

基于混合离散粒子群优化的多时间因素作业车间调度研究^①

李继明^② 徐震浩^③ 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 应用粒子群优化(PSO)进行了考虑机器调整时间、工件运输时间以及提前/拖期惩罚的作业车间调度问题的研究,分析了各时间约束对调度的影响,在此基础上设计了一种解决多时间约束调度问题的混合离散粒子群(HDPSO)算法。该算法在初始阶段采用反向学习机制初始化以提高初始解质量,引入记忆池的概念,在每次迭代中利用记忆池中精英解对当代种群搜索加以指导,以增加粒子与优秀群体间的交流并提高收敛速度及跳出局部最优的能力,最后采用一种针对问题的变邻域搜索策略提高了算法收敛精度。实例仿真验证了该算法的有效性。

关键词 作业车间调度, 调整时间, 运输时间, 提前/拖期, 混合离散粒子群

0 引言

作业车间调度是指对生产过程进行计划和调节,有效的调度方法是提高企业竞争力的核心因素,因而对其研究具有重要的理论意义和应用价值。由于准时(just in time, JIT)制生产模式的广泛接受,针对带交货期的车间调度问题已有大量研究,例如Essafi等采用基于邻域搜索的遗传算法求解带释放期与交货期的作业车间调度问题^[1]; Thiagarajan等采用加权惩罚系数呈阶梯分布的求解规则解决带交货期的动态装配作业车间调度问题^[2]; Mebarki等针对单机提前/拖期调度问题提出了一种优先分派规则,对拖期指标进行了分析并提出了一种新的评价准则^[3]; Khalouli等采用蚁群算法对带提前/拖期惩罚的混合流水车间进行求解^[4]; Rahmani等针对提前/拖期流水车间问题考虑了工件到达时间以及加工时间不确定的情况,并建立了两阶段预测/反应模型^[5]。但是现有的研究中多数都对车间生产进行了大量简化,例如假设调整时间被包含在加工时

间中,不考虑工件的运输时间等。然而实际加工过程中辅助加工时间占了相当比例,据福特汽车统计数据,工厂加工过程中零件加工时间仅占 5%,其他 95% 时间用于搬运和存储^[6],因此忽略辅助加工时间所得到的调度方案一定不是最优方案。为了解决以上问题,本文在现有提前/拖期惩罚模型的基础上考虑了工件运输时间、机器调整时间,形成了多时间因素调度问题,同时引入了粒子群算法。

粒子群算法是受鸟群觅食启发提出的一种启发式算法,其收敛速度快,已成功应用于多个领域^[7-10]。粒子群算法在车间调度邻域也有广泛的应用,如叶寒锋、雷德明、Liu 等通过改进连续粒子群优化(particle swarm optimization, PSO)分别求解作业车间调度问题和流水车间调度问题,但他们均采用映射方式将连续 PSO 转化为离散 PSO,在转换过程中可能会丢失信息,因此算法优化效果不理想^[11-13]。潘全科等重新定义了粒子位置更新方式,使具有连续本质的粒子群算法可直接用于调度问题,并将其与模拟退火结合以克服容易陷入局部最优的缺点,最终取得了较好的效果^[14]。为解决多时

^① 国家自然科学基金(61104178, 61174040)资助项目。

^② 男,1990 年生,硕士生;研究方向:生产调度,智能计算;E-mail: li_jiming@163.com

^③ 通讯作者,E-mail: xuzhenhao@ecust.edu.cn

(收稿日期:2015-06-18)

间约束调度问题,本文提出了新的粒子更新模型,将基本粒子群算法离散化,采用变异策略解决粒子群算法容易陷入局部最优的缺陷。为进一步提高算法性能,本文还融入了反向学习初始化机制、免疫记忆机制以及变邻域搜索,最终形成了混合离散粒子群优化(hybrid discrete PSO, HDPSO)算法。

1 多时间约束车间调度模型

本文中所提到的多时间约束调度是指在满足工艺路线、机床能力的前提下,同时考虑工件运输时间、机器调整时间,合理安排工件在机器上的加工序列和开工时间,使得惩罚成本最小。

1.1 目标函数

以工件提前/拖期惩罚和最小为优化目标,目标函数表示如下:

$$\min(\text{schedule cost}) = \sum_{i=1}^n (\text{Tard}_i + \text{Inv}_i) \quad (1)$$

$$\text{Tard}_i = \text{tard}_i \times \max(0, C_i - \text{deadline}_{i2}) \quad (2)$$

$$\text{Inv}_i = \text{inv}_i \times \max(0, \text{deadlin}_{i1} - C_i) \quad (3)$$

式中 deadlin_{i1} 、 deadline_{i2} 分别为工件 J_i 的最早、最晚交货期; inv_i 、 tard_i 分别为工件 J_i 的提前、拖期惩罚系数,为固定已知值; C_i 为工件 J_i 的完工时间; Inv_i 、 Tard_i 分别为提前库存成本和拖期处罚成本。

1.2 时间约束

1.2.1 调整时间约束

同一台机器,相邻两道加工工序之间存在一段设备调整时间,设备未调整好不能开始加工,因而有

$$C_{\theta_{kl}} - t_{\theta_{kl}} \geq C_{\theta_{ij}} + \text{set}_{\theta_{ij}, \theta_{kl}} \quad (4)$$

