cases} \tag{13}$$

进行求和得到工位加工等待时间 Twt，作为负荷平衡问题的辅助评价指标。

在式(13)中 $Tms_{j,k}$ 表示的是工位 $WS_{j,k}$ 上每两个连续加工工件之间的工位等待时间之合。$\max\{C^{t_i}_{i,j,k} \times At^{t_i}_{i,j,k}\}$ 表示在工位 $WS_{j,k}$ 上进行加工工件中的最后一次加工的完工时间，$\min\{S^{t_i}_{i,j,k} \times At^{t_i}_{i,j,k}\}$ 表示在工位 $WS_{j,k}$ 上进行加工的工件中最早一次加工的开工时间，二者的差表示工位 $WS_{j,k}$ 上加工过程的时间跨度，再减去该工位有效加工时间 $\sum_{i=1}^{n} \sum_{t_i=1}^{on_i} (Tw^{t_i}_{i,j,k} \times At^{t_i}_{i,j,k})$，得到工位 $WS_{j,k}$ 的加工等待时间之和 $Tms_{j,k}$。

混合流水车间中所有工位等待时间之和 Twt 用下式表示：

$$Twt = \sum_{j=1}^{m} \left(\sum_{k=1}^{M_j} Tms_{j,k} \right) \tag{14}$$

2.3 RHFS 负荷平衡综合评价指标

混合流水车间的综合负荷平衡评价指标是将基于工位加工时间的负荷平衡代价 Nlb 和工位加工等待时间 Twt 进行加权得到总负荷平衡代价 f_{LB}，它反映了混合流水车间调度结果要达到负荷平衡所需的代价。在加权前使用公式

$$f_{Nlb} = \begin{cases} \dfrac{Nlb - Nlb_{\min}}{Nlb_{\max} - Nlb_{\min}}, & Nlb > 0 \\ 0, & Nlb = 0 \end{cases} \tag{15}$$

$$f_{wt} = \begin{cases} \dfrac{Twt - Twt_{\min}}{Twt_{\max} - Twt_{\min}} & Twt > 0 \\ 0 & Twt = 0 \end{cases} \tag{16}$$

进行归一化处理，f_{Nlb} 表示基于工位加工时间负荷平衡代价 Nlb 进行归一化处理后的数值，f_{wt} 表示工位加工等待时间 Twt 进行归一化处理后的数值。引入 α_1 和 α_2 权值，满足条件 $\alpha_1 + \alpha_2 = 1$，进一步建立公式

$$f_{LB} = \alpha_1 \cdot f_{Nlb} + \alpha_2 \cdot f_{wt} \tag{17}$$

作为负荷平衡代价。f_{LB} 为混合流水车间加权总负荷平衡代价。根据式(17)，混合流水车间负荷平衡调度问题的优化目标是 $\min f_{LB}$。通过归一化处理使得两个代价的数值在一个数量级上，有利于更好地控制两者在负荷平衡优化中的作用，其中 $Nlb_{\min}$ 和 $Nlb_{\max}$ 分别表示初始种群中负荷不平衡代价的最小值和最大值，$Twt_{\min}$ 和 $Twt_{\min}$ 分别表示初始种群中工位加工等待时间的最小值和最大值。

3 动态自适应差分进化算法的设计

动态自适应差分进化(DSADE)算法在经典差分进化(differential evolution，DE)算法[8]的基础上，采用了参数自适应策略与种群动态更新机制，以巩固算法的进化活力与种群个体的多样性，进一步提高算法的搜素精度与全局搜索能力。

3.1 编码和解码

种群中每一个个体对应问题的一个解，为了反映可重入工序的特征，个体中基因编码依托工件的加工流程，即每个种子由多段编码构成，每段编码表述的是每个工件 J_i 在加工过程中所走过的加工流程 fl_i。种群的基因 a_{i,t_i} 表示的是一个工件 J_i 在其流程中排 t_i 位置的工序 OP_j 选择加工的工位 $WS_{j,k}$，$j \in \{1,2,\cdots,m\}$。a_{i,t_i} 的取值为区间 $(1, M_j + 1)$ 上的一个随机数，将包含基因 a_{i,t_i} 的个体作为混合流水车间排产优化解时，使用的是 $\lfloor a_{i,t_i} \rfloor$ 的值，“$\lfloor\ \rfloor$”表示取下整数，即表示出工件 J_i 在工序 OP_j 选择加工的工位。每个工件 J_i 依据加工流程 FL_i 构成一个基因编码段，代表了加工过程中指定加工工位的数列，多个基因片段合并到一起构成一个个体 X_{np}。$X_{np} = \{a_{1,1}, a_{1,2}, \cdots, a_{i,on_i}, \cdots, a_{n,on_n}\}$，$np \in \{1,2,\cdots,NP\}$，表示出全部工件经过加工流程的工位分配情况。若出现 $\lfloor a_{i,ti} \rfloor = \lfloor a_{k,ti} \rfloor (i \neq k)$，表示多个工件的同一道工序在同一个工位加工，则根据剩余加工时间最大规则确定其加工顺序，即把每个工件剩余的待加工工时求和，最大者优先加工。

