

DOI:10.13196/j.cims.2016.02.027

改进 DE 算法求解混合流水车间负荷平衡问题

韩忠华^{1,2}, 董晓婷¹⁺, 史海波², 朱一行¹

(1. 沈阳建筑大学 信息与控制工程学院, 辽宁 沈阳 110168;

2. 中国科学院 沈阳自动化研究所, 辽宁 沈阳 110016)

摘要:为解决混合流水车间不相关并行机负荷平衡排产优化问题,建立了混合流水车间负荷平衡优化问题数学模型,以并行工位加工时间负荷平衡代价与总工位等待时间加权求和之值作为负荷平衡评价指标。全局优化算法采用双种群自适应差分进化算法,该算法设计了新的双种群结构和协同进化方式,并引入随停止代数自适应调整进化参数的策略,以增强跳出局部极值、保持进化活力的能力。为进一步提高算法搜索最优解效率,设计了一种基于负荷平衡选择概率的初始种群建立方法,以提高初始种群中初始解的质量、缩小有效解空间。基于汽车生产中的实例数据,将双种群自适应差分进化算法与遗传算法、差分进化算法、自适应差分进化算法进行仿真比较,结果表明,双种群自适应差分进化算法的负荷平衡评价指标有显著的降低。

关键词:混合流水车间排产问题;负荷平衡;选择概率;差分算法;双种群自适应差分进化算法

中图分类号:TH186 文献标识码:A

Improved DE algorithm for hybrid flow shop load balancing scheduling problem

HAN Zhonghua^{1,2}, DONG Xiaoting¹⁺, SHI Haibo², ZHU Yihang¹

(1. Faculty of Information and Control Engineering, Shenyang Jianzhu University, Shenyang 110168, China;

2. Department of Digital Factory, Shenyang Institute of Automation, CAS, Shenyang 110016, China)

Abstract:To solve the Hybrid Flow Shop with non-identical parallel machine Load Balancing Scheduling Problem (HFS-LBSP), a mathematical model was formulated. The weighted summation of processing time load balancing cost and total parallel machine waiting time was taken as the load balancing comprehensive evaluation index. A Double Population Self-adaptive Differential Evolution (DPSADE) algorithm was proposed for global optimization purpose. In DPSADE algorithm, a new bi-population construction methodology and cooperative evolution mechanism was devised for the usage of maintaining the population diversity and evolution energy, and a self-adaptive parameter adjusting strategy in terms of stop iterations was introduced to enhance the capacity in escaping from the local optimal and keeping evolution alive. To improve the searching efficiency of the algorithm, a new population initialization method based on load balancing selection probability was designed at the initialization stage. Two example of production scheduling problem for car manufacturing and steel smelting processing were simulated, and the results showed that the load balance evaluation index of DPSADE algorithm had a significant reduction compared with Genetic Algorithm(GA), Differential Evolution algorithm(DE) and Self-Adaptive Differential Evolution algorithm (SADE).

Keywords: hybrid flow shop scheduling problem; load balancing; selection probability; differential evolution algorithm; double population self-adaptive differential evolution algorithm

收稿日期:2015-02-10;修订日期:2015-05-25。Received 10 Feb. 2015;accepted 25 May 2015.

基金项目:国家重大科技专项资助项目(2011ZX02601-005);辽宁省教育厅资助项目(L2013237)。**Foundation items:**Project supported by the National Key Technological Special Project, China(No. 2011ZX02601-005), and the Project of Liaoning Provincial Education Department, China(No. L2013237).

0 引言

混合流水车间 (Hybrid Flow Shop, HFS) 是经典流水车间 (flow shop) 与并行机作业车间 (parallel machine shop) 的结合, 具有多任务、多工序、多并行机的特点, 广泛存在于汽车生产、半导体封装、大型装备制造等诸多国民经济的关键行业。受并行机工作能力和不同加工任务工艺差异的影响, HFS 中常出现并行机之间负荷不平衡的现象, 导致部分设备空闲或过载, 降低了设备利用率, 延长了生产周期。因此, 研究混合流水车间负荷平衡排产优化问题 (Hybrid Flow Shop-Load Balancing Scheduling Problem, HFS-LBSP) 对提高生产效率和设备利用率具有重要意义。

车间负荷平衡问题自 1970 年被提出以来^[1], 已取得较大研究进展。例如: Shimoyashiro 指出, 工位负荷是否平衡以及加工任务的投入量将直接影响作业车间的加工效率^[2]; Houghton 为提高准时制生产系统的生产效率, 设计了一类规划程序以平衡负荷变化与在制品库存^[3]; Mezmaz 根据基于网格的分支定界算法设计了一种有效的负荷平衡策略, 并将其用于解决流水车间双目标优化问题^[4]; Timur Keskinurk 采用蚁群优化算法减小并行机环境下的负荷相对不平衡率问题^[5]; Samia Kouki 提出一种基于并行搜索策略的 GALB (grid algorithm with load balancing) 算法, 以更好地平衡置换流水车间加工机器间的负载^[6]; Rong-Hwa Huang 指出平衡各加工工位负载是企业复杂竞争环境下降低制造成本、提高产能利用率的必要方法之一^[7]。

HFS-LBSP 受自身复杂性影响, 其研究进展相对较慢; 文献[8]设计了一种人工免疫算法, 在合理范围内均衡各阶段机器的加工时间, 但将负荷平衡作为约束处理 (将每道工序每台设备加工时间与该工序平均工时的相对误差限设为 15%), 主要优化的是 makespan 问题; 文献[9]研究了不相关并行机 HFS-LBSP, 以每道工序任意两台并行机最大负荷之差与该工序总负荷水平之比作为负荷平衡优化目标, 设计了混合遗传算法进行求解。HFS-LBSP 研究文献表明, 当前该问题的研究主要集中在均衡各并行机时间负荷, 并未考虑各个工位等待时间对负荷平衡的影响, 而且工位等待时间越长、车间生产效率越低。因此, 本文对 HFS-LBSP 进行深入研究, 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, 并将其作为全局优化算法的适应度函数, 采用双种群自适应差分进化算法 (Double Population Self-Adaptive Differential Evolution algorithm, DPSADE) 作为全局优化方法对该问题进行求解。

1 HFS 排产问题描述与参数设计

如图 1 所示, 包含 n 个工件的加工队列在车间顺序进行 m 道工序的加工, m 道工序中至少有一道工序包含多个并行工位, 工件在并行工位上的加工时间可能不同, 但要求工件每道工序都选择在一个工位加工。各道工序间存在缓冲区, 当工件完成前一道工序的加工后进入下一道工序缓冲站中的加工等待队列。通过确定每道工序上工件的加工顺序及其在并行机上的分配情况, 进而确定工件在每道工序上的开工时间和完工时间。

