

DOI:10.16356/j.1005-2615.2018.03.016

# 基于可变重调度区间的柔性作业车间动态调度策略

王 雷 蔡劲草

(安徽工程大学机械与汽车工程学院, 芜湖, 241000)

**摘要:**针对动态柔性作业车间调度问题,提出了基于可变重调度区间的动态重调度策略。建立了柔性作业车间调度数学模型。提出一种初始化机器、初始化工序和随机初始化相结合的改进种群初始化的方法,进一步提高初始种群解的质量。实际生产案例仿真对比分析结果表明,利用本文重调度策略和改进遗传算法后得到的结果比企业实际运行时间缩短了 47.8%,比他人的调度策略所得到的优化结果提高了 5.4%,从而验证了本文所提出算法的可行性和有效性。

**关键词:**柔性作业车间;动态调度;种群初始化;遗传算法;可变重调度区间

**中图分类号:**TP278      **文献标志码:**A      **文章编号:**1005-2615(2018)03-0397-07

## Dynamic Flexible Job Shop Scheduling Problem Based on Variant Rescheduling Interval Strategy

WANG Lei, CAI Jingcao

(School of Mechanical and Automotive Engineering, Anhui Polytechnic University, Wuhu, 241000, China)

**Abstract:** Aiming at the dynamic flexible job shop scheduling problem, a dynamic rescheduling method based on variable rescheduling interval strategy is proposed. A mathematical model of the job shop scheduling problem is established. A method to initialize the population is proposed by combining machine initialization, process initialization and random initialization, and therefore the quality of the initial population solution can be made a further improvement. The contrast analysis results for an actual production case indicate that the makespan is 47.8% shorter than the actual makespan of the enterprise by using our proposed dynamic rescheduling method. Meanwhile, compared with other literature, the optimization result is improved by 5.4%. Therefore the results can prove the feasibility and validity of the proposed strategy.

**Key words:** flexible job shop; dynamic scheduling; population initialization; genetic algorithm; variant rescheduling interval strategy

如何提高效益和降低成本是制造企业永远追求的目标。相关资料研究表明,在制造过程中,90%以上的时间将消耗在运输、等待加工等非切削过程中<sup>[1]</sup>,因此制造企业需要合理进行生产调度才能缩短加工时间、降低生产成本和提高效益<sup>[2]</sup>。作

为制造企业车间生产管理的核心内容,生产调度在很大程度上是企业能否盈利的关键环节。生产调度的任务在于如何为每一个工件的工序在有限的加工机器上确定其加工顺序,以使一个或者多个目标达到最优<sup>[3]</sup>。其中,柔性作业车间调度问题

**基金项目:**安徽省自然科学基金(1708085ME129)资助项目;国家自然科学基金(51305001)资助项目;安徽省高校优秀青年人才支持计划(gxyqZD2016125)资助项目。

**收稿日期:**2017-01-02; **修订日期:**2017-03-23

**通信作者:**王雷,男,博士,副教授,E-mail:wangdalei2000@126.com。

**引用格式:**王雷,蔡劲草.基于可变重调度区间的柔性作业车间动态调度策略[J].南京航空航天大学学报,2018,50(3):397-403. WANG Lei, CAI Jingcao. Dynamic flexible job shop scheduling problem based on variant rescheduling interval strategy[J]. Journal of Nanjing University of Aeronautics & Astronautics, 2018, 50(3): 397-403.

(Flexible job shop scheduling problem, FJSP) 由于突破了机器约束条件的限制, 每一个工件的不同工序可以在多台机器上进行加工, 选择不同的加工机器所得到的优化结果不同。因此, FJSP 更加符合实际的生产, 对其更具有重要的指导意义。大多数对柔性作业车间调度模型的研究都是在静态环境下进行的, 如黄志清等以完工时间和能量消耗为优化目标, 对基于改进算法的柔性作业车间集成调度进行研究<sup>[4]</sup>。然而, 实际的调度过程中经常会出现一些随机的因素, 如紧急订单的加入、订单取消、订单优先级改变、机器故障等, 在这种情况下静态调度模型不再适用于实际的生产环境。所以, 动态环境下柔性作业车间调度问题开始得到许多学者的关注。刘爱军等<sup>[3]</sup>提出基于事件和周期驱动的混合再调度策略, 并利用自适应遗传算法求解多目标柔性作业车间动态调度, 通过生产实例仿真验证了该模型和算法的有效性和可行性。刘国宝等<sup>[5]</sup>提出了基于事件和周期混合驱动的滚动时域优化策略, 并进一步通过算例验证了该方法的有效性。文献<sup>[6]</sup>以工件延期及完工时间为目标, 对机器故障和工件随机到达等动态作业车间调度进行研究。Liu 等<sup>[7]</sup>对模糊处理时间的动态柔性作业车间调度进行了研究。Kundak 和 Kulak 考虑到机器故障、新工件插入及加工时间变化的动态事件, 以最小完工时间为目标, 利用混合遗传算法进行求解, 仿真结果表明该算法的有效性<sup>[8]</sup>。Zhang 等<sup>[9]</sup>以新工件的加入、机器故障为动态事件, 利用混合遗传禁忌搜索算法对多目标动态车间调度进行优化。刘明周等<sup>[10]</sup>结合滚动时域窗口, 利用混合粒子群算法对作业车间动态重调度问题进行研究并通过具体案例验证所提出方法的有效性。上述文献对动态柔性作业车间调度问题的研究, 基本上都是基于滚动时域优化策略, 利用固定周期驱动的重调度策略, 然而并没有考虑到可变重调度区间对动态调度优化目标的影响。

