

文章编号:1000-6893(2007)02-0464-06

基于免疫算法的柔性制造单元动态调度研究

余建军, 孙树栋, 王军强, 吴秀丽

(西北工业大学 机电学院, 陕西 西安 710072)

Flexible Manufacturing Cell Dynamic Scheduling by Immune Algorithm

YU Jian-jun, SUN Shu-dong, WANG Jun-qiang, WU Xiu-li

(School of Mechatronic Engineering, Northwestern Polytechnical University, Xi an 710072, China)

摘 要: 为了将调度理论实用化, 研究接近生产实际的柔性制造单元动态调度, 既解决路径选择问题, 又优化工序调度方案。不仅考虑了零件分批、工艺路线柔性可变、不相容指标协同优化等问题, 而且探讨了由于设备故障、加工任务临时变动等扰动因素引起的动态调度。建立了柔性制造单元动态调度的数学模型; 针对这个组合优化问题的难处理性, 提出了一种免疫算法; 引入了滚动-扰动混合再调度策略。采用提出的方法求解西安航空发动机(集团)有限公司的柔性制造单元动态调度问题, 得到满足实际生产要求的优化调度方案。实例仿真结果表明柔性制造单元动态调度问题采用提出的算法、策略和方法基本能够得到解决。

关键词: 柔性制造单元动态调度; 免疫算法; 疫苗; 滚动-扰动混合再调度策略

中图分类号: TP278; TP301.6 **文献标识码:** A

Abstract: The flexible manufacturing cell dynamic scheduling (FMCDS) is studied in order to apply the scheduling theory to the production practice. Not only the process route selection problem is solved, but also the operation scheduling is optimized. The work piece batching, the flexible process route and the contradiction objectives optimizations are taken into account. The dynamic scheduling inspired by the disturbance factors such as the machine breakdown and the process task change is discussed. The FMCDS model is firstly set up. In view of the stubborn nature of the FMCDS, an improved immune algorithm is brought forward and the rolling-disturb hybrid rescheduling policy is introduced to treat with the dynamic scheduling. Finally the above method is applied to solve the FMCDS of Xi an Aero-engine (Group) Ltd. in China, and the optimization scheduling solution obtained can meet the production practice expectation. The simulation results indicate that the FMCDS can be solved by the proposed algorithm, policy and method.

Key words: flexible manufacturing cell dynamic scheduling; immune algorithm; vaccine; rolling-disturb hybrid rescheduling policy

制造单元调度是实现先进制造和提高生产效益的基础和关键, 成为机械工程、运筹学等学科研究热点之一, 多种调度方法在文献[1]有详细分析。为了降低处理难度, 很多调度研究都进行简化处理。但是更接近实际情况的调度一般具有柔性, 近年来, 有学者对柔性 Job-Shop 调度问题做了一些研究, Kacem 等^[2]提出一种以局部最小化方法为分配模型的遗传算法求解柔性 Job-shop 调度问题; 夏蔚军等^[3]提出一种基于微粒群优化和模拟退火的分解方法, 先后处理柔性 Job-shop 调度中的路径子问题和调度子问题。然而, 实际生产中的调度不仅有路径柔性的需求, 而且随着各种扰动因素的出现需要进行动态调度。因此,

研究柔性制造单元动态调度 (FMCDS) 具有更重要的理论意义和工程价值^[4]。

受生物免疫系统启发而提出的人工免疫系统 (AIS) 给包括调度问题在内的众多工程问题开辟了新的求解方法。Alisanto 等^[5]采用免疫算法解决柔性 PCB Flow-shop 问题; 余建军等^[6]研究了用免疫算法求解标准调度问题, 对 FT10 × 10 问题进行了仿真。上述文献基本限于研究经过理想化处理的调度问题, 没有涉及更复杂的柔性制造单元动态调度。

本文将免疫思想引入柔性制造单元动态调度, 提出了一种免疫算法, 对柔性制造单元动态调度问题进行了求解。

1 柔性制造单元动态调度模型

在柔性制造单元中, 各工序可以在多台可选

收稿日期: 2005-12-07; 修订日期: 2006-07-18
基金项目: 863 计划 (2003AA411110, 2002AA414060); 教育部博士
点基金 (20040699025)
通讯作者: 张树栋 E-mail: sdsun@nwpu.edu.cn