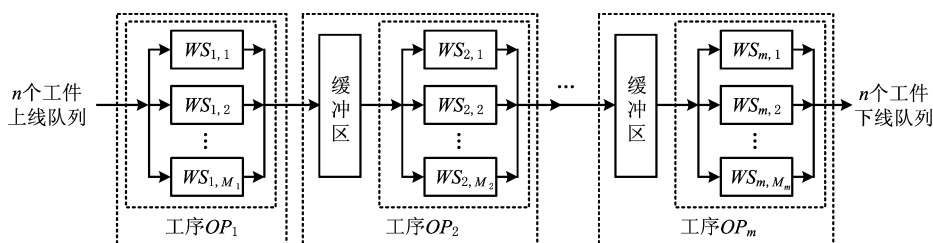


图1 混合流水车间排产问题数学模型图

1.1 模型参数

n 表示加工的工件总数;

J_i 表示第 i 个工件 $i \in \{1, \dots, n\}$;

m 表示加工的工序总数;

OP_j 表示第 j 道工序, $j \in \{1, \dots, m\}$;

M_j 表示工序 OP_j 的并行工位数, $j \in \{1, \dots,$

$m\}$;

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

$S_{i,j,k}$ 表示工件 J_i 在工序 OP_j 中的工位 $WS_{j,k}$ 上的开始加工时间;

$C_{i,j,k}$ 表示工件 J_i 在工序 OP_j 中的工位 $WS_{j,k}$

上的结束加工时间;

$n_{j,k}$ 表示 n 个工件中分配到工序 OP_j 的工位 $WS_{j,k}$ 上加工的工件总数;

$T\omega_{i,j,k}$ 表示工件 J_i 在工序 OP_j 的工位 $WS_{j,k}$ 上进行加工所需的时间。

1.2 假设变量和基本约束

$$At_{i,j,k} = \begin{cases} 0, & \text{工件 } J_i \text{ 在工序 } OP_j \\ & \text{未被分配到工位 } WS_{j,k} \text{ 上加工;} \\ 1, & \text{工件 } J_i \text{ 在工序 } OP_j \text{ 已被} \\ & \text{分配到工位 } WS_{j,k} \text{ 上加工。} \end{cases}$$

假设变量 $At_{i,j,k}$ 表示工件 J_i 是否在工序 OP_j 中的工位 $WS_{j,k}$ 上进行加工。

$$C_{i,j,k} = S_{i,j,k} + T\omega_{i,j,k}, i \in \{1, 2, \dots, n\}, j \in \{1, 2, \dots, m\}; \quad (1)$$

$$C_{i,j,k} \leq S_{i,j+1,k'}, i \in \{1, 2, \dots, n\}, j \in \{1, 2, \dots, m-1\}, k \in \{1, \dots, M_j\}, k' \in \{1, \dots, M_{j+1}\}; \quad (2)$$

$$n_{j,k} = \sum_{i=1}^n At_{i,j,k} \quad (3)$$

其中:式(1)说明工件在每道工序的开始加工时间、加工时间和结束加工时间之间的关系;式(2)说明同一工件在连续加工工序的开始加工时间和结束加工时间之间的关系;式(3)说明工位上加工的工件总数等于所有工件中分配到该工位上加工的工件总数。其他约束如下:①不可中断约束,若工序在设备上已开始加工则不能中断,直到工序完成;②设备唯一性约束,一台设备在同一时刻只能加工一道工序;③工件独立性约束,各个待加工工件之间相互独立,没有优先级别的差异;④设备可用性约束,所有设备在调度时刻都是空闲可用的;⑤时间简化约束,工序的处理时间包括相应操

作的准备时间(如设置设备参数、换刀具等)、前后工序之间的物流/转移时间,即在生成调度结果时只需考虑工序的处理时间。

2 HFS-LBSP 数学模型

各并行工位的有效加工时间是否均衡是车间负荷平衡的主要判断依据,而且工位上加工任务之间的等待时间体现了车间的运作效率,因此在研究 HFS-LBSP 时,需要综合考虑工位加工时间和工位加工等待时间,两者的重要程度通过权值体现。

2.1 并行工位加工时间负荷平衡代价

工序 OP_j 的工位 $WS_{j,k}$ 上的工件加工时间之和 $T_{S_{j,k}}$ 为

$$T_{S_{j,k}} = \sum_{i=1}^n (T\omega_{i,j,k} \cdot At_{i,j,k}) \quad (4)$$

在工序 OP_j 的 M_j 个并行机上的工件平均加工时间 $\overline{T\omega_j}$ 为

$$\overline{T\omega_j} = \left[\frac{\sum_{k=1}^{M_j} \sum_{i=1}^n (T\omega_{i,j,k} \cdot At_{i,j,k})}{M_j} \right] \quad (5)$$

将车间中每一个工位 $WS_{j,k}$ 上的总加工时间 $T_{S_{j,k}}$ 与其所处工序的并行工位加工时间平均值 $\overline{T\omega_j}$ 的差值进行平方后再求和,建立并行工位加工时间负荷平衡代价 Tlb ,作为 HFS 并行工位负荷平衡程度的主要评价指标。

$$Tlb = \sum_{j=1}^m \left(\sqrt{\sum_{k=1}^{M_j} ((T_{S_{j,k}} - \overline{T\omega_j})^2)} \right) \quad (6)$$

2.2 总工位加工等待时间

在工位 $WS_{j,k}$ 上连续加工工件之间的等待时间之和为

$$Tms_{j,k} = \begin{cases} (\max\{C_{i,j,k} \cdot At_{i,j,k}\} - \min\{S_{i,j,k} \cdot At_{i,j,k}\}) - \sum_{i=1}^n (T\omega_{i,j,k} \cdot At_{i,j,k}), & n_{j,k} \geq 2; \\ 0, & n_{j,k} < 2. \end{cases} \quad (7)$$

式中: $n_{j,k} = \sum_{i=1}^n At_{i,j,k}$ 表示工位上分配的加工工件数; $\max\{C_{i,j,k} \cdot At_{i,j,k}\}$ 表示分配到工位 $WS_{j,k}$ 上加工工件的最大完工时间, $\min\{S_{i,j,k} \cdot At_{i,j,k}\}$ 表示分配到工位 $WS_{j,k}$ 上加工的工件中的最早开工时间,二者的差表示工位 $WS_{j,k}$ 上加工任务的时间跨度, $\sum_{i=1}^n (T\omega_{i,j,k} \cdot At_{i,j,k})$ 表示该工位上的有效工作时间。HFS 中所有工位的加工等待时间之和为

