

# 一种启发式算法和改进遗传混合算法 在流水车间重调度中的应用

王 森, 熊福力, 李 志

(西安建筑科技大学 信息与控制工程学院, 西安 710055)

**摘要:** 在解决以合同惩罚和存储成本最小化为优化目标的流水车间重调度问题时, 提出了一种启发式算法和改进的遗传混合算法; 传统的遗传算法是一种基于优胜劣汰的随机、自适应的优化算法; 通过复制, 交叉和变异, 将问题解编码所表示的“染色体”群在逐代进化, 最终收敛到最合适的群体, 从而得到问题的最优或满意解; 但缺点是求解结果依赖于初始值, 且运行时间过长; 因此对传统遗传算法做了相应的改进, 考虑到启发式算法的快速性, 为充分发挥两种算法的优势, 提出启发式算法和改进遗传混合算法; 最后对性能进行分析; 试验结果表明: 该算法运行时间短, 且在大规模数据集下, 更易于靠近全局最优解。

**关键词:** 遗传算法; 染色体; 初始值; 启发式算法

## Application of a Heuristic Algorithm and Improved Genetic Hybrid Algorithm in Production Rescheduling

Wang Sen, Xiong Fuli, Li Zhi

(College of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an 710055, China)

**Abstract:** A heuristic algorithm and an improved genetic hybrid algorithm are proposed to solve the rescheduling problem of flow shop with the objective of minimizing the contract penalty and storage cost. The traditional genetic algorithm is a random and adaptive optimization algorithm based on the survival of the fittest. By means of replication, crossover and mutation, the “chromosome” group represented by the solution coding is evolved from generation to generation, and finally converges to the most appropriate group, so as to obtain the optimal or satisfactory solution of the problem. But the disadvantage is that the solution depends on the initial value, and the running time is too long. In order to give full play to the advantages of the two algorithms, a heuristic algorithm and an improved genetic hybrid algorithm are proposed. Finally, the performance of the algorithm is analyzed, and the experimental results show that the algorithm runs in a short time, and is easier to approach the global optimal solution in a large data set.

**Keywords:** genetic algorithm; chromosome; initial value; heuristic algorithm

## 0 引言

流水车间重调度问题是以流水车间调度问题 (flow shop scheduling problem, FSP) 为基础模型的一种动态调度问题, 多年来, 研究者们大多关注以最大完工时间为优化目标, 较少关注工件交货期对流水车间调度问题的影响。在考虑交货期时, 一般认为交货期是一个固定的值, 而不是一个模糊变量。遗传算法作为一种群体智能算法, 具有全局搜索的特点, 是求解流水车间调度问题常用的一种算法。然而, 传统遗传算法存在收敛速度慢、稳定性差等

缺点<sup>[1]</sup>。

在实际生产过程中, 干扰事件的突然到来往往导致原调度不在可行, 继而重新生成新调度方案, 即为重调度。重调度即在尽量小损耗原调度性能水平的同时追求调度稳定性的优越, 如何既经济又高效地实施干扰, 重调度一直以来备受国内外调度学者与实践者的关注。常见的生产干扰事件有机器突然故障、一批新工件到达、工件返工、订单撤销、交货期提前、误操作等等<sup>[2]</sup>。

Chan 和 Hu<sup>[3]</sup>介绍了一种基于人工智能的制造业流水调度方法, 并通过对预制生产特点的分析, 建立了预制生产流水车间排序模型。Benjaoran<sup>[4]</sup>等人研究了模具数量对预制生产车间进度的影响, 提出了定制预制生产车间进度模型。Ko 和 Wang<sup>[5]</sup>通过包括缓冲区大小的限制, 即临时存储位置的大小, 改进了人工智能调度的可行性, 在机器之间, 将待完工的部分完工预制件 (以下简称在制品) 存储到优化模型中, 并开发出相应的调度系统。Yang<sup>[6]</sup>等人提出了多条预制生产线的流水车间调度模型, 以促进多条预制生产线的优化调度。动态调度出现较早, Jackson 对静态调度和动态调度做了区分。Yang<sup>[7]</sup>提出了流水车间预制

收稿日期:2020-04-02; 修回日期:2020-05-06。

**基金项目:**国家自然科学基金项目(61473216); 陕西省教育厅科学研究计划项目(17JK0459); 陕西省自然科学基金(2015JM6337); 陕西省自然科学基金面上项目(2020JM-489); 西安建筑科技大学基础研究项目(ZR18049)。