式中 θ_{ij} 表示工件 θ_i 第 j 道工序, $C_{\theta_{ij}}$ 表示工序 θ_{ij} 完工时间, $\text{set}_{\theta_{ij}, \theta_{kl}}$ 表示工序 θ_{ij} , θ_{kl} 间的调整时间(θ_{ij} 在 θ_{kl} 前加工且相邻), $t_{\theta_{ij}}$ 为工序 θ_{kl} 加工时间。

1.2.2 运输时间约束

同一工件相邻两工序在不同机器上加工,两道工序间有运输时间,工件未运达不能开始加工,因而有

$$C_{\theta_{kl}} - t_{\theta_{kl}} \geq C_{\theta_{k(l-1)}} + \text{tran}_{\theta_{k(l-1)}, \theta_{kl}} \quad (5)$$

式中 $\text{tran}_{\theta_{k(l-1)}, \theta_{kl}}$ 表示工件 θ_k 相邻工序间的运输时间。

1.3 工艺约束

同一工件的工序有先后约束,上一工序完成之后才能开始下一工序,因而有

$$C_{\theta_{ij}} - t_{\theta_{ij}} \geq C_{\theta_{ij-1}} \quad (6)$$

1.4 资源约束

每台机器同一时刻只能加工一个工件,即

$$C_{\theta_{kl}} - t_{\theta_{kl}} \geq \max \{ C_{\theta_{ij-1}} + \text{tran}_{\theta_{ij-1}, \theta_{ij}}, C_{\theta_{ij}} + \text{set}_{\theta_{ij}, \theta_{kl}} \} \quad (7)$$

其中工序 θ_{ij} 在 θ_{kl} 前加工且相邻。

2 混合离散粒子群优化(HDPSO)调度算法

多时间约束作业车间调度问题相比于经典的作业车间调度问题更为复杂,用单一的启发式算法求解此类复杂调度问题效果不理想。因此,本文针对变邻域搜索(variable neighborhood search, VNS)和离散粒子群优化(discrete particle swarm optimization, DPSO)算法分别进行了改进,并提出了混合离散粒子群优化(HDPSO)算法。

2.1 算法思路

2.1.1 编码与解码

基于工序的编码不仅能映射到所有解空间,而且执行交叉操作时也不会产生非法解,因此选用基于工序的编码方式。提前、拖期惩罚最小化为非正规性能指标问题,最优解有可能出现在半主动解码中,为了不丢失潜在最优解,选择半主动解码。解码时先将基因转化为一个有序操作表,然后按照此表以及工艺约束条件对各工件的工序以最早允许加工时间逐一进行加工。

2.1.2 基于反向学习的初始化

随机产生的初始解分布不均匀,为提高初始解质量,本文引入基于反向学习的机制,随机产生一对反向解并取两者中较好的作为初始解。对于序列 $\pi = (k_1, k_2, \dots, k_N)$,其反向解为 $\bar{\pi} = (\bar{k}_1, \bar{k}_2, \dots, \bar{k}_N)$,其中 $\bar{k}_i = 1 + N - k_i$ 。较好的初始解有助于提高收敛速度。

2.1.3 改进的离散粒子群优化(DPSO)

基本的粒子群适合于对连续域求解,对于组合

优化问题本文采用 DPSO。粒子群中各个粒子只向一个全局最优粒子学习,导致粒子与种群优秀个体交流偏少而容易陷入局部最优,因此本文引入记忆池的概念,利用记忆池保留搜索过程中出现的较好解。这里重新定义粒子更新模型:

$$x_i^{t+1} = p_i^t \text{cross} x_i^t + \text{memory}_i^t \text{cross} x_i^t \quad (8)$$

式中 cross 是交叉算子, $+$ 是优选算子, x_i^{t+1} 是粒子 i 第 $t+1$ 次迭代后位置, p_i^t 是粒子 i 第 t 次迭代时历史最优位置, memory_i^t 是第 t 次迭代中从记忆池中随机选择的精英粒子。交叉操作采用文献[15]中的部分映射交叉方法。

2.1.4 基于问题的局部搜索

在调度算法中加入局部搜索可以提高搜索速度,但普通局部搜索一般采用随机方式进行插入、交换操作,效率不高,此处采用针对问题的变邻域搜索(VNS)方法。由于基于工序的编码方式和半主动解码方式的特殊性,不同的解序列可能对应同一加工过程,而对于同一台机器上加工的工件,任意调整其在机器上的加工顺序都会导致机器调整时间的变化,最终导致惩罚的变化,因此每台机器上的任意工序都属于关键路径上的工序。变邻域搜索过程中的交换、插入操作只对在同一机器上加工的工序进行;逆序操作可对任意两操作序号间工序进行,无需限制在同一机器上,这样扩大了搜索范围。对于 $n \times m$ 规模问题,每个精英解需搜索其附近 $n \times m$ 个邻域解。考察所有邻域解适应度,若最优邻域解适应度优于当前解,则用其替换当前精英解,否则不做改动。