$$T\omega t = \sum_{j=1}^m \left(\sum_{k=1}^{M_j} Tms_{j,k} \right) \quad (8)$$

通过对每个工位在加工过程中的等待时间 $Tms_{j,k}$ 进行求和,得到总工位加工等待时间 $T\omega t$,作为负荷平衡问题的辅助评价指标。

2.3 HFS 负荷平衡综合代价

HFS 负荷平衡综合评价指标是将并行工位加工时间负荷平衡代价 Tlb 和总工位加工等待时间 $T\omega t$ 进行加权求和,得到 HFS 负荷平衡综合代价

f_{LB} 。在加权前使用式(9)和式(10)进行归一化处理,使得两个评价指标的数值在一个数量级,从而有利于更好地控制两者在 HFS-LBSP 优化结果中的作用。其中: Tlb_{min} 和 Tlb_{max} 分别表示初始种群中并行工位加工时间负荷平衡代价的最小值和最大值, Twt_{min} 和 Twt_{max} 分别表示初始种群中总工位加工等待时间的最小值和最大值。

$$f_{TLB} = \begin{cases} \frac{Tlb - Tlb_{min}}{Tlb_{max} - Tlb_{min}}, & Tlb > 0; \\ 0, & Tlb = 0; \end{cases} \quad (9)$$

$$f_{WT} = \begin{cases} \frac{Twt - Twt_{min}}{Twt_{max} - Twt_{min}}, & Twt > 0; \\ 0, & Twt = 0; \end{cases} \quad (10)$$

$$f_{LB} = \alpha_1 \cdot f_{TLB} + \alpha_2 \cdot f_{WT} \quad (11)$$

式中权值 α_1 和 α_2 满足条件 $\alpha_1 + \alpha_2 = 1$ 。HFS-LBSP 的优化目标是 $\min f_{LB}$, f_{LB} 的值越小, HFS-LBSP 的优化结果越好。

3 算法设计

采用 DPSADE 算法求解 HFS-LBSP。DPSADE 算法是对标准差分进化(Differential Evolution algorithm, DE)算法^[10]的改进,其核心在于初始种群的建立、编码解码,以及参数自适应策略与进化机制的设置。

3.1 建立初始种群

初始种群的每个个体都对应调度的一个可行解,若初始种群能反映优化目标特征,则能缩小寻优空间、提高搜索效率。近年来,一些学者研究了很多初始种群建立方法,如基于 GT(generate and test)^[11]、NEH(Nawaz-Enscore-Ham)^[12]等启发式规则生成初始种群,也有学者将群体智能算法用于初始种群的建立,如蚁群优化^[13]算法。本文依据负荷平衡优化目标设计了一种工件选择加工工位的选择概率,使优化目标特征能够很好地体现在建立初始种群的过程中,起到提高全局优化效果的目的。

车间排产仿真过程是对车间生产过程的一种模拟,随着生产时间的推进,车间的各种生产状态会发生改变。在排产过程中,工件 J_i 被分配到工序 OP_j 的并行机上,即将工件被分配到工位的事件定义为 t , $t \in \{1, \dots, n\}$ 。将 t 加入 HFS-LBSP 数学模型的相关公式中,通过分析这些事件发生时车间中各种生产加

工状态的变迁,建立工件分配事件相关的选择概率。

(1)基于工件分配事件的并行工位加工时间负荷平衡代价

在工件分配事件 t 发生后,工序 OP_j 的工位 $WS_{j,k}$ 上已分配工件的加工时间之和为

$$Ts_{j,k}(t) = \sum_{i=1}^n (T\omega_{i,j,k} \cdot At_{i,j,k}(t)) \quad (12)$$

式中 $At_{i,j,k}(t)$ 是假设变量,表示当事件 t 发生后工件 J_i 是否被分配到工位 $WS_{j,k}$ 进行加工。若在事件 t 发生后在工序 OP_j 中工件 J_i 被分配到工位 $WS_{j,k}$ 上,则加工 $At_{i,j,k}(t) = 1$, 否则 $At_{i,j,k}(t) = 0$ 。

在事件 t 发生后,工序 OP_j 的并行工位上已分配工件的加工工时的均值

$$\overline{T\omega_j}(t) = \left[\frac{\sum_{k=1}^{M_j} \sum_{i=1}^n (T\omega_{i,j,k} \cdot At_{i,j,k}(t))}{M_j} \right] \quad (13)$$

当事件 t 发生后,在工序 OP_j 的并行工位上已分配工件的加工时间负荷平衡代价

$$Tlb_j(t) = \sqrt{\sum_{k=1}^{M_j} (Ts_{j,k}(t) - \overline{T\omega_j}(t))^2} \quad (14)$$

(2)基于连续工件分配事件的并行工位加工时间负荷平衡代价

为进一步研究连续两次工件分配事件的相互影响和关系,设 t_1 和 t_2 为在工序 OP_j 的排产过程中,两个连续工件分配到工位事件。

当事件 t_1 发生后, $\sum_{i=1}^n \sum_{k=1}^{M_j} At_{i,j,k}(t_1) = num'$, $num' \in [0, n-1]$, 有 num' 个工件已经被分配。下一个事件 t_2 是为工件 $J_{i'}$ 分配工位 ($i' \in \{1, \dots, n\}$ 且 $i' \notin \{i | \sum_{i=1}^n \sum_{k=1}^{M_j} At_{i,j,k}(t) = 1, 1 \leq t \leq t_1\}$) 的事件,工件 $J_{i'}$ 表示在工序 OP_j 的排产过程中,事件 t_1 发生后还没有被分配过工位的工件。

当事件 t_2 发生后已经分配到工位 $WS_{j,k}$ 上的工件的加工时间之和

$$Ts_{j,k}(t_2) = \sum_{i=1}^n (T\omega_{i,j,k} \cdot At_{i,j,k}(t_1)) + T\omega_{i',j,k} \cdot At_{i',j,k}(t_2) \quad (15)$$

当事件 t_2 发生后,若工件 $J_{i'}$ 在工序 OP_j 被分配到工位 $WS_{j,k}$ 上加工,则并行工位上已分配工件的加工工时的均值为

$$\overline{T\omega_{j,k}}(t_2) = \left[\frac{\sum_{k=1}^{M_j} (\sum_{i=1}^n (T\omega_{i,j,k} \cdot At_{i,j,k}(t_1)) + T\omega_{i',j,k} \cdot At_{i',j,k}(t_2))}{M_j} \right] = \overline{T\omega_j}(t_1) + \frac{T\omega_{i',j,k}}{M_j} \quad (16)$$

若工件随机被分配到一个工位,则当事件 t_2 发生后,工件 J_i 可能被分配到工序 OP_j 的并行机中的任意一个工位 $WS_{j,k}$ ($k \in \{1, M_j\}$) 上加工。因为工件 J_i 在并行机中每个工位上的加工工时都不同,并且事件 t_1 发生后并行工位中的每个工位上已分配工件的情况不同,导致工件 J_i 分配到不同的工位

$$Tlb_{i',j,k}(t_2) = \sqrt{\sum_{k=1}^{M_j} ((Ts_{j,k}(t_2) - \overline{T\omega_{j,k}}(t_2))^2)} \\ = \sqrt{\sum_{k=1}^{M_j} ((\sum_{i=1}^n (T\omega_{i,j,k} \cdot At_{i,j,k}(t_1)) + T\omega_{i',j,k} \cdot At_{i',j,k}(t_2) - (\overline{T\omega_j}(t_1) + \frac{T\omega_{i',j,k}}{M_j}))^2)} \quad (17)$$