一个个体 X_{np} 中包含的基因个数 N_{Gene} 是全部工件的流程中工序数之和，即 $N_{Gene} = card(X_{np}) =$

$\sum_{i=1}^{n} om_i$。这种种群个体构造方法是基于矩阵的编码方法而来[9]，巧妙地依据加工流程中包含的工序进行编码，以向量序列代替矩阵表示染色体，克服了基于矩阵的编码方法受矩阵位置信息与元素的限制，无法描述加工流程中包含可重入工序的缺陷。

3.2 种群初始化

进化(DE)算法通常采用随机方法产生初始种群中的解，本文同样采用随机初始化产生种群，以保证初始种群的分散性。

3.3 动态更新机制

在染色体编码设计中，汉明距离常被用来描述两个染色体间的不相似程度。基于染色体编码的两个个体间的汉明距离是所有对应位置不同基因的个数[10]。汉明距离越大，相似性越低。当种群进化到一定代数后，种群中相似个体增多，进化活力降低，此时引进种群动态更新机制能有效改善解空间中个体的分布性，以增强算法的全局搜索性能。具体操作步骤如下：

步骤 1：将所有个体按照适应度进行排序。

步骤 2：从最优个体开始，依照式

$$SI_{i,t_i} = \begin{cases} 0 & \lfloor a_{i,t_i} \rfloor \neq \lfloor a'_{i,t_i} \rfloor \\ 1 & \lfloor a_{i,t_i} \rfloor = \lfloor a'_{i,t_i} \rfloor \end{cases} \tag{18}$$

$$N_{SI} = \sum_{i=1}^{n} \sum_{t_i=1}^{om_i} SI_{i,t_i} \tag{19}$$

$$SI = \frac{N_{SI}}{N_{Gene}} \tag{20}$$

计算其他个体与这个体的相似度 SI，如果 SI 大于阀值 Rt 就说明两个个体是相似个体，找到的相似个体构造一个临时子种群 $stPop_1$。

假设变量 SI_{i,t_i} 表示种群两个不同的个体 X_{np}、X'_{np} 中对应基因片段 $\lfloor a_{i,t_i} \rfloor$、$\lfloor a'_{i,t_i} \rfloor$ 取整后是否相同，如果相同，表示工件选择加工工位相同，则 $SI_{i,t_i} = 1$，否则 $SI_{i,t_i} = 0$。在式(19)中 N_{SI} 表示两个个体相同基因片段的个数。在式(20)中 SI 是个体之间的相似度，是两个个体相同基因片段的个数 N_{SI} 与个体的基因总数 N_{Gene} 的比值，如果大于相似度阀值 Rt，则说明两个个体相似。

步骤 3：对剩余个体，重复步骤 1，2，构造临时子种群 $stPop_i$，直到筛选完全部个体。

步骤 4：采用初始种群中个体建立方法建立新个体，并通过汉明距离检测新个体与种群中个体的相似度如果与种群中任意一个个体相似度大于阀值 Rt 则舍弃这个个体，继续重新生成新个体。

3.4 进化参数自适应调整策略

为了进一步改善进化停滞的问题，防止早熟，文献[11，12]提出了基于不同随机分布函数的自适应策略，能有效平衡算法的全局探索和局部开发能力，但这些进化策略没有进化过程的指导，具有很大的盲目性。对此，本文根据 DE 算法实际进化规律，引进了随停止代数自适应调整交叉因子 CR 与变异因子 F 的策略。如式(15)、(16)所示，在进化过程中，如果一代进化没有提高进化效果，则随着进化停止代数 $StopGen$ 自适应地调整交叉因子 CR 和变异因子 F。CR 和 F 用下式表示：

$$CR = CR \times rand(0,1) \times 2^{\sin(\frac{x}{StopGen} \times \frac{\pi}{2})} \quad x \in \{1, StopGen\} \tag{21}$$

$$F = F \times rand(0,2) \times 2^{\sin(\frac{x}{StopGen} \times \frac{\pi}{2})}, \quad x \in \{1, StopGen\} \tag{22}$$

