

基于混合差分进化算法的并行机 批处理调度问题研究*

王万良, 范丽霞, 徐新黎, 张 静

(浙江工业大学 计算机科学与技术学院, 浙江 杭州 310023)

摘要: 考虑到实际生产中产品多、批量小的特点,建立了一种带工艺约束的并行机批处理调度优化模型。为解决调度中的分批问题,提出了一种新的基于产品需求量的批量划分方案及批量染色体编码方式,采用两级差分进化算法来解决批量划分和批次调度问题;针对标准差分进化算法收敛速度慢、易出现早熟现象等问题,引入动态随机搜索和随机变异的局部搜索策略,以增强标准差分进化算法的局部搜索能力。测试算例及调度实例的仿真结果表明,该算法能有效地提高算法收敛速度,平衡其全局搜索和局部探索能力。

关键词: 差分进化算法;动态随机搜索;局部搜索;并行机调度;批处理

中图分类号:TH166;TP183

文献标志码:A

文章编号:1001-4551(2012)02-0125-06

New hybrid differential evolution for parallel machines batch scheduling

WANG Wan-liang, FAN Li-xia, XU Xin-li, ZHANG Jing

(College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China)

Abstract: Considering the characteristics of more products, small batches in practical production, a parallel machines batch scheduling model with process constraint was established. In order to solve the problem of splitting, a new batch splitting method based on demand and new chromosome representation was put forward. A new parallel encoding was brought forward to solve both the batch splitting problem and batch scheduling problem. To the problem of low searching speed and premature convergence appeared in standard differential evolution, a new hybrid differential evolution (DE), based on dynamic random search and Chaos optimization, was proposed to enhance local search ability of standard DE. Performance of the proposed algorithm on classic benchmark function and shop scheduling demonstrate that, the proposed algorithm can effectively improve searching speed, balancing the abilities of global search and local search.

Key words: differential evolution (DE); dynamic random search; local search (LS); parallel machines scheduling; batch process

0 引 言

调度的任务是在满足装置设备和工艺要求的条件下,根据市场需求,合理、最佳地安排与组织生产过程,为企业带来显著的经济效益^[1]。随着市场对产品个性化的需求以及市场的多变性,多品种、小批量的生产方式已逐渐成为发展主流,在这种生产方式下,批处理过程已成为现代工业过程的一个重要分支^[2],且生产车间往往存在并行机的情况。由于其不可忽略的产

品切换时间及设备处理柔性等特点,使带工艺约束的并行机批处理调度成为学术界的研究热点。苏生等^[3]为解决实际计划需求的分批问题,提出了一种新颖的单阶段并行机多产品厂短期调度连续时间混合整数规划模型,同时处理需求分批与批次调度任务。文献^[4]针对最小化加权完工时间的等同和非等同并行多级调度问题,提出了一种阶梯遗传算法。文献^[5]研究受工艺约束的非等同机环境下的调度问题,并用遗传算法求解。

收稿日期:2011-09-06

基金项目:国家自然科学基金资助项目(61070043, 60874074);浙江省自然科学基金资助项目(Y1090592)

作者简介:王万良(1957-),男,江苏高邮人,教授,博士生导师,主要从事人工智能、生产调度、网络控制等方面的研究。E-mail: wangwanliang@

批处理调度问题涉及批量划分 (batch splitting 或 lot streaming) 和批次优化 (batch scheduling)。由于批处理过程是按批次进行生产, 且通常设备的生产能力有限, 即批量大小有最高上限和最低下限, 而实际产品计划需求量大小不一, 故批处理调度的任务就是确定每种产品计划需求所需划分的批次数, 以及每批的批量大小, 称批量划分。批量划分完后就是安排这些批次在各设备上的加工顺序, 以及各个批次的开始时间和结束时间, 称为批次优化。

在各类调度问题中, 进化算法作为调度问题的求解算法被广泛研究, 并得到了成功的应用。作为一种新型的智能算法, 差分进化算法^[6]采用实数矢量编码, 原理简单, 受控参数少, 鲁棒性好, 易于实现, 一经提出便受到众多学者的关注和研究, 已在函数优化、车间调度、化工等多个领域取得了广泛领域^[7], 并取得了较好的结果。但标准 DE 的一个重要缺陷就是在进化后期收敛速度较慢, 容易陷入局部最优的情况。因此, 如何避免算法早熟收敛, 增强算法的鲁棒性, 提高算法的全局搜索能力, 一直是工程应用领域所关心的重点。