2.2 算法流程

HDPSO 算法采用基于工序的编码方式并随机产生初始解,以扩大初始解覆盖空间,算法伪代码如下:

```

初始化参数  $\delta$ 、 $\beta$ 、 $\gamma$ 、 $number$ 、 $\varepsilon_1$ 、 $\varepsilon_2$ 、 $\varepsilon_3$ 
采用反向学习方法初始化种群  $population$ 
更新记忆池  $memory$ 、个体历史最优解  $pbest$ 
for  $i = 1$ :  $iter$ 
    for  $j = 1$ :  $number$ 
        从记忆池  $memory$  中随机选择一个个体  $memory(i)$ 
        作为最优解  $gbest$ , 将当前个体分别与  $pbest(i)$ 、
         $gbest$  交叉, 所得最好个体取代当前个体

```

end

用轮盘赌法从 $memory$ 中选取精英解集 $elite$
对 $elite$ 中所有个体进行变邻域搜索
更新 $pbest$ 与 $memory$
对非最优解以概率 γ 进行对换变异

end

3 仿真试验与分析

以经典的 LA、FT、ABZ 算例为基础,通过加入调整时间 $set_{\theta_{ij}, \theta_{kl}}$ 、运输时间 $tran_{\theta_{i(j-1)}, \theta_{ij}}$ 以及交货期窗 $[deadline_{i1}, deadline_{i2}]$ 构造多时间因素车间调度算例。

(1) 机器 $M_{\theta_{ij}}$ 上的调整时间 $set_{\theta_{ij}, \theta_{kl}}$ 在 $[1, p]$ 之间随机产生,其中 p 为所有工件最长加工时间 $\max(T_{\theta_{ij}})$, 因为调整时间约占加工时间的 30% ~ 100%^[16]。

(2) 运输时间 $tran_{\theta_{i(j-1)}, \theta_{ij}}$ 依据 $[1, 1.2p] \times dis_{\theta_{i(j-1)}, \theta_{ij}} / (m - 1)$ 随机产生,其中 $dis_{\theta_{i(j-1)}, \theta_{ij}}$ 为工件 θ_i 相邻两工序加工所在机器的序号差值。这样设定是因为运输时间约占加工时间的 40% ~ 120%^[16],且运输时间与运输距离及工件大小相关。假定典型算例中每个工件加工前都已准备就绪即第一道工序的运输时间为 0。

(3) 各工件提前惩罚系数在 $[0.1, 0.3]$ 之间随机产生,拖期惩罚系数在 $[0.4, 0.8]$ 之间随机产生,因为订单拖期带来的损失一般大于提前完工带来的库存积压成本。

(4) 交货期 $deadline_{i1}$ 依据 $p1 \times \sum_{j=1}^n (T_{\theta_{ij}}) + set + tran$ 产生, $deadline_{i2}$ 依据 $deadline_{i1} + p2 \times \sum_{j=1}^n (T_{\theta_{ij}})/n$ 产生,其中 $tran$ 为所有工件运输时间均值, set 为调整时间均值, $p1$ 为交货期松弛系数,用以调节各个工件的交货期松紧程度, $p1$ 越小交货越紧迫,工件拖期可能性越高, $p2$ 为交货期集中程度系数, $p2$ 越小交货期窗越窄,工件也越可能提前或拖期。

本试验在 Intel core i5 2.53GHz、内存为 6G 的 PC 上进行。采用 MATLAB R2014a 编程。

3.1 参数整定

本文以适应度计算次数 $iter$ 为算法终止准则, 取 $iter = 10000 + 1000 \times m \times n$ 。影响算法的主要参数有记忆池大小比例 β , 种群数量 $number$, 精英比例 δ , 插入、交换和逆序选取概率 $\varepsilon_1, \varepsilon_2, \varepsilon_3$, 变异概率 γ 。考虑到计算时间和空间的限制, 每个参数的水平限定为 5 水平, 如表 1 所示。

由于插入效果优于交换和逆序^[17,18], 因此主要测试不同水平的插入概率 ε_1 对效果的影响。 $\varepsilon_2, \varepsilon_3$ 随机取值, 但须满足 $\varepsilon_1 + \varepsilon_2 + \varepsilon_3 = 1$ 。本文中 5 因素 5 水平的试验选择 $L_{25}(5^6)$ 等水平正交表, 其中空列为误差列。以 FT20 为例进行试验, 试验中每种参数的组合均运行 20 次, 交货期松弛系数、交货期集中程度系数分别取 2 和 1。试验结果如表 2。

表 1 参数水平

水平	参数				
	number	δ	β	γ	$\varepsilon_1, \varepsilon_2, \varepsilon_3$
1	50	0.45	0.3	0.4	0.15 0.3 0.55
2	100	0.15	0.15	0.5	0.6 0.25 0.15
3	200	0.6	0.6	0.3	0.3 0.3 0.4
4	150	0.3	0.45	0.1	0.75 0.15 0.1
5	120	0.75	0.75	0.2	0.45 0.35 0.2

表 2 算例正交试验结果

指标	number	空列	δ	β	γ	$\varepsilon_1, \varepsilon_2, \varepsilon_3$
k_1	1904.1	2211	2238	2186	2337	2352.2
k_2	2107.7	2257	2078	2148	2621	2344.2
K_3	2603.6	2171	2283	2231	2161	2186.9
k_4	2275.1	2265	2115	2194	1971	2155.7
k_5	2190.7	2177	2367	2323	1992	2042
R	699.5	94.4	288.4	175.5	650.4	310.2
F	34.1	1	7.4	2.3	37.9	9

