

# 基于遗传算法的某航空发动机 装配车间优化调度

范金松,严洪森,周久海,蒋南云

(东南大学 复杂工程系统测量与控制教育部重点实验室,江苏 南京 210096;  
东南大学 自动化学院,江苏 南京 210096)

**摘要:**航空发动机装配车间装配生产线的调度问题,是一类比较典型的混合 Flowshop 问题,同时还带有工件可重入等特点,这就区别于一般的 Flowshop 和 Jobshop 调度问题,因此,将可重入混合车间调度问题划为第三类调度问题。关于重入式混合车间生产调度的优化问题通常来说都是属于 NP 难问题。文中通过某航空发动机装配车间生产线的研究,以最小化最大完工时间为目标函数,借助随机矩阵的编码方式和改进的交叉方法与变异方法,提出了基于遗传算法的调度优化方法。最后实验结果表明,文中提出的改进算法能够有效地实现装配车间调度的优化。

**关键词:**遗传算法;可重入混合车间;生产调度

中图分类号:TP391;TH166

文献标识码:A

文章编号:1673-629X(2012)09-0205-05

## Optimizing Aeroengine Assembly Shop Schedule Based on Genetic Algorithm

FAN Jin-song, YAN Hong-sen, ZHOU Jiu-hai, JIANG Nan-yun

(Ministry of Education Key Laboratory of Measurement and Control of CSE, Southeast University,  
Nanjing 210096, China;  
School of Automation, Southeast University, Nanjing 210096, China)

**Abstract:** Assembly line scheduling problem of aeroengine assembly plant is a typical class of hybrid flowshop problem, and also features a piece re-entrant, which is different from the general flowshop and jobshop scheduling. Therefore, designate re-entrant hybrid flowshop as a third of scheduling problems. Re-entry hybrid plant production scheduling optimization problems are usually NP-hard ones. In this paper, through a study on an aero engine assembly plant production line, minimizing the makespan as the objective function and by means of random matrix encoding and improved crossover and mutation methods, an approach to optimize schedules is proposed based on genetic algorithm. Finally, simulation experiments show that the proposed improved algorithm can effectively achieve the assembly shop scheduling optimization.

**Key words:** genetic algorithm; re-entrant hybrid flowshop; production scheduling

## 0 引言

在航空发动机装配车间生产过程中,完成一台发动机的装配任务,往往要有上百道工序,还会涉及多次检验试车等反复加工情况。通过对装配车间装配生产线的研究,将其中的一些工序进行划分归类,具有相类似或连续的一类操作工序看作是一个组,对装配流程中几个连续的组看作是一个班。每个组内装配工序操

作是固定的,同时每个班内组与组之间的装配操作顺序也是固定的。这样装配车间生产线以班组来划分共有:部件、传装、总装、试车、分解、故检、返件等七个班组。为了增加产能,班中还有多个并行机组同时在进行加工装配。由于航空发动机是飞机的关键核心部件,对其合格率要求极高。因此,一台发动机装配过程中,往往会经过多次分解后再重新装配的过程。

通过以上分析,航空发动机装配车间调度优化问题属于可重入混合车间调度问题。一般可重入混合车间调度问题可描述为:有多个加工工件,每个工件都需经过多道加工工序,且工艺路线都相同,至少有一个工序可在多台并行加工机器上加工,至少存在一个工序在一台机器上或一组机器上加工至少一次以上,根据

收稿日期:2012-02-06;修回日期:2012-05-11

基金项目:国家自然科学基金资助项目(60934008)

作者简介:范金松(1987-),男,江苏沭阳人,硕士研究生,研究方向为生产计划与调度;严洪森,博士,教授,博士生导师,研究方向为知识化制造、生产计划与调度、预测等。