本研究首先考虑实际生产过程中产品多、批量小的特点及存在并行机的情况, 建立一种带工艺约束的并行机批处理调度优化模型; 为解决订单式生产批量大小不一致的问题及设备容量约束的难点, 提出一种新的基于产品需求量的批量划分方案及批量染色体编码方式, 用以解决在尽可能满足设备加工能力的情况下实现最优分批问题。本研究采用两级差分进化算法解决批量划分和批次调度问题; 针对标准差分进化算法易陷入局部最优的缺陷, 设计基于动态随机搜索和随机变异相结合的局部搜索策略, 以平衡算法的全局搜索能力和局部搜索能力。最后将混合差分进化算法应用于测试算例和调度实例。

1 并行机批处理调度模型建立

1.1 问题描述

在并行机生产环境中, 设车间有 M 台设备, 每种设备的容量不尽相同, 每台设备都有最小生产量和最大生产量; 设备按批次生产, 且每台设备可对任意类型的产品进行加工。企业需要加工的订单有 L 个, 每个订单要求加工一种或多种产品, 每种产品的交货期不同, 不同设备加工不同类型产品的时间、成本也不一样。同时产品切换时还需考虑切换时间和切换成本。根据客户重要程度约束, 每个订单具有不同的超期惩罚权重。带工艺约束的并行机批处理过程要解决的问题是: 如何将接收到的订单并单、拆单, 确定每种产品划分的批数、每批次的大小以及各批次加工顺序和加工设备, 使

得在满足各种约束条件下, 目标函数最优。

1.2 变量定义

本研究中的变量定义如下:

P	加工生产的产品类型
N	产品类型数量
J	设备集合
M	设备数量
O	需要生产加工的订单集合
L	订单个数
D_p	产品 p 的需求量
$BD_{j,p}$	某一时间段设备 j 生产产品 p 的量
$\min V_j$	设备 V_j 允许的最小批量, 即设备的最小生产量
$\max V_j$	设备 V_j 允许的最大批量, 即设备的最大生产量
$\max Cost_{j,p}$	设备 j 在最大生产量下, 生产产品 p 的成本
$con Cost_{j,p}$	设备 j 生产产品 p 的常数成本;
$unit Cost_{j,p}$	设备 j 生产单位产品 p 的成本;
$\max Time_{j,p}$	设备 j 在最大生产量下, 生产产品 p 的时间
$con Time_{j,p}$	设备 j 处理产品 p 所需的常数时间
$unit Time_{j,p}$	设备 j 处理单位产品 p 所需的时间
$tr Cost_{j,p,p'}$	设备 j 处理产品 p 和 p' 的切换成本
$tr Time_{j,p,p'}$	设备 j 处理产品 p 和 p' 的切换时间
β_{p,o_j}	订单 o_j 中产品 p 的超期惩罚系数
Pe_{p,o_j}	订单 o_j 中产品 p 的完工时间
d_{p,o_j}	订单 o_j 中产品 p 的交货期时间
$TS_{j,p}$	设备 j 加工产品 p 的开始时间
$TE_{j,p}$	设备 j 加工产品 p 的结束时间
$po Cost_{p,o_j}$	订单 o_j 中产品 p 的超期惩罚成本

