

## 基于DEPSO的模糊时间ZW多产品厂间歇调度

徐震浩, 李青青, 顾幸生

(华东理工大学 化工过程先进控制和优化技术教育部重点实验室, 上海 200237)

**摘要:** 研究以最小化完工时间为目标的模糊加工时间零等待多产品厂间歇调度问题, 提出一种基于差分进化粒子群优化(DEPSO)的间歇调度算法. 以基本粒子群算法为整体进化框架, 采用基于反向学习的方法初始化种群, 引入群体极值保持代数作为阈值, 利用基于排序的差分进化算法优化粒子个体极值位置, 改变粒子的搜索范围, 防止粒子陷入局部极值. 仿真实验验证了所提算法在解决模糊加工时间零等待多产品厂间歇调度问题上的有效性和优越性.

**关键词:** 粒子群算法; 离散差分进化; 模糊时间; 零等待; 间歇调度

中图分类号: TP301

文献标志码: A

## A study of the DEPSO-based multiproduct plants batch scheduling under uncertainty with zero wait

XU Zhen-hao, LI Qing-qing, GU Xing-sheng

(Key Laboratory of Advanced Control and Optimization for Chemical Processes, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China. Correspondent: XU Zhen-hao, E-mail: xuzhenhao@ecust.edu.cn)

**Abstract:** For minimizing the total flow time of multiproduct plants batch scheduling under uncertain processing time with zero wait, a batch scheduling algorithm based on the differential evolution particle swarm optimization is proposed. With basic particle swarm optimization as the algorithm's framework, the opposition-based learning algorithm is used to initialize population, the group extreme is introduced to keep algebra as a threshold, and the permutation-based differential evolution algorithm is used to optimize the individual extremum so as to change the search range and prevent the particle falling into local extremum. The simulation results show the effectiveness of the proposed algorithm in solving multiproduct plants batch scheduling under uncertainty with zero wait.

**Keywords:** particle swarm optimization; discrete differential evolution; uncertain processing time; zero wait; batch scheduling

### 0 引言

多产品厂间歇生产过程比连续生产过程更加灵活, 操作柔性更高, 同一流程线可以生产各种规格不同的产品, 因此多产品厂间歇生产过程特别适合多品种、小批量、工艺复杂和附加值高的化学品<sup>[1]</sup>. 在许多实际生产过程中会产生不稳定的中间产品, 需要采用零等待的生产方式来保证产品和生产安全<sup>[2]</sup>. 同时, 在生产过程中, 往往存在很多不确定性, 如加工时间不确定、生产设备性能不稳定、周围环境的影响以及操作技术水平等因素, 这些都会对生产过程产生影响. 在生产调度中充分考虑不确定因素, 有助于提高生产过程的效率和抗干扰能力, 因此, 考虑零等待多

产品厂间歇调度问题的不确定性具有重要的学术价值和现实意义.

粒子群算法和差分进化算法都是性能优异的群体优化算法, 两者的混合算法有多种变体且优势显著: Hendtlass<sup>[3]</sup>提出了DEPSO算法, 以基于Gbest模型的PSO算法为主体, 将DE算法以周期 $T$ 作用于每个个体的当前位置; Zhang等<sup>[4]</sup>提出的DEPSO算法先将基于Gbest模型的PSO算法作用于整个种群, 再利用DE算法对所有的个体最优位置进行改进; Epitropakis等<sup>[5]</sup>提出的DEPSO算法利用PSO改变每代粒子的位置, 将DE算法选择性地作用于个体最优位置发生改变的粒子; Caponio等<sup>[6]</sup>提出了一种两阶段DEPSO算

收稿日期: 2014-11-13; 修回日期: 2015-03-17.

基金项目: 国家自然科学基金项目(61104178, 61174040).

作者简介: 徐震浩(1976—), 女, 副研究员, 博士, 从事生产计划与调度等研究; 李青青(1986—), 女, 硕士生, 从事生产调度的研究.

