

Stability Rescheduling of Virtual Cell Based on Backtracking Search Algorithm under New Order Insertion

Wenmin Han^{1,a*}, Junjie Fan^{2,b}, Chenglong You^{3,c}

¹Institute of Economics and Management, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu, China

²Institute of Economics and Management, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu, China

³Institute of Economics and Management, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu, China

^{a*} wlmh63@163.com

^b 327582654@qq.com

^c 2392246491@qq.com

ABSTRACT

In view of the frequent adjustment of virtual cell scheduling plan brought about by the arrival of new orders, and considering the impact of rescheduling on production stability, this paper proposes an insertion rescheduling strategy based on slack time. The slack time is subdivided into idle time, partial Relaxation time and total relaxation time. Under the guidance of this strategy, considering the constraints of order delivery time, a nonlinear integer programming model with the goal of minimizing the maximum completion time and the minimum instability coefficient is constructed, and is solved by introducing a backtracking search algorithm of non-dominated sorting. Finally, the feasibility of the scheme proposed in this paper is verified through a numerical example study.

Keywords: virtual cell; rescheduling; slack time; backtracking search algorithm

新订单插入下基于回溯搜索算法的虚拟单元稳定性重调度

韩文民^{1,a*} 范钧杰^{2,b} 尤程龙^{3,c}

¹ 江苏科技大学经济与管理学院, 镇江, 江苏, 中国

² 江苏科技大学经济与管理学院, 镇江, 江苏, 中国

³ 江苏科技大学经济与管理学院, 镇江, 江苏, 中国

^{a*} wlmh63@163.com

^b 327582654@qq.com

^c 2392246491@qq.com

摘要

针对新订单到达带来的虚拟单元调度计划频繁调整问题, 考虑到重调度对生产稳定性的影响, 本文提出了一种基于松弛时间的插入重调度策略, 将松弛时间细分为空闲时间、部分松弛时间和总松弛时间。在该策略指导下, 考虑到订单交货期的约束, 构建了以最小化最大完工时间、不稳定性系数最小为目标的非线性整数规划模型, 并通过引入快速非支配排序和拥挤度距离计算的回溯搜索算法予以求解。最后, 通过算例研究, 验证了本文所提方案的可行性。

关键词: 虚拟单元; 重调度; 松弛时间; 回溯搜索算法

1. 引言

虚拟单元作为一种先进的生产组织形式,其相较于传统单元制造方式而言,克服了车间布局不灵活、机器利用率低等缺点^[1],能够充分利用现有生产资源,灵活调整单元结构,快速适应多品种、小批量的生产需求^[2]。然而,在实际生产过程中存在着许多扰动因素,如新订单任务的插入、旧工序任务的取消、加工机器的故障等^[3]。为有效应对随机扰动事件的发生,保证生产系统的稳定性与连续性,需要对原生产作业计划做出及时调整,即重调度。

目前,针对生产车间重调度问题的研究主要集中于重调度方法的优化和重调度性能的评估等方面^[4]。近年来,国内外学者关于重调度模型的优化与算法的改进等问题做了大量研究。Al-Hinai 等^[5]针对具有随机设备故障的柔性作业车间调度问题,设计了可生成预测调度的两阶段混合遗传算法。基于类似的方法,Nouiri 等^[6]设计了一种两阶段粒子群优化算法,以解决仅有一个机器故障下的柔性作业车间调度问题。He 等^[7]考虑到单策略重调度在提高鲁棒稳定性方面存在局限性,提出了一种基于路径变换和右移的空闲时间插入策略。Wang 等^[8]设计了一个具有特殊染色体编码的遗传算法,用于解决能够适应机器中断的柔性作业车间调度问题,以更真实地反映现实制造环境。Ahmadi 等^[9]采用基于帕累托的非支配排序遗传算法求解具有机器故障的柔性作业车间调度问题,将最小化完工时间和提高稳定性结合起来。Gao 等^[10]将带有新作业插入的柔性作业车间调度分为两个阶段:初始调度和新作业插入时的重调度,构建了以最小化完工时间、提前和迟到的平均时间、机器工作量为目标的数学规划模型。

新订单插入是虚拟单元生产方式下最为常见的扰动事件之一,若不对其进行及时处理,则有可能导致在制品堆积、订单延误等问题。因此,针对此类扰动事件进行虚拟单元重调度是有必要的^[11]。

一般情况下,为尽可能满足完工时间的要求,车间计划层会在新订单任务到达后,通过完全重调度策略,将还未开始的工序任务与新订单任务进行重新安排。这种方式下所获得的重调度方案在完工时间方面具有一定的优势,但随之产生的重调度成本也相对较大^[12]。此外,完全重调度策略下所生成的计划方案通常与原计划方案相差较大,车间执行层若不能及时做出反应,则可能导致生产系统出现混乱,阻碍生产计划的有效实施。