1.3 数学模型

1.3.1 约束条件

(1) 批量约束。在批量生产过程中, 产品是以一个批次为单位生产的, 每台设备 V_j 都有最小、最大生产量 $[\min V_j, \max V_j]$ 。设备在每批次加工中, 产品 p 的生产量 $BD_{j,p}$ 必须满足:

$$\min V_j \leq BD_{j,p} \leq \max V_j \quad (1)$$

(2) 生产与需求约束。所有设备上加工产品 p 的总量应等于其需求量:

$$\sum_{j \in J} BD_{j,p} = D_p, \forall p \in P \quad (2)$$

(3) 切换成本和切换时间约束。在批量生产时, 每个设备都是柔性生产线, 可以加工任何产品, 但在不同产品间进行切换时, 为保证下一种产品的质量, 需

要在加工产品前对设备进行清洗,因而会产生切换成本,包括清洗成本和切换时间。在设备 j 先后加工两种不同产品时的时间差应大于切换时间:

$$TS_{j,p'} - TE_{j,p} \geq trTime_{j,p,p'} \quad (3)$$

(4) 交货期约束。不同订单有不同的交货期,有时针对同一订单内的产品,客户也有不同的要求,其中某些产品需要加急生产,具有不同的超期惩罚系数 $\beta_{p,o}$, 对企业重要客户的超期惩罚系数会比一般客户要大。在制订设备排产计划时需要考虑产品的交货期。如果产品的完工时间 $Pe_{p,o}$ 超出了交货时间 $d_{p,o}$, 则会产品超期惩罚成本 $poCost_{p,o}$, 如下式所示:

$$poCost_{p,o} = \beta_{p,o} \times \max(0, (Pe_{p,o} - d_{p,o})) \quad (4)$$

1.3.2 目标函数

并行机批处理调度是企业依据接收到的订单,综合以上约束,确定各产品划分的批次,并安排各产品在不同设备上的加工批次和各批次大小,在尽可能满足订单要求的前提下,使企业的总生产成本最小,目标函数如下式所示:

$$f = \sum_{j \in V, p \in P} (conCost_{j,p} + unitCost_{j,p} \times BD_{j,p}) + \sum_{j \in V, p \in P} trCost_{j,p,p'} + \gamma \sum_{p \in P, o \in O} \beta_{p,o} \times \max(0, (Pe_{p,o} - d_{p,o})) \quad (5)$$

式中:生产成本包括加工成本、切换成本和超期惩罚成本; γ 是超期惩罚成本和另外两项成本的比例系数。加工成本由常数成本和变化成本构成。

在实际生产中,设备在最大生产量条件下加工产品 p 的加工成本 $\max Cost_{j,p}$ 和加工时间 $\max Time_{j,p}$ 可以通过经验预先给出。生产加工成本和时间主要由加工产品的数量决定,同时与产品的加工工艺以及设备容量有关。

1.4 批量划分策略

实际生产企业中,每个订单对同一种产品的需求不尽相同,有的订单的需求量可能超出了最大设备的最大生产量,有的订单的需求量可能小于最小设备的最小生产量,因此需要把不同客户的订单合并成一个或多个批次,即并单,或把单个客户的需求分成几个批次,即拆单。

本研究提出了一种新的基于产品需求量的批量划分方案,即不考虑产品来自哪个订单,首先将所有订单的同种产品放在一起加工,总和为 D_p ,并计算全部设备的平均最大生产量 $\bar{V}$,则对产品 p 的分批数目定义为 $B_p = [D_p / \bar{V}]$, $[\cdot]$ 表示向上取整数。

在实际生产中,各任务的生产应尽可能满足设备的最大生产量,因此对待加工生产的某种产品,它最多只可能有一批不满足最大生产量的条件。基于这种思想,本研究假设某种产品 p 的分批数目为 B_p ,则前 $(B_p - 1)$ 批产品需满足某台设备最大生产量的条件,并且最后一批的加工量大小, $BD_{p,B_p} = D_p - \sum_{B_i=1}^{B_p-1} BD_{p,B_i}$ 应保证能在某台设备中生产,即大小应位于最小设备的最小生产量和最大设备的最大生产量间。本研究找出所有满足以上条件的设备组合方式,即产品 p 的分批方案数 K_m^p ,选择其中一种方案对应一种分批方法。

2 求解并行机批处理调度问题的混合差分进化算法

2.1 编、解码方案

研究并行机批处理调度问题,需要确定各产品划分的批次数,以及各批次的加工设备和加工顺序。算法采用两级染色体的编码方式,分别表示为批次染色体和批量染色体。其中,批量划分需要确定产品划分的批次数及各批量的大小,批次优化确定各子批次加工顺序。