法,该算法以DE算法为母体,PSO只在初始阶段运行,从而改进了随机生成的初始种群;Ali等<sup>[7]</sup>提出的两阶段DEPSO以基于Gbest模型的PSO为母体,DE算法负责空间探索,当DE算法搜索达到收敛的阈值时,由DE切换到PSO.

以上的研究主要是针对连续空间问题,而调度问题是一类典型的离散空间优化问题.本文根据模糊加工时间零等待多产品厂间歇调度模型,提出一种改进的差分进化粒子群优化(DEPSO)算法,以最小化完工时间为目标,对模糊加工时间零等待多产品厂间歇调度问题进行求解.算法以基本粒子群优化(GPSO)算法为母体,当粒子最优值保持代数达到阈值时,搜索陷入局部最优,采用离散的差分进化算法对个体最优位置进行优化,使粒子能够开辟更好的搜索区域,增强粒子的多样性,同时在每一代进化过程中,对粒子进行变邻域搜索,提高算法的搜索精度.

## 1 问题描述与模型建立

零等待操作方式要求产品的中间产物在某个设备上加工完成后,立即转移到下个加工设备上.一般对零等待多产品厂间歇生产过程做如下假设:1)所有产品在每个设备上的加工次序相同;2)产品之间没有优先性;3)一个设备不能同时加工多个产品,一种产品不能同时被多个设备加工;4)产品加工过程不允许中断,工件的传送时间及设备的清洗时间都计算在加工时间内;5)原料和产品有足够的存储容量.

根据以上假设,加工时间为模糊数的零等待多产品厂间歇生产过程可以描述为:有 $N$ 个待加工产品在 $M$ 个设备上加工;第 $i$ 个产品在第 $j$ 个设备上的加工时间为 $\tilde{T}_{i,j}$ ,该时间包括进出料时间、清洗时间和加工处理时间等,其值是在一定范围内变化的量,每个产品加工完成时间为 $\tilde{C}_{i,j}$ ,两者都采用模糊数表示.以最小化完工时间为目标,即 $\min(\tilde{C}_{k_N,M})$ ,处理此问题的模型如下.

假设 $\pi = (k_1, k_2, \dots, k_N)$ 为所有产品的一个排序, $k_i$ 为整数,且 $1 \leq k_i \leq N$ , $\tilde{C}_{k_i,j}$ 为产品 $k_i$ 在设备 $j$ 上的完工时间, $\tilde{C}_{k_N,M}$ 为此序列的总完工时间,则目标函数即为 $\min(\tilde{C}_{k_N,M})$ ,递推公式表示如下:

$$\tilde{C}_{k_1,1} = \tilde{T}_{k_1,1}; \quad (1)$$

$$\tilde{C}_{k_1,j} = \tilde{C}_{k_1,j-1} + \tilde{T}_{k_1,j}, \quad j = 2, 3, \dots, M; \quad (2)$$

$$\tilde{C}_{k_i,1} = \tilde{C}_{k_{i-1},1} + \tilde{T}_{k_i,1} + \Delta \tilde{t}_{k_i}, \quad i = 2, 3, \dots, N; \quad (3)$$

$$\Delta \tilde{t}_{k_i} = \max \left\{ 0, \max_{2 \leq j \leq M} \left[ \tilde{C}_{k_{i-1},j} - \left( \tilde{C}_{k_{i-1},1} + \sum_{l=1}^{j-1} \tilde{T}_{k_i,l} \right) \right] \right\}, \quad i = 2, 3, \dots, N, \quad j = 2, 3, \dots, M; \quad (4)$$

$$\tilde{C}_{k_i,j} = \tilde{C}_{k_i,1} + \sum_{l=2}^j \tilde{T}_{k_i,l}, \quad i = 2, 3, \dots, N, \quad j = 2, 3, \dots, M. \quad (5)$$