**作者简介:**王 森(1994-), 男, 陕西渭南人, 硕士, 主要从事重调度优化方向的研究。

**通讯作者:**熊福力(1974-), 男, 黑龙江肇东人, 硕士生导师, 副教授, 主要从事人工智能与系统优化、生产计划与调度优化、智能建筑方向的研究。

生产多条生产线优化重调度模型。

本文针对传统遗传算法的不足之处对其作了相应的改进, 在选择阶段引入了精英保留策略来确保算法的收敛性, 在进行重调度时引入启发式算法; 最后通过实验分析了该算法的优点, 描述了流水车间重调度模型, 并且通过实验分析了启发式算法和改进遗传混合算法求解流水车间重调度问题。

## 1 重调度的数学模型

FSP 一般描述如下: 有  $m$  台机器和  $n$  个作业; 每个作业由  $m$  个操作组成, 每个操作需要不同的机器。必须在所有机器上以相同的顺序处理  $n$  个作业<sup>[8]</sup>。其目的是找出作业顺序, 使最大流动时间最小化, 这在技术上被称为生产计划的最大完成时间。两个重要的约束为: ① 一台机器上不能同时加工多个工件, 且一个工件不能同时在多台机器上加工<sup>[9]</sup>; ② 同一机器上工件的加工次序不限制, 每个工件的加工路线亦无限制。预制构件都要经过 6 个生产步骤, 即 6 道工序, 它们分别为成型 (M1)、放置钢筋和埋件 (M2)、浇注 (M3)、养护 (M4)、脱模 (M5) 和精加工 (M6)。其中, 养护是一个并行过程, 可以同时处理多个作业。除养护外, 其余步骤是串行过程。浇筑, 养护不可中断, 其余步骤可中断;

### 1) 优化目标

合同惩罚和存储成本是优化目标中, 将随着工件数量线性增加。即优化目标表示为:

$$\min f_{\infty} = \alpha_i * \text{Max}[0, \text{due date}(i) - C(J_i, M_k)] + \beta_i * \text{Max}[0, C(J_i, M_k) - \text{due date}(i)] \quad (1)$$

其中:  $f_{\infty}$  代表合同惩罚和存储成本;

$J_i$  是按顺序  $i$  生产的工件号,  $M_k$  是处理第  $k$  步的机器的序号;

$C(J_i, M_k)$  是预制件  $J_i$  在机器  $M_k$  上的离开时间;

$\alpha_i, \beta_k$  为早期惩罚与延期惩罚的惩罚系数;

$\text{due date}(i)$  是预制件  $J_i$  的交货期。

### 2) 优化约束

#### (1) 机器生产率约束

在调度过程中, 不仅要考虑正常生产对机器生产率的约束, 还要考虑生产紧急情况对机器生产率的制约。约束取决于生产条件变化的原因。式 (2) 和式 (3) 表示生产率的约束;

$$S(J_i, M_k) \geq \begin{cases} \text{Max}[C(J_{i-1}, M_k), C(J_i, M_{k-1})] & \text{if } k \neq 4 \\ C(J_i, M_{k-1}) & \text{if } k = 4 \end{cases} \quad (2)$$

$$C(J_i, M_k) \geq S(J_i, M_k) + P_{i,k} \quad (3)$$

其中:  $C(J_i, M_k)$ ,  $S(J_i, M_k)$ ,  $P_{i,k}$  是预制件  $J_i$  在机器  $M_k$  上的离开时间、进入时间、处理时间。

#### (2) 八小时工作制的约束

式 (4) 表示浇注工序每天八小时的工作; 式 (5) 表示养护工序每天八小时的工作; 式 (6) 表示其他工序每天

八小时的工作。

$$C(J_i, M_3) \geq \begin{cases} T, & \text{if } T \leq 24D + H_w + H_E \\ 24(D+1) + P_{i,k} & \text{if } T > 24D + H_w + H_E \end{cases} \quad (4)$$

$$C(J_i, M_4) \geq \begin{cases} T & \text{if } T < 24D + H_w \text{ or } T > 24(D+1) \\ 24(D+1) & \text{if } 24D + H_w \leq T < 24(D+1) \end{cases} \quad (5)$$

$$C(J_i, M_k) \geq \begin{cases} T & \text{if } T < 24D + H_w \text{ and } k = 1, 2, 5, 6 \\ T + H_N, \text{ if } T \geq 24D + H_w \text{ and } k = 1, 2, 5, 6 \end{cases} \quad (6)$$

其中:  $H_w, H_N, H_E$  是每天允许工作时间、非工作时间和加班时间。  $T$  是在不考虑 8 小时工作制的情况下计算  $C(J_i, M_k)$  得到的。  $D = (T/24)$  是从生产开始到  $C(J_i, M_k)$  的总天数,  $D$  取整。