由于负荷平衡问题需要考虑并行工位上以前分配工件的情况,式(17)也可以表示成包含事件 t_1 的表达形式。

(3)设计负荷平衡选择概率

考虑 t_1 和 t_2 都是工件分配工位的状态,用 $Tlb_{i',j,k}(t)$ 直接表示事件 t 发生后工件被分配到不同工位的并行工位加工时间负荷平衡代价。采用并行工位加工时间负荷平衡代价 $Tlb_{i',j,k}(t)$ 构建负荷平衡选择概率 $P_{i',j,k}^{LB}(t)$,式(18)表示在工序 OP_j 上,当事件 t 发生后工件 J_i 选择分配到工位 $WS_{j,k}$ 上,对并行工位的负荷平衡的影响程度。在工件第一次分配到工位上时,由于没有以前分配结果的影响,负荷平衡选择概率 $P_{i',j,k}^{LB}(t)$ 是工序 OP_j 的并行工位数 M_j 的倒数,表示工件被分配到每个工位的几率都均等。设置 $P_{i',j,k}^{LB}(t)$ 的限定条件,如式(19)所示,每个工件在并行工位上的选择概率 $P_{i',j,k}^{LB}(t)$ 之和为 1。

$$P_{i',j,k}^{LB}(t) = \begin{cases} \frac{1}{M_j}, & t = 1; \\ \frac{\sum_{k=1}^{M_j} Tlb_{i',j,k1}(t) - Tlb_{i',j,k}(t)}{\sum_{k=1}^{M_j} Tlb_{i',j,k1}(t) \cdot (M_j - 1)}, & t > 1 \cup t \leq n; \end{cases} \quad (18)$$

$$\sum_{x=1}^{M_j} P_{i',j,k}^{LB}(t) = 1. \quad (19)$$

在建立初始种群过程中考虑负荷平衡优化目标的影响,应该让工件选择负荷平衡概率大的工位进行加工,即使负荷更加均衡的工位优先分配工件。

3.2 编码解码

本文基于矩阵编码方法^[14],用向量序列替代矩阵表示染色体,更加直观、简便。种群的基因 a_{ij} 表

上, $Tlb_j(t_2)$ 的值可能会不相同,所以重新定义负荷不平衡代价 $Tlb_{i',j,k}(t_2)$,用以表示工件被分配到不同工位时的并行工位加工时间负荷平衡代价,如式(17)所示。式中 $Tlb_{i',j,k}(t_2)$ 表示事件 t_2 发生后,工件 J_i 被分配到工序 OP_j 中工位 $WS_{j,k}$ 上的并行工位加工时间负荷平衡代价。

示工件 J_i 在工序 OP_j 中选择加工的工位, $i \in \{1, 2, \dots, n\}$, $j \in \{1, 2, \dots, m\}$, $a_{i,j} = rand(1, M_j + 1)$, $a_{i,j}$ 的取值为区间 $(1, M_j + 1)$ 上的一个随机数。基因 $a_{i,j}$ 的 $\lfloor a_{i,j} \rfloor$ 值($\lfloor \cdot \rfloor$ 表示向下取整)为工件 J_i 在工序 OP_j 选择加工的工位。每个个体 $X_{np} = \{a_{1,1}, a_{1,2}, \dots, a_{i,j}, \dots, a_{n,m}\}$, $np \in \{1, 2, \dots, NP\}$, 表示全部工件经过 HFS 加工过程的工位分配情况。确定了工件的加工工位,又已知工件在该工位的加工时间,再通过先入先出(First In First Out, FIFO)规则控制工件在每个工序开工完工顺序,就可以将初始种群中的个体解析为 HFS 排产结果。

3.3 进化机制

在每一代进化运算前,根据适应度值将种群分为精英种群和平凡种群,符合进化趋势的种群分到精英种群,否则分到平凡种群。

(1)种群各自独立进化方式中,种群的自进化采用 DE/rand/1/bin^[15] 变异方式。

(2)两个种群交流进化方式中,种群交流进化采用 DE/best/2/bin^[15] 变异方式,从两个种群中各取出两个个体进行交叉进化,生成新个体。

3.4 进化参数自适应调整策略

$$CR' = CR \cdot rand(0, 1) \cdot 2^{\sin(\frac{StopGen}{StopGen_{max}} \cdot \frac{\pi}{2})}, \\ StopGen \in \{1, StopGen_{max}\}; \quad (20)$$

$$F' = F \cdot rand(0, 1) \cdot 2^{\sin(\frac{StopGen}{StopGen_{max}} \cdot \frac{\pi}{2})}, \\ StopGen \in \{1, StopGen_{max}\}. \quad (21)$$

式(20)和式(21)是交叉因子 CR' 和变异因子 F' 随停止代数 $StopGen$ 自适应调整的公式。交叉因子 CR' 和变异因子 F' 的波动幅度随停止代数的增加呈指数增长,算法更容易跃出局部极值。由于普通种群和优秀种群独自进化和交流进化的程度不同,对每种进化方式的 CR' 和 F' 都设置不同的调整范围,如表 3 所示。当 CR' 和 F' 超出给定范围时,

分别取其相应的上下限值。

3.5 DPSADE 算法流程描述

DPSADE 算法的具体步骤如下:

步骤 1 设置初始化进化参数和记录进化迭代次数初值 $gen=0$ 。

步骤 2 将初始种群根据个体适应度值分为精英种群 $Pop1$ 和平凡种群 $Pop2$, $Pop1$ 的个体总数为 $NP1$, $Pop2$ 的个体总数为 $NP2$, $NP = NP1 + NP2$ 。

步骤 3 对 $Pop2$ 进行 DE 进化运算。在 $Pop2$ 中取 3 个个体, 采用 DE/rand/1/bin 方式生成新个体 X_{new} 。

步骤 4 计算新个体 X_{new} 的适应度, 如果新产生的个体 X_{new} 优于 $Pop1$ 中的任意一个个体, 则用该新个体替换 $Pop1$ 中按适应度排序最差的个体; 如果不优于 $Pop1$ 中的个体, 则判断新个体是否优于 $Pop2$ 中原来的个体 X_1 , 如果优于原个体 X_1 , 则替代原个体 X_1 , 否则舍弃新个体 X_{new} 。

步骤 5 对 $Pop1$ 进行 DE 进化运算。在 $Pop1$ 中取 3 个个体, 采用 DE/rand/1/bin 方式生成新个体 X''_{new} 。

步骤 6 计算由 $Pop1$ 产生的新个体 X''_{new} 的适应度, 如果新产生个体 X''_{new} 的适应度值优于 $Pop1$ 中的任意一个个体, 则用该新产生的个体代替 $Pop1$ 按适应度排序最差的个体, 否则舍弃新个体 X''_{new} 。