其中:式(1)和(2)为产品 $k_i$ 在每个设备上的模糊完工时间;式(3)为产品 $k_i$  ( $i = 2, 3, \dots, N$ )在第1个设备上的模糊完工时间,由于产品采用零等待的加工方式,除了第1个产品外,其他产品在第1个设备上的开始加工时间都可能会有延迟;式(4)为模糊延迟时间;式(5)为工件 $k_i$  ( $i = 2, 3, \dots, N$ )在设备 $j$  ( $j = 2, 3, \dots, M$ )上的模糊完工时间,当 $i = N, j = M$ 时,即为最后一个工件在最后一个设备上的模糊完工时间,也就是整个加工工序的最大模糊完工时间.模糊加工时间的零等待多产品间歇调度的最终目标是找到最优序列,使得 $\tilde{C}_{k_N,M}$ 最小.

在实际情况下,很难估计不确定的加工时间的概率分布规律,只能根据经验或历史数据大致估计它的区间分布,应用三角模糊数来处理加工时间的不确定性,将加工时间考虑为某一个区间范围可以使调度模型更加符合生产实际<sup>[8-9]</sup>.假设模糊加工时间为 $\tilde{T} = (T^L, T^M, T^U)$ .其中: $T^L$ 为最优加工时间, $T^M$ 为最可能加工时间, $T^U$ 为最坏加工时间.基于三角模糊数的加法运算为

$$\tilde{T}_1 + \tilde{T}_2 = (T_1^L + T_2^L, T_1^M + T_2^M, T_1^U + T_2^U). \quad (6)$$

基于三角模糊数的加法运算能够计算出产品的模糊完工时间 $\tilde{C} = (C^L, C^M, C^U)$ ,其隶属度函数为

$$\mu_C(x) = \begin{cases} \frac{x - C^L}{C^M - C^L}, & C^L < x \leq C^M; \\ 0, & x \leq C^L \text{ 或 } x > C^U; \\ \frac{C^U - x}{C^U - C^M}, & C^M < x \leq C^U. \end{cases} \quad (7)$$

根据文献[10],定义 $\tilde{C} = (C^L, C^M, C^U)$ 的左积分为

$$I_L(C) = C^M - \int_{C^L}^{C^M} \mu_C(x) dx = \frac{C^L + C^M}{2}, \quad (8)$$

右积分为

$$I_R(C) = C^M - \int_{C^M}^{C^U} \mu_C(x) dx = \frac{C^M + C^U}{2}, \quad (9)$$

则 $\tilde{C}$ 的全积分为

$$I_\beta(C) = \beta I_L(C) + (1 - \beta) I_R(C), \quad 0 \leq \beta \leq 1. \quad (10)$$

其中: $I_L(C)$ 为 $\tilde{C}$ 最好的状态; $I_R(C)$ 为 $\tilde{C}$ 最差的状态; $\beta$ 为乐观系数,由调度决策者根据实际生产状态来调整, $\beta$ 越大,表示决策者的态度越乐观.通过以上定义可以对不同的调度方案进行评价,如果 $I_\beta(C_1) < I_\beta(C_2)$ ,则 $\tilde{C}_1 < \tilde{C}_2$ ,进而通过优化可以找到一个调度方案,使得 $\tilde{C}_{k_N,M}$ 最小.

## 2 差分进化粒子群混合算法

基于粒子群算法的特点和离散差分进化算法在解决组合优化问题上的优势,本文提出一种差分进化粒子群混合算法(DEPSO).

### 2.1 算法思路

1) 基于反向学习的种群初始化.

一般情况下, 调度问题都是在解空间随机产生一些候选解作为初始种群. 为了提高初始种群的质量, 同时保证其分散度, 本文引入基于反向学习的机制, 为每个初始解产生相对应的反向解, 并从这两种解中选择适应度较优的解作为最终的初始解<sup>[11]</sup>. 对于一个随机加工序列  $\pi = (k_1, k_2, \dots, k_N)$ , 其反向解为  $\pi = (\bar{k}_1, \bar{k}_2, \dots, \bar{k}_N)$ , 其中  $\bar{k}_i = 1 + N - k_i$ , 然后分别计算两者的适应度值, 选择适应度值较小的个体作为初始种群粒子有助于改进进化过程的收敛速度.