从表 2 可知所有算例空白列极差 R 均小于其他因素列极差, 表明因素间交互影响相比于这些参数可以忽略; 由于 $F_{0.05}(4,4) = 6.39$, 所以对于显著性水平 $\delta = 0.05$, 除记忆池比例 β 外所有参数对试验结果都有显著影响。根据上表可得各参数最优取值如下: 种群数量 $number = 50$, 精英比例 $\delta = 0.15$, 记忆池选取比例 $\beta = 0.15$, 变异概率 $\gamma = 0.1$ 。适当的变异概率有助于跳出局部最优, 过大的变异概率会导致搜索随机性强收敛速度慢; 插入比例过小或过大导致操作的多样性降低。

3.2 优化分析试验

3.2.1 时间约束分析

以经典的 FT06 调度问题为例, 模型中机器工序调整时间如表 3 所示。

表 3 算例调整时间表

机器 1						机器 2						机器 3						机器 4					
工件	1	2	3	4	5	6	工件	1	2	3	4	5	6	工件	1	2	3	4	5	6			
1	0	6	1	8	3	5	1	0	1	7	4	2	4	1	0	7	8	1	9	4			
2	4	0	9	7	8	9	2	8	0	1	5	6	3	2	5	0	5	9	4	10			
3	1	2	0	9	9	10	3	7	6	0	2	3	3	3	9	8	0	8	7	6			
4	4	6	4	0	1	10	4	9	1	5	0	8	6	4	7	5	2	0	3	7			
5	6	1	4	6	0	5	5	2	9	7	6	0	4	5	8	1	7	7	0	8			
6	9	1	2	10	6	0	6	7	2	3	2	8	0	6	8	1	3	5	1	0			
机器 4						机器 5						机器 6						机器 1					
工件	1	2	3	4	5	6	工件	1	2	3	4	5	6	工件	1	2	3	4	5	6			
1	0	4	3	8	7	3	1	0	8	6	8	6	10	1	0	6	9	8	5	3			
2	8	0	9	6	3	1	2	3	0	9	1	8	4	2	2	0	8	10	5	2			
3	5	6	0	4	5	7	3	5	5	0	4	9	10	3	8	4	0	4	6	4			
4	9	3	3	0	10	5	4	3	10	8	0	8	9	4	7	10	4	0	1	5			
5	6	10	9	5	0	8	5	8	5	2	10	0	1	5	5	9	9	10	0	7			
6	2	6	3	10	2	0	6	10	5	2	4	2	0	6	1	9	2	2	1	0			

运输时间、交货期及惩罚系数如表 4 所示(由于工件加工顺序已知,所以各个工序间的运输距离也可以确定)。

表 4 工件运输及交货参数

工件	工序					
	1	2	3	4	5	6
1	0	2	2	4	3	0
2	0	2	2	1	11	2
3	0	0	0	2	2	1
4	0	2	1	1	1	2
5	0	2	2	2	9	1
6	0	4	5	1	4	2
工件号	1	2	3	4	5	6
最早交货期	92	134	108	110	90	100
最晚交货期	96	142	114	116	94	105
提前惩罚系数	0.27	0.15	0.19	0.12	0.22	0.2
拖期惩罚系数	0.67	0.79	0.59	0.42	0.45	0.78

当不考虑调整时间以及运输时间,利用 HDPSO 算法对此问题进行优化,得出其中一最优解 $S_1 = [4, 2, 5, 3, 3, 2, 3, 3, 1, 2, 3, 6, 5, 4, 5, 5, 6, 6, 6, 4, 1, 1, 1, 1, 4, 5, 6, 4, 4, 2, 2, 5, 2, 3, 6]$, 惩罚为 0, 甘特图如图 1 所示。