当有紧急加工工件时, 对其进行重调度处理, 对紧急工件的处理方式本文的做法为: 给予紧急加工工件更高的加工权重, 使其先加工; 在重调度过程中尽可能早地完工紧急加工工件; 而对于重调度的开始时间的设置, 假设原调度的开始时间从零开始, 在急件到达时刻  $RT > 0$ , 而工件交付日期不变, 故而重调度的开始时间  $T_{\text{start}}$  即为  $RT$  时刻当前工件的 6 个工序的离开时间:

$$T_{\text{start}} = C(J_{RT}, M_k) \quad (7)$$

其中:  $J_{RT}$  取整数;

重调度策略的步骤如下所示:

① 在已排好的调度计划基础上, 有紧急订单到来;

② 假设有多个紧急订单, 确定影响订单优先级的各个因素, 将所有订单按优先级进行排序;

③ 根据紧急订单策略, 将紧急订单依次插入到待加工工序, 找到库存成本与延期交货的惩罚之和最小的位置;

④ 找到所有紧急订单的插单位置, 得出最小惩罚, 插单结束。

## 2 启发式算法和改进遗传混合算法

### 2.1 改进遗传算法

遗传算法是一种模拟达尔文生物进化论的模型, 是通过模拟自然进化来寻找最优解的一种群体优化算法。改进遗传算法在传统遗传算法的基础上, 加入精英保留策略以及动态变异率, 最后经过调参, 找出结果最优的遗传算法各个参数, 从而使得改进遗传算法结果更优。

#### 2.1.1 编码

本文采用工件号编码来表示个体 (染色体), 一组编码代表一种加工方案, 如对于 6 个工件中的工序 [4 1 3 2 6 5], 其中数字的顺序代表工件在各个机器上的顺序, 数字代表着相应的工件号, 如数字 “4” 表示在机器上第一个加工工件为工件 4, 数字 “1” 表示机器上第二个加工工件为工件 1, 以此类推, 数字 “5” 表示在机器上第六个加工工件为工件 5。所有工件在机器上的加工顺序相同, 且每台机器在同一时刻只能加工一个工件。

### 2.1.2 初始种群的产生

在本文中,初始种群是随机产生的。使用 MATLAB 软件中的随机排序函数“randperm”对染色体进行随机编码,给定种群大小,通过多次调用该函数得到所需大小的初始种群。

### 2.1.3 计算个体适应度值

以目标函数的倒数作为个体的适应度值,并采用精英保留策略<sup>[10]</sup>使最优个体存活下来。

### 2.1.4 选择

本文采用合同惩罚和存储成本最小的指标来对调度个体的适应度值进行评价,即个体适应度值大的为优良个体。在计算完每个调度个体的适应度值后,采用精英保留策略进行个体选择。

### 2.1.5 交叉

本文采用部分映射交叉 (Partial-Mapped Crossover, PMX) 进行交叉,首先,随机选择一对染色体 (父代) 中多个基因的起始和终止位置 (在同一位置选择两条染色体);然后,交换两组基因的位置;最后,进行重复检测,并建立基于两组交换基因的映射关系,如图 1 所示,以 1-6-3 这一映射关系为例,可以看到第二步结果中 Child1 存在两个重复基因 1,则将基因 1 先变为基因 6,在变为基因 3,Child2 存在两个重复基因 3,与 Child1 基因转变顺序相反,这时将其通过映射关系将 Child1 中的重复基因 1 转变为基因 6,将 Child2 中的重复基因 3 转变成基因 6,可以看到第三步 Mid1 存在两个重复基因 6, Mid2 也存在两个重复基因 6,再通过映射关系将 Mid1 中的重复基因 6 转变为基因 3, Mid2 中的重复基因转变为基因 1。以此类推至没有重复基因为止,即图 2 中第四步 Final1 和 Final2 所示。最后所有重复的基因都会经过映射转变为另一个基因,保证形成的新子代基因无重复基因。

Parent1	1	2	3	4	5	6	7	8	9
Parent2	5	4	6	9	2	1	7	8	3

Child1	1	2	6	9	2	1	7	8	9
Child2	5	4	3	4	5	6	7	8	3

1 ↔ 6 ↔ 3  
2 ↔ 5  
9 ↔ 4

Mid1	6	5	6	9	2	1	7	8	9
Mid2	2	9	3	4	5	6	7	8	6

Final1	3	5	6	9	2	1	7	8	4
Final2	2	9	3	4	5	6	7	8	1

图 1 部分映射交叉

### 2.1.6 变异

交叉操作后形成的新个体具有一定的基因突变概率。与选择操作一样,这种操作是基于概率的,这就变成了变异概率  $p_m$ 。一般来说,变异概率设置得很小,一般  $p_m \leq 0.05$ 。