批量染色体,记为 $chrome1$,本研究对批量染色体提出了一种新的编码方案,该染色体分为前后两部分,前一部分表示各产品选择的分批方案号 $K_i^p (1 \leq i \leq m)$,后一部分表示具体的每种产品每批的大小,以无意义的“0”作为间隔,如图1所示。其中 id 中的数字 k 表示第 k 中产品。

批次染色体,记为 $chrome2$,采用实数编码方式。解码时首先需要将批次染色体 $chrome2$ 转化成离散加工序列,本研究将批次染色体上每一位的值从小到大排列,得到序列 $chrome2'$,并根据每种产品的划分批数按顺序依次分配给各产品。假设共有3种产品,产品划分的批数分别为3、2、3,批次染色体转化为离散加工序列的一个示例如表1所示, $chrome2'$ 表示 $chrome2$ 中各位大小的排序, $product$ 表示对应产品号, $batch$ 表示各种产品对应的批次。

表1 批次染色体转化为离散加工序列的示例

$chrome2$	0.31	0.75	0.53	0.40	0.28	0.64	0.19	0.84
$chrome2'$	3	7	5	4	2	6	1	8
$product$	1	3	2	2	1	3	1	3
$batch$	1	1	1	2	2	2	3	3

$$chrome1 \quad K_1^1, K_1^2, \dots, K_1^p \mid 0 \mid BD_{1,1}, \dots, BD_{1,B_1} \mid BD_{2,1}, \dots, BD_{2,B_2} \mid BD_{k,1}, \dots, BD_{k,B_k} \mid BD_{p,1}, \dots, BD_{p,B_p}$$

$$id \quad 1, \dots, 1 \mid 2, \dots, 2 \mid k, \dots, k \mid p, \dots, p$$

图1 批量染色体

本研究根据 *chrome1* 和 *chrome2'*, 按以下步骤得到解码方式:

Step 1: 设置各设备的开始空闲加工时间 $MST_j=0, j=1, \wedge M$ 。设置 $len=1$, 执行 Step 2;

Step 2: 选择 *chrome2'* 上从左到右第 len 位基因 l , 获得其加工产品 p , 批次号 k 。结合 *chrome1*, 根据基因位获得该批批量大小 BD_{pk} ;

Step 3: 根据 Step 3.1~Step 3.3 为待安排任务分配设备, 进行准备、加工操作;

Step 3.1: 根据批量大小 BD_{pk} 选择合适的加工设备, 若该批量满足某台设备最大生产量的条件, 则安排批量到该类型的设备上; 否则, 充分考虑设备的利用率, 将其安排到合适的设备上加工;

Step 3.2: 确定好加工设备类型后, 将其安排到某一具体的设备中。若该类型的设备只有一台, 则直接加工, 否则, 选择在该类型设备中最早完工的设备上加工。计算该批量加工的开始时间和结束时间;

Step 3.3: 如果设备上还没有加工任务, 则不需计算清洗时间及清洗成本; 否则, 比较该次任务和设备上一次的加工任务是否相同, 若不相同, 则需计算清洗时间及清洗成本; 否则, 不需要;

Step 4: 刷新各设备的最早开始空闲时间;

Step 5: 设置 $len=len+1$, 如果 $len \leq length$, 转到 Step 2。

2.2 局部搜索策略

在搜索过程中基本差分进化算法易出现局部收敛的现象, 严重影响算法的性能。针对这一问题, 本研究设计了一种新的局部搜索策略, 以增强算法的局部搜索能力。动态随机搜索策略用于调整批次染色体, 在批量划分不变的情况下, 搜索工序排序的邻域解; 随机变异操作用于调整批量染色体, 在批次加工顺序不变的情况下, 对产品重新进行批量划分。

动态随机搜索^[9], 该算法由一般搜索 (general search) 和局部搜索 (local search) 两部分组成。在局部搜索阶段, 从全局搜索得到的解出发, 通过不断缩小随机搜索空间, 进一步得到在该解邻域的最优解, 因此具有很强的局部搜索能力。其算法流程如下:

Step 1: 初始化局部搜索迭代次数 $Epoch$, 搜索步长 r , 每个步长搜索次数 N 。 $X_{current}=X_{best}$, $epoch=0$;