基于上述分析, 针对 FJSP 问题, 本文建立了以完工时间最小化为优化目标的柔性作业车间调度模型。提出基于可变重调度区间的动态调度策略, 并利用改进遗传算法进行求解。提出一种改进的种群初始化方法, 进一步提高初始解的质量。仿真测试案例验证了所提出动态调度算法的有效性, 从而为生产实际提供参考。

## 1 柔性作业车间调度模型

柔性作业车间调度可以描述为: 有  $n$  个工件

( $J_1, J_2, J_3, \dots, J_n$ ) 在  $m$  台机器 ( $M_1, M_2, M_3, \dots, M_m$ ) 上加工, 工件  $J_i$  包含一道或者多道加工工序, 每个工件的各个工序之间满足工艺的顺序要求, 同时每道工序可能有多台机器可以被选择。FJSP 研究的就是如何为每道工序选择一台加工机器, 以及如何确定每台机器上的多个工序的加工顺序及加工开始结束时间, 这种顺序不仅要满足工件工序之间的工艺顺序约束, 还需要满足机器的占用约束, 以最优化某个或者某些性能指标<sup>[2]</sup>。另外, 工序的装夹时间包含在工序的加工时间中, 不考虑工件的运输时间。

首先对相关数学符号说明如下:

$n$ —工件总数;

$J$ —工件集合, 其中  $J = \{J_i | (i=1, 2, 3, \dots, n)\}$ ;

$m$ —机器总数;

$M$ —机器集合, 其中  $M = \{J_i | (i=1, 2, 3, \dots, m)\}$ ;

$O_{ij}$ —工件  $J_i$  的第  $j$  道工序;

$T_{ijk}$ —工件  $J_i$  的第  $j$  道工序在机器  $k$  上的加工时间;

$S_{ijk}$ —工件  $J_i$  的第  $j$  道工序在机器  $k$  的加工开始时间;

$E_{ijk}$ —工件  $J_i$  的第  $j$  道工序在机器  $k$  的加工结束时间;

$n_i$ —工件  $J_i$  的工序数;

$C_i$ —工件  $J_i$  的完工时间。

如果  $O_{ij}$  在机器  $k$  上加工, 则  $x_{ijk} = 1$ ; 否则  $x_{ijk} = 0$ 。

以最大完工时间最小为优化目标, 建立柔性作业车间调度数学模型如下

$$f = \min C_{\max} = \min(\max_{i=1}^n (C_i)) \quad (1)$$

同一个工件的相邻工序之间必须满足先后顺序约束, 因此, FJSP 必须满足以下工序顺序约束

$$E_{ijk} \leq S_{i(j+1)k'} \quad (2)$$

式中:  $\forall i \in [1, n], \forall j \in [1, n_i - 1], \forall k, k' \in [1, m]$ 。

机器占用约束表明机器  $k$  加工一个工序完成后才能加工其他工序

$$E_{ijk} < E_{i'j'k} \quad (3)$$

式中:  $x_{ijk} = 1, x_{i'j'k} = 1, \forall i \in [1, n], \forall j \in [1, n_i]$ 。

加工过程中工序不允许中断, 有

$$E_{ijk} - S_{ijk} = T_{ijk} \quad (4)$$

式中:  $x_{ijk} = 1, \forall i \in [1, n], \forall j \in [1, n_i - 1], \forall k \in [1, m]$ 。

