

# 基于元胞机和改进 GA 的大型零件 柔性作业车间调度算法

陈 勇 阮幸聪 鲁建厦 王亚良

浙江工业大学特种装备制造与先进加工技术教育部重点实验室,杭州,310032

**摘要:**针对大型零件柔性作业车间调度问题,采用改进遗传算法优化元胞机局部演化规则,提出了元胞机和改进遗传算法相结合的混合调度算法。依据总加工时间最短、各工位负荷率高、同一工位组各工位负荷平衡率高的优化目标,建立了离散化后单个静态调度单元的遗传算法优化模型,并结合算例具体说明了优化过程。通过文献实例演算验证了混合算法求解大型零件柔性作业车间调度问题的可行性和有效性。

**关键词:**元胞机;遗传算法;大型零件;柔性作业车间调度

**中图分类号:**TB497

**DOI:**10.3969/j.issn.1004-132X.2012.02.012

## Flexible Job—Shop Scheduling Algorithm of Large Parts Based on Cellular Automata and Improved GA

Chen Yong Ruan Xingcong Lu Jiansha Wang Yaliang

Key Laboratory of Special Purpose Equipment and Advanced Processing Technology,  
Ministry of Education, Zhejiang University of Technology, Hangzhou, 310032

**Abstract:** According to the flexible job—shop scheduling problem of large parts, the local evolution rule of cellular automata was optimized using an improved GA and a hybrid scheduling algorithm combining cellular automata with improved GA was proposed. Based on three optimization objectives of minimizing the total processing time, maximizing load rates of all the work stations and maximizing balance rate of a group of work stations of the same type, A GA optimization model for all the static scheduling units was built, and the optimization process was explained by use of a numerical example. The feasibility and efficiency of the hybrid scheduling algorithm of cellular automata and improved GA was verified through applying it into actual example referenced from literature.

**Key words:** cellular automata; genetic algorithm(GA); large part; flexible job—shop scheduling

## 0 引言

大型零件柔性作业车间调度问题作为一类亟待解决的调度问题,具有以下特点<sup>[1]</sup>:①单件生产和小批量生产相结合,工件陆续到达;②经常受到插单、返工返修和设备故障等随机因素的干扰,动态性强;③每个工件每道工序所需加工时间长,加工的设备不唯一,并且不同加工设备所需的加工时间不同;④工件体积大,生产过程中的缓存空间不可忽略。

柔性作业车间调度的方法大致分为运筹学方法、启发式搜索算法、仿真模型法。运筹学方法属于精确方法,能保证得到全局最优解,但存在实际生产环境的不确定因素、动态因素描述困难和求

解空间太大等问题。启发式搜索算法适用于大规模调度问题,能较快得到问题的较优解,满足解决实际问题的需要,但存在早熟、局部收敛、寻优效率低等缺点。生产调度系统的仿真建模法有框图法、Petri 网技术法<sup>[2]</sup>、系统结构重现法<sup>[3]</sup>、Multi Agent 系统开发法<sup>[4]</sup>等。

现有研究表明,上述的单一方法已经不能满足柔性作业车间调度问题研究的需要,两种或两种以上方法的结合成为当前研究的新趋势,如混合算法、仿真模型嵌套算法、重调度模型嵌套算法等<sup>[5]</sup>。文献<sup>[1]</sup>提出了基于元胞机建模的方法,建立了大型零件柔性作业车间调度的仿真模型,通过简单的局部演化规则模拟复杂的动态调度,并通过实例验证了模型的可行性和有效性。该模型获得的调度方案虽优于实际方案,但结果并非最优。本文在此基础上嵌入改进遗传算法,优化元

收稿日期:2011—01—20

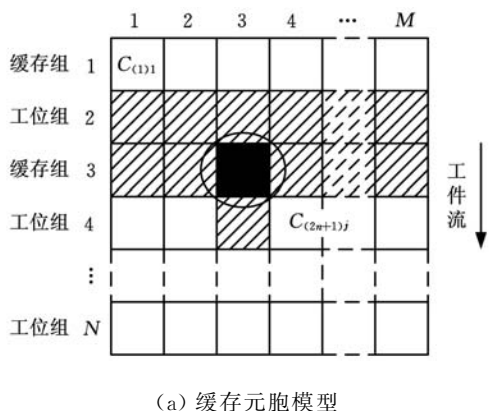
基金项目:国家自然科学基金资助项目(70971118);浙江省自然科学基金资助项目(Y607456,Y6090475)

胞机的局部演化规则,以获得更优的调度方案。

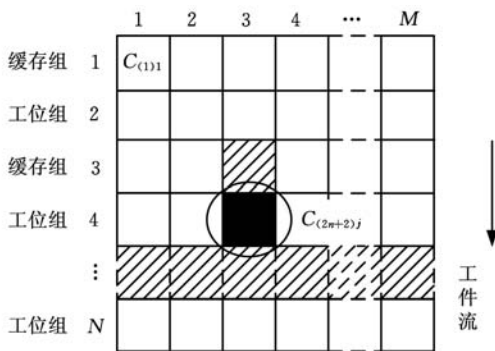
### 1 大型零件柔性作业车间调度元胞机建模

元胞机是一个时间与空间都离散的动力系统,它在一个由有限状态的元胞组成的离散空间上,按照一定的局部演化规则进化<sup>[6]</sup>。标准元胞机由元胞、元胞空间、邻域和局部演化规则构成,具有离散性、同质性、并行性、局部性和维数高等特点。

如图 1 所示,大型零件柔性作业车间调度的元胞机模型由缓存元胞  $C_{(2n+1)j}$  ( $j, n$  均为整数)、工位元胞  $C_{(2n+2)j}$ ,以及不断进出元胞空间的工件粒子组成。



(a) 缓存元胞模型



(b) 工位元胞模型

图 1 大型零件柔性作业车间调度元胞机模型图

根据元胞状态函数  $S_i^{(t+1)} = f(S_i^{(t)}, S_N^{(t)})$ ,任一缓存元胞  $t + 1$  时刻的状态:

$$S_{C_{(2n+1)j}}^{(t+1)} = f[S_{C_{(2n+1)j}}^{(t)}, (S_{C_{(2n)j}}^{(t)}, S_{C_{(2n+1)m}}^{(t)}, S_{C_{(2n+2)j}}^{(t)})] \quad (1)$$

式中,  $f$  为局部状态转换规则即作业调度规则;  $S_{C_{(2n+1)j}}^{(t)}$  为缓存元胞  $C_{(2n+1)j}$  在  $t$  时刻的状态;  $S_{C_{(2n)j}}^{(t)}$  为缓存元胞  $C_{(2n+1)j}$  上游工位组在  $t$  时刻的状态集合;  $S_{C_{(2n+1)m}}^{(t)}$  ( $m \neq j$ ) 为缓存元胞  $C_{(2n+1)j}$  所在缓存组中其他元胞在  $t$  时刻的状态集合;  $S_{C_{(2n+2)j}}^{(t)}$  为缓存元胞  $C_{(2n+1)j}$  下游工位组中与其对应的工位元胞在  $t$  时刻的状态。