步骤 7 找到 $Pop1$ 中的最优个体, 在 $Pop1$ 和 $Pop2$ 中各自取出两个个体, 使用 DE/best/2/bin 方式产生新个体 X''_{new} 。

步骤 8 计算新产生个体 X''_{new} 的适应度, 如果新产生个体 X''_{new} 的适应度值优于 $Pop1$ 中的任意一个个体, 则代替 $Pop1$ 适应度值排序最差的个体, 否则舍弃新个体 X''_{new} 。

步骤 9 在 $Pop2$ 中的所有个体完成一代进化后, 如果没有更新 $Pop1$ 中的最优个体, 则需要记录停止进化代数 $StopGen$, 重新调整 $Pop1$ 自进化交叉因子 $CR1$ 和变异因子 $F1$ 、 $Pop2$ 自进化交叉因子 $CR2$ 和变异因子 $F2$ 、双种群交流进化的交叉因子 $CR3$ 和变异因子 $F3$ 。

步骤 10 判断是否到达终止条件, 如果未达到终止条件, 则转步骤 2, 继续进化过程。

4 仿真对比分析

采用两组不同规模的 HFS 实例进行性能测试

和算法比较, 实例为汽车发动机车间调度问题^[16]。HFS 中包括 3 个工序 $\{OP_1, OP_2, OP_3\}$, 工序包含的工位数为 $\{3, 3, 3\}$, 随机解空间大小为 27^{12} , 基于负荷平衡选择概率的解空间大小为 2.18×10^9 。所有工件在全部工位上的加工时间如表 1 所示。

表 1 工件加工时间表

工 序	工 位	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}
OP_1	$WS_{1,1}$	2	4	6	4	4	6	5	3	2	3	5	6
	$WS_{1,2}$	2	5	5	3	5	5	2	5	5	6	2	5
	$WS_{1,3}$	3	4	4	4	3	4	4	4	4	4	4	4
OP_2	$WS_{2,1}$	4	3	4	6	3	2	4	7	1	3	3	5
	$WS_{2,2}$	5	4	2	5	1	3	6	5	2	4	5	4
	$WS_{2,3}$	2	3	3	3	3	4	3	3	7	4	6	3
OP_3	$WS_{3,1}$	3	4	4	6	4	3	4	3	8	8	7	4
	$WS_{3,2}$	2	5	2	5	6	9	3	6	6	6	6	7
	$WS_{3,3}$	3	4	5	8	5	5	5	4	7	7	5	5

4.1 仿真方案

通过与遗传算法 (Genetic Algorithm, GA)、DE 算法、SADE 算法进行比较, 分析 DPSADE 算法的优化效果, 设计仿真方案 1~4。为了分析基于负荷平衡选择概率的初始种群建立方法对提高优化效果的影响, 对应设计仿真方案 5~8。8 组仿真方案信息如表 2 所示。

表 2 8 组仿真方案信息

仿真方案	全局优化方法	初始种群构造方法
方案 1	GA 算法	工件选工位的初始种群建立方法
方案 2	DE 算法	工件选工位的初始种群建立方法
方案 3	SADE 算法	工件选工位的初始种群建立方法
方案 4	DPSADE 算法	工件选工位的初始种群建立方法
方案 5	GA 算法	基于负荷平衡选择概率的初始种群建立方法
方案 6	DE 算法	基于负荷平衡选择概率的初始种群建立方法
方案 7	SADE 算法	基于负荷平衡选择概率的初始种群建立方法
方案 8	DPSADE 算法	基于负荷平衡选择概率的初始种群建立方法

4.2 设置进化参数

设置参数如表 3 所示, DPSADE 算法除了种群个体数 NP 、进化代数 Gen 、停止代数 $StopGen$ 等公

共参数外,两个种群都配置各自的交叉因子和变异因子,并根据 DE 算法参数的取值范围设定交叉因子和变异因子的自适应调整范围;精英种群的交叉因子和变异因子调整范围小,有利于保持优秀个体

稳定;平凡种群的交叉因子和变异因子调整范围大,有利于增强平凡种群的进化活力。虽然设定了三对交叉因子和变异因子并限定了调整范围,但将参数设置在推荐的取值附近^[17]有利于提高优化效果。

表 3 群体进化算法配置参数

群体进化算法	算法特有参数	公共参数
GA 算法	交叉概率 $C_{max}=0.7$, 变异概率 $P_m=0.8$	
DE 算法	$F=0.9, CR=0.7$	
SADE 算	$CR=0.7, CR \in [0.6, 0.9], F=0.9$	种群规模 $NP=30$, 最大进化代数
	精英种群自进化 $CR_1=0.5, F_1=0.8, CR'_1 \in [0.3, 0.7], F'_1 \in [0.6, 1]$; 平凡种群	$Gen_{max}=1000$, 最大停止进化代数
DPSADE 算法	自进化 $CR_2=0.7, F_2=1, CR'_2 \in [0.5, 0.9], F'_2 \in [0.8, 1.2]$	$StopGen_{max}=500$
	双种群交流进化 $CR_3=0.7, F_3=0.9, CR'_3 \in [0.5, 0.9], F'_3 \in [0.7, 1.1]$	

4.3 评价指标设计

为了更好地研究分析 HFS-LBSP 的优化结果,将并行工位加工时间负荷平衡代价 Tlb 、总工位加工等待时间 Twt 和 HFS 负荷平衡综合代价 f_{LB} 作为评价指标,并增加总设备利用率 f_{UR} ,从而更好地评价 HFS-LBSP 的优化效果。

(1)总设备利用率

$$f_{UR} = \frac{\sum_{j=1}^m (\sum_{k=1}^{M_j} (\sum_{i=1}^n (T\omega_{i,j,k} \cdot At_{i,j,k})))}{\sum_{j=1}^m (\sum_{k=1}^{M_j} (\max\{C_{i,j,k} \cdot At_{i,j,k}\} - \min\{S_{i,j,k} \cdot At_{i,j,k}\}))} \quad (22)$$

式中 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的跨度之比,该时间跨度是从工位开始的第一个加工任务到最后加工任务之间的跨度,是一个时间段。在实际工程应用中,时间跨度范围也可以是一个固定的时间段。

(2)最大完工时间

$$C_{max} = \max\{C_{i,m,k} \mid i \in \{1, 2, \dots, n\}, k \in \{1, \dots, M_j\}\} \quad (23)$$

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

4.4 仿真结果分析

测试程序采用 WPF (Windows presentation foundation) 技术开发,使用可扩展应用程序标记语言 (Extensible Application Markup Language, XAML)、C# 语言编程,面向企业网络化应用环境的浏览器/服务器 (Browser/Server, B/S) 模式下运行,测试环境为 CPU Core2 P8600,主频 2.4 G,内存 2 G,20 次仿真结果的平均值如表 4 所示。群体智能算法的适应度函数采用式(9),负荷不平衡综合代价 f_{LB} 的权值 $\alpha_1=0.7, \alpha_2=0.3$ 。初始种群的个体中,并行工位加工时间负荷平衡代价的最大值 $Tlb_{max}=53.1$,总工位加工等待时间的最大值 $Twt_{max}=70$,同时设定这两个评价指标的最小值 $Tlb_{min}=0, Twt_{min}=0$ 。