在同一个时刻同一个工序只能允许在一台机器上加工, 有

$$\sum_{i=1}^m x_{ijk} = 1 \quad (5)$$

式中:  $\forall i \in [1, n], \forall j \in [1, n_i], \forall k \in [1, m]$ 。

$$T_{ijk} \geq 0, S_{ijk} \geq 0, E_{ijk} \geq 0 \quad (6)$$

式中:  $\forall i \in [1, n], \forall j \in [1, n_i], \forall k \in [1, m]$ 。

## 2 基于可变重调度区间动态重调度策略

动态作业车间调度是指在执行优化后的静态调度过程中,遇到譬如紧急任务插入、设备故障等动态事件时,系统能够做出及时响应,对加工任务进行再调度,以适应动态的加工环境。目前常用的动态重调度策略有基于周期性重调度、基于事件驱动重调度、基于周期与事件驱动相结合的重调度 3 种。基于周期的重调度能够根据每个加工周期的生产情况,全局对调度过程进行考虑,但它不能对动态事件及时有效得进行处理。基于事件的重调度对动态事件反应及时,但该策略缺乏从全局观方面进行考虑与分析。基于周期与事件驱动相结合的重调度策略则将两者很好地融合,但是该方法得到的结果也不一定是最优的。

基于以上分析,针对 FJSP 问题,本文提出基于可变重调度区间的重调度策略,即在动态事件发生以后,只对一定范围内的工件进行重调度,范围的大小根据动态事件直接影响到的工件来确定,其他区间中不被动态事件直接涉及到的工件的加工顺序不变,工序所在的加工机器也不变。

确定重调度区间的步骤如下:(1)确定动态事件影响到的工件;(2)将影响到的工件从调度区删除;(3)将剩余未受影响的工件合并;(4)将动态事件影响到的工件和未影响到的工件在一起进行重调度;(5)确定重调度后受动态事件直接影响到的工件的开始时间  $t_1$  与完成时间  $t_2$ ;(6)重调度区间即为  $[t_1, t_2]$ 。

以表 1 的加工信息为例,首先利用遗传算法获得柔性作业车间静态调度的一个调度方案:2—3—2—2—5—1—2—4—2—5—3—3—3—3—5—3—3—5—1—1—4—1—2—1—4—3—1—2—4—5—7—3—2—6—9—8—6—4—10—6—5—3—5—10—8—1—7—9—4—3—2—1—9—2—10—5—7—8—1—4(数字表示工件号,该数字在此序列中出现的次数表示其加工工序),调度甘特图如图 1 所示。自定义动态事件为机器  $M_1$  损坏且不可修复,直接影响到的工件有 2,7,8 和 9,发生时刻为 5 时刻,如图 2 所示。利用本文提出的可变重调度区间的动态调度策略获得的重调度方案如图 3 所示。从图 3 可以看出,工件 5 的第 2 道工序进行了后移,工件 3 的第 3 道工序和

工件 1 的第 1 道工序进行了后移,使得不受动态事件直接影响的工件的加工顺序不变,工序所在的加工机器也不变。这里给出的只是一种基于可变重调度区间动态重调度策略方案,但是该方案并不一定是最优的,因此需要进一步通过优化算法对其进行优化处理。