目标函数来确定并行机器的分配情况以及加工机器上工件的加工顺序。Kumar 于 1993 年首次提出了可重入生产线 (Re-entrant lines), 并把它作为区别于 Flowshop 和 Jobshop 的第三类调度问题<sup>[1]</sup>。航空发动机装配车间调度问题属于可重入混合 Flowshop 问题 (Re-entrant Hybrid Flowshop, RHFS)。对于一般 Flowshop 的调度问题都是 NP 难问题, 而可重入混合车间除了具有一般 Flowshop 特点外, 还具有工件可重入、工艺约束更加复杂等特点。针对混合 Flowshop 问题, Brah 和 Hunsucker<sup>[2]</sup>提出了分枝定界法的解决方案、Wittrock<sup>[3]</sup>则给出了启发式算法, 但是他们对规模较小的混合 Flowshop 问题给出了较好的解决方案。针对可重入 Flowshop 问题, Chen 等<sup>[4]</sup>给出了混合遗传算法, 但他们并没有给出混合 Flowshop 问题的解。

因为遗传算法相比其他算法具有更强大的全局搜索能力, 在求解诸如可重入混合车间调度问题方面表现出非常大的优势。为此, 文中以最小化最大完工时间作为航空发动机装配车间调度问题的目标函数, 应用遗传算法对该问题进行求解。实验结果表明, 文中给出的基于遗传算法的航空发动机装配车间智能优化调度方法是非常有效的。

## 1 航空发动机装配车间调度模型

### 1.1 航空发动机装配车间分析

航空发动机装配车间加工工序如图 1 所示。图 1 中, 装配一台发动机至少需要经过十二道工序, 其中部件、传装、总装和试车等四个工序是可重入的, 在发动机第二次经过该工序时, 在图中分别以部件\*、传装\*、总装\*和检试来区别第一次经过此工序。同时在图中还用虚线和虚线框用来说明“返件”和“试车情况”这两件随机概率事件。

当发动机完成“故检”工序任务后, 如果发动机是合格的, 那么就进行“部件\*”工序的加工任务; 如果发动机出现故障等问题需要返件, 此时依据故障问题的具体情况, 有“返回原厂”和“重新加工”两种情况。当发动机完成“检试”工序任务后, 经“试车情况”工序检测发现发动机试车情况不符合相关技术标准, 此时需要重新进入分解等加工工序操作; 否则顺利完成装配任务, 这也是随机概率事件。如果在可重入混合车

间调度问题中再考虑可能发生的随机事件, 会大大增加问题的复杂度, 这将在以后文章中进一步讨论。下面将对装配车间生产线作适当地简化, 假设每台发动机都不会出现返件情况, 且每台发动机的试车情况符合相关技术标准, 即只经过一次工厂阶段加工和一次检验阶段加工。简化后的加工工序模型如图 2 所示。因此, 发动机装配车间调度问题可描述为有  $n$  个工件,  $s$  个工序, 每个工序有  $m_i$  台并行机器 ( $1 \leq i \leq s$ ) 的可重入混合 Flowshop 问题。不考虑发动机每道工序加工前的准备时间, 同时认为工序之间的缓冲区能力为无限大。

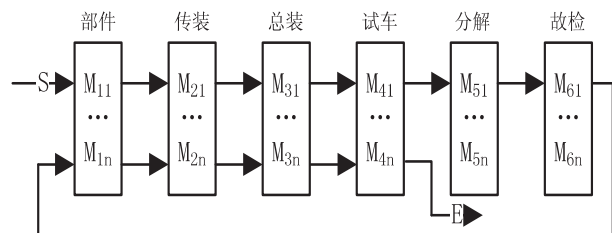


图 2 发动机装配车间加工工序简化模型

### 1.2 航空发动机装配车间调度问题定义与建模

下面介绍在文中发动机装配车间调度问题定义时出现的一些变量, 如下:

$m$ : 加工机器的数量

$i$ : 工序号

$s$ : 工序的数量

$m_i$ : 第  $i$  道工序的并行加工机器数量

$j$ : 装配的发动机号

$n$ : 需装配的发动机数量

$s_{ij}$ : 第  $j$  台发动机的第  $i$  道工序的开始时间

$p_{ij}$ : 第  $j$  台发动机的第  $i$  道工序的作业时间

$k$ : 机器号

$m_{ik}$ : 第  $i$  道工序的  $m_i$  台并行机器中的第  $k$  台机器

$l$ : 工件的加工顺序号

$m_{ikl}$ : 第  $i$  道工序的  $m_i$  台并行机器中的第  $k$  台机器上的第  $l$  个作业

$ms_{ikl}$ : 第  $i$  道工序的  $m_i$  台并行机器中的第  $k$  台机器上的第  $l$  个作业的开始时间

$mp_{ikl}$ : 第  $i$  道工序的  $m_i$  台并行机器中的第  $k$  台机器上的第  $l$  个作业的处理时间

$x_{ikl}$ : 第  $i$  道工序的  $m_i$  台并行机器中的第  $k$  台机

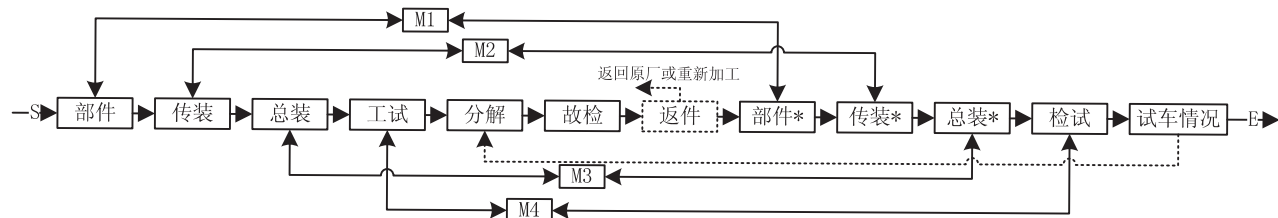


图 1 某航空发动机装配车间加工工序图

器上的第  $l$  个作业处理的是否为第  $j$  台发动机的第  $i$  道工序

$c_{ij}$ : 第  $j$  台发动机的第  $i$  道工序的完成时间

$c_j$ : 第  $j$  台发动机的装配完成时间

根据上述定义,为使重入式混合流水车间调度问题的最大完工时间 (Makespan) 达到最小,我们对问题进行建模如下:

$$\text{Obj: } f = \min \{ \max \{ c_j \mid j = 1, 2, \dots, n \} \} \quad (1)$$

s. t.

$$\sum_{i=1}^s m_i = m \quad (2)$$

$$s_{ij} + p_{ij} \leq s_{i+1,j} \quad i = 1, 2, \dots, s \quad (3)$$

$$x_{ijkl} \in \{0, 1\} \quad (4)$$

$$\sum_{k=1}^{m_i} \sum_{l=1}^n x_{ijkl} = 1 \quad (5)$$

$$s_{ij} = ms_{ikl} \times x_{ijkl} \quad (6)$$

$$s_{ij} \times x_{ijkl} < s_{ij'} \times x_{ij'kl}$$

$$\forall (j, j') \in \{(j, j') \mid j < j'\} \quad (7)$$

$$\forall (l, l') \in \{(l, l') \mid l < l'\} \quad (7)$$

$$c_{ij} = s_{ij} + p_{ij} \quad (8)$$

$$c_j = \max(s_{ij} + p_{ij}) \quad (9)$$

$$i = 1, 2, \dots, s; j = 1, 2, \dots, n \quad (10)$$

$$k = 1, 2, \dots, m_i; l = 1, 2, \dots, n \quad (11)$$

## 2 遗传算法

### 2.1 基本遗传算法

美国 Michigan 大学 J. Holland 教授于 1975 年在其专著《Adaptation in Natural and Artificial Systems》<sup>[5]</sup>中首次提出遗传算法 (Genetic Algorithm), 其思想是模拟 Darwin 的生物进化论自然选择和 Mendel 的遗传学机理生物进化过程, 是一种通过模拟生物进化过程来搜索最优解的方法。J. Holland 教授所提出的 GA 通常为简单遗传算法 (SGA)。