3.5 DSADE 算法流程

图 1 所示是动态自适应差分进化(DSADE)算法的流程，其包括以下步骤：

步骤 1 设置初始化进化参数和记录进化迭代次数初值 。

步骤 2 进化代数 是否大于最大进化代数 ，如果大于，则保存最优解退出优化；如果不大于，则继续执行进化。

步骤 3 采取 DE/rand/1/bin 方式对种群 Pop 进行差分进化运算。

步骤 4 判断是否找到最优个体和满足停止进化条件，如果满足停止进化条件，则退出优化；如果不满足停止进化条件，则继续执行进化。

步骤 5 如果这代进化种群没有更新最优个体，则需要记录停止进化代数 = +1，重新调整精英种群自进化交叉算子 和变异算子 ；否则，=0。

步骤 6 判断是否进化过程到达开启动态更新种群相似个体的代数 ，如果 < ，则表示未到开启动态运 算过程的代数，则转到步骤2执行；否则，开始动

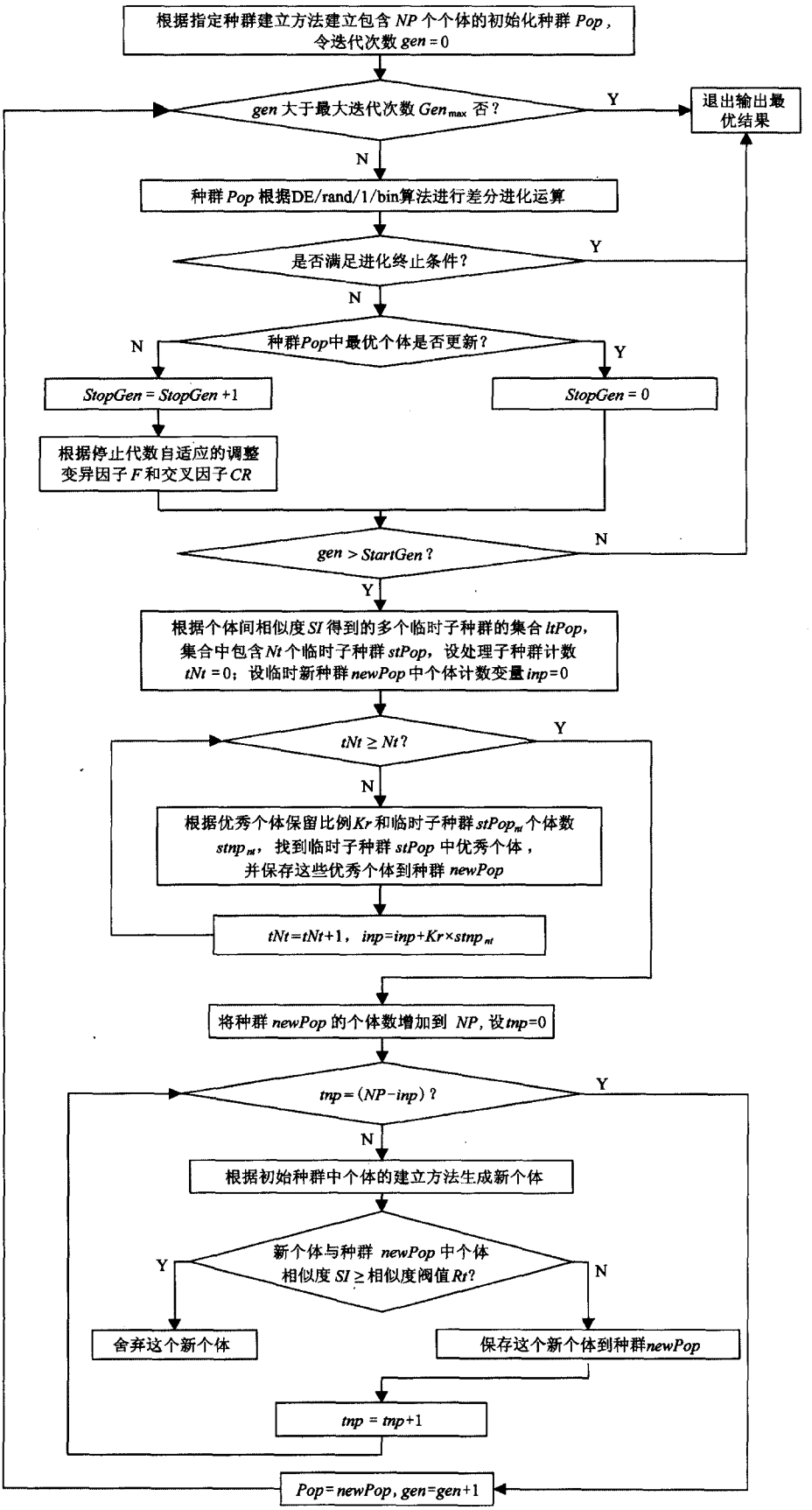

图 1　动态自适应差分进化算法流程

态更新相似性个体运算过程。

步骤 7 计算个体之间的相似度，将大于相似度阈值的个体组成临时子种群，这些临时子群集合为，临时子种群集合中包含个子种群和每个临时子种群中个体数。

步骤 8 设一个临时新种群中个体计数变量，依次根据优秀个体保留比例 提取每个临时子种群中优秀的个体，即提取适应度值符合进化趋势的个体，并将提取最优秀个体放入到一个新种群中，然后令 = +。

步骤 9 要使得中个体数达到设定的种群规模，设一个新增加个体计数变量。

步骤 10 判断新增个体计数是否大于等于，如果大于等于，则转到步骤 12；如果小于，则继续执行。

步骤 11 根据种群个体建立方法，生产新个体，判断这个新个体与新种群中每个个体的相似度，只要新个体与中个体相似度大于相似度阈值，则舍弃这个新个体；如果新个体与中个体相似度小于等于相似度阈值，则保留这个新生成的个体，并将这个个体加入到，并将 = +1，然后转到步骤 10。