表 1 10 个工件 5 个机器的加工信息

Tab. 1 Processing information for ten jobs and five machines

| 工件       | 工序        | $M_1$ | $M_2$ | $M_3$ | $M_4$ | $M_5$ |
|----------|-----------|-------|-------|-------|-------|-------|
| $J_1$    | $O_{11}$  | 2     | 3     | 2     | 5     | 2     |
|          | $O_{12}$  | 5     | 3     | 2     | 4     | 3     |
|          | $O_{13}$  | 5     | 2     | 5     | 2     | 4     |
| $J_2$    | $O_{21}$  | 2     | 2     | 2     | 2     | 4     |
|          | $O_{22}$  | 5     | 3     | 4     | 4     | 3     |
|          | $O_{23}$  | 3     | 3     | 3     | 3     | 5     |
| $J_3$    | $O_{31}$  | 2     | 2     | 3     | 2     | 5     |
|          | $O_{32}$  | 3     | 5     | 5     | 3     | 4     |
|          | $O_{33}$  | 4     | 2     | 3     | 5     | 4     |
| $J_4$    | $O_{41}$  | 4     | 4     | 4     | 4     | 2     |
|          | $O_{42}$  | 2     | 4     | 3     | 5     | 2     |
|          | $O_{43}$  | 2     | 4     | 2     | 3     | 3     |
| $J_5$    | $O_{51}$  | 3     | 4     | 2     | 3     | 5     |
|          | $O_{52}$  | 5     | 3     | 2     | 2     | 4     |
|          | $O_{53}$  | 4     | 4     | 4     | 5     | 2     |
| $J_6$    | $O_{61}$  | 3     | 3     | 2     | 3     | 5     |
|          | $O_{62}$  | 2     | 4     | 2     | 3     | 5     |
|          | $O_{63}$  | 4     | 2     | 2     | 2     | 4     |
| $J_7$    | $O_{71}$  | 5     | 2     | 4     | 5     | 5     |
|          | $O_{72}$  | 3     | 5     | 4     | 5     | 4     |
|          | $O_{73}$  | 3     | 4     | 5     | 2     | 4     |
| $J_8$    | $O_{81}$  | 2     | 2     | 4     | 5     | 3     |
|          | $O_{82}$  | 4     | 2     | 3     | 5     | 5     |
|          | $O_{83}$  | 2     | 2     | 4     | 3     | 4     |
| $J_9$    | $O_{91}$  | 4     | 3     | 5     | 4     | 4     |
|          | $O_{92}$  | 2     | 3     | 2     | 4     | 5     |
|          | $O_{93}$  | 2     | 2     | 3     | 2     | 5     |
| $J_{10}$ | $O_{101}$ | 4     | 2     | 3     | 4     | 3     |
|          | $O_{102}$ | 4     | 4     | 2     | 4     | 5     |
|          | $O_{103}$ | 2     | 5     | 2     | 3     | 3     |

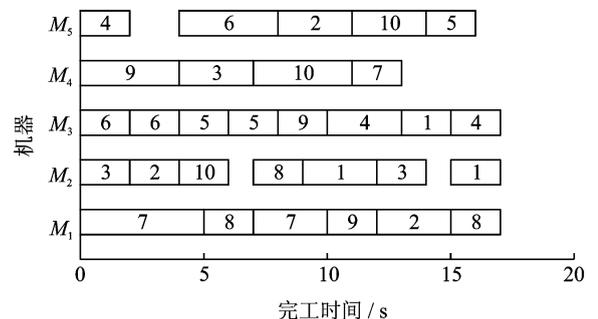


图 1 静态调度结果

Fig. 1 Static scheduling result

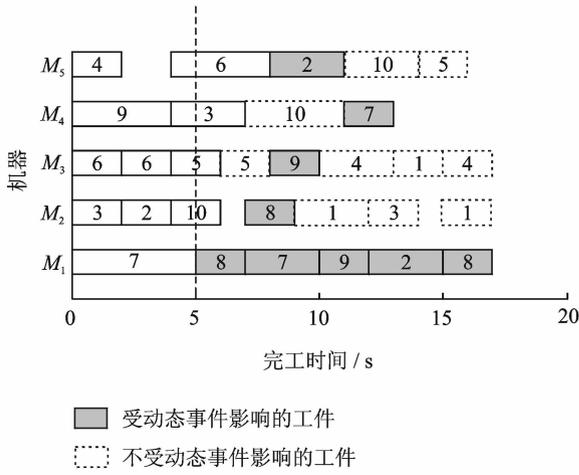


图2 动态事件发生后工件受影响情况

Fig. 2 Affected workpiece condition after occurrence of dynamic event

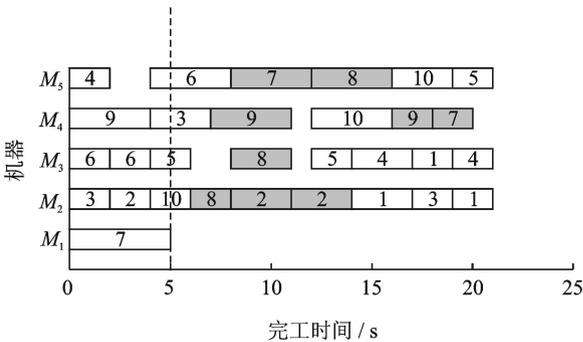


图3 动态重调度结果

Fig. 3 Dynamic scheduling result

### 3 改进遗传算法求解柔性作业车间动态调度

#### 3.1 编码与解码

编码主要用来确定各工序所在的加工机器和加工顺序,因此采用双层编码方式可以很好地进行表达,具体表现形式如图4所示。机器部分确定工序所在的加工机器,工序部分确定工序的加工顺序。举例:若工序  $O_{11}, O_{12}, O_{13}, O_{21}, O_{22}, O_{23}$  所在的加工机器分别为  $M_3, M_1, M_2, M_2, M_1, M_3$ , 工序的加工顺序为  $O_{21}, O_{11}, O_{12}, O_{22}, O_{13}, O_{23}$ , 则编码为 312213211212, 如图5所示。解码为编码的逆操作,即根据编码确定工序的加工机器和工序的加工顺序的过程。