对交叉后代集中每个后代的每一位,产生一个随机数

$rand \in [0, 1]$ ,若  $rand < p_m$  则将该位取反,否则该位不变。

改进遗传算法的流程图如图 2 所示。

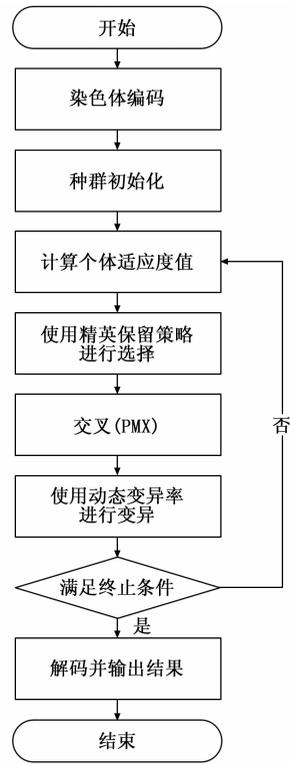


图 2 改进遗传算法过程

## 2.2 启发式算法和改进遗传混合算法

对于启发式算法,本文选取 NEH 快速启发式算法,该算法为快速求解生产调度问题的算法。由于遗传算法具有可扩展性,容易与其他算法结合的优点。因此当干扰事件到来时,采用启发式算法与改进遗传混合算法,改进遗传算法得出的最优结果作为初始解传入快速启发式算法中,作为快速启发式算法的初始解,将紧急加工工件依次插入到每个位置上,得到不同位置的解,找出最小惩罚的调度工序的位置,可以快速解决干扰事件带来的影响,而且两种算法的优点都体现出来,不仅运行时间短,而且得到合适解与对应的调度工序。

## 2.3 启发式算法和改进遗传混合算法流程

- 1) 利用 MATLAB 软件中的随机排序函数“randperm”产生染色体的初始种群,每个染色体代表一个生产安排;
- 2) 生成与染色体相对应的条件最优调度;
- 3) 基于优化目标对它们进行评估并确定是否终止;
- 4) 如果要继续,则基于前一个染色体,通过变异和交叉并返回到第二个步骤来生成新的染色体群;否则,结束计算并输出由步骤 3 选择的最优调度。

当迭代次数达到调度器给定的极限时,优化终止,得到最合适的调度工序。

- 5) 先提取出调度工序,在上述步骤完成后,当有紧急加工工件即在 RT 时刻有新工件到来,将 RT 时刻之后的未加工工件的工序提取;

6) 保持原有调度工序, 将紧急工件依次插入所有位置, 每次找出最优解的位置, 逐次进行到最后一个紧急加工工件, 最终得出最优调度。

### 3 实验结果及分析

程序运行平台是 Windows10 操作系统中的 MATLAB 软件, 经过调参可得各种参数如下: 种群规模大小取为  $NIND=50$ , 迭代次数取为  $MAXGEN=200$ , 改进遗传算法采用部分映射交叉方法进行交叉操作, 采用换位法进行变异操作, 交叉率和变异率分别取作:  $P_c=0.85$ ,  $P_m=0.02$ 。以下面 5 种规模来验证改进遗传混合算法的性能。本仿真实验包含  $10*6$ 、 $20*6$ 、 $30*6$ 、 $50*6$ 、 $70*6$  五种问题规模, 每个规模在相同条件下运行 30 次。通过模拟紧急加工工件干扰下的重调度情景, 统计了合同违约金与仓储成本 (Earliness/Tardiness, E/T) 的实施方案, 在相同运行时间下与不同运行时间下的不同方案, 在不同大小规模下的方案。

表 1 为部分结果, 由表 1 可得,  $10*6$ ,  $20*6$ ,  $30*6$ ,  $50*6$ ,  $70*6$  的最优解分别为 1 263.9, 2 024.8, 1 909.5, 9 369.1, 22 462, 紧急订单规模分别取  $4*6$ ,  $10*6$ ,  $14*6$ ,  $20*6$ ,  $30*6$ 。结果分析为在小规模如  $10*6$  与  $20*6$  规模情况下, 蚁群算法结果较好, 在  $30*6$ ,  $50*6$ ,  $70*6$  的其他大规模情况下, 改进遗传算法结果较好, 但所需时间远远大于提出的改进遗传混合算法的时间, 以  $70*6$  为例, 蚁群算法与改进遗传算法单次运行耗时 50 s 左右, 而启发式算法和改进遗传混合算法单次运行耗时 15 s 左右, 时间差距比较大, 因此, 对 3 种算法在相同运行时间下进行比较。