步骤 12 将作为新一代进化的种群，并，然后转到步骤 2 执行。

4 DSADE 算法求解 RHFS 负荷平衡调度问题验证分析

在客车生产企业的涂装车间多遍彩条工序段中，客车车体上喷绘的不同彩条图案是企业技术人员事先依据彩条模板复杂程度，将彩条分解成多种颜色，每经过一遍“贴彩条”、“喷漆”、“烘漆”工序喷绘一种颜色，经过多遍喷绘形成整体图案，所以多遍彩条工序段是典型的可重入生产阶段[13]。该工序段中包含多个工序、多个并行工位，并且由于技术人员工作能力不同导致客车在并行工位上加工工时不同，所以该工序段又具有典型的混合流水车间特征。针对涂装车间多遍彩条工序段的排产问题应用案例，运用遗传算法（GA）、差分进化（DE）、自适应差分进化（SADE）、动态自适应差分进化（DSADE）算法进行全局优化，求解这类具有可重入工序的混合流水车间的负荷平衡排产问题。

4.1 排产数据

排产数据采用某客车制造企业的涂装车间中多遍彩条工序段的实际生产数据，图 2 所示是多遍彩条工序段工位配置图。

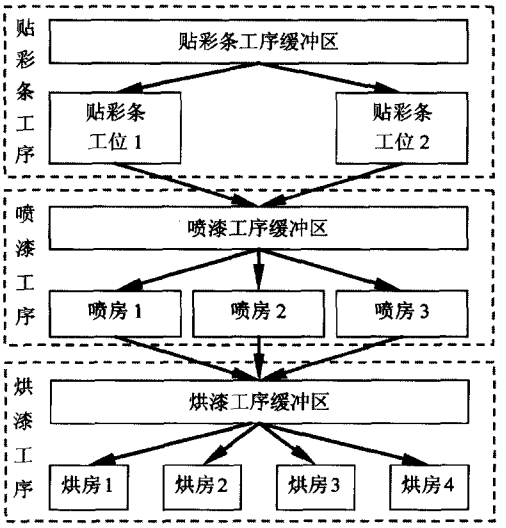

图 2 多遍彩条工序段工位配置图

求解包含 15 辆客车车体的在多遍彩条工序段的负荷平衡排产优化问题，工序 $\{OP_1, OP_2, OP_3\}$ 表示多遍彩条工序段中“贴彩条”、“喷漆”、“烘漆”3 个工序，这 3 个工序包含的并行工位数是 $\{2, 3, 4\}$，在排产过程中将“喷漆”工序的漆房和“烘漆”工序的烘房都当成工位处理。工件 $\{J_1, J_2, J_3, J_4, J_5, J_6, J_7, J_8, J_9, J_{10}, J_{11}, J_{12}, J_{13}, J_{14}, J_{15}\}$ 代表 15 辆待加工客车的车体，15 辆客车分属 15 个车型，每个车型按照其加工流程进行加工，客车加工流程信息如表 1所示。

多遍彩条工序段客车生产加工时间如表 2 所示，每辆客车根据车型不同，在各工序的加工工时也不相同。“烘漆”工序的烘房在烘漆过程没有人工参与，同一车型在不同烘房烘漆的时间相同，工时差别只与车型相关。表 3 给出了两个工序时间窗的时间范围。

4.2 设置进化参数

验证过程中采用 GA 算法、DE 算法、SADE 算法和 DSADE 算法进行全局优化，三种算法设置的参数如表 4 所示，除公共参数外，还要设置每种算法特有参数。

表1 客车加工流程信息表

待加工的客车	流程	流程中工序	加工工序的总数	是否重入	彩条遍数
J_1, J_2, J_3	FL_1, FL_2, FL_3	$\{OP_1^1, OP_2^2, OP_3^3, OP_1^4, OP_2^5, OP_3^6, OP_1^7, OP_2^8, OP_3^9\}$	$om_i = 9, i \in \{1,2,3\}$	是	3
J_4, J_5, J_6, J_7, J_8	$FL_4, FL_5, FL_6, FL_7, FL_8$	$\{OP_1^1, OP_2^2, OP_3^3, OP_1^4, OP_2^5, OP_3^6\}$	$om_i = 6, i \in \{4,5,\cdots,8\}$	是	2
$J_9, J_{10}, J_{11}, J_{12}, J_{13}, J_{14}, J_{15}$	$FL_9, FL_{10}, FL_{11}, FL_{12}, FL_{13}, FL_{14}, FL_{15}$	$\{OP_1^1, OP_2^2, OP_3^3\}$	$om_i = 3, i \in \{9,10,\cdots,15\}$	否	1

表2 涂装车间多遍彩条工序段加工时间表 (单位: min)