缓存元胞  $C_{(2n+1)j}$  的邻域就是图 1a 中的阴影部分,同样,任一工位元胞  $t + 1$  时刻的状态表示为

$$S_{C_{(2n)j}}^{(t+1)} = f[S_{C_{(2n)j}}^{(t)}, (S_{C_{(2n-1)j}}^{(t)}, S_{C_{2n+1}}^{(t)})] \quad (2)$$

式中,  $S_{C_{(2n)j}}^{(t)}$  为工位元胞在  $t$  时刻的状态;  $S_{C_{(2n-1)j}}^{(t)}$  为工位元胞  $C_{(2n)j}$  上游工位组中与其对应的缓存元胞  $C_{(2n-1)j}$  在  $t$  时刻的状态;  $S_{C_{2n+1}}^{(t)}$  为工位元胞  $C_{(2n)j}$  下游缓存组所有缓存元胞在  $t$  时刻的状态集合。

工位元胞  $C_{(2n)j}$  的邻域就是图 1b 中的阴影部分。

元胞状态属性根据仿真过程中的变化情况可分为静态属性和动态属性两类。静态属性值的设置始于仿真开始前,是模型基本参数,不随仿真时间的变化而变化;动态属性值用于记录仿真过程中系统的变化,随着仿真的运行而变化。

缓存元胞  $t$  时刻的状态属性表示为

$$S_{C_{(2n+1)j}}^{(t)} (\omega_{ct}, \omega_{co}, \omega_{cl}, l_q, \omega_q) \quad (3)$$

$$\omega_{cl} = \omega_{ct} - \omega_{co}$$

$$\omega_q = \sum (\omega_{L_i} / e_p)$$

式中,  $\omega_{ct}$  为缓存元胞的在制品空间总容量,是静态属性,不同缓存元胞所含有的空间总容量不同;  $\omega_{co}$  为缓存元胞中已被占用的容量,是动态属性,  $0 \leq \omega_{co} \leq \omega_{ct}$ ;  $\omega_{cl}$  为该缓存元胞未被占用的容量,是动态属性;  $l_q$  ( $l_q \in \mathbf{N}$ ) 为队列长度,是动态属性,表示缓存元胞中等待加工的工件个数;  $\omega_q$  为等待时间,是动态属性,等于缓存元胞的工件队列中最后一个加工工件需要等待的时间;  $d_i$ ;  $\omega_{L_i}$  为零件对应的工序所需的加工能力;  $e_p$  为加工效率,是静态属性,为一天能完成的代表工序数。

工位元胞  $t$  时刻的状态属性表示为

$$S_{C_{(2n)j}}^{(t)} (s_i, e_p, T, s_{ct}, s_{co}, s_{cl}) \quad (4)$$

式中,  $s_i$  为所属工位组,是静态属性;  $T$  为工位一个调度周期内总共能用于加工的时间,是静态属性;  $d_i$ ;  $s_s$  为工位忙碌状态,是动态属性,  $s_s \in \{0, 1, 2\}$ , 0 表示空闲, 1 表示忙碌, 2 表示故障;  $s_{ct}$  为工位一个调度周期内的总加工能力,是静态属性,  $s_{ct} = e_p T$ ;  $s_{co}$  为工位已被占用的加工能力,是动态属性,  $0 \leq s_{co} \leq s_{ct}$ ;  $s_{cl}$  为工位元胞剩余的加工能力,是动态属性,  $s_{cl} = s_{ct} - s_{co}$ 。

工件粒子  $t$  时刻的状态属性表示为

$$S_p^{(t)} (p_i, p_t, p_n, s_{ft}, \omega_{L_i}, s_n, d_p, q_n, t_n) \quad (5)$$

式中,  $S_p^{(t)}$  为粒子  $p_k$  在  $t$  时刻的状态;  $k$  为工件编号;  $p_i$  为粒子所需加工的工序总数,是静态属性;  $p_t$  为粒子已经完成的工序数,是动态属性;  $p_n$  为粒子的下一道工序编号,是动态属性;  $s_n$  为粒子下道工序所属的工位组,是动态属性;  $s_n$  为粒子占用的空间,是静态属性,对应缓存区的  $\omega_{cl}$ ;  $d_p$  为工件加工优先级,是静态属性,由工件的交货期决定,交货期越紧优先级越高,对应的  $d_p$  数值越大;  $q_n$  为队列排序的序号,是动态属性;  $t_n$  为到达元胞的时间,是动态属性。

模型的局部演化规则有三条,即工位选择规则、工件排序规则和任务触发规则,具体如表 1 所示。

表 1 大型零件柔性作业车间调度元胞机模型局部演化规则表

规则名称	符号表示	作用	相关因素
工位选择规则	$R_{p \rightarrow c}$	确定工件下道工序加工所在的工位, 决定粒子下一时步所要进入的缓存元胞。	$s_{ik}, s_{is}, e_{ip}, \tau_{\omega_q}, \tau_{\omega_l}, \tau_{\omega_{cl}}, s_n$
工件排序规则	$R_{c \rightarrow p}$	确定工位下一个加工的工件, 即决定与该工位元胞绑定的缓存元胞中等待粒子的排列顺序。	$t_n, d_p, l_q, q_n$
任务触发规则	$R_{ta}$	确定工位元胞的加工任务发生与否	$s_s, l_q$

模型的初始演化规则基于排队论和工件的加工优先级, 工位选择主要依据加工完成时间的长短和工件体积的大小; 工件排序主要根据工件到达的先后和加工优先级。

## 2 改进 GA 优化元胞机局部演化规则的建模

元胞机模型已经将柔性作业车间调度问题在时间、空间上作了离散化处理, 在同一调度周期内, 整个系统的调度转化为多个工位组的静态调度, 研究对象变成多个由一个工位元胞组和一个缓存元胞组组成的离散单元。直接将一个调度周期定义为一个仿真时步。大型零件生产车间工件加工时间长, 因此仿真时步一般较长。

设一个工位组和其对应的缓存组为一个单元, 每一个时步内, 每个单元的调度机制相同, 整个模型转化为多个静态柔性调度问题。对应描述如下:  $n$  个工件  $\{p_1, p_2, \dots, p_n\}$  要在  $m$  个工位  $\{s_1, s_2, \dots, s_m\}$  上加工; 每个工件只包含一道待加工的工序, 每道工序可选择的工位不止一个, 且不同工位所需的加工时间不同; 单个工件各道工序的加工顺序已定。各单元每一时步的调度目标有两个: ① 确定所有工序的加工工位; ② 确定每个工位上工件的加工顺序, 即工位对应缓存中工件的队列顺序。本文的最优目标有 3 个: 各道工序完工时间早、各工位负荷率高、同一工位组中所有工位负荷平衡。下面, 利用遗传算法对工位选择规则  $R_{p \rightarrow c}$  和工件排序规则  $R_{c \rightarrow p}$  进行编码, 优化局部演化规则。

### 2.1 元胞机静态调度单元建模

柔性作业车间调度的特点是存在一定的扰动因素, 如设备故障、插单、返工等。插单和返工主要体现在调度单元的窗口粒子群随仿真时步的推进不断更新, 进而实现系统再调度; 设备故障作为