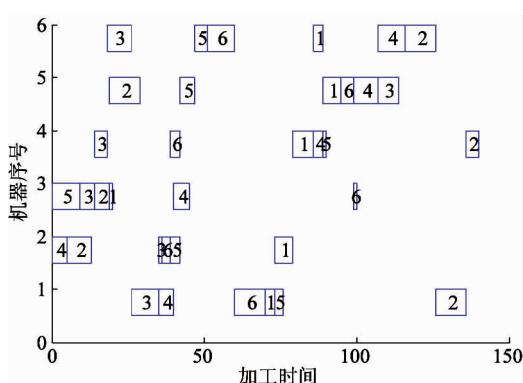


图 1 时间甘特图

对此最优解直接加入调整时间以及运输时间,进行仿真得到的甘特图如图 2, 惩罚为 200, 因此若按此调度方案进行调度显然不具合理性。

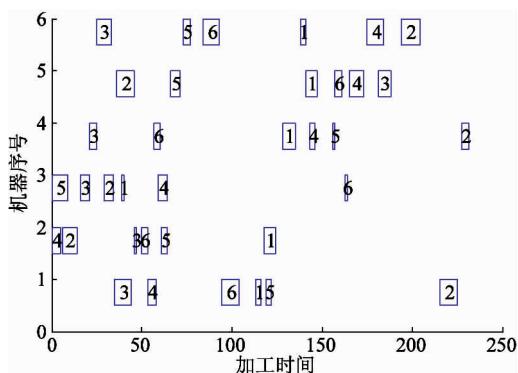


图 2 按原最优解加入时间约束的甘特图

利用本文建立的多时间约束模型,并采用 HDPSO 算法对此问题进行优化,得到一最优解 $S_2 = [1, 2, 3, 4, 5, 6, 6, 6, 1, 5, 1, 6, 2, 2, 4, 3, 6, 3, 1, 1, 5, 3, 4, 4, 5, 4, 5, 1, 6, 2, 2, 4, 5, 3, 6, 2, 3]$, 甘特图如图 3, 惩罚为 2.58, 显然全面考虑时间约束后的优化结果更加符合实际要求。

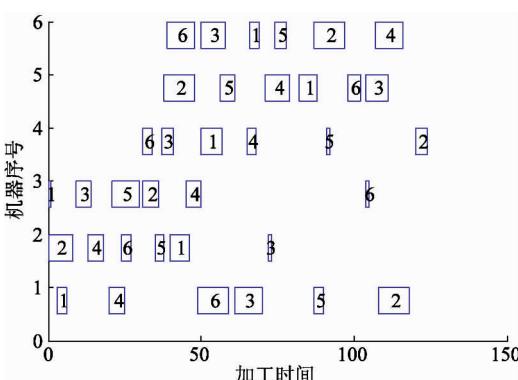


图 3 考虑时间约束并进行优化后的甘特图

为单独分析调整时间和运输时间对调度的影响,分别对各算例(取 $p_1 = p_2 = 2$)在考虑调整/运输时间前、后采用 HDPSO 算法进行优化,为消除两种时间因素的交互影响,在分析调整时间的影响时假设运输时间为 0, 分析运输时间时调整时间设为 0。分别对不同规模的算例进行仿真分析,算例取不同条件时均运行 20 次,仿真结果如表 5。

根据以上结果可知,调整时间和运输时间对各算例都有重大影响,不同算例中调整时间与运输时间影响相差不大。当调整时间或运输时间不考虑时,实际与理论上工件加工一道工序所花费时间会有所不同,若不考虑辅助加工时间进行优化,而实际

表 5 辅助时间对调度影响

算例	辅助时间	改进前	改进后	改进比例
LA03	setup	2710.96	65.35	97.59%
	tran	957.00	26.82	97.20%
	setup/tran	4249.21	72.03	98.30%
LA11	setup	13641.25	365.57	97.32%
	tran	1859.93	1.96	99.89%
	setup/tran	6009.22	7.50	99.88%
LA17	setup	4365.16	0.96	99.98%
	tran	2752.77	62.06	97.75%
	setup/tran	16116.48	511.54	96.83%
LA23	setup	8593.70	58.72	99.32%
	tran	3072.97	8.03	99.74%
	setup/tran	11709.83	98.58	99.16%
LA29	setup	17191.67	242.23	98.59%
	tran	5095.61	31.19	99.39%
	setup/tran	20059.46	303.85	98.49%

加工过程按优化结果进行,会导致工件实际完成时间早于或晚于理论时间,惩罚值偏大。因此,在对生

产过程进行建模时,这两种时间不可忽略。

3.2.2 典型算例仿真

为验证算法的有效性,分别采用离散粒子群优化(DPSO)算法、标准基因算法(GA)以及文献[9]中自适应学习粒子群优化(ALPSO)算法、文献[19]中结合模拟退火的粒子群优化(PSOSA)算法对不同规模的算例进行测试。测试运行参数设置是:DPSO 种群数量为 50,变异概率为 0.1;GA 种群数量为 50,交叉概率为 0.88,变异概率为 0.1;PSOSA、ALPSO 参数取值与原文相同。以相对 HDPSO 的偏离误差,即剩余预测偏差(residual predictive deviation, RPD)为评价指标,RPD 计算公式如下:

$$RPD = \frac{Obt - HDPSO}{HDPSO} \times 100 \quad (9)$$

其中 Obt 表示当前算法得到的平均解, $HDPSO$ 表示 $HDPSO$ 算法所得平均值。仿真结果如表 6 所示,表中所列为各算法运算 20 次所得平均惩罚。