图4 编码方式

Fig. 4 Encoding method

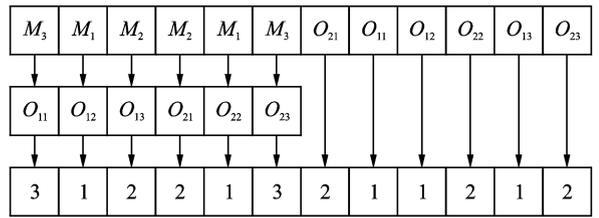


图5 编码示例

Fig. 5 Encoding example

#### 3.2 初始化种群

相比传统的作业车间调度问题, FJSP 是一类更复杂的 NP-hard 问题。种群初始化的好坏对于遗传算法求解 FJSP 影响很大,而以往大都采用随机种群初始化方法进行初始种群的生成,但是这种方法提供解的质量往往不一定有保证。

本文提出以初始化机器、初始化工序和随机初始化相结合的方法对种群进行初始化,具体方法如下。

##### (1) 初始化机器

由于优化目标是完工时间,因此优先考虑加工效率高的机器,即对于加工的工序,优先选择加工时间少的机器。此阶段不考虑工件的加工顺序,具体步骤如下:(a)读取工序的加工时间表  $T$ ; (b)求每道工序的最小值  $T_{\min}$ ; (c)根据  $T_{\min}$  利用式(7)确定选取各工序的加工机器的概率; (d)根据各个工序所在加工机器的选取概率随机选择机器; (e)重复步骤(d)直到机器编码部分完全初始化。

$$p_i = \frac{1}{T_{\min_i}} \sum_{i}^{apq} \frac{1}{T_{\min_i}} \quad (7)$$

##### (2) 初始化工序

当机器部分初始化以后,开始对工序部分进行初始化。按照顺序,从工序编码的第一位开始。每选择一个工件,则形成一个调度方案,根据选取可选工件后的完工时间来确定编码的初始值。由于优化的问题是完工时间,所以选取工件后的完工时间越小,选取此工件的概率就越大。(a)读取当前加工信息表  $a$ , 具体描述如图6所示; (b)计算选取可选工件后的完工时间  $\text{mak}_{\text{cur}}$ ; (c)根据式(8)确定选取各工件的概率; (d)根据概率确定选取的工件; (e)重复(d)直到工序编码部分完全初始化。

$$p_i = \frac{1}{\text{mak}_{\text{cur}_i}} \sum_i^n \frac{1}{\text{mak}_{\text{cur}_i}} \quad (8)$$

##### (3) 随机初始化

随机初始化步骤如下:(a)随机选取可选的加工机器; (b)从可选工件集中随机选择加工工件; (c)重复(a, b), 直到种群完全初始化。

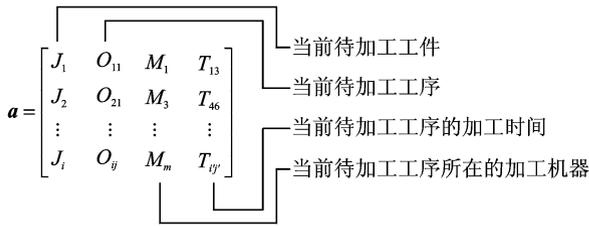


图6 当前加工信息

Fig. 6 Current processing information

### 3.3 选择操作

选择操作是种群中适应度值大的个体将被选择进化到下一代种群中,适应度值小的个体被淘汰掉,它反映了“优胜劣汰”的进化精髓。本文采用常用的轮盘赌法进行选择操作。

### 3.4 交叉操作

对工序的交叉操作采取张超勇提出的 POX 交叉<sup>[11]</sup>。POX 交叉过程如下:将工件任意分成  $J_1$  和  $J_2$  两个集合,交叉操作时,父代 1 保留集合  $J_1$  中基因的位置不变,其他位置基因则由父代 2 中  $J_2$  集合中相应位置基因来依次填补,具体实例如图 7 所示(其中  $J_1$  由工件 1 和 4 组成,  $J_2$  由工件 2 和 3 组成)。