Step 2: $n=0$;

Step 3: 产生随机向量 dx , 其中 $dx \in [-r, r]$;

Step 4: 更新 $epoch$, $epoch=epoch+1$;

Step 5: $f_{new}=f(X_{current}+dx)$, 如果 $f_{new} < f_{best}$, 则 $f_{best}=f_{new}$, $X_{best}=X_{current}+dx$, 转到 Step 7;

如果 $f_{new} < f_{current}$, 则 $f_{current}=f_{new}$, 转到 Step 7;

Step 6: $f_{new}=f(X_{current}-dx)$, 如果 $f_{new} < f_{best}$, 则 $f_{best}=f_{new}$, $X_{best}=X_{current}-dx$, 转到 Step 7;

如果 $f_{new} < f_{current}$, 则 $f_{current}=f_{new}$, 转到 Step 7;

Step 7: $n=n+1$, 如果 $n < N$, 转到 Step 3;

Step 8: $r=r \times 0.9$;

Step 9: 如果 $epoch < Epoch$, 转到 Step 2, 否则结束;

其中, X_{best} 为搜索到的全局最好解, $f_{new}, f_{best}, f_{current}$ 为对应解处的适应度值。

同时引入随机变异操作对批量染色体进行局部搜索。具体操作为: 随机选择一条染色体, 并随机选取一个变异位, 重新对该产品批量划分。

2.3 算法实现

本研究将 DE 与 LS 相结合得到改进的 DLHDE 算法。为简便起见, 将 DE 种群进化记为操作 $Oper_1$, LS 记为操作 $Oper_2$ 。

DLHDE 用于求解并行机批处理调度的具体步骤如下:

Step 1: 设置参数。设定种群规模 N_p , 缩因子 F , 交叉因子 Cr , 最大迭代次数 G_{max} , 局部搜索迭代次数 $Epoch$, 搜索步长 r , 每个步长搜索次数 N , 变异概率 p_m 等参数;

Step 2: 种群初始化。采用随机初始化的方式产生批量染色体和批次染色体;

Step 3: 操作 $Oper_1$ 。对批量染色体和批次染色体分别进行变异、交叉操作, 分别生成变异个体 V_i^g , 试验个体 U_i^g ;

Step 4: 计算适应度值, 选择下一代种群 X_i^{g+1} , 更新最优值和最优个体;

Step 5: 操作 $Oper_2$ 。对种群 10% 个体执行局部搜索策略, 搜索其邻域解; 计算适应度函数, 更新最优值和最优个体;

Step 6: 若最优值连续指定代未进化, 对种群 10% 个体进行重新初始化, 并用全局最优解代替种群中的一个解;

Step 7: $G=G+1$, 如果 $G < G_{max}$, 转到 Step 2, 否则结束。

2.4 算法复杂度分析

从上面的流程可以看出, DLHDE 算法的时间主要花费在迭代过程中。对种群规模为 N_p , 迭代次数为 G_{max} , 动态随机搜索次数为 $Epoch$ 的问题, 每次迭代中都包含了批量染色体和批次染色体的差分操作、解码、选择和局部搜索操作。差分操作和解码的时间复杂度都为 $o(N_p \times len1)$, 选择操作的时间复杂度为 $o(N_p)$, 局部搜索操作的时间复杂度为。因此, 整个算法的时间复杂度为:

$$o(N_p, G_{max}) = G_{max} \times (o(N_p \times len1) + o(N_p \times len1) + o(N_p) + o(len1 \times Epoch)) = G_{max} \times (2o(N_p \times len1) + o(len1 \times Epoch) + o(N_p))$$

可见, DLHDE 的时间复杂度与染色体长度 $len1$, 种群规模 N_p , 迭代次数 G_{max} 和局部搜索次数 $Epoch$ 有关。

3 实验仿真与分析

3.1 算例测试

为了验证本研究提出的混合差分进化算法 (DLHDE) 的性能, 笔者采用文献[10]的两种不同规模的并行机调度实例进行仿真。

本研究采用 VC6.0 编程,运行环境 Intel Pentium CPU,主频 2.66 GHz,1 GB 内存。算法参数设置:种群规模 50,迭代次数 100。缩放因子 $F=0.5$,交叉因子 $CR=0.7$ 。