表 6 典型算例仿真结果

算例	算法	$p1 = 1 p2 = 1$	$p1 = 1 p2 = 2$	$p1 = 2 p2 = 1$	$p1 = 2 p2 = 2$
ABZ5(10 × 10)	HDPSO	0.00%	0.00%	0.00%	0.00%
	ALPSO	20.73%	10.25%	10.11%	6.61%
	PSOSA	77.95%	111.78%	41.68%	76.86%
	DPSO	118.18%	185.26%	81.05%	187.60%
	GA	547.02%	965.15%	1792.63%	6429.75%
FT10(10 × 10)	HDPSO	0.00%	0.00%	0.00%	0.00%
	ALPSO	26.08%	39.21%	36.03%	25.00%
	PSOSA	79.20%	109.53%	506.25%	882.86%
	DPSO	78.63%	107.51%	550.74%	2455.71%
	GA	328.10%	439.30%	4194.49%	14482.86%
FT20(20 × 5)	HDPSO	0.00%	0.00%	0.00%	0.00%
	ALPSO	46.32%	55.94%	55.23%	85.22%
	PSOSA	123.80%	124.38%	156.79%	180.10%
	DPSO	95.57%	109.97%	113.86%	164.95%
	GA	387.61%	418.86%	592.13%	680.10%
LA01(10 × 5)	HDPSO	0.00%	0.00%	0.00%	0.00%
	ALPSO	30.80%	-1.87%	-7.65%	20.34%
	PSOSA	71.90%	171.82%	43.43%	22.88%
	DPSO	87.05%	208.44%	114.98%	147.46%
	GA	555.60%	1256.98%	2276.15%	2822.03%
LA06(15 × 5)	HDPSO	0.00%	0.00%	0.00%	0.00%
	ALPSO	27.80%	71.23%	41.49%	44.69%
	PSOSA	71.86%	105.79%	76.88%	58.29%
	DPSO	138.83%	152.14%	244.75%	219.32%
	GA	493.91%	705.77%	1631.12%	1672.06%

	HDPSO	0.00%	0.00%	0.00%	0.00%
LA12(20×5)	ALPSO	106.53%	77.12%	62.19%	104.37%
	PSOSA	166.26%	133.44%	122.02%	149.20%
	DPSO	204.73%	196.41%	284.91%	427.50%
	GA	878.46%	849.48%	1519.66%	2001.43%
LA20(10×10)	HDPSO	0.00%	0.00%	0.00%	0.00%
	ALPSO	26.80%	7.81%	139.77%	121.62%
	PSOSA	56.25%	87.73%	944.32%	386.49%
	DPSO	95.47%	151.35%	664.77%	656.76%
LA21(15×10)	GA	502.54%	861.52%	11798.86%	22724.32%
	HDPSO	0.00%	0.00%	0.00%	0.00%
	ALPSO	16.18%	12.74%	105.99%	228.20%
	PSOSA	116.07%	90.33%	400.21%	877.07%
DPSO	114.14%	124.88%	642.77%	1896.62%	
	GA	543.79%	567.42%	3913.22%	15684.59%

从试验结果可以看出,对于以上 8 个算例在交货期控制参数取不同值的情况下,HDPSO 算法的效果明显优于其他算法,32 组测试中取得 30 组最优值,ALPSO 算法取得 2 组最优值,剩余 3 种算法中 PSOSA 算法略优于 DPSO 算法,而 DPSO 算法明显优于 GA 算法。从仿真结果明显可以看出,交货期集中程度系数 p_2 相同时,交货期松弛系数 p_1 越大平均惩罚越小; p_1 相同时, p_2 越大平均惩罚越小,这些仿真结果和变化规律与实际生产相符合,这在一定程度上也说明了算法的有效性。

3.2.2 动态环境仿真

本文同时研究了工件动态到达且到达时间已知的情况,假设各工件到达时间在 $[0, 100]$ 之间服从均匀分布。针对各个算例 p_1, p_2 分别取 1、2, 仿真结

果如表 7 所示。

由以上结果可知算法优化效率从高到低依次为 HDPSO、ALPSO、PSOSA、DPSO、GA, 与典型算例试验的结果一致。PSOSA 与 HDPSO 差异在于 HDPSO 使用了改进的邻域搜索而 PSOSA 使用模拟退火, 结果表明改进的邻域搜索更适于求解此类问题; ALPSO 采用 random-key 机制将连续解映射到离散空间, 虽然引入了退火机制、邻域搜索, 并采用了自适应步长调整策略, 但是效果仍差于 HDPSO, 说明离散 PSO 相比于连续 PSO 在求解此类问题时更有优势; PSOSA 相比于 DPSO 效果有了显著提升, 说明在 DPSO 基础上加入模拟退火是有效的; 相比于 GA, DPSO 中个体均与种群最优个体和历史最优个体有信息交流, 因此 DPSO 效果远优于 GA。仿真结果并

表 7 动态环境下仿真结果

算例	算法	$p_1 = 1$ $p_2 = 1$	$p_1 = 1$ $p_2 = 2$	$p_1 = 2$ $p_2 = 1$	$p_1 = 2$ $p_2 = 2$
ABZ5(10×10)	HDPSO	0.00%	0.00%	0.00%	0.00%
	ALPSO	15.06%	209.96%	22.50%	0.68%
	PSOSA	108.87%	355.91%	122.99%	15.93%
	DPSO	180.56%	1221.65%	246.80%	103.73%
FT10(10×10)	GA	1472.57%	8981.27%	4674.06%	2143.05%
	HDPSO	0.00%	0.00%	0.00%	0.00%
	ALPSO	32.78%	21.55%	51.28%	143.52%
	PSOSA	108.87%	127.41%	511.72%	944.04%
DPSO	189.88%	264.08%	1211.36%	1306.74%	
	GA	851.41%	1000.50%	17290.84%	9631.61%