在文中, 使用遗传算法来解决以最大完工时间为适应度函数的 RHFS 调度问题。

### 2.2 参数设置

在遗传算法中参数包括种群规模、迭代次数、交叉概率、变异概率和其他一些操作参数。

### 2.3 编码

使用遗传算法最重要的工作就是编码。对于调度问题, 已经有多种编码方法。在众多编码方法中, 基于工件的编码、基于加工机器的编码和基于操作的编码都是优化问题中经常使用的编码方法。孙承夏等<sup>[6]</sup>提出“带有标签机制”的编码方式, 以此来抑制不可行解的生成, 较为复杂; 王永等<sup>[7]</sup>提出预先对生产线考虑了八种调度策略规则, 然后使用 3 位二进制来对八种调

度策略规则进行编码, 虽然编码简单, 但调度策略需要预先设计且不够灵活。在文中, 采用一种巧妙的基于加工机器的编码方法来进行编码操作。

文献[8,9]给出了一种随机键表示方法来求调度优化问题, 该编码方法是对各个工序的加工机器号进行编码, 采用浮点数表示法, 其中整数部分用于确定加工该工序的并行机器号, 而小数部分则用于确定当有多个工件在同一台机器上加工时的选择加工情况决策。文献[10]中指出, 基于实数编码的遗传算法比基于二进制编码的遗传算法的收敛速度快且精度高。由于染色体中每个基因值都是随机实数, 不仅很好地解决了机器冲突问题, 而且能保证每条染色体都对应一个可行的调度解。

假设有  $n$  个工件,  $s$  道工序, 每个工序的并行加工机器数为  $m_i$  ( $1 \leq i \leq s$ ), 且并行加工机器数至少为一台 ( $\forall m_i \geq 1, \exists m_i > 1$ )。下面构造一个  $s \times n$  维的随机实数矩阵:

$$A_{s \times n} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{s1} & \cdots & a_{sn} \end{bmatrix} \quad (12)$$

上式中, 矩阵元素  $a_{ij}$  是区间  $(1, m_i + 1)$  上的一个随机实数,  $i$  表示工序号,  $j$  表示工件号,  $|a_{ij}|$  表示并行加工机器号, 符号  $|x|$  表示对实数  $x$  向下取整。当  $|a_{ij}| = |a_{i\beta}|$ ,  $j \neq \beta$  时, 说明同一台加工机器需要对多个工件进行同一个工序的加工。对于第一个工序 ( $i = 1$ ), 按照  $a_{1j}$  的升序来加工工件; 其余工序 ( $i > 1$ ), 则需要根据前一个工序的完成时间和  $a_{ij}$  值综合考虑来确定加工顺序。染色体由随机矩阵来确定, 染色体由  $s$  个小段组成, 即矩阵的  $s$  行, 每段包括  $n$  个基因, 段与段之间用间隔符“0”隔开, 长度为  $s \times n + s - 1$ 。根据上述分析, 染色体可表示为:

$$[a_{11}, a_{12}, \dots, a_{1n}, 0, a_{21}, \dots, a_{2n}, 0, \dots, 0, a_{s1}, a_{s2}, \dots, a_{sn}] \quad (13)$$

### 2.4 种群初始化与适应度函数定义

根据式(12)所述的方法得到以随机实数为元素的随机矩阵, 依据所得随机矩阵用式(13)表示, 即得到一条染色体。根据算法参数中种群规模的大小, 依次生成染色体数, 即可初始化种群。

很显然, 航空发动机装配车间调度是多目标优化调度问题, 比如, 最小化最大完成时间、加工周期、瓶颈机器利用率等, 同时这些目标之间又存在着各种各样的复杂联系。在文中, 定义如下函数作为适应度函数:

$$f_{\tau} = \frac{1}{c_{\max}(\tau)} \quad (14)$$

其中,  $c_{\max}(\tau)$  是第  $\tau$  条染色体所代表的一个调度

的最大加工周期。

2.5 遗传算法操作

2.5.1 选择操作

遗传算法的原理从本质上来说基于达尔文的自然选择学说。选择是遗传算法的驱动力。如果驱动力太大,遗传搜索将过早地终止;如果驱动力太小,进化过程将缓慢难以接受。通常需要在遗传操作搜索的早期采用较小的选择压力,而在晚期则推荐采用较大的选择压力。选择将遗传操作引导至搜索空间中有前途的区域中。在过去的 20 年中提出、验证了许多选择方法。通常采用的选择方法有:轮盘赌选择、 $(\mu + \lambda)$  选择、竞争选择、稳态复制、排序与比例变换和共享<sup>[11]</sup>。