一个约束条件, 影响工位的加工能力。

元胞机静态调度单元模型可描述为:  $n$  个工件粒子在包含  $m$  个同类工位的工位组中加工,  $m$  个工位分别对应  $m$  个缓存。每个工位的加工效率不同, 每个工件的工序根据实际要求可选择工位组中某几个工位或全部工位, 每道工序在不同工位上加工所需时间不同。设工件粒子集  $P = \{p_1, p_2, \dots, p_n\}$ ; 工位集  $S = \{s_1, s_2, \dots, s_m\}$ ; 缓存集  $B = \{b_1, b_2, \dots, b_m\}$ ; 工序  $O_{ij}$  表示第  $i$  个工件的第  $j$  道工序, 静态调度单元中, 每个工件只有一道工序, 即每个工件对应一个  $j$ ; 第  $x$  个工位组的加工时间矩阵用  $T$  表示,  $T_{ijxy}$  为矩阵中的一个元素, 表示第  $i$  个工件的第  $j$  道工序在第  $x$  个工位组的第  $y$  个工位上加工所需要的时间; 每个工位对应一个设备, 工位  $y$  上设备的故障率表示为  $\lambda_{xy}$ ;  $\omega_1, \omega_2, \omega_3$  为子目标权重。

前述三个目标对应的目标函数如下:

(1) 所有工序总加工时间最短, 即

$\min \sum_{i=1}^n T_i A_i$ , 用  $F_1$  表示,  $0 < F_1 \leq 1$ 。为了配合后面的两个子目标  $F_2$  和  $F_3$ , 使之相对统一, 对  $\min \sum_{i=1}^n T_i A_i$  做如下处理, 转化为最大值, 且不至于过分接近于 0:

$$F_1 = \sum_{i=1}^n \min T_{iy} / \sum_{i=1}^n T_i A_i \quad y = 1, 2, \dots, m \quad (6)$$

式中,  $\sum_{i=1}^n \min T_{iy}$  为所有工件都选择加工时间最短的工位进行加工时总的加工时间, 是理想状态的最小值;  $\sum_{i=1}^n T_i A_i$  为每个可行解的实际总加工时间;  $T_i$  为一列包含  $m$  个元素的行矩阵, 每个元素对应一个  $T_{ijxy}$ ;  $A_i$  为一列包含  $m$  个元素的列矩阵, 元素  $a_{iy} \in \{0, 1\}$ 。

(2) 各工位负荷率高, 转化为所有工位总利用率高, 用  $F_2$  表示:

$$F_2 = \sum_{i=1}^n (T_i A_i) / \sum_{y=1}^m [T_s (1 - \lambda_y)] \quad (7)$$

式中,  $T_s$  为每个仿真步长所代表的实际时间, 仿真开始之前设定, 仿真过程中不变。

(3) 同一工位组中所有工位负荷平衡, 平衡率用  $F_3$  表示:

$$F_3 = \sum_{i=1}^n T_i A_i / (m C_{\max}) \quad (8)$$

式中,  $C_{\max}$  为负荷最大的工位所耗费的加工时间。

通过权重加合法得到元胞机的每个静态调度单元的调度总目标函数:

$$F = \max(\omega_1 F_1 + \omega_2 F_2 + \omega_3 F_3) \quad (9)$$

加工约束条件如下:

(1) 资源约束。同一台设备同一时刻只能加工一个工件,表示为

$$a_{iy} = \begin{cases} 1 & \text{选择工位组的第 } y \text{ 个工位} \\ & \text{加工工件 } i \text{ 的第 } j \text{ 道工序} \\ 0 & \text{不选择工位组的第 } y \text{ 个工位} \\ & \text{加工工件 } i \text{ 的第 } j \text{ 道工序} \end{cases}$$

同一工件的同一道工序只能被一台设备加工,表示为

$$\sum_{y=1}^m a_{iy} = 1$$

每个工件的每道工序一旦开始就不能停止。

(2) 工艺约束。不同工件的工序之间没有先后约束;同一工件的工序之间有先后约束。

(3) 权重约束。每个子目标的权重  $w_i (i=1, 2, 3)$  介于 0 和 1 之间,即  $0 \leq w_i \leq 1$ ;所有子目标的权重之和为 1,即  $w_1 + w_2 + w_3 = 1$ 。

沿用文献[1]的算例 1 做具体说明。已知元胞机模型仿真的第  $t$  个时步有 8 个工件( $p_1, p_2, \dots, p_8$ ) 进入 2 号工位组,该工位组包含 5 台同类型通用设备( $s_1, s_2, \dots, s_5$ ),但设备的性能、效率不同。各工件在各台设备上加工所需的时间如表 2 所示,“—”表示工件  $p_i$  不能在工位  $s_j$  上加工。虽然静态调度单元中每个工件只有一道工序,但从全局看,每个工件都包含多道工序,因此给出工序编号,便于最终全局调度方案的说明。算例 1 对应的元胞机模型如图 2 所示。

表 2 算例 1 设备加工时间表

工件	工序	可选择加工工位				
		$s_1$	$s_2$	$s_3$	$s_4$	$s_5$
$p_1$	$O_{13}$	12	6	6	4	3
$p_2$	$O_{22}$	6	—	3	2	—
$p_3$	$O_{32}$	—	9	—	6	4.5
$p_4$	$O_{41}$	4	—	2	—	1
$p_5$	$O_{53}$	—	9	9	6	—
$p_6$	$O_{65}$	10	—	5	3.3	2.5
$p_7$	$O_{72}$	6	3	—	2	1.5
$p_8$	$O_{87}$	8	4	4	—	2

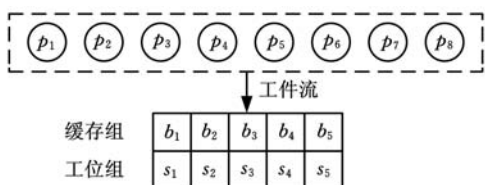


图 2 算例 1 元胞机模型

### 2.2 染色体编码和解码

染色体编码和解码是两个互逆的过程,是实际问题的解和染色体的相互转换。根据已确定的两个目标(确定静态调度单元所有工序的加工工位,确定每个工位上工件的加工顺序),将调度编码分为两部分:① 工位染色体基于工位分配的编

码确定所选的加工工位,对应工位选择子问题。

② 工序染色体基于工序的编码确定工序间的先后加工顺序,对应工件排序子问题。融合这两种编码,形成一条染色体(对应单元调度的一个可行解)。根据张国辉等[7]提出的编码方式和 Witkowski 等[8]应用的遗传编码方法,结合大型零件柔性作业车间静态调度单元自身的特点,采用如下染色体编码方式。

(1) 工位染色体。在每个静态调度单元中,各工件均只包含一道工序,因此工件数等于工序数  $n$ ,工序编号和工件编号一致,分别是  $1, 2, \dots, n$ 。各工序可选择的工位子集分别为  $S_1, S_2, \dots, S_n$ 。算例 1 中,工件  $p_1$  对应工序  $O_{13}$  的可选工位集合  $S_1 = \{s_1, s_2, s_3, s_4, s_5\}$ ,工件  $p_2$  对应的工序  $O_{22}$  的可选工位集合  $S_2 = \{s_1, s_3, s_4\}$ 。第一部分的基因串长为  $n$ ,用  $g_1, g_2, \dots, g_n$  表示,  $g_i \in \{1, 2, \dots, m\}$ ,  $i (i=1, 2, \dots, n)$  为工位的编号。根据可选工位集合中随机选中的工位编号确定每道工序加工的工位,若  $g_1 = 5$ ,则工件  $p_1$  对应的工序  $O_{13}$  的加工工位为  $s_5$ ,依次类推确定余下工序的加工工位。第一部分的基因串为  $5-4-2-1-4-5-2-3$ ,这 8 个工件 8 道工序的加工工位的有序排列为  $s_5 - s_4 - s_2 - s_1 - s_4 - s_5 - s_2 - s_3$ 。