## 2) 基于阈值的 DE 操作.

传统差分进化算法在求解连续优化问题上有很好的效果, 但在求解具有离散特征的调度优化问题时, 需要引入修复操作来保证进化过程中候选解的可行性, 这样对算法的求解性能有很大的影响. 针对这个问题, 文献 [12] 提出了基于排列的离散差分进化 (PDE) 算法. 这种操作方式不需要选择缩放因子和交叉概率, 是一种通过改变位置关系的新型变异交叉操作, 过程更加简单, 也不需要解进行修复, 直接产生可行解, 更加适合解决具有离散特征的生产调度问题.

在进化初期, PSO 算法下降速度比较快, 但对于多峰问题, 又非常容易陷入局部极小值, 使搜索停滞. 而基于排序的离散差分进化算法, 通过变异交叉操作, 能够对粒子产生局部扰动, 使其跳出局部极小值. 为了在保证群体进化速度的同时, 使粒子能够不断地跳出局部极小值, 本文提出的 DEPSO 算法总体上遵循 PSO 算法的步骤, 对粒子进行全局搜索, 保持粒子的多样性, 同时, 在每一代的进化过程中, 记录全局最优值保持的代数  $dir$ , 当其保持代数超过设定阈值  $stage$  时, 对每个粒子最优位置进行离散 DE 搜索, 然后对  $dir$  置零. 阈值  $stage$  一般选择小于 10 的整数, 既能够给 PSO 算法一些进化空间, 又不会消耗太多的时间, 本文选择  $stage = 5$  来判定搜索进入停滞状态, 开始差分进化操作. DE 采用改进的 DE/rand/1 变异机制, 通过引入变异系数 CV 来决定变异操作基向量的选择范围.

$$CV = CV_{\max} - \frac{CV_{\max} - CV_{\min}}{n_{\max}} \times \text{gen}. \quad (11)$$

其中:  $[CV_{\min}, CV_{\max}]$  为变异系数调整区间, 通常取值  $[0.15, 0.95]$  来控制变异机制的选择;  $n_{\max}$  为最大迭代次数;  $gen$  为本次进化代数. 假设种群粒子数为 NIND, 则变异基向量  $X_{r,1}$  的选择范围由  $CV \times NIND$  取整的结果来决定. 进化初期, CV 值较大, 变异机制近似为 DE/rand/1, 所有粒子的变异向量都从种群中随机选择, 能够在变异的过程中保持粒子的多样性, 使算法在进化早期能够有较好的全局搜索能力; 进化后期, CV 值渐小, 变异机制近似于 DE/best/1, 粒子近似收敛于问题的最小值, 粒子之间差异变小, 所有粒子的变异向量都接近于全局最优值, 此时对粒子进行变异操作, 对工件序列进行微调, 提高算法的局部搜索能力, 使其更接近问题的最小值.

## 3) 基于问题的局部搜索.

大量研究表明, 在调度算法中加入局部搜索能够有效提高搜索速度, 同时对邻域搜索有更彻底的深度, 常用的 3 种邻域结构有 Insert、Interchange 和 Swap. 本文在每代进化的最后, 对每个粒子的个体最优值进行局部搜索, 从而提高搜索效率.