文中采用轮盘赌选择方法。Holland 提出的轮盘赌选择是最知名的选择方式<sup>[5]</sup>,其基本原理是根据每个染色体适应值的比例来确定该个体的选择概率或生存概率。设  $f_{\tau}$  为群体  $\text{pop}(t)$  中染色体个体  $y_{\tau}$  的适应度函数,群体规模为  $\text{pop\_size}$ ,  $\sum_{\tau=1}^{\text{pop\_size}} f_{\tau}$  (简记为  $\sum f_{\tau}$ ) 为群体中所有染色体个体适应度函数之和。令  $\alpha = 2, 3, \dots, \text{pop\_size}$ , 计算  $\sum_{\tau=1}^{\alpha} f_{\tau}$ , 形成  $\text{pop\_size}$  个区间:

$$\left[0, \frac{f_1}{\sum f_{\tau}}\right] \left[\frac{f_1}{\sum f_{\tau}}, \frac{f_1 + f_2}{\sum f_{\tau}}\right] \dots \left[\frac{\sum_{\tau=1}^{\alpha-1} f_{\tau}}{\sum f_{\tau}}, \frac{\sum_{\tau=1}^{\alpha} f_{\tau}}{\sum f_{\tau}}\right] \dots$$
$$\left[\frac{\sum_{\tau=1}^{\text{pop\_size}-1} f_{\tau}}{\sum f_{\tau}}, 1\right]$$

对于每次随机产生的染色体,如果有

$$\frac{\sum_{\tau=1}^{\alpha-1} f_{\tau}}{\sum f_{\tau}} \leq \frac{\sum_{\tau=1}^{\alpha} f_{\tau}}{\sum f_{\tau}}$$

成立,则表示染色体个体  $y_{\tau}$  被选中。

2.5.2 交叉操作

所谓交叉就是从父辈种群中选取两个个体进行基因交换从而获得新个体的过程。在传统遗传算法中,交叉操作是主要操作,遗传算法的性能在很大程度上依赖于交叉操作的性能。依据上述编码方式,当对于每个基因值满足  $a_{ij} \in (1, M_i + 1)$  时,就可保证个体的合法性。每条染色体都是由各基因段组成,同时为了达到较好的交叉操作性能,采取分段交叉的方式,依据基因子段的长度随机产生每段的交叉点,对每个小段分别进行随机点数交叉<sup>[12]</sup>。为了充分保证染色体个体的多样性,首先从种群中随机选取两个染色体,作为本次交叉操作的父代染色体;然后在两个选出的父代染色体的各段中随机找到一个基因位,再随机生成

用来进行交叉的基因长度,只要此基因段长度不超过子基因段长度即可。然后交换各段中随机选取的一部分基因,得到子代染色体个体。分段交叉操作示意图如图 3 所示。

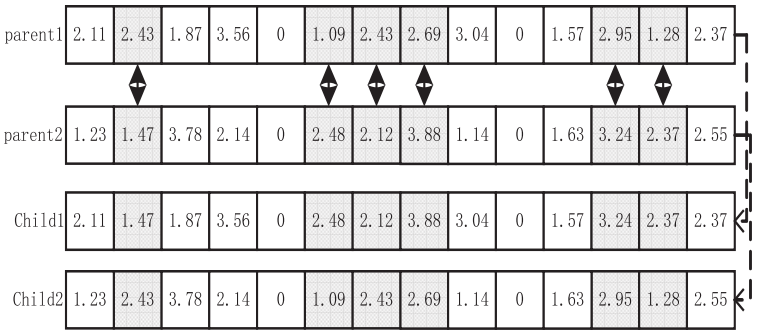


图 3 分段交叉操作示意图

2.5.3 变异操作

每条染色体都是由各基因段组成,因此变异操作也必须采取分段变异的方式,具体变异操作步骤如下:

- a)  $d = \text{rand}(-1, 1)$  ;
- b) 如果  $d = 1$ , 则  $r = \text{rand}(0, m_i - a_{ij})$  , 否则  $r = \text{rand}(0, a_{ij} - 1)$  ;
- c)  $a'_{ij} = a_{ij} + d \times r$  .

其中,  $d$  是随机整数,  $r$  是随机实数,  $a_{ij}$  是变异前的基因值,  $a'_{ij}$  是变异后的基因值。

3 实验仿真

现假设要装配 8 种型号的发动机,每台发动机有 10 个加工装配工序,包括 4 个需要重复进行的加工工序。这里为了减少实验用例的复杂度,还假设每道工序的并行机都是 3 台。为了使用例更具实际意义,假设每种发动机在每道工序上的加工时间都不相同,但同一工序内各并行机加工时间相同。各工序的作业时间如表 1 所示。

表 1 各工序的作业时间

机型	作业时间								
	部件	传装	总装	工试	分解	故检	部件*	传装*	总装*
1	2	3	3	3	2	2	2	3	3
2	3	3	2	2	3	2	3	3	2
3	2	3	2	3	1	1	2	3	2
4	1	2	3	3	3	2	1	2	3
5	2	2	3	2	2	1	2	2	3
6	3	3	2	3	1	2	3	3	2

本算法采用 VS2008 C#编程,算法中使用的参数为  $P_c = 0.80$ ,  $P_m = 0.32$ , 种群规模  $\text{pop\_size} = 50$ , 种群经过 500 代的进化,得到最好的染色体的甘特图如图 4 所示。由甘特图很容易得到最大完工时间为 39。

图 4 清晰地反应了各并行加工机器上的加工任务分配情况和各工件工序的加工时间情况。一般分析可



知,为了保证加工时间尽可能地短,需满足加工任务分配尽可能均匀,工件的加工工序尽可能保持连续。从图 4 可知,在调度前期基本上满足了这些条件,但是随着加工的进行,后面加工工序任务分配稍有分散,这是混合制造车间的必然现象。

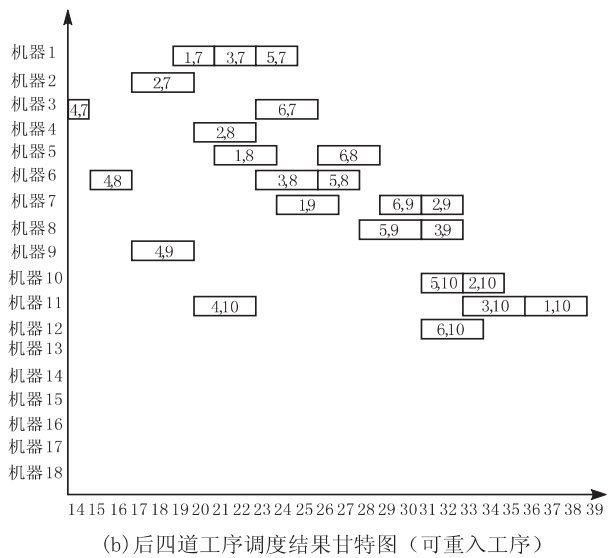
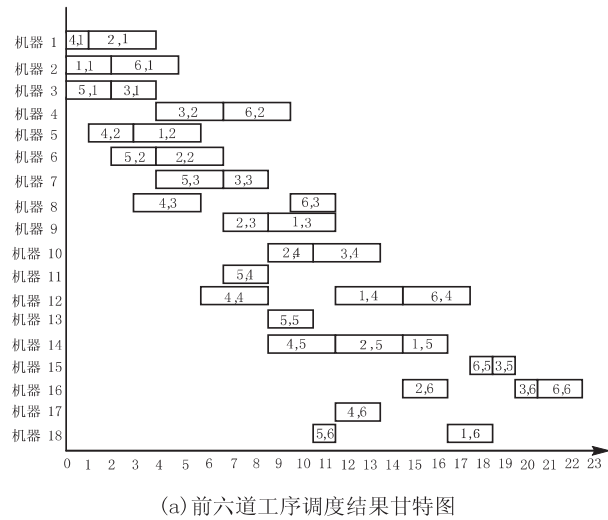