本研究对每个问题独立运行 20 次,结果如表 2 所示。其中, $n \times m$ 表示 n 个工件、 m 台机器, BE、WT、AV、SD、RT 分别表示 20 次独立运行过程中算法所取得的最优解、最差解、平均值、标准差和运行时间。

表 2 DE、DLHDE 算法运行结果

$n \times m$	7×3					30×10				
	BE	WT	AV	SD	RT	BE	WT	AV	SD	RT
DE	10	10	10	0	0.03	40	41	40.6	0.49	0.15
DLHDE	10	10	10	0	0.27	39	40	39.65	0.48	1.67
文献[10]	10	--	--	--	--	41	--	--	--	--

从表 2 中可以看出,对 7×3 的小规模问题,3 种算法都能取得最优解,但对于大规模问题,DLHDE 相对于 DE 和文献中算法具有更好的优化性能,可见 DLHDE 具有较好的局部搜索能力。

对 30×10 问题,两种算法独立运行 20 次最优值结果如图 2 所示,横坐标表示迭代次数,纵坐标表示最优值。从图 2 中可看出,局部搜索策略的加入可有效加强算法的搜索能力,进一步提高了算法的稳定性。

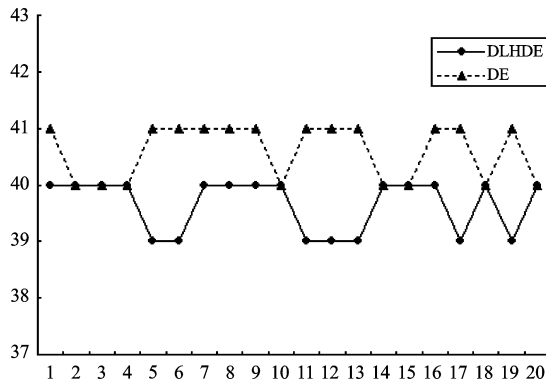


图 2 30×10 两种算法独立运行 20 次结果

3.2 调度实例

本研究对印染企业的某一个染色车间生产实际中获得一定量的样本数据进行仿真实验。染色加工是印染企业生产的核心,因此染色车间调度可看作是单阶段并行机批处理过程。假设该车间有 20 台大小类型不同的设备,设备信息、设备在其最大生产量下清洗不同颜色深浅度的产品所需的时间(单位:h)、加工时间(单位:h)和成本如表 3 所示。企业某一时间接收 9 个订单,需加工生产 10 种产品,其所要求加工生产的产品的颜色深浅度、需求量、交货时间、超期惩罚权重如表 4 所示。另外,取切换成本系数 α, β 分别为 0.9、0.7,在目标函数中加工成本、切换成本和超期惩罚成本之间的比例系数 γ 根据经验值取 10。

同时假设常数成本 $conCost_{j,p} = maxCost_{j,p} / 3$,单位成

表 4 需求排产的订单信息

订单代号	产品编号	颜色深浅度	需求量/kg	交货时间/h	超期惩罚权重
o_1	1	24	90	26	2
	2	89	100	27	1
	3	60	40	30	1.5
o_2	1	24	50	26	1.1
	2	89	90	35	1.6
o_3	1	24	130	48	1.3
	2	89	90	40	1.2
o_4	3	60	110	60	1.9
	4	25	60	7	8
o_5	5	62	120	9	8.5
	6	88	500	55	1
o_7	7	98	60	18	1.4
	8	100	80	55	1.2
o_8	7	98	88	50	2.3
	8	100	100	80	2.1
	9	120	135	65	1.8
o_9	10	145	90	72	9.5