工序		工位	涂装车间多遍彩条工序段加工时间														
			J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9	J_{10}	J_{11}	J_{12}	J_{13}	J_{14}	J_{15}
			FL_1	FL_2	FL_3	FL_4	FL_5	FL_6	FL_7	FL_8	FL_9	FL_{10}	FL_{11}	FL_{12}	FL_{13}	FL_{14}	FL_{15}
一遍彩条	OP_1^1	$WS_{1,1}$	12	15	15	12	15	15	15	20	12	18	15	15	15	20	15
		$WS_{1,2}$	15	18	12	15	10	12	20	18	15	15	12	18	20	18	18
	OP_2^2	$WS_{2,1}$	15	18	20	18	15	18	20	20	15	18	15	18	20	25	25
		$WS_{2,2}$	18	20	16	20	15	15	20	20	20	22	15	15	25	25	18
		$WS_{2,3}$	15	15	15	18	18	20	22	22	18	20	18	20	20	20	20
	OP_3^3	$WS_{3,1}$	20	20	20	20	20	20	25	25	25	25	20	20	25	25	25
		$WS_{3,2}$															
		$WS_{3,3}$															
		$WS_{3,4}$															
二遍彩条	OP_1^4	$WS_{1,1}$	15	15	12	10	12	15	15	15							
		$WS_{1,2}$	12	12	15	12	10	15	10	12							
	OP_2^5	$WS_{2,1}$	18	15	15	15	15	20	15	15							
		$WS_{2,2}$	15	18	18	15	10	15	15	15							
		$WS_{2,3}$	18	15	15	15	15	15	20	20							
	OP_3^6	$WS_{3,1}$	20	18	18	18	18	20	20	20							
		$WS_{3,2}$															
		$WS_{3,3}$															
		$WS_{3,4}$															
三遍彩条	OP_1^7	$WS_{1,1}$	10	12	12												
		$WS_{1,2}$	12	10	10												
	OP_2^8	$WS_{2,1}$	10	15	12												
		$WS_{2,2}$	15	10	15												
		$WS_{2,3}$	15	15	10												
	OP_3^9	$WS_{3,1}$	18	18	18												
		$WS_{3,2}$															
		$WS_{3,3}$															
		$WS_{3,4}$															

表3 每个工序时间窗时间范围表 (单位: min)

	工序 OP_1	工序 OP_2	工序 OP_3
工序开始时间 Twb_j	0	35	75
工序间时间间隔 Pt_j		35	40

表 4 进化算法配置参数

群体进化算法	算法特有参数	公共参数
GA 算法	交叉概率 $P_c = 0.7$，变异概率 $P_m = 0.8$	种群规模 $f_{UR} = 30$，最大进化代数 $Gen_{max} = 2000$，$StopGen_{max} = 1000$
DE 算法	变异算子 $F = 0.9$，交叉因子 $CR = 0.7$	
SADE 算法	变异算子 $F = 0.9$，交叉算子 $CR = 0.7$	
DSADE 算法	变异算子 $F = 0.9$，交叉算子 $CR = 0.7$，动态更新相似性个体算法启动代数 $StartGen = 300$，个体相似性阈值 $Rt = 0.6$，保留比例 $Kr = 0.5$。	

4.3 评价指标设计

为了更好研究分析 HFS 负荷平衡排产结果，将 HFS 总负荷平衡代价 f_{LB}、工位加工等待时间 Twt 和基于工位加工时间的负荷平衡代价 Nlb 作为评价指标，并且增加总设备利用率 f_{UR} 和最大完工时间 C_{max} 个评价指标，从而体现 HFS 负荷平衡排产优化效果。f_{UR} 用下式表示：

$$f_{UR} = \frac{\sum_{j=1}^{m}\left(\sum_{k=1}^{M_j}\left(\sum_{i=1}^{n}\left(Tw_{i,j,k}\times At_{i,j,k}\right)\right)\right)}{\sum_{j=1}^{m}\left(\sum_{k=1}^{M_j}\left(\max\{C_{i,j,k}\times At_{i,j,k}\}-\min\{S_{i,j,k}\times At_{i,j,k}\}\right)\right)} \tag{23}$$

f_{UR} 表示 HFS 中所有工位的总设备利用率，是全部工位有效加工时间与工位工作时间跨度之比，这个时间跨度从工位开始第一个加工任务到最后加工任务之间跨度，是一个时间段。

最大完工时间 C_{max} 用下式表示：

$$C_{max} = \max\{C_{i,m,k} \mid i \in \{1,2,\cdots,n\}, k \in \{1,\cdots,M_j\}\} \tag{24}$$

C_{max} 表示全部工件在最后一道工序加工完工时间的最大值，也是全部工件完成加工的时间。

4.4 仿真结果与分析

验证程序采用 WPF(Windows Presentation Foundation)技术开发，使用 XAML、C#语言编写程序代码，面向企业网络化应用环境的瘦客户端(Browser/Server，B/S)模式下运行，测试环境 CPU Core2 P8600，主频 2.4G，内存为 2G，将 20 次仿真运算的平均统计结果放入到表 4 中。其中优化目标的适应度函数采用式(12)，其中权值 $\alpha_1 = 0.6$ 和 $\alpha_2 = 0.4$。排产结果的评价指标见表 5。