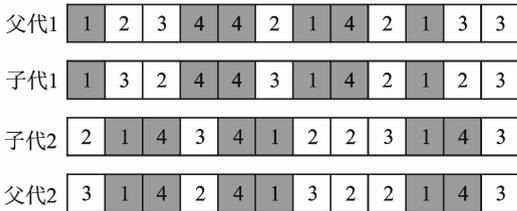


图7 POX交叉实例

Fig. 7 Crossover example based on POX

对机器的交叉操作在此采取任意产生 0-1 的方法。具体过程为:随机生成一个由 0-1 组成的染色体,该染色体的长度等于机器染色体的长度。保持两个机器染色体上的基因与 0 位置相对应的不变,交换两个机器染色体上与 1 位置相对应的基因,具体实例如图 8 所示。

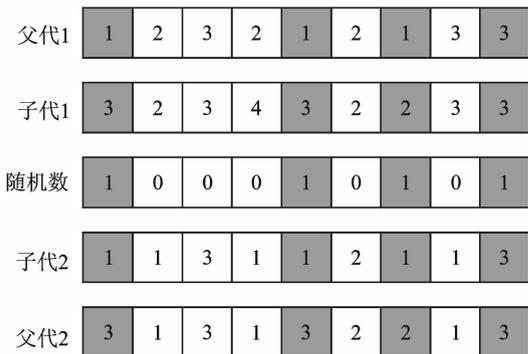


图8 机器交叉实例

Fig. 8 Crossover example for machine

### 3.5 变异操作

利用随机选择一个基因位置,然后顺序交换两个基因片段,以保证解可行,具体实例如图 9 所示。

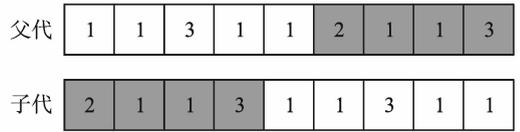


图9 对工序进行变异

Fig. 9 Mutation for operation

对机器部分变异则采取依次后移基因的方法(最后一个基因则移动到首位置上),以保证解可行,具体实例如图 10 所示。

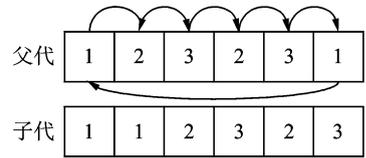


图10 对机器进行变异

Fig. 10 Mutation for machine

最终得到利用改进初始种群的遗传算法及可变重调度区间求解柔性作业车间动态调度流程图,如图 11 所示。

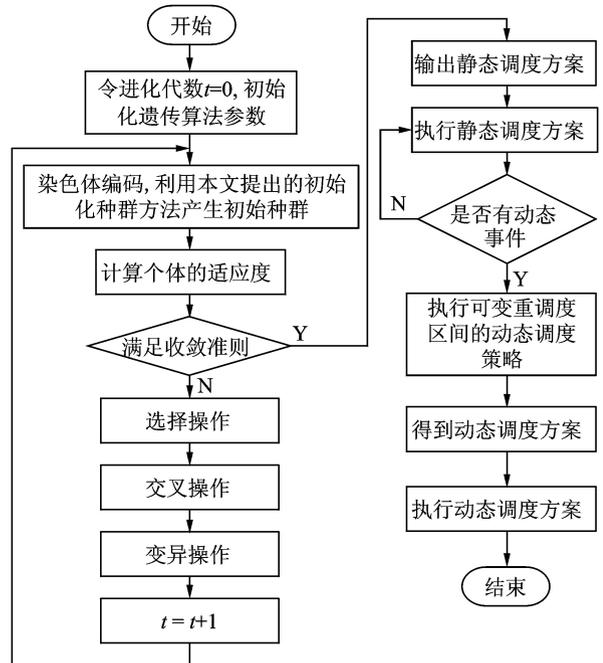


图11 改进的遗传算法流程图

Fig. 11 Flow chart of improved genetic algorithm

## 4 实验与结果分析

为了证明本文提出的改进初始化种群方法的

遗传算法和动态调度策略的有效性和可行性,利用文献[12]中的案例对柔性作业车间动态调度问题进行验证。遗传算法的参数设置:种群规模 100,进化代数 100,交叉概率 0.8,变异概率 0.1。

表 2 显示的是利用本文提出的初始化方法和随机初始化方法获得的初始种群的性能对比结果。从表 2 中可以看出本文提出的初始化种群方法可以获得优质的个体。

图 12 是利用传统随机初始化种群方法获得适应度值的优化结果,图 13 是利用本文改进初始化种群方法获得适应度值的优化结果。从图中可以看出,利用未改进的方法需要近 80 代才能获得最优解 40,而利用本文改进的初始化种群的方法仅需要 40 多代便可获得最优解 35。利用本文算法得到的静态调度最优解的甘特图如图 14 所示。本文方法比文献[12]中利用改进算法获得的静态调度的最优解还减少了 2 个时间单位,所以本文提出的初始化种群方法是有效性。