(2) 工序染色体。静态调度单元中,每个工件只包含一道工序,所以工序染色体中基因的个数等于单元中的总工序数(总工件数),即染色体的串长为  $n$ 。工序染色体表示为  $g'_1, g'_2, \dots, g'_n$ ,其中,  $g'_j \in \{1, 2, \dots, n\}$ 。算例 1 中,假设工序染色体部分的一个可行解为  $1-2-3-4-8-6-7-5$ ,则工件的加工顺序为  $p_1 - p_2 - p_3 - p_4 - p_8 - p_6 - p_7 - p_5$ ,对应工序的加工顺序为  $O_{13} - O_{22} - O_{32} - O_{41} - O_{87} - O_{65} - O_{72} - O_{53}$ 。加工顺序是缓存元胞内的队列排序的关键依据。

解码就是编码的一个逆向过程,先根据工位染色体部分确定每道工序的加工工位,再根据工序染色体部分确定已分配到每个工位的相关工序的加工顺序。结合所有静态调度单元的调度方案,获得元胞机模拟的整个车间的调度方案。调度方案将随着仿真时步的推进实时更新,形成动态调度。

### 2.3 获取初始种群

遗传算法初始解的优劣直接关系到获得最优解的迭代次数和收敛速度。在调度模型中,遗传算法嵌套于元胞机模型,用于寻找最优演化规则。由于染色体包含工位选择和工序排序两部分,且模型对收敛速度要求高,因此在编码过程

中,工位选择染色体的初始解根据  $\omega_q + \omega_L/e_p$  值最小的工位选择标准获得,工序排序染色体的初始解则根据 FCFS 规则并结合工件加工优先级获得。单元调度研究的是时间、空间离散的单步调度,所以每个研究单元的工件可视为同时到达。算例 1 按照  $p_1 - p_2 - p_3 - p_4 - p_5 - p_6 - p_7 - p_8$  的顺序选择工位,得到的其中一个工位选择初始

解为  $5 - 4 - 5 - 3 - 4 - 3 - 2 - 2$ ,对应设备为  $s_5 - s_4 - s_5 - s_3 - s_4 - s_3 - s_2 - s_2$ ;工序排序初始解为  $1 - 2 - 4 - 7 - 5 - 6 - 3 - 8$ ,对应工件  $p_1 - p_2 - p_4 - p_7 - p_5 - p_6 - p_3 - p_8$ ,对应工序为  $O_{13} - O_{22} - O_{41} - O_{72} - O_{53} - O_{65} - O_{32} - O_{87}$ 。具体执行过程如图 3 所示。

工件	$p_i$ 在设备 $s_j$ 上所需的加工时间	设备时间数组	$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	所选设备
$p_1$ :	[ 12 6 6 4 3 ]	+ [ 0 0 0 0 0 ]	= [ 12	6	6	4	(3)	] $\Rightarrow s_5$
$p_2$ :	[ 6 - 3 2 - ]	+ [ 0 0 0 0 3 ]	= [ 6	-	3	(2)	- ] $\Rightarrow s_4$	
$p_3$ :	[ - 9 - 6 4.5 ]	+ [ 0 0 0 2 3 ]	= [ -	9	-	8	(7.5)	] $\Rightarrow s_5$
$p_4$ :	[ 4 - 2 - 1 ]	+ [ 0 0 0 2 7.5 ]	= [ 4	-	(2)	-	8.5 ] $\Rightarrow s_3$	
$p_5$ :	[ - 9 9 6 - ]	+ [ 0 0 2 2 7.5 ]	= [ -	9	11	(8)	- ] $\Rightarrow s_4$	
$p_6$ :	[ 10 - 5 3.3 2.5 ]	+ [ 0 0 2 8 7.5 ]	= [ 10	-	(7)	11.3	10 ] $\Rightarrow s_3$	
$p_7$ :	[ 6 3 - 2 1.5 ]	+ [ 0 0 7 8 7.5 ]	= [ 6	(3)	-	10	9 ] $\Rightarrow s_2$	
$p_8$ :	[ 8 4 4 - 2 ]	+ [ 0 3 7 8 7.5 ]	= [ 8	(7)	11	-	9.5 ] $\Rightarrow s_2$	
		[ 0 7 7 8 7.5 ]						

图 3 算例 1 初始解获取过程图

流程中,设置一个设备时间数组来记录每个工位的累计加工时间。初始化状态时,该数组的每个元素值均为 0,第一个工件直接选择加工时间最短的设备,将该工件所选工位的加工时间加到数组中对应的元素上。若工件  $p_1$  选择了  $s_5$ ,则工位时间数组中的第 5 个元素就加 3,数组由 [0 0 0 0 0] 转变成 [0 0 0 0 3]。再以新的时间数组为基础,加上下一工件可选工位集中对应的加工时间,获得一个判断数组。算例 1 中,第二个工件的判断数组等于 [6 - 3 2 -] 加上 [0 0 0 0 3],得 [6 - 3 2 -],其中的每个时间值均为  $\omega_q + \omega_L/e_p$  对应的值,因此直接选择最小元素对应的工位,这里选择 2 对应的设备  $s_4$ 。以此类推,获得上述工位选择的染色体。

工序染色体的初始解根据设备时间数组中的时间变化点确定,若两个工件加工开始时间相同,则任取一个放在前面。因为同一工位同一时间只能加工一个工件,所以不影响整体加工排序。若两个工件起始加工时间相同,则工件必定位于两个不同的加工工位,即使排序标有先后,实际是两个工件在不同的工位上同时开始加工。

根据上述方法,随机安排工件选择工位的先后顺序,8 个工件有 40 320 种排法,可获得多个不同的可行解,组成遗传算法的初始种群。设群体个数  $N=10$ ,随机抽取 10 种工件选择工位顺序,获得 10 条染色体,组成算例 1 的初始解。计算过程见图 3,结果见表 3。