表 3 设备在最大生产量下加工不同颜色深浅度产品所需的加工时间/h-加工成本-清洗时间

设备代号	数量	最小(最大)生产量	产品颜色深浅度									
			24	60	89	25	62	88	98	100	120	145
v_1	3	1(10)	0.8-9 -0.1	0.9-9.8 -0.15	1.0-10.2 -0.25	0.8-9 -0.15	0.9-9.8 -0.5	1.0-10.2 -0.25	1.1-10.3 -0.3	1.1-10.5 -0.3	1.2-10.7 -0.35	1.4-11 -0.45
v_2	1	5(30)	2.1-26 -0.1	2.4-29 -0.15	2.7-30.5 -0.3	2.1-26 -0.35	2.4-29 -0.25	2.7-30.5 -0.25	2.8-32 -0.4	2.8-32 -0.45	3.1-33.5 -0.5	3.4-35 -0.55
v_3	5	5(50)	3.8-45 -0.2	4.3-48 -0.3	4.6-50 -0.45	3.8-45 -0.2	4.3-49 -0.3	4.6-50 -0.45	4.7-50.5 -0.5	4.7-50.6 -0.5	5.0-52.5 -0.6	5.2-53.5 -0.65
v_4	1	10(85)	6.4-76 -0.25	7.2-83 -0.35	8.0-86.5 -0.45	6.4-76 -0.25	7.2-83.1 -0.35	8.0-86 -0.45	8.2-87 -0.55	8.2-87.5 -0.6	8.8-89 -0.65	9.5-90.5 -0.75
v_5	1	10(90)	7.0-80 -0.3	7.8-88 -0.45	8.5-91 -0.55	7.1-80 -0.3	7.8-88.2 -0.45	8.5-90.8 -0.55	9-92 -0.65	9-92.2 -0.65	9.6-94 -0.8	10.8-96 -0.8
v_9	7	20(100)	7.8-88 -0.35	8.8-97 -0.5	9.8-100 -0.65	7.8-88.5 -0.35	8.8-97.5 -0.5	9.8-100 -0.65	10-101 -0.75	10.1-101 -0.85	10.6(103) 1.0	11.4-103.5 -1.05
v_7	2	50(500)	33.5-455 -0.45	35.8-495 -0.65	37.8-515 -0.8	34.1-455 -0.45	36.2-495 -0.65	37.8-515 -0.8	38.5-530 -0.95	38.5-531 -0.95	40(550) 1.15	41.5-568 -1.2

本 $unitCost_{j,p}=2\max Cost_{j,p}/3\max V_j$, 设备生产单位产品 p 的常数加工时间 $conTime_{j,p}=2\max Time_{j,p}/3$, 单位加工时间 $unitTime_{j,p}=\max Time_{j,p}/3\max V_j$ 。

为了测试算法性能,本研究将标准差分算法和本研究提出的混合差分算法结果进行仿真,参数设置一样,运行结果如图 3 所示。DLHDE 算法 30 次独立运行的结果如图 4 所示。