另外,假设在时刻 20 机器 6 出现故障,利用本文改进的遗传算法和可变重调度区间方法进行动态重调度,动态事件发生后的重调度方案如图 15 所示,动态重调度的结果对比如表 3 所示。

表 2 不同初始化方法获得的初始种群的性能对比结果  
Tab. 2 Performance comparison of initial population obtained by different initialization methods

| 方案      | 初始化种群独立运行次数 | 最佳个体的适应度值 | 种群的平均适应度值 |
|---------|-------------|-----------|-----------|
| 本文初始化方法 | 100         | 41.7      | 74.83     |
| 随机初始化方法 | 100         | 58.79     | 90.94     |

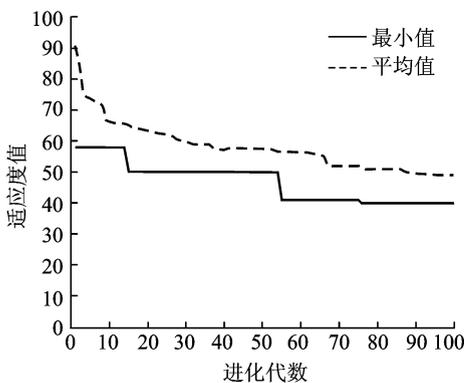


图 12 随机初始化种群方法获得的适应度值的优化过程  
Fig. 12 Evolutionary process of fitness values obtained by randomly initialization population method

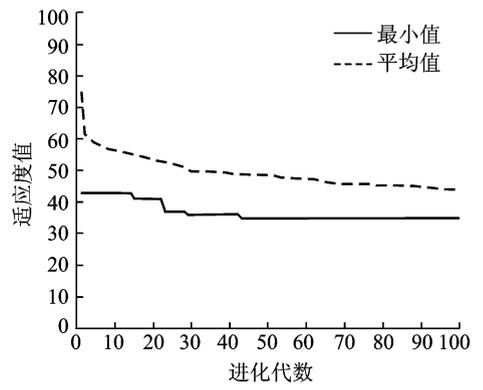


图 13 利用改进初始化种群方法获得的适应度值的进化过程  
Fig. 13 Evolutionary process of fitness values obtained by improved initialization population method

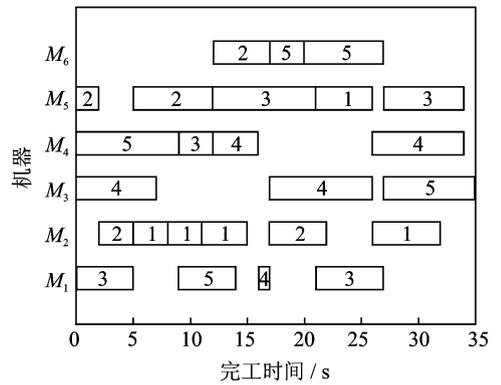


图 14 静态调度甘特图

Fig. 14 Static scheduling Gantt chart

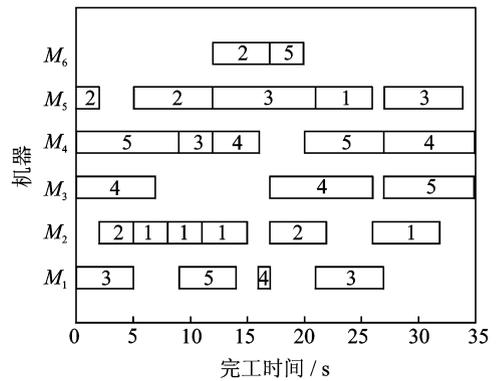


图 15 重调度甘特图

Fig. 15 Rescheduling Gantt chart

表 3 重调度的结果对比

Tab. 3 Comparison of rescheduling results

| 时间/s   | 实际企业调度 | 文献[12] | 本文 |
|--------|--------|--------|----|
| 故障时刻   | 20     | 20     | 20 |
| 维修机器时间 | 30     | 0      | 0  |
| 理想完成时间 | 37     | 37     | 35 |
| 实际完成时间 | 67     | 37     | 35 |