表 4 8 组方案排产结果的评价指标对照表

评价指标	解的质量	排产仿真方案							
		方案 1	方案 2	方案 3	方案 4	方案 5	方案 6	方案 7	方案 8
f_{LB}	最优解	0.086 08	0.076 87	0.063 46	0.044 65	0.071 02	0.065 29	0.058 29	0.024 15
	最差解	0.119 54	0.114 01	0.089 69	0.072 42	0.078 93	0.078 93	0.078 93	0.061 02
	平均值	0.102 91	0.091 95	0.075 61	0.063 55	0.075 19	0.074 96	0.071 70	0.054 87
Tlb	最优解	4.082 48	3.265 98	2.976 74	2.943 92	3.644 92	3.644 29	4.461 42	1.632 99
	最差解	8.928 58	8.146 23	7.367 70	5.374 65	8.062 15	7.534 89	5.789 70	5.136 99
	平均值	6.407 83	6.039 34	5.042 71	4.326 68	6.038 24	5.188 62	4.872 70	3.136 70

续表 4

	最优解	2	1	0	0	2	1	0	0
T_{wt}	最差解	12	7	6	6	9	5	5	4
	平均值	5.96	3.64	3.29	2.63	4.01	3.27	3.14	1.52
	最优解	32	30	33	31	31	29	27	30
C_{max}	最差解	38	40	39	40	36	35	39	37
	平均值	35.10	34.72	34.91	35.34	31.26	32.18	32.63	32.69
	最优解	0.979 87	0.993 93	1	1	0.980 39	0.982 63	1	1
f_{UR}	最差解	0.925 93	0.954 28	0.961 04	0.964 27	0.950 14	0.955 478	0.965 27	0.971 05
	平均值	0.963 68	0.976 75	0.979 58	0.981 86	0.973 67	0.978 13	0.978 64	0.988 51
	平均执行时间/s	25.20	23.35	26.17	39.76	26.44	23.42	28.04	40.58

从表 4 可以看出,采用 DPSADE 算法的方案 4 和方案 8 的 HFS 负荷平衡综合代价均值比采用 GA 的方案 1 和方案 5 分别降低 38.24% 与 27.02%,比采用 DE 算法的方案 2 和方案 6 分别降低 30.89% 与 26.80%,比采用 SADE 算法的方案 3 和方案 7 分别降低 15.95% 与 23.47%。采用 DP-SADE 算法的方案 4 和方案 8 的总设备利用率 f_{UR} 的平均值比采用 GA,DE,SADE 算法的方案均有提高。对比方案 1~4 或方案 5~8 可知,负荷平衡会使最大完工时间 C_{max} 稍微延长,但最大偏差不得超过 4.6%,在可接受范围内。采用 DPSADE 算法的方案 8 比采用基本 DE 算法的方案 2 的程序平均执行时间延长了 17.23 s,说明 DPSADE 算法运行过程中的重构双种群、自适应调整参数与初始种群建立等相关计算需要花费一些程序执行时间。

采用基于负荷平衡选择概率的初始种群建立方法的方案 5,6,8 中,HFS 负荷平衡综合代价 f_{LB} 的平均值比采用工件选工位的初始种群建立方法的方案 1,2,4 的 f_{LB} 平均值分别减小 26.94%,18.47% 和 13.66%;方案 7 的 HFS 负荷平衡综合代价 f_{LB} 与方案 3 的 f_{LB} 比较接近,这是因为采用基于负荷平衡选择概率的初始种群建立方法能够获得高质量的初始种群,在寻优初始阶段已为后续进化过程打下良好的基础,但由于后续进化过程不能有效强化这种优势,初始积累优势也会在进化过程中逐渐丧失,导致最终的进化搜索结果取决于算法本身的优化性能。采用基于负荷平衡选择概率的初始种群建立方法的方案 8,因为采用的 DPSADE 算法具有有效的双种群交流进化机制与自适应参数调整策略,能够很好地保持和巩固进化过程中累计的优势,所以方案 8 的各项评价指标都有显著提高。另外,采

用基于负荷平衡选择概率的初始种群建立方法的方案 5,6,7,8 的最大完工时间比采用工件选工位的初始种群建立方法的方案 1,2,3,4 都有所减小,说明采用负荷平衡选择概率建立初始种群能更好地均衡各并行机负荷,更合理地安排工件加工工位,从而减少整体完工时间。

图 2 所示为方案 8 的排产结果的甘特图,横坐标为时间轴,纵坐标为每个工序的工位。其中工件 J_1 的工艺路线为 $\{WS_{1,2}, WS_{2,2}, WS_{3,2}\}$,从图中可以直观地看出各个工件经过的工艺路线,可知排产结果能够达到很好的负荷平衡,总工位等待时间为 0。

从图 3 可以看出,在基于工序变工位的随机方法生成初始种群的方案 1~4 中,方案 4 的 DP-SADE 算法在进化过程中的下降趋势明显且在 163 代就取得较优解。方案 3 的 SADE 算法较方案 2 的 DE 算法在 1~120 代间下降明显,而 DE 算法在 231 代时陷入局部极值,停止进化,SADE 算法在 120~665 代间有不同程度的进化,取得的解优于 DE 算法。方案 8 与方案 1~4 初始种群的建立方法不同,因此进化初始点不同。

从进化曲线图 3 和图 4 可见,基于负荷平衡选择概率建立初始种群的方案 5~8 有较好的进化起点 0.079 27,方案 2,3,4 要分别进化 309 代、672 代、166 代才能小于该值,而方案 1 始终未进化到该值。而且随着进化代数的增加,方案 5~8 能获得较小的适应度函数值,说明基于负荷平衡选择概率能构建高质量的初始种群,从而获得更好的解。对比图 4 和图 5 两组进化曲线可知,DPSADE 算法的优化效果均优于其他几种算法,在达到相同进化效果时,DPSADE 算法的进化代数最少。