## 2.2 算法流程

DEPSO 混合算法流程如下: 1) 设置参数, 包括种群粒子数 NIND、最大进化代数  $n_{\max}$ 、阈值  $stage$ 、变异参数范围  $[CV_{\min}, CV_{\max}]$  和 PSO 算法中  $c_1$ 、 $c_2$ 、 $w$ 、 $x$ 、 $v$  的取值范围; 2) 根据反向学习机制, 对种群进行初始化, 并计算每个粒子的适应度值; 3) 采用 PSO 算法进行全局搜索, 更新粒子个体极值  $p_i$  和群体极值  $p_g$ ; 4) 判断  $P_g$  保持代数  $dir$  是否大于阈值, 如果  $dir > stage$ , 则执行 5), 否则执行 6); 5) 计算变异参数 CV, 对粒子个体极值  $p_i$  进行离散 DE 搜索, 并更新粒子个体极值  $p_i$  和群体极值  $p_g$ ; 6) 对粒子个体极值  $p_i$  进行局部搜索, 并更新粒子个体极值  $p_i$  和群体极值  $p_g$ ; 7) 判断群体最优适应度值是否等于上一代的值, 若相等, 则  $dir = dir + 1$ , 否则  $dir = 0$ ; 8) 判断进化代数是否大于  $n_{\max}$ , 如果没有, 则返回 3) 并继续搜索, 否则算法结束.

## 3 仿真实验

### 3.1 实验设置

为了验证算法的有效性和优越性, 在双核 Intel Core 2.20 GHz, 内存为 1.99 GB 的 PC 机上进行实验. 对不同规模的典型调度问题的加工时间进行模糊化. 三角模糊数加工时间为  $\tilde{T} = (T^L, T^M, T^U)$ . 其中:  $T^M$  为每个典型问题的加工时间,  $T^L = (T^M - t_1)$ ,  $T^U = (T^M + t_2)$ ,  $t_1$  与  $t_2$  为  $(0, T^M/2)$  之间的随机整数. 采用 Matlab R2008a 对算法进行编程.

### 3.2 算法性能比较和实验结果

#### 1) 算法性能研究.

从上文的算法原理中可以看出: 基于排序的离散差分进化算法通过变异交叉操作, 改变了粒子各个分量位置, 同时能够保留好的结构块, 最后通过选择操作, 得到最优子代个体; 变邻域搜索则通过随机的方式改变粒子个别分量的位置. 两者虽有相似之处, 但差分进化的变异交叉操作是基于个体与极值之间的差异来改变粒子的各分量, 而变邻域搜索则是通过随机方式, 所以两者的优化结果会有所不同.

下面通过对不同规模调度问题的仿真来说明差分进化粒子群算法的有效性.

PSO-DE 和 PSO-VNS 两种改进算法在解决小规模时间零等待多产品厂间歇调度问题时的调度结果如表 1 所示.

在表 1 中:  $N \times M$  为调度规模的大小,  $C^L/C^M/$

$C^U$  为模糊调度结果的最优加工时间、最可能加工时间和最坏加工时间,  $C^*$  为零等待问题在确定时间下已知的最优下界。

表1 不同改进策略算法性能对比(一)

Problem ( $N \times M$ )	$C^*$	Algorithm	$\tilde{C}$	$C^L/C^M/C^U$
Car 2 (13×4)	8242	PSO-DE	8565.5	6848/8247/10920
		PSO-VNS	8565.5	6848/8247/10920
Car 4 (14×4)	9195	PSO-DE	9308.5	7168/9195/11676
		PSO-VNS	9430.8	7247/9402/11672
Car 6 (8×9)	9690	PSO-DE	9897	8012/9765/12046
		PSO-VNS	9897	8012/9765/12046
Car 8 (8×8)	9372	PSO-DE	9539.5	9179/9372/12235
		PSO-VNS	9539.5	9179/9372/12235

表1中的两种算法都是以基本 PSO 算法为框架: PSO-DE 是按照本文算法的改进思路, 只在算法进化进入停滞状态时进行差分进化操作, 不进行变邻域搜索; 而 PSO-VNS 是在每代进化后对每个粒子的个体极值进行变邻域搜索. 从表1可以看出, 两种改进策略都有很好的效果, 且两者相差不大, 差分进化改进策略仅在计算 Car 4 时, 略优于变邻域搜索改进策略, 说明在解决小规模模糊时间零等待多产品厂间歇调度问题上, 两种改进策略都有很好的效果。