表 3 算例 1 初始解集

个体	工件加工随机排序	染色体
1	$p_1 - p_2 - p_3 - p_4 - p_5 - p_6 - p_7 - p_8$	5-4-5-3-4-3-2-2, 1-2-4-7-5-6-3-8
2	$p_2 - p_6 - p_5 - p_4 - p_7 - p_8 - p_3 - p_1$	3-4-5-3-4-5-2-5, 2-4-6-7-5-1-8-3
3	$p_1 - p_5 - p_7 - p_6 - p_4 - p_3 - p_8 - p_2$	5-3-2-1-4-3-5-5, 1-5-6-4-3-7-8-2
4	$p_7 - p_2 - p_6 - p_4 - p_5 - p_1 - p_8 - p_3$	2-4-5-3-4-5-5-3, 7-2-4-1-6-5-8-3
5	$p_5 - p_3 - p_6 - p_7 - p_4 - p_2 - p_8 - p_1$	2-3-5-1-4-3-2-5, 5-3-6-7-4-1-8-2
6	$p_4 - p_5 - p_7 - p_8 - p_2 - p_1 - p_3 - p_6$	5-3-5-5-4-3-5-2, 4-5-8-2-7-1-6-3
7	$p_5 - p_6 - p_8 - p_1 - p_3 - p_4 - p_2 - p_7$	5-3-5-3-4-5-1-2, 5-6-8-4-7-2-1-3
8	$p_6 - p_5 - p_3 - p_1 - p_2 - p_8 - p_7 - p_4$	2-3-5-5-4-5-1-3, 6-5-1-2-7-3-8-4
9	$p_3 - p_8 - p_2 - p_4 - p_5 - p_6 - p_1 - p_7$	2-4-5-3-4-5-1-2, 3-8-2-4-7-5-1-6
10	$p_8 - p_6 - p_5 - p_4 - p_2 - p_1 - p_7 - p_3$	5-3-5-3-2-4-4-5, 8-6-5-4-2-1-7-3

### 2.4 适应度函数

多目标遗传算法的适应度函数比单目标遗传算法的适应度函数复杂得多,各子目标之间存在一定的冲突,优化只能达到相对的整体最优,而无法同时令每个子目标都达到最优。本文采用传统的优先权值设定法,事先设置各子目标的优先权值,将整体目标按照权值合成一个标量效用函数,把多目标优化问题转化成单目标优化问题。根据前述 3 个目标的重要性,分别设  $\omega_1 = 0.4; \omega_2 = 0.3; \omega_3 = 0.3$ 。直接将目标函数设为适应度函

数,有

$$F = 0.4 \frac{\sum_{i=1}^n \min T_{iy}}{\sum_{i=1}^n T_i A_i} + 0.3 \frac{\sum_{i=1}^n T_i A_i}{T_s \sum_{y=1}^m (1 - \lambda_y)} + 0.3 \frac{\sum_{i=1}^n T_i A_i}{mC_{\max}} \quad (10)$$

对于上面所得的 1 号染色体 5-4-5-3-4-3-2-2, 1-2-4-7-5-6-3-8, 有

$$\sum_{i=1}^n \min T_{iy} = 3 + 2 + 4.5 + 1 + 6 + 2.5 + 1.5 + 2 = 22.5; \sum_{i=1}^n T_i A_i = 29.5; T_s \sum_{y=1}^m (1 - \lambda_y) = 50 \text{ (设 } T_s = 10, \text{ 设备故障率均为 } 0); mC_{\max} = 40; \text{ 适应度 } F = 0.703335. \text{ 其余 9 个染色体的适应度也以此类推, 结果见表 4.}$$

表 4 算例 1 轮盘赌法选择基表

初始解	工件加工随机排序	染色体	适应度	选择概率	累积概率
1	$p_1 - p_2 - p_3 - p_4 - p_5 - p_6 - p_7 - p_8$	5-4-5-3-4-3-2-2, 1-2-4-7-5-6-3-8	0.703335	0.103018	0.103018
2	$p_2 - p_6 - p_5 - p_4 - p_7 - p_8 - p_3 - p_1$	3-4-5-3-4-5-2-5, 2-4-6-7-5-1-8-3	0.676095	0.099029	0.202047
3	$p_1 - p_5 - p_7 - p_6 - p_4 - p_3 - p_8 - p_2$	5-3-2-1-4-3-5-5, 1-5-6-4-3-7-8-2	0.692990	0.101503	0.303550
4	$p_7 - p_2 - p_6 - p_4 - p_5 - p_1 - p_8 - p_3$	2-4-5-3-4-5-5-3, 7-2-4-1-6-5-8-3	0.687966	0.100767	0.404317
5	$p_5 - p_3 - p_6 - p_7 - p_4 - p_2 - p_8 - p_1$	2-3-5-1-4-3-2-5, 5-3-6-7-4-1-8-2	0.692990	0.101503	0.505820
6	$p_4 - p_5 - p_7 - p_8 - p_2 - p_1 - p_3 - p_6$	5-3-5-5-4-3-5-2, 4-5-8-2-7-1-6-3	0.657429	0.096294	0.602115
7	$p_5 - p_6 - p_8 - p_1 - p_3 - p_4 - p_2 - p_7$	5-3-5-3-4-5-1-2, 5-6-8-4-7-2-1-3	0.662323	0.097011	0.699126
8	$p_6 - p_5 - p_3 - p_1 - p_2 - p_8 - p_7 - p_4$	2-3-5-5-4-5-1-3, 6-5-1-2-7-3-8-4	0.718227	0.105200	0.804326
9	$p_3 - p_8 - p_2 - p_4 - p_5 - p_6 - p_1 - p_7$	2-4-5-3-4-5-1-2, 3-8-2-4-7-5-1-6	0.668727	0.097949	0.902275
10	$p_8 - p_6 - p_5 - p_4 - p_2 - p_1 - p_7 - p_3$	5-3-5-3-2-4-4-5, 8-6-5-4-2-1-7-3	0.667195	0.097725	1.000000

### 2.5 选择

群体中每个个体都有一个选择概率,它取决于群体中个体的适应度及其分布。本文采用按比例适应度分配法,选择方法采用轮盘赌选择法。算例 1 中的 10 个个体使用按比例适应度分配法获得的选择概率见表 4。

通过函数rand()产生 10 个随机数,将该随机数和计算获得的累积概率进行比较,得到被选中的个体,如表 5 所示。在第一次竞争迭代中,个体 6 和个体 9 被淘汰,取而代之的是适应度较高的个体(个体 1 和个体 8)。

表 5 轮盘赌法选择结果表

初始解	累积概率	随机数	选择个体
1	0.103018	0.194385	2
2	0.202047	0.801047	7
3	0.303550	0.502893	5
4	0.404317	0.008378	1
5	0.505820	0.702235	8
6	0.602115	0.924626	10
7	0.699126	0.753263	8
8	0.804326	0.262206	3
9	0.902275	0.318402	4
10	1.000000	0.095080	1

### 2.6 交叉

本文中,工位选择染色体采用双切点交叉法,对应部分两两交叉<sup>[9]</sup>;工序排序部分染色体的交

叉选用张超勇等<sup>[10]</sup>提出的 POX 交叉法,具体流程如下<sup>[8]</sup>:①随机划分工件集{1,2,...,n}为两个非空子集 J<sub>1</sub> 和 J<sub>2</sub>;②复制 Parent1 包含在 J<sub>1</sub> 的工件到 Children1,复制 Parent2 包含在 J<sub>1</sub> 的工件到 Children2,保留它们的位置;③复制 Parent2 包含在 J<sub>2</sub> 的工件到 Children1,Parent1 包含在 J<sub>2</sub> 的工件到 Children2,保留它们的顺序。以算例 1 中的 1 号和 3 号染色体为例,交叉示意如图 4 所示。