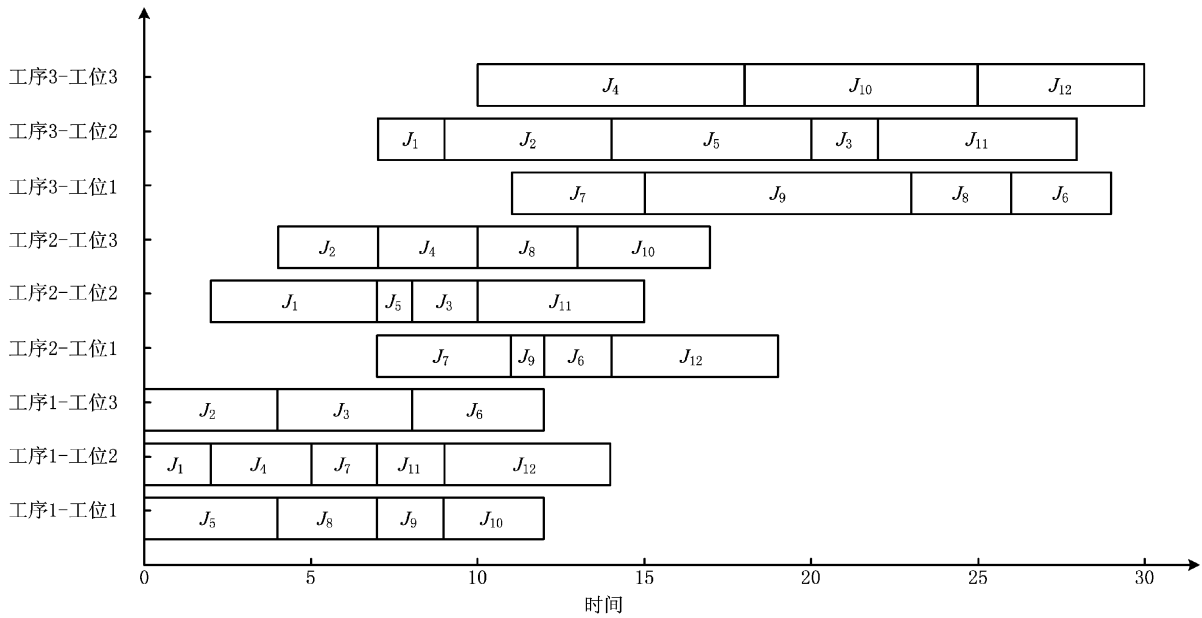


图2 方案8排产结果甘特图

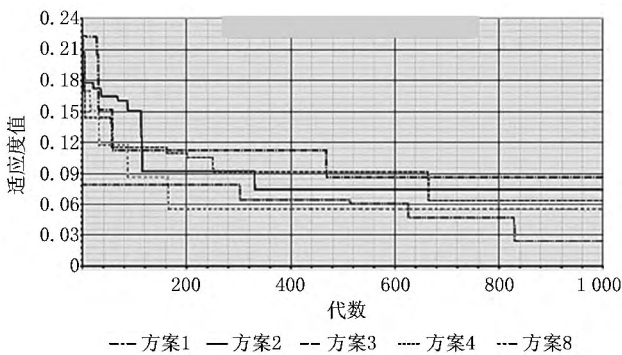


图3 方案1, 2, 3, 4, 8的适应度值训练代数关系图

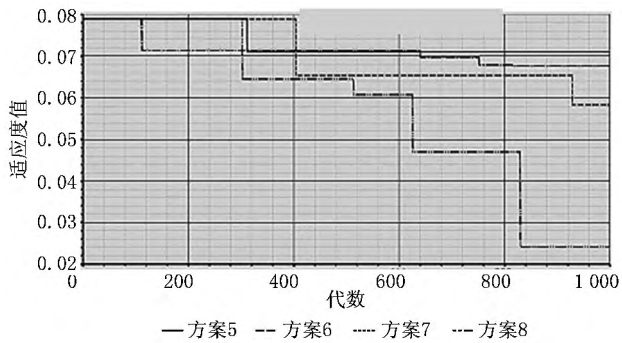


图4 方案5~8的适应度值与训练代数关系图

5 结束语

本文通过将 HFS-LB 优化目标的特征体现在建立初始种群个体的过程中,提出基于负荷平衡选择概率的初始种群建立方法,该方法能够有效提高初始种群中个体的质量,为进化寻优过程打下了良好的基础。全局优化方法采用 DPSADE 算法,该算

法设计了一种新双种群构造和交流进化方式,通过优秀个体从平凡种群向精英种群流动,起到巩固进化优势和保持进化活力的作用,并通过加入随停止代数自适应调整进化参数的方法,提高跳出局部极值的能力。仿真结果证明,同时运用基于负荷平衡选择概率的初始种群建立方法和 DPSADE 算法,能更好地解决混合流水车间负荷平衡排产问题。

下一步待解决的问题是,当增大进化代数时,DPSADE 算法是否还能取得更好的寻优效果,同时如何更好地提高 DPSADE 算法的寻优速度。

参考文献:

[1] THOMOPOULOS N T. Mixed model line balancing with smoothed station assignments [J]. Management Science, 1970,16(9):593-603.

[2] SHTMOYASHIRO S, TSODA K, AWANE H. Input scheduling and load balance control for a job shop[J]. The International Journal of Production Research,1984,22(4):597-605.

[3] HOUGHTON E, PORTOUGAL V. Trade-offs in JIT production planning for multi-stage systems:balancing work-load variations and WIP inventories[J]. International Transactions in Operational Research,1997,4(5/6):315-326.

[4] MEZMAZ M, MELAB N, TALBI E G. An efficient load balancing strategy for grid-based branch and bound algorithm[J]. Parallel Computing,2007,33(4):302-313.

[5] KESKINTURK T, YILDIRIM M B. An ant colony optimization algorithm for load balancing in parallel machines with sequence-dependent setup times[J]. Computers and Operations Research,2012,39(6):1225-1235.