基于两种改进策略的算法在解决较大规模模糊时间零等待多产品厂间歇调度问题时的调度结果如表2所示. 从表2可以看出, 差分进化改进策略明显优于变邻域改进策略, 这是因为差分进化算法也是基于群体内各粒子间的合作和竞争的智能搜索过程, 在解决大规模调度问题时, 这种智能的搜索方式比变邻域搜索的随机方式更加有优势。

表2 不同改进策略算法性能对比(二)

Problem ( $N \times M$ )	$C^*$	Algorithm	$\tilde{C}$	$C^L/C^M/C^U$
Rec 17 (20×15)	2740	PSO-DE	2634	1974/2627/3308
		PSO-VNS	2709.5	2066/2670/3432
Rec 23 (30×10)	3030	PSO-DE	2797	2164/2783/3458
		PSO-VNS	2960.3	2288/2899/3755
Rec 29 (30×15)	3583	PSO-DE	3414	2724/3304/4324
		PSO-VNS	3634.5	2791/3508/4587
Rec 35 (50×10)	4718	PSO-DE	4795	3888/4630/6032
		PSO-VNS	5214	4117/5135/6469

综上所述, 本文的混合差分进化粒子群算法将两种改进策略与粒子群算法融合在一起, 充分发挥了每种策略的优势, 使算法有更好的优化性能。

## 2) 算法性能比较.

在仿真实验中发现, 与其他解决零等待多产品厂调度问题的方法相比, GPSO 和 GA 是比较成熟且效果较好的算法, 因此本文将提出的 DEPSO 与 GPSO 和 GA 算法的结果进行比较. 选取不同调度规模的经典调度问题 Car 1 (11×5)、Car 5 (10×6)、Rec 01 (20×5)、Rec 07 (20×10)、Rec 13 (20×15) 和 Rec 19 (30×10), 对这些不同规模的模糊调度问题求解, 并根据规模的大小来确定合适的进化代数  $n_{Max}$  和种群粒子数 NIND.

上述3种算法都采用实数编码, 在 GPSO 算法和 DEPSO 算法的粒子群算法中:  $c_1 = c_2 = 1.49445$ , 微粒速度区间为  $[-2, 2]$ ,  $w$  由 0.9 线性变化到 0.4; 遗传算法的参数交叉概率为 0.9, 变异概率为 0.02; DEPSO 算法中, 进行差分进化变异交叉操作的进化代数阈值  $stage = 5$ , 变异参数范围  $[CV_{min}, CV_{max}]$  为  $[0.15, 0.95]$ .

每种算法对不同规模问题运行 10 次, 记录每次运行结果,  $C_{min}$  和  $C_{max}$  分别为 10 次运行结果中得到的最优值和最差值,  $C_{ave}$  为平均完成时间,  $C^*$  为经典调度问题的已知最优下界. 然后采用最优相对误差 (ORE)、最差相对误差 (WRE) 和平均相对误差 (ARE) 3 个评价指标对算法进行评价, 即

$$ORE = \frac{C_{min} - C^*}{C^*} \times 100, \quad (12)$$

$$WRE = \frac{C_{max} - C^*}{C^*} \times 100, \quad (13)$$

$$ARE = \frac{C_{ave} - C^*}{C^*} \times 100. \quad (14)$$

仿真结果如表3所示, 其中  $\tilde{C}$  为调度结果在乐观系数  $\beta = 0.5$  时的模糊调度结果. 由表3可以看出, 在解决模糊加工时间的零等待多产品厂间歇调度问题上, DEPSO 算法相对于其他两种算法有很大的优势, 也比以往的搜索结果更加精确, 特别是在模型规模较大时, DEPSO 算法的优越性更加明显。