	HDPSO	0.00%	0.00%	0.00%	0.00%
FT20(20×5)	ALPSO	68.87%	74.32%	94.77%	61.06%
	PSOSA	125.50%	163.02%	152.64%	135.49%
	DPSO	117.45%	128.72%	217.56%	129.88%
	GA	504.96%	517.13%	911.75%	609.96%
LA01(10×5)	HDPSO	0.00%	0.00%	0.00%	0.00%
	ALPSO	26.54%	28.70%	4.15%	10.60%
	PSOSA	58.97%	184.91%	60.24%	24.50%
	DPSO	471.25%	337.57%	292.58%	115.23%
LA06(15×5)	GA	5549.14%	5662.43%	3144.51%	2143.05%
	HDPSO	0.00%	0.00%	0.00%	0.00%
	ALPSO	39.17%	96.04%	17.29%	42.84%
	PSOSA	75.83%	116.00%	59.25%	59.94%
LA12(20×5)	DPSO	359.12%	417.57%	198.35%	385.46%
	GA	2027.28%	2445.66%	2376.97%	1761.72%
	HDPSO	0.00%	0.00%	0.00%	0.00%
	ALPSO	114.00%	53.32%	44.32%	82.35%
LA20(10×10)	PSOSA	195.46%	140.05%	147.44%	292.12%
	DPSO	424.55%	376.23%	523.23%	448.02%
	GA	2499.34%	1898.27%	3134.29%	2465.67%
	HDPSO	0.00%	0.00%	0.00%	0.00%
LA21(15×10)	ALPSO	66.43%	6.79%	4.53%	266.67%
	PSOSA	734.28%	33.58%	16.58%	1894.44%
	DPSO	1646.11%	143.09%	55.71%	2533.33%
	GA	11564.13%	1369.43%	894.19%	47522.22%
	HDPSO	0.00%	0.00%	0.00%	0.00%
	ALPSO	20.53%	9.41%	124.25%	117.60%
	PSOSA	239.15%	74.96%	482.88%	632.76%
	DPSO	438.22%	159.65%	900.57%	1271.15%
	GA	2600.07%	954.12%	8863.78%	9312.71%

未随着 p_1 、 p_2 的增大而减小,因为在 p_1 、 p_2 取不同值情况下工件到达时间不一致导致各工件准时交货难度不同。

为更清晰显示 HDPSO 算法的优越性,绘制了以上算法在求解 FT10 问题时的收敛曲线,如图 4 和图 5 所示。此两图的横坐标分别为适应度计算次数和计算时间。从图中看出,由于采用了反向学习的初始化机制,HDPSO 算法初始解质量优于其他算法;进化初期,HDPSO 算法与 DPSO 算法相比,添加了局部搜索过程,所以收敛速度稍慢,PSOSA 算法与 HDPSO 算法相比采用模拟退火策略接受了部分劣解,所以收敛速度变慢,ALPSO 算法采用连续域搜索方式,所以收敛速度最慢;进化中期,DPSO、PSOSA 不同程度的陷入局部最优,ALPSO 采用了邻域搜索、自适应步长等策略,跳出局部最优能力较

强,HDPSO 相比于其他算法更能够跳出局部最优,最终 HDPSO 的收敛精度最高。图 5 还显示了各算法的运算速度,从快到慢依次为 ALPSO、HDPSO、DPSO、PSOSA。综合收敛精度与计算速度,HDPSO 和 ALPSO 优于其他算法。

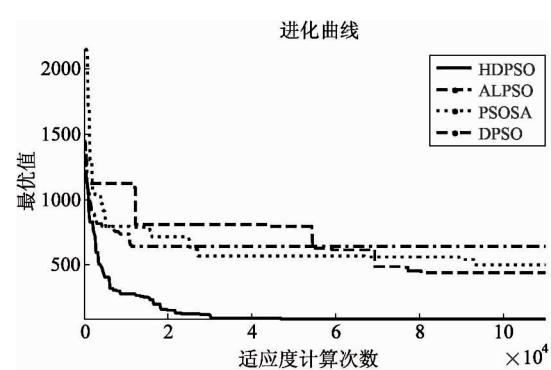


图 4 典型算例 FT10 收敛曲线

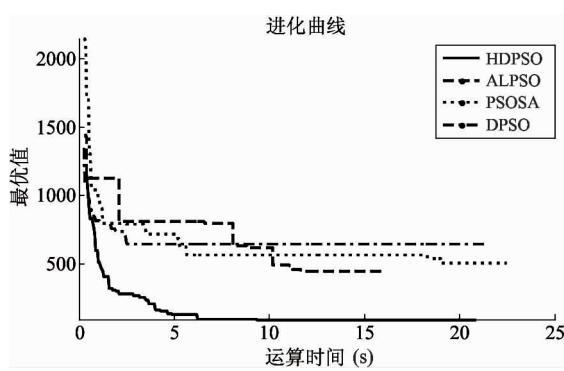


图 5 动态环境 LA06 收敛曲线

4 结论

本文在研究车间调度问题时不仅考虑了加工时间,同时也考虑了工件的交货期、调整时间、运输时间,并分析了它们对调度结果的影响。结果表明,工件交货期的紧急程度、集中程度与惩罚总成本密切相关,而调整时间、运输时间是调度问题中不可忽视的因素;为解决以上多时间因素车间调度问题,本文通过将局部搜索与全局搜索相结合,同时利用了免疫记忆机制以及基于反向初始化方法设计了混合离散粒子群算法,并通过仿真试验及对比分析验证了算法的有效性。本文在研究多时间因素的调度问题时做了很多假设,但实际生产中有诸多不确定因素,如交货期的变化、加工过程的故障等,考虑更多实际加工过程中的约束将更具研究意义。