表 5 排产结果的评价指标对照表

评价指标		优化算法			
		GA	DE	SADE	DSADE
f_{LB}	最优解	0.1122	0.10137	0.09903	0.05742
	最差解	0.13433	0.12704	0.11694	0.09639
	平均值	0.11961	0.11569	0.10544	0.08607
Nlb	最优解	22.13307	19.76097	17.49322	9.51797
	最差解	41.96178	38.67445	33.20842	26.66119
	平均值	29.46243	29.23144	28.84854	23.35518
Twt	最优解	88	61	75	50
	最差解	140	135	117	96
	平均值	112	105.9	89.25	73.5
C_{max}	最优解	232	235	226	230
	最差解	256	260	263	286
	平均值	248.2	252	251.75	251.2
f_{UR}	最优解	0.93863	0.95737	0.94744	0.96441
	最差解	0.90546	0.90878	0.9203	0.93407
	平均值	0.92353	0.92730	0.93803	0.94854
平均执行时间(S)		206.4235	187.4821	219.7638	224.3247

从表5中可以看出,采用DSADE算法的RHFS加工时间负荷平衡代价 Nlb 平均值比采用GA、DE、SADE算法分别减少20.73%、20.1%、19.04%。采用DSADE算法的RHFS工位加工等待时间比采用GA、DE、SADE算法分别减少34.38%、30.59%、17.65%。采用DSADE算法的RHFS负荷平衡调度综合代价 f_{LB} 平均值比采用GA、DE、SADE算法分别减少28.04%、25.60%、18.37%。采用DSADE算法的RHFS总设备利用率比采用GA、DE、SADE算法都有所提高,最大完工时间 C_{max} 无明显变化,说明并行机负载越均衡,设备总闲置时间越小,设备利用率越高,且并未影响车间的生产效率。实验结果证明了在采用相同进化代数条件下,DSADE算法能更好地平衡全局探索能力与局部开发之间的矛盾,在进化停滞时能自动调节控制参数与种群分布密度及时跃出局部极值,用该算法求解RHFS负荷平衡调度问题,能取得更好的优化效果。图3、图4给出了排产结果Gantt图。