的设备上加工,中途可以随时随地增加或取消加工任务,设备也可能突然出现故障。本文研究的柔性制造单元动态调度可以描述为

制造单元有 H 台设备 M_1, M_2, \dots, M_H , 可以加工 N 个零件类型 L_1, L_2, \dots, L_N , 每个零件类型分别有 N_1, N_2, \dots, N_N 个批次, 并且分别由 J_1, J_2, \dots, J_N 个有工艺约束的工序组成。现假设这 N 个零件类型分别有批次和批量为 $(L_{i,j}, P_{i,j}), \dots, (L_{i,N}, P_{i,N})$, 其中 $L_{i,j}$ 表示第 i 个零件类型的第 j 个批次, $P_{i,j}$ 是 $L_{i,j}$ 的零件数。

目标函数包括时间、成本和设备利用率等,可以具体由以下两个子目标函数体现:

对工件提前/拖期完工的惩罚最小

$$F_1 = \sum_{i=1}^N \sum_{j=1}^{N_i} [b_{i,j} \max(0, D_{i,j} - T_{f,i,j}) + a_{i,j} \max(0, T_{f,i,j} - D_{i,j})] \quad (1)$$

使所有设备的利用率最高

$$F_2 = \frac{\sum_{m=1}^H \sum_{i=1}^N \sum_{j=1}^{N_i} \sum_{k=1}^{J_i} T_{i,j,k} z_{m,i,j,k} C_m}{\sum_{m=1}^H (T_{f,m} - T_{s,m}) C_m} \quad (2)$$

上述两个子目标一个需要极小化优化,另一个需要极大化优化,不采用简单的加权方式处理多目标,而采用带权重的乘商方式构造总目标函数,其中 w_1 和 w_2 是子目标的权重系数,借鉴文献 [7] 的方法来确定 w_1 和 w_2 ,

$$F = \min \left(w_1 F_1 w_2 \frac{1}{F_2} \right) \quad (3)$$

约束条件:

资源约束 同一设备上一个批操作结束后才能进行下一个批操作

$$\begin{cases} T_{s,i,j,k} & T_{f,p,q,r} \\ z_{m,i,j,k} = z_{m,p,q,r} = 1 \end{cases} \quad \text{或} \quad \begin{cases} T_{s,p,q,r} & T_{f,i,j,k} \\ z_{m,i,j,k} = z_{m,p,q,r} = 1 \end{cases} \quad (4)$$

工艺约束 同一批号的前一个批操作结束后才能进行下一个批操作

$$T_{s,i,j,k+1} \geq T_{f,i,j,k} \quad (5)$$

式中:

$$z_{m,i,j,k} = \begin{cases} 1 & O_{i,j,k} \\ & \text{在设备 } m \text{ 上加工} \\ 0 & \text{其他} \end{cases} \quad (6)$$

$$T_{s,m} = \min(T_{s,i,j,k}), \quad z_{m,i,j,k} = 1 \quad (7)$$

$$T_{f,m} = \max(T_{f,i,j,k}), \quad z_{m,i,j,k} = 1 \quad (8)$$

式中: $O_{i,j,k}$ 为第 i 类零件第 j 个批次的第 k 个批操作; $T_{s,i,j,k}$ 为批操作 $O_{i,j,k}$ 的开始时间; $T_{f,i,j,k}$ 为批

操作 $O_{i,j,k}$ 的完成时间; $T_{i,j,k}$ 为批操作 $O_{i,j,k}$ 的加工时间; $T_{f,i,j}$ 为第 i 类零件第 j 个批次的完成时间; $D_{i,j}$ 为第 i 类零件第 j 个批次的交货期; H 为设备总数; N 为零件类型数; N_i 为第 i 类零件的批次数; J_i 为第 i 类零件每批次的操作数; $a_{i,j}$ 为第 i 类零件第 j 个批次拖期完成惩罚系数; $b_{i,j}$ 为第 i 类零件第 j 个批次提前完成惩罚系数; C_m 为第 m 台设备的关键性系数。

2 免疫算法

免疫算法是受到生物免疫系统启发提出的新型智能算法,把要解决的问题当作抗原,相应的解当作抗体。为了避免算法陷入局部优化导致“早熟”,同时促进算法向着好的方向快速进化,通过临时解集对抗体动态评价;采用疫苗抽取和接种机制,使搜索更具有针对性。免疫算法流程如图 1 所示,算法具体步骤如下:

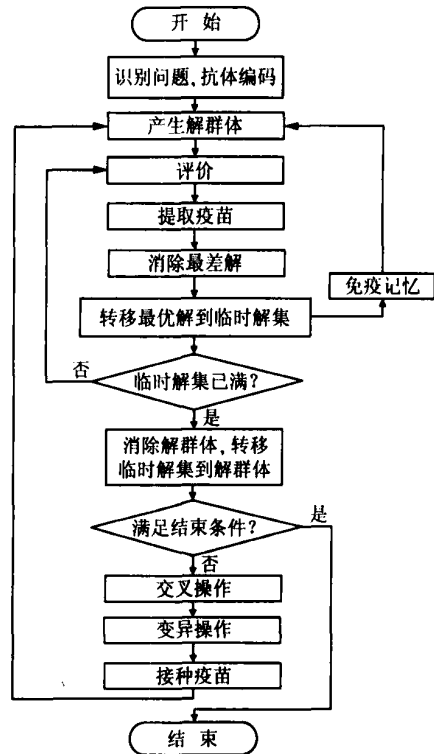


图 1 免疫算法流程

Fig. 1 Immune algorithm flow

步骤 1 识别抗原(问题), 抗体编码

柔性调度比较复杂,不能采用一般的抗体编码。为了编码和解码简便,每个基因由 4 部分组成,依次代表零件类型、该零件类型的一个批次号、该批次的工序号、该工序所占用的设备号。

步骤 2 产生解群体

如果是首次迭代,直接在解空间中随机产生

初始抗体群体;否则,对已有抗体群体通过免疫算子操作产生一些新抗体,再从记忆库中提取 N_m 个较优抗体构成新一代抗体群体。

步骤3 对抗体进行评价

采用抗体浓度来抑制规模比较大又不是最优解的抗体。计算浓度时首先需要对抗体之间的相似性进行度量,而目标函数值无法度量抗体之间的相似性。信息熵通过数学概率统计所有对象上各个位置出现各个符号的几率,非常适合于采用与数值大小无关的符号编码的对象之间相似性的度量,而且以严格的数学理论为基础,并且计算比较简单,所以采用信息熵作为评价抗体相似性的指标。

步骤4 提取疫苗

充分利用进化过程历史信息,依据数理统计原理对进化过程个体等位基因进行统计,把出现概率比较高的基因作为当前的疫苗,因为这些频繁出现的基因很可能就是最优抗体上的基因。

步骤5 转移最优解,消除最差解

为了防止浓度较高的较优解一次性全部被消除所导致的后期收敛速度下降,算法引入一个新的群体——临时解集,其容量为 N_l 。将解群体按期望繁殖率 e_v 降序排列,然后消除 e_v 值最小的若干个抗体,同时把 e_v 值最大的若干个抗体作为暂时最优解转移到临时解集中,分多次动态促进和抑制抗体群体。

步骤6 判断临时解集是否已满

如果临时解集已满,则进行下一步操作。否则,临时解集还没有达到 N_l 个,则转到步骤3,对剩余的解群体重新评价,体现进化动态性。

步骤7 清除解群体,转移临时解集到解群体

抛弃剩余解群体,而将临时解集更新到解群体。同时从临时解集取前 N_m 个较优抗体存入记忆库中。最后,清除临时解集。

步骤8 判断是否满足结束条件

若满足,则结束;否则,进行下一步操作。

步骤9 交叉操作和变异操作

由于动态调度逻辑关系的复杂及抗体的特殊表示,在该算法中交叉和变异方法与一般方法有所不同。本文的交叉操作只改变批操作的先后顺序,而不改变批操作的设备安排;变异操作用来改变批操作的设备分配,当然还要确保新抗体是可行的,能满足工艺约束条件。

步骤10 接种疫苗

从种群中选择要进行接种的个体,按照动态

计算的接种率将疫苗的基因片段依次接入,即用疫苗基因置换抗体相应基因位上原有的基因。并且对接种后的抗体进行检验,查看接种是否变优,确定是否重新接种。

从搜索空间来分析,由于疫苗上的基因代表着当前比较优良的基因,抗体经过接种这些优良疫苗之后,迅速有针对性地朝着最优抗体收敛,可见,接种疫苗不仅提高了算法的全局寻优能力,而且强化了算法的局部搜索能力。