从表3重调度结果对比不难看出,运用本文所提出的基于可变重调度区间的动态调度策略后,之前需要在故障设备上加工而未完成的工件能够直接转移到具有相当加工能力的设备上加工,而没有必要等待故障设备修复之后继续加工,从而使完工时间大大减少。从表3也可以看出,利用本文重调度策略和改进遗传算法后得到的任务完成时间比企业实际完成时间缩短了47.8%,比文献[11]的调度策略提高了5.4%,从而验证了本文所提出算法的可行性和有效性。

## 5 结 论

(1)提出了改进初始化种群的遗传算法,进一步提高了初始化种群解的质量。

(2)建立了柔性作业车间动态调度数学模型,提出了可变重调度区间的动态重调度策略,具体对比案例验证了所提出算法的可行性和有效性,进一步对生产实际提供很好的指导作用。

### 参考文献:

- [1] GUTOWSKI T, MURPHY C, ALLEN D, et al. Environmentally benign manufacturing: Observations from Japan, Europe and the United States [J]. *Journal of Cleaner Production*, 2005, 13(1):1-17.
- [2] 刘志虎. 基于改进蚁群算法的柔性车间调度研究[D]. 芜湖:安徽工程大学,2016.  
LIU Zhihu. Research of flexible job shop scheduling problem based on improved ant colony algorithm [D]. Wuhu: Anhui Polytechnic University, 2016.
- [3] 刘爱军,杨育,邢青松,等. 柔性作业车间多目标动态调度[J]. *计算机集成制造系统*, 2011, 17(12):2629-2637.  
LIU Aijun, YANG Yu, XING Qingsong, et al. Dynamic scheduling on multi-objective flexible Job Shop [J]. *Computer Integrated Manufacturing Systems*, 2011, 17(12):2629-2637.
- [4] 黄志清,唐敦兵,戴敏. 基于改进算法的工艺规划与车间调度的双目标优化模型[J]. *南京航空航天大学学报*, 2015, 47(1):88-95.  
HUANG Zhiqing, TANG dunbing, DAI Min. Bi-objective optimization model for integrated process planning and scheduling based on improved algorithm [J]. *Journal of Nanjing University of Aeronautics & Astronautics*, 2015, 47(1):88-95.
- [5] 刘国宝,张洁. 基于改进滚动时域优化策略的动态调度方法[J]. *机械工程学报*, 2013, 49(14):182-190.  
LIU Guobao, ZHANG Jie. Dynamic schedule method based on improved rolling time domain optimization strategy [J]. *Journal of Mechanical Engineering*, 2013, 49(14):182-190.
- [6] ADIBI M A, ZANDIEH M, AMIRI M. Multi-objective scheduling of dynamic job shop using variable neighborhood search [J]. *Expert Systems with Applications*, 2010, 37(1):282-287.
- [7] LIU B, FAN Y, LIU Y. A fast estimation of distribution algorithm for dynamic fuzzy flexible job-shop scheduling problem [J]. *Computers & Industrial Engineering*, 2015, 87:193-201.
- [8] KUNDAKC N, KULAK O. Hybrid genetic algorithms for minimizing makespan in dynamic job shop scheduling problem [J]. *Computers & Industrial Engineering*, 2016, 96(C):31-51.
- [9] ZHANG Liping, GAO L, LI X. A hybrid genetic algorithm and tabu search for multi-objective dynamic job shop scheduling problem [J]. *International Journal of Production Research*, 2013, 51(12):3516-3531.
- [10] 刘明周,单晖,蒋增强,等. 不确定条件下车间动态重调度优化方法[J]. *机械工程学报*, 2009, 45(10):137-142.  
LIU Mingzhou, SHAN Hui, JIANG Zengqiang, et al. Dynamic rescheduling optimization of job-shop under uncertain conditions [J]. *Journal of Mechanical Engineering*, 2009, 45(10):137-142.
- [11] 张超勇,饶运清,李培根,等. 柔性作业车间调度问题的两级传算法[J]. *机械工程学报*, 2007, 43(4):119-124.  
ZHANG Chaoyong, RAO Yunqing, LI Peigen, et al. Bilevel genetic algorithm for the flexible job-shop scheduling problem [J]. *Journal of Mechanical Engineering*, 2007, 43(4):119-124.
- [12] 吴正佳,何海洋,黄灿超,等. 带机器故障的柔性作业车间动态调度[J]. *机械设计与研究*, 2015, 31(3):94-98.  
WU Zhengjia, HE Haiyang, HUANG Canchao, et al. Flexible job shop dynamic scheduling problem research with machine fault [J]. *Machine Design and Research*, 2015, 31(3):94-98.