表3 不同算法仿真结果对比

Problem ( $N \times M$ )	$C^*$	Algorithm	$\tilde{C}$	ORE	ARE	WRE
Car 1 (11×5)	8142	DEPSO	8436.5	-0.10	-0.10	-0.10
		GPSO	8714	2.63	5.50	8.03
		GA	8765.8	3.40	5.15	8.24
Car 5 (10×6)	9159	DEPSO	9203.5	0	0	0
		GPSO	9222	0.55	5.88	9.24
		GA	9337	0.32	5.61	9.97
Rec 01 (20×5)	1590	DEPSO	1519	-7.92	-7.33	-6.73
		GPSO	1581.8	-4.09	0.11	3.65
		GA	1564.2	-4.78	0.48	4.59
Rec 07 (20×10)	2119	DEPSO	2123	-3.30	-2.55	-2.08
		GPSO	2216	3.11	5.02	7.98
		GA	2197.5	1.65	5.99	10.38
Rec 13 (20×15)	2709	DEPSO	2630.7	-5.76	-5.63	-5.54
		GPSO	2797.5	1.81	4.32	7.49
		GA	2801.8	1.14	3.29	8.38
Rec 19 (30×10)	3157	DEPSO	2964	-8.27	-7.72	-6.91
		GPSO	3350	2.85	6.83	10.17
		GA	3283	0.41	4.04	8.05

为了更加清晰地显示 DEPSO 算法在收敛性上的优越性, 选择 Car 1 (11×5)、Car 5 (10×6)、Rec 07 (20×10) 和 Rec 19 (30×10) 调度问题进行仿真, 绘制 3 种算法的进化曲线如图 1 所示。

从图 1 能够看出, 由于采用了基于反向学习机制, DEPSO 算法初始解的质量比其他两种算法优秀, 同时, DEPSO 算法进化初期的曲线下落速度也有明显优势, 在进化过程中能够不断地跳出局部极小值, 最终的收敛精度也明显优于 GPSO 和 GA。

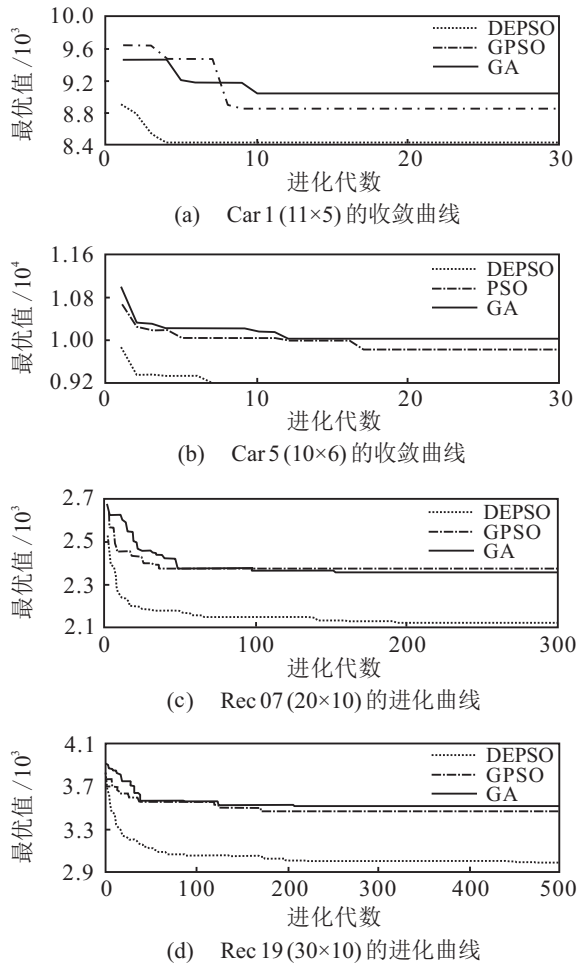


图1 不同算法的进化曲线

在求解调度问题时,算法的时间复杂度主要与迭代次数、种群中的粒子个数和生产规模有关,3种算法的时间复杂度均为 $o(n_3)$ ,本文算法对基本粒子群算法进行了改进,每一代的计算时间必然有所增加。但是从图1~图4所示的进化曲线来看,无论是曲线下下降速度还是最终收敛效果,本文提出的DEPSO算法在求解模糊时间零等待多产品厂间歇调度问题上,相对于GA和GPSO算法有明显的优势。