3 柔性动态调度策略

由于在柔性动态调度中,可能中途插入或取消加工任务,也可能发生设备故障等突发事件,这就需要引入相应的柔性动态调度策略^[8]。

3.1 滑动窗口技术

因为在柔性动态不确定环境下,各种扰动因素不可预知,优化需要反复进行,考虑远时段内的优化没有意义。滑动窗口技术是根据预测控制中滚动优化原理而提出,是有限时段的滚动优化。在每个采样时刻性能指标只涉及到从该时刻起的有限时间,而到下一个采样时刻,这一优化时刻相应地向前推移,如此反复滚动进行。

3.2 调度任务池

采用滑动窗口技术进行柔性动态调度优化,需要设置一个调度任务池。调度任务池划分为3个前后衔接的小池:待调度任务池、窗口任务池和已完工任务池。待调度任务池里存放着已经下达但还未调度的工件;窗口任务池存放着需要进行优化调度的工件;已完工任务池里存放着已经加工完成的工件。根据工件优先级从待调度任务池中选择工件进入窗口任务池,选用松弛度 L 作为工件优先级,计算公式为

$$L = \frac{D - S}{T} \quad (9)$$

式中: D 为工件的交货期; S 为调度时刻; T 为工件有待加工的总加工时间。

用所提出的免疫算法对窗口任务池的工件进行优化调度,然后下达生产线进行加工,加工完成后就转入已完工任务池。工件在调度任务池中的流动过程如图2所示。

3.3 滚动-扰动混合再调度

滚动再调度是每隔一段时间周期性地进一

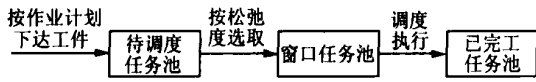


图 2 工件在调度任务池中的流动示意图

Fig. 2 The flow of the work pieces in task pool

次调度;扰动再调度是系统状态一旦出现扰动发生变化就立即进行一次调度。由于系统的灵活性和动态性,本文同时考虑以上两种再调度,采用滚动-扰动混合再调度策略。

当一个调度时刻到来时,窗口任务池里的工件不一定全部结束,这时从待调度任务池里按照优先级选择一定数量工件进入窗口任务池中,与窗口任务池里原先剩余的工件或正在加工工件的剩余工序一起进行下一轮调度优化。如果调度系统一直没有出现扰动,则有周期地滚动再调度。但是如果突然出现扰动,比如设备出现故障,为了保证生产能够继续进行,正在该设备上加工的工件可以安排到其它设备上继续加工,这样也必然会影响到窗口任务池中其他工件对设备的选取,所以需要立即进行一次再调度。

4 免疫算法求解柔性动态调度实例

以西安航空发动机(集团)有限公司的柔性制造单元动态调度为例进行分析,本制造单元共有 15 台加工设备,加工 10 类典型零件,每类零件又包括若干个批次。设备号与设备关键性系数 c_m 如表 1 所示。零件号、批次、拖期完成惩罚系数 a_{ij} 、提前完成惩罚系数 b_{ij} 等零件信息如表 2 所示。工艺信息如表 3 所示。

表 1 本制造单元的设备信息

Table 1 Machine information in the manufacturing cell

| 设备类别 | 设备号 | c_m |
|--------------------|-----|-------|
| 车削加工柔性单元 MOC200MS3 | 1 | 3 |
| 自动液压车床 ZZC100 | 2 | 2 |
| 数控加工中心 UCP1350 | 3 | 2 |
| 数控加工中心 UCP1350 | 4 | 2 |
| 数控加工中心 UCP1350 | 5 | 2 |
| 立卧两用型加工中心 XH7910T2 | 6 | 3 |
| 重型龙门刨床 B2032 *8000 | 7 | 3 |
| 钻铣镗磨床 ZXTM-40 | 8 | 1 |
| 钻铣镗磨床 ZXTM-40 | 9 | 1 |
| 钻铣镗磨床 ZXTM-40 | 10 | 1 |
| 柔性加工单元 TR6340 | 11 | 1 |
| 柔性加工单元 TR6340 | 12 | 1 |
| 柔性制造单元 QH1-FMC001 | 13 | 1 |
| 柔性制造单元 QH1-FMC001 | 14 | 2 |
| 柔性制造单元 QH1-FMC001 | 15 | 1 |