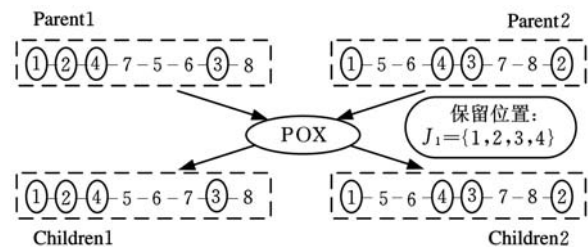


图 4 工序排序染色体 POX 交叉示意图

### 2.7 变异

变异是在种群中按照变异概率 P<sub>m</sub> 任选若干基因位改变其位值。本文的变异也分为两种情况:①在工位选择染色体的基因串中随机选择一个位置,在此工序的工位集中随机选择一个与先前所选位置上的数不相等的整数,替换当前基因,这样得到的解可以保证是可行解。②采用基因

位置互换的方法,即从工序排序染色体中随机选择两个位置的基因,将其位置进行互换,这样可以保证同一条工序排序染色体中不会出现相同的数字,即解可行。以算例 1 为例,1 号染色体的变异如图 5 所示。

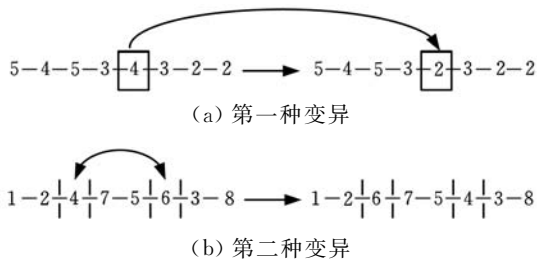


图 5 算例 1 中 1 号染色体变异示意图

第一种变异中,随机选择 5 号基因进行变异,由于第 5 道工序的可选择工位集为  $\{s_2, s_3, s_4\}$ , 现状基因为 4, 变异可选基因集合就是  $\{2, 3\}$ , 随机选择 2, 完成第一种基因变异, 见图 5a。第二种变异随机选择的两个基因是 3 号基因和 6 号基因, 两者相互交换位置, 完成第二种基因变异, 见图 5b。

2.8 遗传算法优化元胞机演化规则子程序设计

遗传算法优化演化规则在工位选择子程序和工件排序子程序的基础上进行, 其程序流程如图 6 所示, 其中,  $N_p$  为种群个体数,  $N_G$  为遗传算法迭代次数。基于元胞机和改进遗传算法的大型零件柔性作业车间调度算法求解通过 MATLAB 编

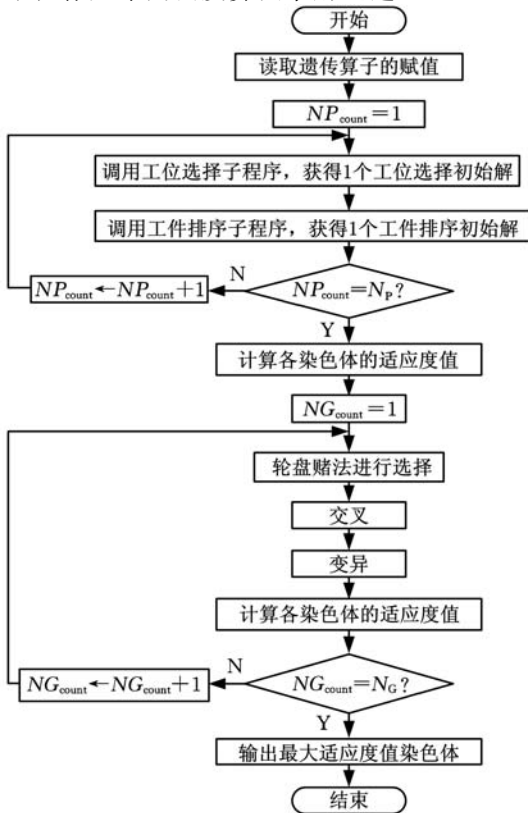


图 6 遗传算法优化元胞机演化规则子程序流程图

程实现。

3 实例验证

将上述模型应用于文献[1]中的案例, 建立元胞机模型并进行扩展, 以该车间调度瓶颈工位组——镗铣工位组的 9 台镗铣床为例, 对二季度 43 个工件共 89 道工序进行 4 次再调度。缓存元胞和工位元胞初始状态属性值分别如表 6、表 7 所示, 工件粒子的初始状态属性值如表 8 所示。仿真中, 实例元胞机模型缓存元胞初始状态属性值  $w_{co} = l_q = w_q = 0$ ; 工位元胞初始状态属性值  $s_s = s_{co} = 0$ ; 工件粒子初始状态属性值  $p_f = q_n = 0$ ,  $p_n = s_{ft} = 1, d_p = 500$ , 开始时间均为 7 月 1 日。

表 6 仿真实例元胞机模型缓存元胞初始状态属性值表

缓存元胞	状态属性		缓存元胞	状态属性		缓存元胞	状态属性	
	$w_{ct}$	$w_{cl}$		$w_{ct}$	$w_{cl}$		$w_{ct}$	$w_{cl}$
$C_{(1)1}$	5	5	$C_{(3)5}$	2	2	$C_{(5)5}$	2	2
$C_{(1)2}$	5	5	$C_{(3)6}$	2	2	$C_{(5)6}$	4	4
$C_{(1)3}$	5	5	$C_{(3)7}$	2	2	$C_{(5)7}$	2	2
$C_{(1)4}$	5	5	$C_{(3)8}$	2	2	$C_{(5)8}$	2	2
$C_{(1)5}$	5	5	$C_{(3)9}$	4	4	$C_{(7)1}$	2	2
$C_{(3)1}$	2	2	$C_{(5)1}$	2	2	$C_{(7)2}$	2	2
$C_{(3)2}$	2	2	$C_{(5)2}$	2	2	$C_{(7)3}$	2	2
$C_{(3)3}$	2	2	$C_{(5)3}$	6	6	$C_{(7)4}$	2	2
$C_{(3)4}$	2	2	$C_{(5)4}$	4	4	$C_{(7)5}$	3	3
						$C_{(11)6}$	4	3
						$C_{(7)7}$	4	4
						$C_{(9)1}$	2	2
						$C_{(9)2}$	2	2
						$C_{(11)1}$	7	7
						$C_{(11)2}$	3	3
						$C_{(11)3}$	3	3
						$C_{(11)4}$	3	3
						$C_{(11)5}$	5	5