## 4 结 论

本文研究了模糊时间零等待多产品厂间歇调度问题,以最小化产品模糊完工时间为优化目标,设计了一种差分进化粒子群优化算法。算法主要通过将粒子群算法和基于排序的差分进化算法融合在一起,有效地改善了粒子群算法易于陷入局部最优的缺点,同时基于反向的初始化方法和变邻域搜索都有效地提高了算法的搜索精度。通过仿真实验,以及对仿真结果的分析对比,验证了所提算法的有效性和优越性。

## 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,2007:3-9.  
(Wang W L, Wu Q D. Production scheduling intelligent algorithm and its application[M]. Beijing: Science Press, 2007: 3-9.)

[2] 杨玉珍,顾幸生.多目标零等待间歇生产过程多任务调度[J].化工学报,2013,64(12):4578-4584.  
(Yang Y Z, Gu X S. Multi-objective no-wait multi-task scheduling problem of batch process[J]. CIESC J, 2013, 64(12): 4578-4584.)

[3] Hendtlass Tim. A combined swarm differential evolution algorithm for optimization problems[J]. Lecture Notes in Computer Science, 2001, 2070: 11-18.

[4] Zhang Wenjun, Xie Xiaofeng. DEPSO: Hybrid particle swarm with differential evolution operator[C]. Proc of the IEEE Int Conf on Systems, Man, and Cybernetics. Washington: IEEE Press, 2003: 3816-3821.

[5] Epitropakis M G, Plagianakos V P, Vrahatis M N. Evolving cognitive and social experience in particle swarm optimization through differential evolution: A hybrid approach[J]. Information Sciences, 2012, 216: 50-92.

[6] Caponio Andrea, Neri Ferrante, Tirronen Ville. Super-fit control adaptation in memetic differential evolution frameworks[J]. Soft Computing, 2009, 13(8/9): 811-813.

[7] Ali Musrrat, Pant Millie, Abraham Ajith. Inserting information sharing mechanism of PSO to improve the convergence of DE[C]. 2009 World Congress on Nature and Biologically Inspired Computing. Coimbatore: IEEE Computer Society, 2009: 282-287.

[8] 耿兆强,邹益仁.基于遗传算法的作业车间模糊调度问题的研究[J].计算机集成制造系统,2002,8(8):616-620.  
(Geng Z Q, Zou Y R. Study on job shop fuzzy scheduling problem based on genetic algorithm[J]. Computer Integrated Manufacturing Systems, 2002, 8(8): 616-620.)

[9] 徐震浩,顾幸生.不确定条件下具有零等待的流水车间免疫调度算法[J].计算机集成制造系统,2004,10(10):1247-1251.  
(Xu Z H, Gu X S. Immune scheduling algorithm for flow shop under uncertainty with zero wait[J]. Computer Integrated Manufacturing Systems, 2004, 10(10): 1247-1251.)

[10] Ghrayeb Omar A. A bi-criteria optimization: Minimizing the integral value and spread of the fuzzy makespan of job shop scheduling problems[J]. Applied Soft Computing J, 2003, 2(3): 197-210.

[11] 吴昱,李元香,徐星.基于群智能的新型反向混合差分进化算法[J].小型微型计算机系统,2009,30(5):903-907.  
(Wu Y, Li Y X, Xu X. Novel oppositional hybrid differential evolution algorithm based on swarm intelligence[J]. J of Chinese Computer Systems, 2009, 30(5): 903-907.)

[12] 董明刚.基于差分进化的优化算法及应用研究[D].杭州:浙江大学控制科学与工程学院,2012.  
(Dong M G. Research on differential evolution-based optimization algorithm and applications[D]. Hangzhou: College of Control Science and Engineering, Zhejiang University, 2012.)