表 2 本制造单元的零件编号

Table 2 Work piece information in the manufacturing cell

| 零件类型 | 批次数 | a_{ij} | b_{ij} | 零件序号 |
|------|-----|----------|----------|----------|
| 1 | 2 | 2 | 2 | 1,2 |
| 2 | 2 | 1 | 1 | 3,4 |
| 3 | 4 | 2 | 1 | 5,6,7,8 |
| 4 | 3 | 1 | 1 | 9,10,11 |
| 5 | 1 | 1 | 1 | 12 |
| 6 | 1 | 3 | 1 | 13 |
| 7 | 3 | 2 | 1 | 14,15,16 |
| 8 | 1 | 1 | 2 | 17 |
| 9 | 2 | 3 | 1 | 18,19 |
| 10 | 1 | 1 | 3 | 20 |

表 3 本制造单元的工艺信息

Table 3 Technology information in the manufacturing cell

| 零件类型 | 交货期 | 工序号 | 可选设备 | 加工时间/h |
|------|-----|-----|-------------|--------|
| 1 | 38 | 1 | 1,2,11,12 | 3 |
| | | 2 | 3,4,5 | 8 |
| | | 3 | 6,8,9,10 | 4 |
| | | 4 | 7,13,4 | 6 |
| | | 5 | 11,12 | 2 |
| | | 6 | 15 | 5 |
| | | 7 | 11,12,13,14 | 7 |
| 2 | 28 | 1 | 1,2,11,12 | 2 |
| | | 2 | 8,9,10 | 3 |
| | | 3 | 11,12 | 4 |
| | | 4 | 7,13,14 | 7 |
| 3 | 26 | 1 | 3,4,5 | 5 |
| | | 2 | 7,11,12 | 4 |
| | | 3 | 13,14,15 | 2 |
| 4 | 18 | 1 | 3,4,5 | 9 |
| | | 2 | 6,8,9,10 | 5 |
| 5 | 14 | 1 | 1,2 | 6 |
| | | 2 | 13,14,15 | 2 |
| | | 3 | 7,10 | 5 |
| | | 1 | 8,9,10 | 2 |
| | | 2 | 6 | 6 |
| 6 | 20 | 3 | 11,12,15 | 3 |
| | | 4 | 3,4,5 | 1 |
| | | 5 | 10 | 4 |
| | | 1 | 2 | 5 |
| | | 2 | 1,3,4,5 | 6 |
| 8 | 28 | 1 | 1 | 4 |
| | | 2 | 11,12 | 1 |
| | | 3 | 15 | 6 |
| | | 4 | 13,14 | 2 |
| | | 5 | 3,4,5 | 5 |
| 9 | 17 | 1 | 3,4,5 | 3 |
| | | 2 | 2 | 5 |
| | | 3 | 11,12 | 2 |
| 10 | 25 | 1 | 13,14 | 4 |
| | | 2 | 2 | 2 |
| | | 3 | 11,12 | 6 |
| | | 4 | 15 | 3 |
| | | 5 | 3,4,5 | 3 |
| | | 6 | 8,9,10 | 8 |

运用免疫算法对该制造单元的 20 个零件批次进行了调度,共运行 10 次,每次调度时进化 250 代。在优化总目标函数之前首先需要确定两个子目标函数的权重系数,通过综合分析之后确定 w_1 为 1.8, w_2 为 1。最终计算结果如表 4 所示。

表 4 计算结果

Table 4 Calculated results

| 目标函数最优值 | 目标函数平均值 | 收敛到最优值的次数 | 平均时间/s |
|---------|---------|-----------|--------|
| 129 | 143.6 | 6 | 387 |

其中,算法的一次进化过程如图 3 所示,最优值是本次运行过程每代总目标函数的最小值,平均值是本次运行过程每代种群所有个体总目标函数的平均值,从进化曲线可以看出,本算法求解该类调