表 1 不同运行时间下合同违约金与仓储成本 E/T 的比较

		蚁群算法	启发式算法和改进遗传混合算法	改进遗传算法
$10*6$	最小值	1 263.9	1 315.2	1 263.9
	均值	1 265.68	1 595.7	1 266.5
$20*6$	最小值	2 024.8	3 819.9	2 494.9
	均值	3 172.51	4 964.8	2 756
$30*6$	最小值	2 338.8	4 289.9	1 909.5
	均值	3 163.66	5 944.2	3 133.7
$50*6$	最小值	15 054.5	20 068	9 369.1
	均值	16 789.2	23 783	10 604
$70*6$	最小值	37 142.1	37 816	22 462
	均值	40 977.9	44 895	23 389

运行结果如表 2 所示,  $10*6$ ,  $20*6$ ,  $30*6$ ,  $50*6$ ,  $70*6$  的最优解分别为 1 363.9, 4 155.6, 3 875, 17 122, 28 644, 结果分析为在相同运行时间下, 在  $10*6$  与  $20*6$  规模情况下, 蚁群算法结果较好, 对于  $30*6$ ,  $50*6$ ,  $70*6$  等其余规模情况下, 启发式算法和改进遗传混合算法结果最优。故而, 在同运行时间下对于大规模情况来说, 本文提出的启发式算法和改进遗传混合算法结果最优。

表 2 同运行时间下合同违约金与仓储成本 E/T 的比较

		蚁群算法	启发式算法和改进遗传混合算法	改进遗传算法
$10*6$	最小值	1 363.9	1 647.9	1 387
	均值	1 673.97	1 647.9	1 700.6
$20*6$	最小值	4 155.6	4 500.8	4 193.5
	均值	4 784.4	4 971.7	5 300.9
$30*6$	最小值	6 021.2	3 875	6 983.4
	均值	7 519.7	4 354.9	8 033.9
$50*6$	最小值	21 549.8	17 122	23 686
	均值	23 498.6	20 088	28 843
$70*6$	最小值	55 631.5	28 644	55 771
	均值	61 781.3	33 256	63 793

### 4 结束语

本文以合同违约金与仓储成本最小化为优化目标的流水车间重调度问题。采用精英保留策略以及动态变异概率对遗传算法进行了相应的改进。使算法具有较强的寻优能力。同时, 考虑到随着个体数目变大遗传算法运行时间过长的缺陷, 以及遗传算法容易与其他算法结合的优点, 提出了启发式算法与改进遗传算法的混合算法, 算法包含了两种算法的优点, 在前期具有较强的寻优能力, 随着个体数目变大运行时间也不会变慢。

#### 参考文献:

- [1] 王 凌. 车间调度及其遗传算法 [M]. 北京: 清华大学出版社, 2003.
- [2] 何 伟. 机器故障下柔性 JobShop 调度研究 [D]. 重庆: 重庆大学, 2012.
- [3] Chan W T, Hu H. Production Scheduling for Precast Plants using a Flow Shop Sequencing Model [J]. Journal of Computing in Civil Engineering, 2002, 16 (3): 165-174.
- [4] Benjaoran V, Dawood N, Hobbs B. Flowshop scheduling model for bespoke precast concrete production planning [J]. Construction Management and Economics, 2005, 23 (1): 93-105.
- [5] Ko C H, Wang S F. GA-based decision support systems for precast production planning [J]. Automation in Construction, 2010, 19 (7): 907-916.
- [6] Zhiliang M, Zhitian Y. Flowshop Scheduling Model of Multiple Production Lines for Precast Production [A]. Cib W78 Conference [C]. 2016.
- [7] Zhiliang M, Zhitian Y, Shilong L, et al. Optimized rescheduling of multiple production lines for flowshop production of reinforced precast concrete components [J]. Automation in Construction, 2018, 95: 86-97.
- [8] 曲 媛, 杨晓伟. 关于流水车间调度问题的综述 [J]. 中小企业科技, 2007 (8): 24-25.
- [9] 周 珍, 车阿大. 自动化制造单元调度算法综述 [J]. 计算机应用研究, 2010, 27 (6): 2001-2005.
- [10] 李海宁, 孙树栋, 郭 杰. 基于混合整数规划的高级计划排程方法研究 [J]. 制造业自动化, 2012, 34 (18): 59-62.