表 7 仿真实例元胞机模型工位元胞初始状态属性值表

工位元胞	状态属性				
	$s_t$	$e_p$	$T(d)$	$s_{ct}$	$s_{cl}$
$C_{(2)1}$	$C_2$	4.00	27	108.00	108.00
$C_{(2)2}$	$C_2$	4.00	27	108.00	108.00
$C_{(2)3}$	$C_2$	4.00	27	108.00	108.00
$C_{(2)4}$	$C_2$	4.00	27	108.00	108.00
$C_{(2)5}$	$C_2$	4.00	27	108.00	108.00
$C_{(4)1}$	$C_4$	0.24	25.65	6.16	6.16
$C_{(4)2}$	$C_4$	0.24	25.65	6.16	6.16
$C_{(4)3}$	$C_4$	0.24	25.65	6.16	6.16
$C_{(4)4}$	$C_4$	0.24	25.65	6.16	6.16
$C_{(4)5}$	$C_4$	0.30	24.98	7.49	7.49
$C_{(4)6}$	$C_4$	0.30	25.65	7.70	7.70
$C_{(4)7}$	$C_4$	0.34	24.98	8.56	8.56
$C_{(4)8}$	$C_4$	0.34	27	9.26	9.26
$C_{(4)9}$	$C_4$	0.34	25.65	8.79	8.79
$C_{(6)1}$	$C_6$	2.00	27	54.00	54.00
$C_{(6)2}$	$C_6$	2.00	27	54.00	54.00
$C_{(6)3}$	$C_6$	2.00	27	54.00	54.00
$C_{(6)4}$	$C_6$	2.00	27	54.00	54.00
$C_{(6)5}$	$C_6$	2.00	27	54.00	54.00
$C_{(6)6}$	$C_6$	2.00	27	54.00	54.00
$C_{(6)7}$	$C_6$	2.00	27	54.00	54.00
$C_{(6)8}$	$C_6$	2.00	27	54.00	54.00
$C_{(8)1}$	$C_8$	0.32	24.98	7.99	7.99
$C_{(8)2}$	$C_8$	0.32	25.65	8.21	8.21
$C_{(8)3}$	$C_8$	0.40	25.65	10.26	10.26
$C_{(8)4}$	$C_8$	0.40	27	10.80	10.80
$C_{(8)5}$	$C_8$	0.48	25.65	12.31	12.31
$C_{(8)6}$	$C_8$	0.60	25.65	15.39	15.39
$C_{(8)7}$	$C_8$	0.69	27	18.51	18.51
$C_{(10)1}$	$C_{10}$	2.00	27	54.00	54.00
$C_{(10)2}$	$C_{10}$	2.00	27	54.00	54.00
$C_{(12)1}$	$C_{12}$	0.60	27	16.20	16.20
$C_{(12)2}$	$C_{12}$	0.60	27	16.20	16.20
$C_{(12)3}$	$C_{12}$	0.60	27	16.20	16.20
$C_{(12)4}$	$C_{12}$	0.60	27	16.20	16.20
$C_{(12)5}$	$C_{12}$	0.60	27	16.20	16.20

表 8 仿真实例工件粒子初始状态属性值表

工件 粒子	状态属性			工件 粒子	状态属性			工件 粒子	状态属性		
	$p_t$	$\omega_L$	$s_n$		$p_t$	$\omega_L$	$s_n$		$p_t$	$\omega_L$	$s_n$
$p_1$	11	2	1	$p_{16}$	20	1	1	$p_{31}$	10	1	1.5
$p_2$	11	2	1	$p_{17}$	20	1	1	$p_{32}$	10	1	1.5
$p_3$	18	1	1	$p_{18}$	23	1	1	$p_{33}$	10	1	1.5
$p_4$	19	1	1	$p_{19}$	23	1	1	$p_{34}$	10	1	1.5
$p_5$	19	1	1	$p_{20}$	20	1	1	$p_{35}$	14	2	2
$p_6$	19	1	1	$p_{21}$	20	1	1	$p_{36}$	14	2	2
$p_7$	19	1	1	$p_{22}$	20	1	1	$p_{37}$	12	2	2
$p_8$	18	1	1	$p_{23}$	27	1	1	$p_{38}$	12	2	2
$p_9$	18	1	1	$p_{24}$	27	1	1	$p_{39}$	12	2	2
$p_{10}$	19	1	1	$p_{25}$	18	1	1	$p_{40}$	12	2	2
$p_{11}$	24	1.5	1	$p_{26}$	18	1	1	$p_{41}$	22	2	1
$p_{12}$	24	1.5	1	$p_{27}$	16	1	1	$p_{42}$	22	2	1
$p_{13}$	18	1	1	$p_{28}$	21	1	1	$p_{43}$	22	2	1
$p_{14}$	18	1	1	$p_{29}$	21	1	1				
$p_{15}$	16	1	1	$p_{30}$	21	1	1				

各静态调度单元中设权重  $\omega_1 = 0.4, \omega_2 = \omega_3 = 0.3$ 。每次调度获取初始解 20 个,并将其作为遗传算法优化的初始种群。遗传算法优化部分迭代次数  $N = 100$ ,交叉概率  $P_c = 0.6$ ,变异概率  $P_m = 0.001$ ,将调度总目标作为适应度函数,直接调用 MATLAB 所编程序,获得优化解集,将其中适应度值最大的作为最终的优化方案。图 7 和图 8 分别为改进遗传算法优化前后元胞机模型获得的调度方案甘特图。对应的三个子目标和一个总目标值如表 9 所示。图 7、图 8 和表 9 显示,将改进遗传算法嵌入元胞机模型,优化局部演化规则后获得的调度方案比优化之前更优,因此本论文所设计的元胞机和改进遗传算法的混合算法可行、有效。