度问题的收敛性好,进化效率高。首次优化调度解的甘特图如图 4 所示,图中括号中的 3 个数依次表示零件号、工序号和开始加工时间。

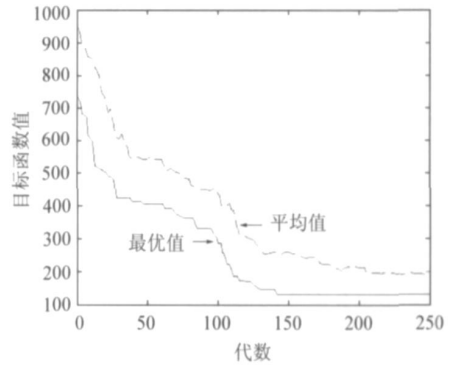


图 3 算法进化曲线

Fig. 3 Evolution curves of the algorithm

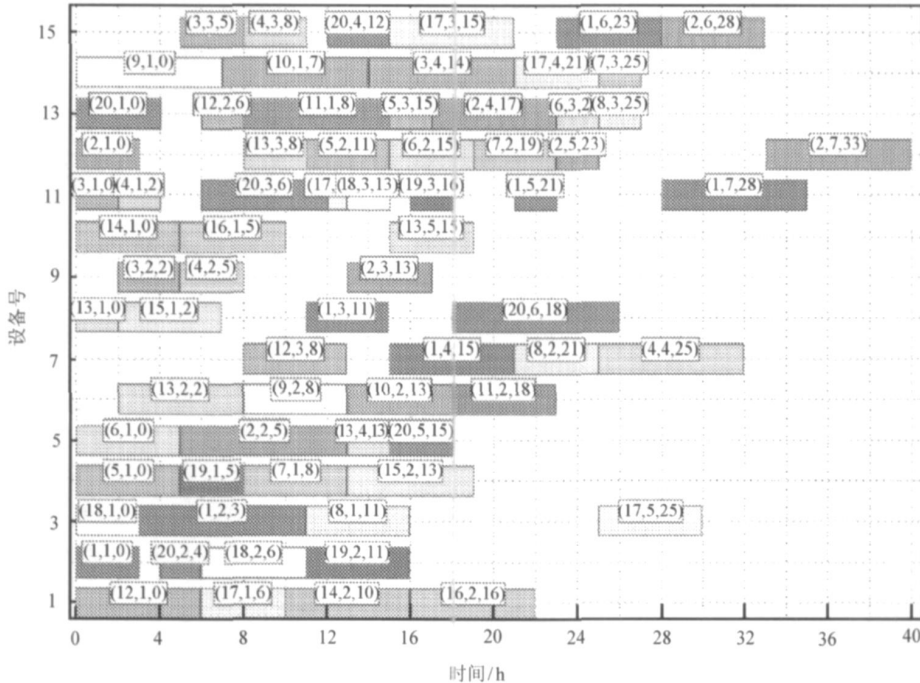


图 4 调度甘特图

Fig. 4 Scheduling gantt chart

由于在柔性动态制造单元的生产过程,可能会不断出现各种扰动,所以需要实行动态调度。这里考虑 3 种类型的扰动,在时刻 16 由于某种原因零件号 1 被取消,这时需要再调度;在时刻 18 突然增加零件类型 10 的一个新批次,零件号定为 21,这时也需要再调度;在时刻 20 设备 13 突然出现故障,这时同样需要再调度。经过多次再调度后,整个动态调度的甘特图如图 5 所示。从甘特图可以看出,在时刻 16 零件号 1 正在设备 7 上加工第 4 道工序,突然被取消,之后的未加工工序就

再没有出现在图中;在时刻 18 增加了零件号 21,但它的第 1 道工序可选的设备 13 和 14 正在加工其他批次零件,所以只能等到时刻 21 才安排到设备 14 上开始加工;在时刻 20 设备 13 突然出现故障,此时零件号 2 的第 4 道工序正在其上加工,只能转移到设备 7 上继续加工。