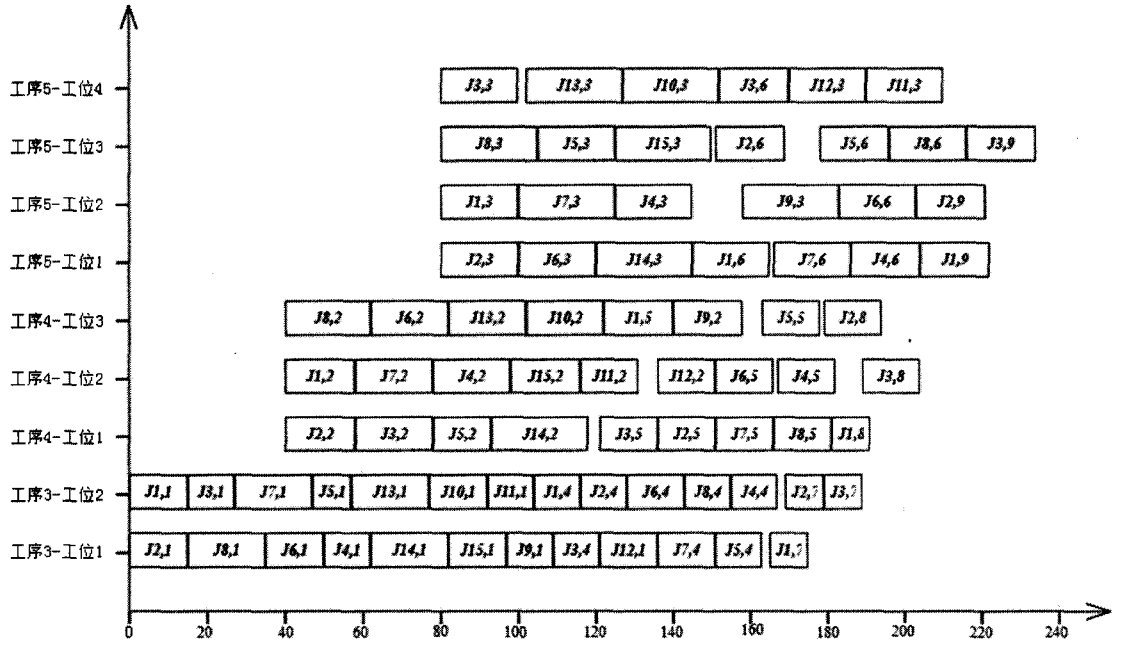

图3 基于时间窗的排产结果Gantt图

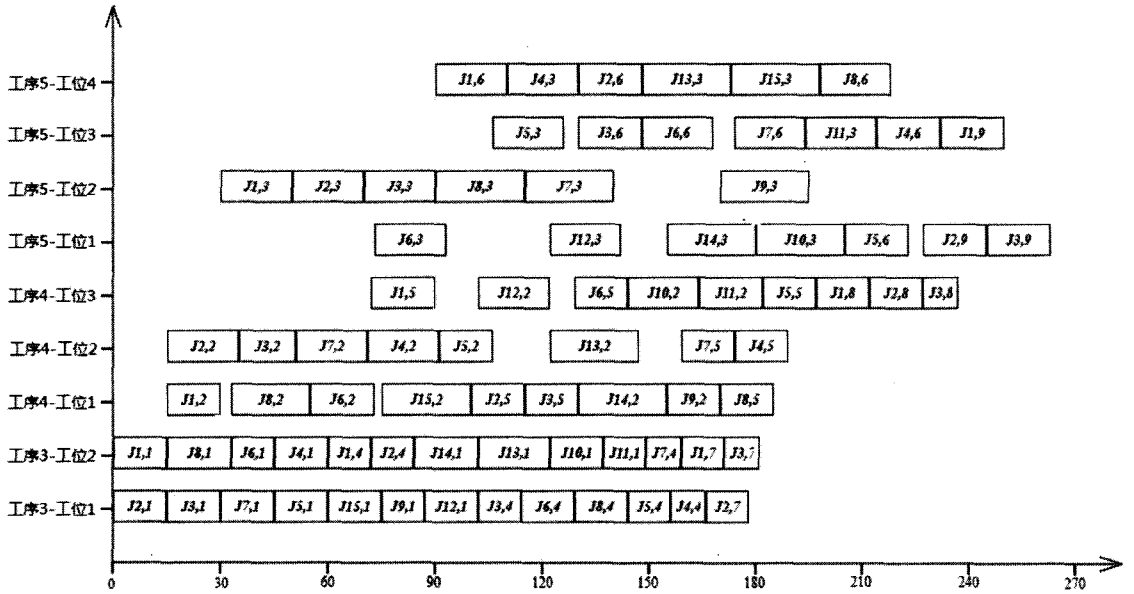

图4 无时间窗约束的排产结果Gantt图

通过图 3、图 4 可以看出，对每道工序设置加工时间节点后，工位加工等待时间减少，各并行工位负荷相当，设备利用率提高。

从图 5 中可以看出，随着训练代数的增加，4 组方案的适应度值都在减少，采用 GA 进化的方案 1 在 93 代之前迅速下降，在 423 到 674 间完成两次进化后停止进化；采用 DE 算法方案 2 在 6 代之前迅速下降趋势，从 382 代开始停滞进化；采用 SADE 算法的方案 3 在 18 代之前持续下降，之后进化速度减慢，但仍保持进化趋势，在 926 代以后停滞进化。采用 DSADE 算法的方案 4，在 6 代和 1443 代内保持进化活力。方案 1 与方案 2、3、4 对比可见，DE 算法比 GA 算法收敛速度快。方案 2 与方案 3 的曲线对比说明，加入参数自适应策略能加强 DE 算法跃出局部极值的能力。方案 4 与方案 3 曲线对照证明，引进动态种群更新机制能在保持 SADE 算法较强局部搜索能力的基础上进一步增强 SADE 算法的全局搜索能力。四组方案曲线对照图证明了采用基于汉明距离动态更新种群的 DSADE 算法能更持久地保持进化活力，增强逃离局部极值的能力，具有更强搜索最优解的能力。从另一个角度来讲，由于本次仿真实验设置的进化代数是 2000 代，四种算法进化 2000 代的时间有所差别，DSADE 算法执行时间稍长，但可以从进化曲线看出，在进化过程中，DSADE 算法有较好的进化活力，且在取得相同进化效果时，DSADE 算法明显速度快，所需进化代数小。当增大进化代数，DSADE 算法能取得更好的进化效果。