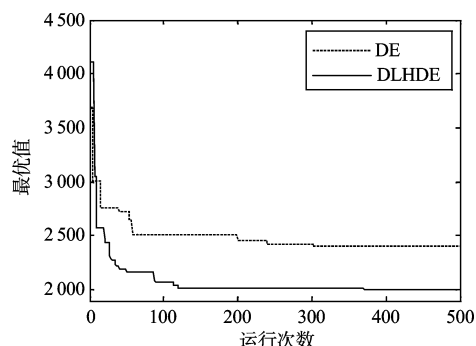


图 3 DLHDE、DE 结果比较

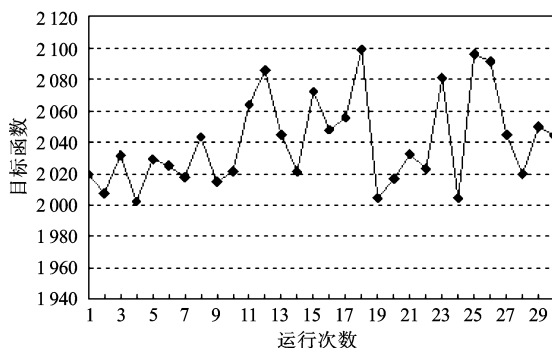


图 4 DLHDE 30 次独立运行

根据订单产品的需求总量及设备信息,可得各产品的划分批次数为 $[3, 2, 3, 1, 2, 5, 2, 2, 2, 1]$, 算法运行结果的最优解包含一条批量染色体和批次染色体,分别为 $[17\ 7\ 18\ 1\ 7\ 3\ 4\ 6\ 7\ 1\ 0\ 85\ 100\ 85\ 100\ 50\ 90\ 100\ 90\ 60\ 100\ 20\ 10\ 10\ 10\ 50\ 420\ 50\ 98\ 90\ 90\ 100\ 35\ 90]$ $[16\ 6\ 9\ 8\ 1\ 17\ 13\ 7\ 20\ 3\ 12\ 2\ 4\ 5\ 10\ 18\ 14\ 19\ 23\ 22\ 2111]$, 最优目标函数为 2 004.473 511, 超期惩罚为 0。

为和文献[11]中的结果作比较,本研究将设备清洗时间设为 0, 独立运行 30 次, 结果如表 5 所示。

表 5 混合差分进化算法和文献[11]设备排产结果比较

算法	最好值	最差值	平均值	标准差	运行时间/s
遗传算法	2 689.39 3	195.91 2	890.181	146.268 2	13.8
量子遗传算法	2 662.37 3	131.39 2	871.162	130.158 4	14.5
混合差分算法	1 735.71 2	148.85 2	037.07	86.497 8	1.2

结果表明,本研究中所提的批量划分方法及批量染色体编码方式的有效性, 不仅明显减小了子批数量, 而且大大减小了生产成本, 运行时间也大大减小, 提高了效益。

从批量染色体可以看出, 各子批次的大小几乎都达到了某台设备最大生产量的条件, 每种产品最多只有一批不满足条件, 这样就实现了最初批量划分的目的。

4 结束语

本研究首先建立了带工艺约束的并行机批处理调度优化模型, 采用两级染色体编码方案, 同时对批量染色体和批次染色体进行了优化; 提出了一种新的基于产品需求量的批量划分方案及批量染色体编码方式, 有效地解决了在尽可能满足设备加工能力的情况下实现最优分批。针对标准差分进化算法易陷入局部最优的缺陷, 设计了基于动态随机搜索和随机变异的局部搜索算子, 分别用于调整批次染色体和批量染色体的邻域解。

本研究通过并行机测试算例对算法性能的分析比较, 验证了局部搜索策略的加入, 有效地提高了算法的优化性能; 最后将其应用于并行机批处理调度实例, 通过仿真及与其他算法的比较分析, 验证了算法及模型的有效性和可行性。

参考文献(References):

- [1] 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.
- [2] 陈昌领, 刘长龄, 袁德成, 等. 单阶段多产品批处理过程的短期调度 1. 基本模型的建立[J]. 信息与控制, 2002, 31(2): 106-111.
- [3] 苏 生, 战德臣, 徐晓飞. 一种并行机多产品厂短期调度连续时间模型[J]. 计算机集成制造系统, 2006, 12(4): 494-498.
- [4] 周辉仁, 郑丕谔, 王海龙. 基于递阶遗传算法最小加权完工时间并行机调度[J]. 系统仿真学报, 2008, 20(13): 3510-3513.
- [5] 何桂霞. 特殊工艺约束下并行机多目标调度问题[J]. 工业工程, 2008, 11(6): 130-135.
- [6] STORN R, PRICE K. Differential Evolution—a Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces[R]. Berkeley: University of California, 1995.
- [7] 刘 波, 王 凌, 金以慧. 差分进化算法研究进展[J]. 控制与决策, 2007, 22(7): 721-729.
- [8] 贾东立, 郑国莘. 基于混沌和高斯局部优化的混合差分进化算法[J]. 控制与决策, 2010, 25(6): 899-902.
- [9] HAMZACEBI C, KUTAR F. Continuous functions minimization by dynamic random search technique [J]. **Applied Mathematical Modelling**, 2007, 31(10): 2189-2198.
- [10] 王 凌. 车间调度及其遗传算法[M]. 北京: 清华大学出版社, 2003.
- [11] 蒋佳颖. 基于量子遗传算法的单阶段多产品批处理调度问题的研究与应用[D]. 杭州: 浙江工业大学信息工程学院, 2011.

[编辑: 罗向阳]