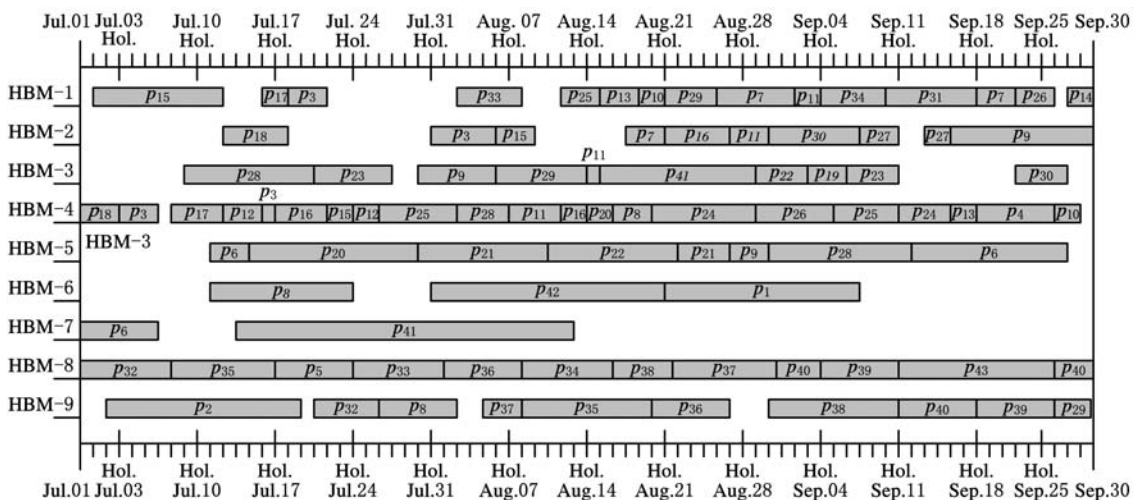


图 7 优化前调度方案甘特图

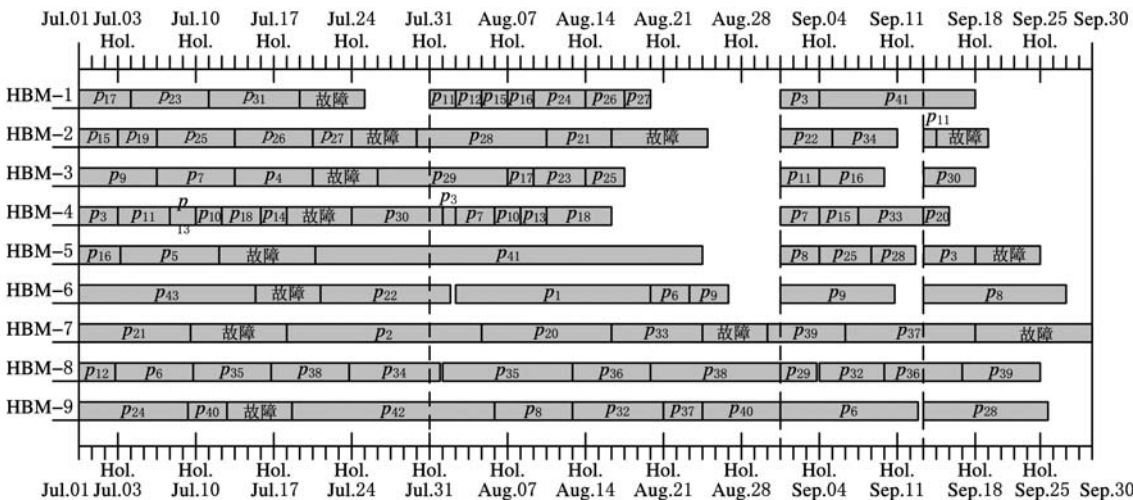


图 8 优化后调度方案甘特图

表 9 遗传算法优化前后目标函数值

目标函数值	$F_1$	$F_2$	$F_3$	$F$
优化前	0.839 243 5	0.843 974 5	0.743 670 9	0.811 991 0
优化后	0.867 546 4	0.816 440 5	0.773 242 6	0.823 923 5

4 结语

本文在大型零件柔性作业车间调度元胞机模  
(下转第 190 页)

- [6] 廖貅武,唐焕文. 动态联盟中伙伴选择的证据推理方法[J]. 计算机集成制造系统—CIMS, 2003, 9(1): 57-61.
- [7] Herrera F, Martinez L. A 2-tuple Fuzzy Linguistic Representation Model for Computing with Words [J]. IEEE Transactions on Fuzzy Systems, 2000, 8(12): 746-752.
- [8] 鲍广宇,付丰科,赵志敏. 一种基于二元语义信息处理的群体决策方法[J]. 解放军理工大学学报, 2009, 10(5): 435-439.
- [9] 姜艳萍,樊治平. 具有语言信息的多指标群体综合评价[J]. 东北大学学报(自然科学版), 2005, 26(7): 703-706.
- [10] 张园林,匡兴华. 一种基于多粒度语言偏好矩阵的多属性群决策方法[J]. 控制与决策, 2008, 23(11): 1296-1300.
- [11] Chen Z F, Berrarieh D. On the Fusion of Multi-granularity Linguistic Label Sets in Group Decision Making[J]. Computers & Industrial Engineering, 2006, 51(10): 526-541.
- [12] Herrera F, Matinez L. A Model Based on Linguistic 2-tuples for Dealing with Multi-granularity Linguistic Contexts in Multi-expert Decision Making [J]. IEEE Transaction on Systems, Man and Cybernetics, Part B: Cybernetics, 2001, 31(2): 227-234.
- [13] 卫贵武,黄登仕,魏宇. 基于 ET-WG 和 ET-OWG 算子的二元语义群决策方法[J]. 系统工程学报, 2009, 24(6): 744-748.
- [14] 徐泽水,达庆利. 一种组合加权几何平均算子及其应用[J]. 东南大学学报, 2002, 32(3): 506-509.
- [15] Xu Z S. An Overview of Methods for Determining OWA Weights[J]. International Journal of Intelligent Systems, 2005, 20(8): 843-865.
- [16] 叶飞,孙东川,张红. 面向虚拟企业合作伙伴的新过程框架研究[J]. 系统工程理论与实践, 2003(11): 88-94.
- [17] 陈菊红,汪应洛,孙林岩. 虚拟企业伙伴选择过程及方法研究[J]. 系统工程理论与实践, 2001, 21(7): 48-54.
- [18] 钱碧波,潘晓弘,程耀东. 敏捷虚拟企业合作伙伴选择评价体系研究[J]. 中国机械工程, 2000, 11(4): 397-401.

(编辑 张 洋)

作者简介:彭安华,男,1973年生。中国矿业大学机电工程学院博士研究生,淮海工学院副教授。主要研究方向为模糊多准则决策及先进制造技术。发表论文 40 余篇。肖兴明,男,1962年生。中国矿业大学机电工程学院教授、博士研究生导师。

(上接第 184 页)

型的基础上嵌入改进遗传算法,优化了元胞机模型的局部演化规则,设计了基于元胞机和改进遗传算法的混合算法,建立了经元胞机模型离散化后的每个静态柔性作业车间调度单元的改进遗传算法优化模型。调度算法的求解由 MATLAB 编程实现。本文将算法应用于文献[1]中的扩展案例,应用该算法的调度方案较原方案更优,从而验证了元胞机和改进遗传算法相结合的混合算法求解大型零件柔性作业车间调度问题的可行性和有效性。

#### 参考文献:

- [1] 陈勇,阮幸聪,鲁建厦. 基于元胞机的大型零件生产车间动态柔性调度仿真建模[J]. 中国机械工程, 2010, 21(21): 2603-2609.
- [2] 廖伟志,古天龙,李文敬,等. 模糊柔性制造系统的混杂 Petri 网建模与调度[J]. 计算机集成制造系统, 2008, 14(11): 2134-2141.
- [3] 孙小明. 生产系统建模与仿真[M]. 上海:上海交通大学出版社, 2006.
- [4] 陈勇,吴国献,林飞龙. 多品种多工艺车间作业调度的 Multi-Agent 建模[J]. 浙江大学学报(工学版), 2009, 43(9): 1672-1678.
- [5] 张超勇,李新宇,王晓娟,等. 基于滚动窗口的多目标动态调度优化研究[J]. 中国机械工程, 2009, 20(18): 2190-2197.
- [6] 闻凯. 元胞自动机的进化与计算研究[D]. 南京:南京航空航天大学, 2008.
- [7] 张国辉,高亮,李培根,等. 改进遗传算法求解柔性作业车间调度问题[J]. 机械工程学报, 2009, 45(7): 145-151.
- [8] Witkowski T, Antczak A, Elzway S, et al. Evolving Cellular Automata-based Flexible Job Shop Scheduling[C]//5th International Conference on Natural Computation. Tianjin, 2009: 8-13.
- [9] 张伟存,郑丕涛,吴晓丹. 基于主-从遗传算法求解柔性调度问题[J]. 计算机集成制造系统, 2006, 12(8): 1241-1245.
- [10] 张超勇,饶运清,刘向军,等. 基于 POX 交叉的遗传算法求解 Job-Shop 调度问题[J]. 中国机械工程, 2004, 15(23): 2149-2153. (编辑 张 洋)

作者简介:陈勇,男,1973年生。浙江工业大学机械工程学院工业工程研究所教授。主要研究方向为物流系统的分析与优化、工业工程。发表论文 30 余篇。阮幸聪,女,1985年生。浙江工业大学机械工程学院工业工程研究所硕士研究生。鲁建厦,男,1963年生。浙江工业大学机械工程学院工业工程研究所教授。王亚良,男,1977年生。浙江工业大学机械工程学院工业工程研究所讲师。