参考文献

- [1] Essafi I, Mati Y, Dauzère-Pérès S. A genetic local search algorithm for minimizing total weighted tardiness in the job-shop scheduling problem. *Computers & Operations Research*, 2008, 35(8): 2599-2616
- [2] Thiagarajan S, Rajendran C. Scheduling in dynamic assembly job-shops to minimize the sum of weighted earliness, weighted tardiness and weighted flowtime of jobs. *Computers & Industrial Engineering*, 2005, 49(4): 463-503
- [3] Mebarki N, Shahzad A. Correlation among tardiness-based measures for scheduling using priority dispatching rules. *International Journal of Production Research*, 2013, 51(12): 3688-3697
- [4] Khalouli S, Ghedjati F, Hamzaoui A. A meta-heuristic approach to solve a JIT scheduling problem in hybrid flow shop. *Engineering Applications of Artificial Intelligence*, 2010, 23(5): 765-771
- [5] Rahmani D, Heydari M. Robust and stable flow shop scheduling with unexpected arrivals of new jobs and uncertain processing times. *Journal of Manufacturing Systems*, 2014, 33(1): 84-92
- [6] 李学诗. 改善工厂物流必须抓住车间生产物流这个重点. 物流技术, 2008, 27(10): 215-217
- [7] Eberhart R C, Shi Y. Particle swarm optimization: developments, applications and resources. In: Proceedings of the 2001 Congress on Evolutionary Computation, IEEE, 2001, 1: 81-86
- [8] Karanki S B, Mishra M K, Kumar B K. Particle swarm optimization-based feedback controller for unified power-quality conditioner. *IEEE Transactions on Power Delivery*, 2010, 25(4): 2814-2824
- [9] Voumvoulakis E M, Hatziargyriou N D. A particle swarm optimization method for power system dynamic security control. *IEEE Transactions on Power Systems*, 2010, 25(2): 1032-1041
- [10] Sun J, Fang W, Xu W. A quantum-behaved particle swarm optimization with diversity-guided mutation for the design of two-dimensional IIR digital filters. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2010, 57(2): 141-145
- [11] 叶寒峰, 李占山, 陈超. 基于具有自适应与自学习能力的粒子群优化算法的车间调度算法. 吉林大学学报(理学版), 2014, 52(01): 93-97
- [12] 雷德明, 吴智铭. 基于粒子群优化的多目标作业车间调度. 上海交通大学学报, 2007, 41(10): 1657-1657
- [13] Liu B, Wang L, Jin Y H. An effective hybrid PSO-based algorithm for flow shop scheduling with limited buffers. *Computers & Operations Research*, 2008, 35(9): 2791-2806
- [14] 潘全科, 王文宏, 朱剑英. 基于粒子群优化和模拟退火的混合调度算法. 中国机械工程, 2006, 17(10): 1044-1046
- [15] 彭传勇, 高亮, 邵新宇等. 求解作业车间调度问题的广义粒子群优化算法. 计算机集成制造系统, 2006, 12(6): 911-917
- [16] 李峥峰. 多时间因素作业车间调度问题的研究与工程应用:[博士学位论文]. 武汉:华中科技大学, 2010.

24-28

- [17] Naderi B, Zandieh M, Roshanaei V. Scheduling hybrid flowshops with sequence dependent setup times to minimize makespan and maximum tardiness. *The International Journal of Advanced Manufacturing Technology*, 2009, 41(11-12): 1186-1198
- [18] Ruiz R, Stützle T. An iterated greedy heuristic for the se-

quence dependent setup times flowshop problem with makespan and weighted tardiness objectives. *European Journal of Operational Research*, 2008, 187(3): 1143-1159

- [19] 葛茂根, 刘明周, 张明伟等. 计及机器调整和物料运送时间的 Job-Shop 调度算法. *农业机械学报*, 2009, 39(11): 127-131

Research on the job-shop scheduling problem with multi-time constraints based on discrete particle swarm optimization

Li Jiming, Xu Zhenhao, Gu Xingsheng

(Key Laboratory of Advanced Control and Optimization for Chemical Processes,
Ministry of Education, Shanghai 200237)

Abstract

The job-shop scheduling considering the time constraints of processing time, setup time, transit time and earliness/tardiness punishment was studied by the application of particle swarm optimization (PSO), and the influences of the constraints on the scheduling were analyzed. Then, a hybrid discrete (HD) PSO (HDPSON) algorithm for solving the job-shop scheduling with multi-time constraints was designed. The algorithm uses the opposition-based learning to initialize population to improve the quality of the initial population. It also implants the memory mechanism into the discrete PSO to speed up convergence. At last the algorithm adopts a modified variable neighborhood search (VNS) to strengthen the local search ability. The effectiveness of the proposed algorithm was demonstrated by the experiments on different simulation examples.

Key words: job-shop scheduling, setup time, transit time, earliness/tardiness, hybrid discrete particle swarm