[6] KOUKI S, JEMNI M, LADHARI T. A load balanced distrib-

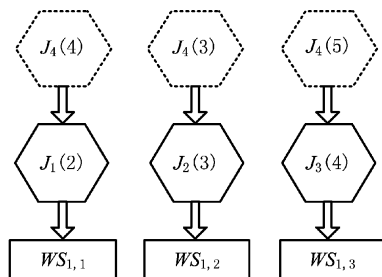
- uted algorithm to solve the permutation flow shop problem using the grid[C]//Proceedings of the 15th IEEE International Conference on Computational Science and Engineering, Washington, D. C., USA, IEEE, 2012; 146-153.
- [7] HUANG R H, YANG C L, CHENG W C. Flexible job shop scheduling with due window—a two-pheromone ant colony approach[J]. International Journal of Production Economics, 2013, 141(2): 685-697.
- [8] LIU Jianguo, ZHU Hengmin. An immune algorithm for load balancing of hybrid flow shop scheduling[J]. Journal of Xidian University, 2006, 33(4): 655-659 (in Chinese). [刘建国, 朱恒民. 混合流水车间负荷平衡调度的免疫算法[J]. 西安电子科技大学学报, 2006, 33(4): 655-659.]
- [9] ZHAN Y, QIU C H. A hybrid genetic algorithm for hybrid flow shop scheduling with load balancing[J]. Key Engineering Materials, 2008, 392-394: 250-255.
- [10] STORN R, PRICE K. Differential evolution—a simple and efficient heuristic strategy for global optimization over continuous spaces[J]. Journal of Global Optimization, 1997, 11(4): 341-359.
- [11] WANG Weiling, LI Junfang. Double-population genetic algorithm for multi-objective job shop scheduling problem[J]. Computer Integrated Manufacturing Systems, 2011, 17(4): 808-815 (in Chinese). [王伟玲, 李俊芳. 求解多目标作业车间调度问题的双种群遗传算法[J]. 计算机集成制造系统, 2011, 17(4): 808-815.]
- [12] MARICHELVAM M K, PRABAHARAN T, YANG X S. Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan[J]. Applied Soft Computing, 2014, 19(2): 93-101.
- [13] BEHNAMIAN J. Development of a hybrid metaheuristic to minimise earliness and tardiness in a hybrid flowshop with sequence-dependent setup times [J]. International Journal of Production Research, 2010, 48(5): 1415-1438.
- [14] WANG Wanliang, YAO Minghai. Hybrid flow-shop scheduling approach based on genetic algorithm[J]. Journal of System Simulation, 2002, 14(7): 863-864 (in Chinese). [王万良, 姚明海. 基于遗传算法的混合 Flow-shop 调度方法[J]. 系统仿真学报, 2002, 14(7): 863-864.]
- [15] WU Lianghong, WANG Yaonan, ZHOU Shaowu, et al. Research and application of pseudo parallel differential evolution algorithm with dual subpopulations[J]. Control Theory and Applications, 2007, 24(3): 453-458 (in Chinese). [吴亮红, 王耀南, 周少武, 等. 双群体伪并行差分进化算法研究及应用[J]. 控制理论与应用, 2007, 24(3): 453-458.]
- [16] WANG Shengyao, WANG Ling. An estimation of distribution algorithm for solving hybrid flow-shop scheduling problem[J]. Acta Automatica Sinica, 2012, 38(3): 437-443 (in Chinese). [王圣尧, 王凌. 求解混合流水车间调度问题的分布估计算法[J]. 自动化学报, 2012, 38(3): 437-443.]
- [17] YANG Zhenyu. An overview of parameter control and adaptation strategies in differential evolution algorithm[J]. CAAI Transactions on Intelligent Systems, 2011, 6(5): 415-423 (in Chinese). [杨振宇. 差分进化算法参数控制与适应策略综述[J]. 智能系统学报, 2011, 6(5): 415-423.]

作者简介:

- 韩忠华(1977—),男,辽宁沈阳人,教授,博士,硕士生导师,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, E-mail: hanzhonghua@sia.cn;
- 董晓婷(1992—),女,四川广元人,硕士,研究方向:车间排产与生产调度算法的工程应用,通信作者, E-mail: dxt199211@163.com;
- 史海波(1966—),男,辽宁沈阳人,研究员,博士,博士生导师,研究方向:生产与运作管理理论、制造过程建模与仿真技术、制造执行系统技术、数字化装备与智能系统技术;
- 朱一行(1990—),男,浙江温州人,硕士,研究方向:车间排产与生产调度算法。

附录:基于负荷平衡选择概率的初始种群构造方法举例

以 4 个工件 $\{J_1, J_2, J_3, J_4\}$ 在工序 OP_1 有 3 个工位 $\{WS_{1,1}, WS_{1,2}, WS_{1,3}\}$ 的分配为例, 即 $M_1 = 3$, 如附图 1 所示。



附图1 基于负荷平衡目标的选择概率控制工件分配到工位

在事件 t_1 发生后, 如果 J_1, J_2, J_3 已被分配, $\sum_{i=1}^4 \sum_{k=1}^3 At_{i,j,k}(t_1) = 3$, 并且已分配任务数为 3, 则工位 $WS_{1,1}$ 上已分配工件 J_1 , 加工时间为 2, 工位 $WS_{1,2}$ 上已分配工件 J_2 , 加工时间为 3, 工位 $WS_{1,3}$ 上已分配工件 J_3 , 加工时间为 4, 表示为 $\{J_1(2), J_2(3), J_3(4)\}$ 的形式。

在事件 t_2 发生时, 要对 J_4 进行加工工位的分配, J_4 在并行工位上的加工时间分别为 $\{4, 3, 5\}$, 即 $Tw_{4,1,1} = 4, Tw_{4,1,2} = 3, Tw_{4,1,3} = 5$, 表示为 $\{J_4(4), J_4(3), J_4(5)\}$ 。

如果工件 J_4 被分配到工位 $WS_{1,1}$, 即 $At_{4,1,1}(t_2) = 1$, 则分配任务加工时间总和 $Ts_{4,1,1}(t_2) = 2 + 3 + 4 + 4 = 13$, 并行工位分配任务加工时间均值

$$\overline{Tw}_{4,1,1}(t_2) = \frac{13}{3},$$

$$Tlb_{4,1,1}(t_2) = \sqrt{(2+4-\overline{Tw}_{4,1,1}(t_2))^2 + (3-\overline{Tw}_{4,1,1}(t_2))^2 + (4-\overline{Tw}_{4,1,1}(t_2))^2} = 2.16。$$

如果工件 J_4 被分配到工位 $WS_{1,2}$, 即 $At_{4,1,2}(t_2) = 1$, 则分配任务加工时间总和 $Ts_{4,1,2}(t_2) = 2 + \overline{Tw}_{4,1,2}(t_2) = 4$, 并行工位分配任务加工时间均值 $3+4+3=12$,

$$Tlb_{4,1,2}(t_2) = \sqrt{(2-\overline{Tw}_{4,1,2}(t_2))^2 + (3+3-\overline{Tw}_{4,1,2}(t_2))^2 + (4-\overline{Tw}_{4,1,2}(t_2))^2} = 2.83。$$

如果工件 J_4 被分配到工位 $WS_{1,3}$, 即 $At_{4,1,3}(t_2) = 1$, 则分配任务加工时间总和 $Ts_{4,1,3}(t_2) = 2 + \overline{Tw}_{4,1,3}(t_2) = \frac{14}{3}$, 并行工位分配任务加工时间均值 $3+4+5=14$,

$$Tlb_{4,1,3}(t_2) = \sqrt{(2-\overline{Tw}_{4,1,3}(t_2))^2 + (3-\overline{Tw}_{4,1,3}(t_2))^2 + (4+5-\overline{Tw}_{4,1,3}(t_2))^2} = 5.35。$$

根据式 (18), 得到 $P_{4,1,1}^{LB}(t_2) = 39.56\%$, $P_{4,1,2}^{LB}(t_2) = 36.31\%$, $P_{4,1,3}^{LB}(t_2) = 24.13\%$ 。工件 J_4 分配给负荷平衡选择概率大的工位 $WS_{1,1}$ 进行加工, 进而可以构造初始种群的个体中基因值, 即表示工件 J_4 在工序 OP_1 上选择工位 $WS_{1,1}$ 的情况, 就是在 (1,2) 之间取一个随机数。