图 4 调度结果甘特图

最后,对该算法连续运行 10 次,每次由运算结果得到的目标函数值:40、39、41、39、40、41、40、39、39、40。上述数据表明本算法在 10 次运行中有 4 次达到最小值,运行结果基本上趋于稳定,由于遗传算法本身的特点,可知本算法是稳定有效的。

4 结束语

文中针对航空发动机装配车间这一类可重入混合 Flowshop 问题提出了基于遗传算法的解决途径。文中采用随机矩阵法对染色体进行编码,这使得遗传操作变得简单,同时不会产生非法染色体,通过实验仿真结果表明遗传算法也可以较好地解决可重入混合 Flowshop 问题。虽然遗传算法可以解决文中提出的调度问题,但是问题的规模还比较小,复杂度还比较低,并不能保证随着问题规模的增加,算法依然保持较好的效果,同时遗传算法本身还存在一些缺陷。因此,在接下来的时间中,笔者将继续研究如何对算法作必要的改进,克服算法本身所存在的一些缺陷,用于解决实际情况下的可重入混合 Flowshop 问题。

参考文献:

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

[2] Brah S A,Hunsucker J L. Branch and bound algorithm for the flowshop with multiple processors [J]. European Journal of Operational Research,1991,51(1):88-99.

[3] Wittrock R J. An adaptable scheduling algorithm for flexible flow lines[J]. Operations Research,1988,36(3):445-453.

[4] Chen J S,Pan J C H,Lin C M. A hybrid genetic algorithm for the re-entrant flow-shop scheduling problem[J]. Expert Systems with Applications,2008,34(1):570-577.

[5] Holland J. Adaptation in Natural and Artificial Systems[M]. [s. l.]:MIT Press,1992.

[6] 孙承夏,郭 禾. 一种有效的遗传算法在重入式生产调度问题中的应用[J]. 软件,2010,31(11):62-67.

[7] 王 永,吴智铭,隋 义. 基于遗传算法的可重入半导体生产线的调度[J]. 计算机仿真,2007,24(12):247-250.

[8] 吴云高,王万良. 基于遗传算法的混合 Flowshop 调度[J]. 计算机工程与应用,2002,38(12):82-84.

[9] 冯碧琤,乔 非,王 坚. 基于遗传算法的半导体生产线方法研究[J]. 计算机工程,2005,31(13):145-147.

[10] 刘 民,吴 澄. 制造过程智能优化调度算法及其应用[M]. 北京:国防工业出版社,2008:43-48.

[11] 玄光男,程润伟. 遗传算法与工程优化[M]. 北京:清华大学出版社,2004:6-7.

[12] 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[J]. 同济大学学报,2011,39(5):726-730.

(上接第 204 页)

少? [N]. 计算机世界,2003-01-17.

[9] 沈贺丹,潘亚楠,邵良杉. 关于搜索引擎的研究综述[J]. 计算机技术与发展,2006,16(4):147-148.

[10] 丁兆贵,金 敏. 基于 Lucene 的个性化搜索引擎研究与实现[J]. 计算机技术与发展,2011,21(2):106-107.

[11] 王 坚,赵恒永. 专业搜索引擎中文分词算法的实现与研究[J]. 福建电脑,2005(7):55-57.

[12] Wen Kunmei,Lu Zhengding,Li Yuhua,et al. A Cooperative Schema between Web Server and Search Engine for Improving Freshness of Web Repository[J]. Wuhan University Journal of Natural Sciences,2006,11(1):11-14.