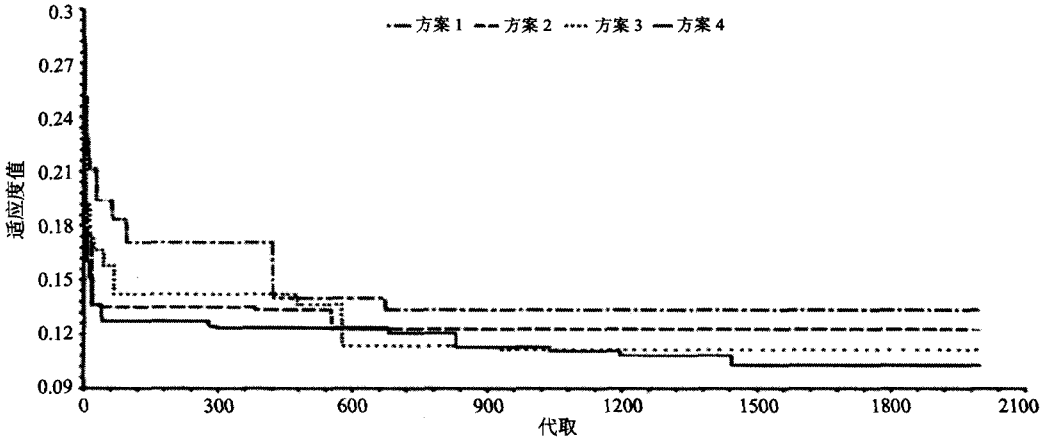

图 5 方案 1,2,3,4 适应度值与训练代数关系图

5 结 论

本文研究了可重入混合流水车间负荷平衡排产优化调度问题，将加工时间负荷不平衡代价和总工位等待时间加权求和后作为负荷平衡综合评价指标。全局优化方法采用 DSADE 算法，为适合可重入生产系统的要求，算法的编码依托工件的加工流程，每个染色体表示全部工件经过加工流程的工位分配情况，并结合最大剩余时间优先加工规则和时间窗约束进行解码。该算法设计了一种基于汉明距离的动态更新种群机制，实时更新具有高相似性个体，起到保持进化活力，增强全局搜索能力的作用，并通过加入随着停止代数自适应调整进化参数的方法，提高跃出局部极值的能力。应用算例的结果表明，相比与经典的 GA 算法、DE 算法，带时间窗约束的 DSADE 算法对可重入混合流水车间负荷平衡优化问题具有更好的求解能力。

参考文献

[1] Cho H M, Bae S J, Kim J, et al. Bi-objective scheduling for reentrant hybrid flow shop using Pareto genetic algorithm. *Computers & Industrial Engineering*, 2011, 61(3):529-541

[2] Kumar P R. Re-entrant lines. *Queueing systems*, 1993, 13(1-3): 87-110

[3] Graves S C, et al. Scheduling of re-entrant flow shops. *Journal of Operations Management*, 1983, 3(4): 197-207

[4] Yalaoui N, Amodeo L, Yalaoui F, et al. Particle swarm optimization under fuzzy logic controller for solving a hybrid reentrant flow shop problem. In: Proceedings of the 2010 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), Atlanta, USA, 2010. 1-6

[5] Hekmatfar M, Fatemi Ghomi S M T, Karimi B. Two stage reentrant hybrid flow shop with setup times and the criterion of minimizing makespan. *Applied Soft Computing*, 2011, 11(8): 4530-4539

[6] 刘小华, 林杰, 邓可. 基于遗传粒子群混合的可重入生产调度优化. 同济大学学报:自然科学版, 2011, 39(5): 726-730

[7] Han Z, Shi H, Yao L. Time-window-based combined objective DE algorithm for hybrid flow-shop. *International Journal of Modelling, Identification and Control*, 2014, 21(3): 295-305

[8] Storn R, Price K. Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 1997, 11(4): 341-359

[9] 王万良, 姚明海. 基于遗传算法的混合 Flow-shop 调度方法. 系统仿真学报, 2002, 14(7): 863-864

[10] 张凯, 肖建华, 耿修堂等. 基于汉明距离的 DNA 编码约束研究. 计算机工程与应用, 2009, 44(14): 24-26

[11] Lee W P, Chiang C Y. A self-adaptive differential evolution algorithm with dimen-sion perturb strategy. *Journal of Computers*, 2011, 6(3): 524-531

[12] 刘兴阳, 毛力. 基于 t 分布变异的自适应差分进化算法. 计算机工程与应用, 2012, 48(2):127-129

[13] 姜丽苹, 史海波, 彭慧. 涂装装配车间计划排产建模与仿真. 信息与控制, 2013, 42(005): 652-656

Research on the load balancing scheduling problem of reentrant hybrid flowshops

Han Zhonghua*,**, Dong Xiaoting*, Shi Haibo**

(*Faculty of Information and Control Engineering, Shenyang Jianzhu University, Shenyang 110168)

(**Department of Digital Factory, Shenyang Institute of Automation, CAS, Shenyang 110016)

Abstract

To solve the load balancing scheduling problem of a reentrant hybrid flowshop (RHFS), a mathematical RHFS model was formulated, and the weighted summation of the processing time load balancing cost and the total parallel machine waiting time was put as an index for comprehensive evaluation of load balancing. Furthermore, a new encoding method based on job processing procedure was designed, coupled with time-window constraints and the largest remaining time rules, to finish the decoding process, and a dynamic self-adaptive differential evolution (DSADE) algorithm was used to complete the global optimization. The DSADE algorithm presents a new dynamic population update mechanism on the basis of hamming distance to increase the diversity of population, and brings in a self-adaptive parameter adjusting strategy along with stop iterations to enhance the ability to jump out of local extreme value. Finally, an example of production scheduling problem for multi-pass color strip procedure in bus manufacturing painting workshop was simulated. The results showed that the load balance evaluation index of the DSADE algorithm was decreased by more than 20% in average compared with the algorithms of GA, differential evolution (DE) and solf-adaptive differential evolution (SADE).

Key words: reentrant hybrid flowshop (RHFS), load balancing, DE algorithm, individual similarity, painting workshop