5 结束语

研究柔性制造单元动态调度,解决具有柔性工艺路线的工序调度难题;协同优化多目标,考虑

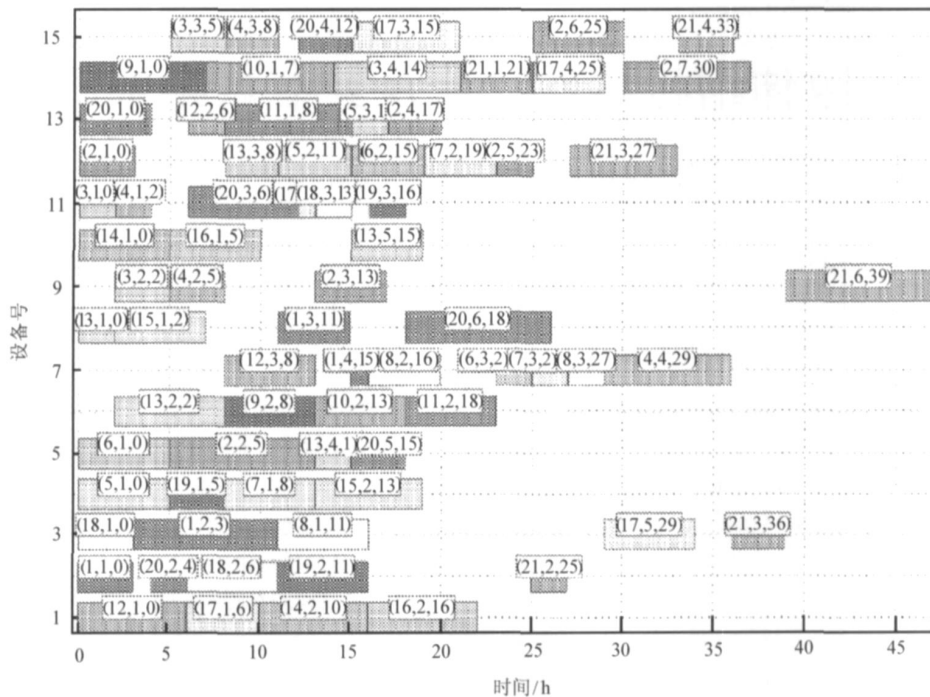


图 5 出现扰动后再调度甘特图

Fig. 5 Rescheduling gantt chart after disturbance

多种扰动因素,建立了柔性制造单元动态调度的数学模型;针对柔性制造单元动态调度提出了一种免疫算法。引入滑动窗口技术和调度任务池的概念,采用了滚动-扰动混合再调度策略。通过西安航空发动机(集团)有限公司柔性制造单元动态调度的具体应用表明,所提出的算法、策略和方法能满足柔性制造单元动态调度的要求。

参 考 文 献

- [1] Jain A S, Meeran S. A state of the art review of job shop scheduling techniques [R]. Technical report, Scotland Dundee: University of Dundee, 1998. 130-140.
- [2] Kacem I, Hammadi S, Borne P. Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic[J]. Mathematics and Computers in Simulation, 2002, 60 (35) : 245-276.
- [3] 夏蔚军,吴智铭.基于混合微粒群优化的多目标柔性 Job-shop 调度[J].控制与决策,2005,20(2):137-141.
Xia W J, Wu Z M. Hybrid particle swarm optimization approach for multi-objective flexible job-shop scheduling problems[J]. Control and Decision, 2005, 20 (2) : 137-141. (in Chinese)
- [4] 周泓,冯允成.一种启发式混合遗传算法及其在车间作业排序问题中的应用[J].航空学报,1998,19(1):74-77.
Zhou H, Feng Y C. A kind of hybrid heuristic genetic algorithm with applications to job-shopscheduling [J]. Acta Aeronautica et Astronautica Sinica, 1998, 19 (1) : 74-77. (in Chinese)
- [5] Alisantoso D, Khoo L P, Jiang P Y. An immune algorithm approach to the scheduling of a flexible PCB flow shop[J]. Advanced Manufacturing Technology, 2003, 22: 819-827.
- [6] 余建军,孙树栋,郑锋.基于动态评价免疫算法的车间作业调度研究[J].机械工程学报,2005,41(3):25-31.
Yu J J, Sun S D, Zheng F. Job-shop scheduling study by dynamic evaluation based immune algorithm [J]. Chinese Journal of Mechanical Engineering, 2005, 41 (3) : 25-31. (in Chinese)
- [7] 李寿安,张恒喜,李曙林,等.飞机生存力评估与综合权衡方法研究[J].航空学报,2005,26(1):23-26.
Li S A, Zhang H X, Li S L, et al. Research on aircraft survivability evaluation and synthetic tradeoff methods[J]. Acta Aeronautica et Astronautica Sinica. 2005, 26 (1) : 23-26. (in Chinese)
- [8] Church L, Uzsoy R. Analy of Periodic and event-driven re-scheduling policies in dynamic shops[J]. International Journal of Computer Intergrated Manufacturing, 1992, 5(3):153-163.

作者简介:



余建军(1978 -) 男,江西南丰人,博士生。主要研究方向:计算机集成制造系统、制造单元调度、免疫算法等。
Tel:13201476547;
E-mail:npu_yjj@163.com

孙树栋(1963 -) 男,教授,博士生导师。主要研究方向:计算机集成制造系统、调度算法和机器人等。
E-mail:sdsun@nwpu.edu.cn

(责任编辑:蔡斐)