因此,针对新订单插入下考虑生产稳定性的虚拟单元重调度问题,本文提出了一种基于松弛时间的插入重调度策略,将松弛时间细分为空闲时间、部分松弛时间和总松弛时间。在该策略指导下,考虑到订单交货期的约束,构建以最小化完工时间、不稳定系数最小为目标的非线性整数规划模型,并通过改进的回溯搜索算法予以解决。算例研究表明,该方法在保证

生产稳定性以及交货期的同时,能够有效缩短完工时间、降低不稳定系数、提升机器平均利用率。

2. 新订单插入下的虚拟单元重调度

2.1. 问题描述

针对新订单插入下的虚拟单元重调度问题,可描述为:在生产车间中,有一组订单任务($i=1,2,L,N$)在多台机器上进行加工,每个订单任务均由多道工序组成,各道工序可在一台或多台具有不同处理时间的机器上进行加工。当新订单到达时,需将原生产作业计划中未开始的工序任务与新订单中的各项加工任务进行综合考虑,以实现虚拟单元重调度。为便于重调度模型的构建,现做出如下假设:(1)各订单任务是相互独立的,它们之间没有优先顺序;(2)每个订单任务都由一组具有先后顺序的工序组成,即后道工序必须在前道工序完成后才能开始;(3)每台机器一次只能进行一道工序的加工任务;(4)当新订单到达时,原调度方案中正在进行的工序任务不可中断;(5)已知各道工序在机器上的处理时间(包括装卸时间、运输时间和加工时间)。

2.2. 重调度模型构建

针对新订单插入下虚拟单元重调度模型中的相关符号参数及变量做出如下定义:

符号:

i ——表示订单任务序号, $i=1,2,L,N$;

j ——表示各订单任务的工序号, $j=1,2,L,n_i$;

k ——表示机器序号, $k=1,2,L,K$;

p ——表示机器上进行工序操作的序号;

参数:

N ——表示订单任务数量;

n_i ——表示完成订单任务 i 的工序数量;

K ——表示机器数量;

d_k ——表示机器 k 上完成工序操作的数量;

O_{ij} ——表示订单任务 i 的第 j 道工序;

P_{ijk} ——表示订单任务 i 的第 j 道工序在机器 k 上的处理时间;

S_{kp} ——表示机器 k 上进行第 p 个工序操作的开始时间;

C_{kp} ——表示机器 k 上进行第 p 个工序操作的完成时间;

B_{ij} ——表示订单任务 i 的第 j 道工序的开始时间;

F_{ij} ——表示订单任务 i 的第 j 道工序的完成时间;

C_i ——表示订单任务 i 的完成时间;

C_k ——表示机器 k 的完成时间;

G_i ——表示订单任务 i 的交货期;

L ——表示一个无穷大的正数;

决策变量:

X_{ijk} ——表示当订单任务 i 的第 j 道工序在机器 k 上加工时为 1, 否则为 0;

Y_{ijkp} ——表示当订单任务 i 的第 j 道工序为机器 k 上处理的第 p 个工序时为 1, 否则为 0。

由于重调度过程中涉及两个阶段, 为便于区分, 本文将重调度阶段的参数与变量的右上角加上符号“*”, 例如 B_{ij}^* 表示重调度后工序 O_{ij} 的开始时间。本文以最小化完工时间、不稳定系数最小为目标, 构建了新订单插入下虚拟单元重调度模型。

2.2.1 目标函数

(1) 最大完工时间最小

$$\min f_1 = \max_{1 \leq k \leq K} C_k \quad (1)$$

(2) 不稳定系数最小

$$\min f_2 = \frac{\sum_{i=1}^N \sum_{j=1}^{n_i} |B_{ij}^* - B_{ij}|}{\sum_{i=1}^N n_i} \quad (2)$$

2.2.2 约束条件

$$\sum_{k=1}^K X_{ijk} = 1, i = 1, 2, L, N, j = 1, 2, L, n_i \quad (3)$$

$$S_{k(p+1)} - C_{kp} \geq 0, k = 1, 2, L, K, p = 1, 2, L, d_k - 1 \quad (4)$$

$$B_{i(j+1)} - F_{ij} \geq 0, i = 1, 2, L, N, j = 1, 2, L, n_i - 1 \quad (5)$$

$$B_{ij} \leq S_{kp} + L(1 - Y_{ijkp}), i = 1, 2, L, N, j = 1, 2, L, n_i, k = 1, 2, L, K, p = 1, 2, L, d_k \quad (6)$$

$$B_{ij} \geq S_{kp} - L(1 - Y_{ijkp}), i = 1, 2, L, N, j = 1, 2, L, n_i, k = 1, 2, L, K, p = 1, 2, L, d_k \quad (7)$$

$$F_{ij} \leq C_{kp} + L(1 - Y_{ijkp}), i = 1, 2, L, N, j = 1, 2, L, n_i, k = 1, 2, L, K, p = 1, 2, L, d_k \quad (8)$$

$$F_{ij} \geq C_{kp} - L(1 - Y_{ijkp}), i = 1, 2, L, N, j = 1, 2, L, n_i, k = 1, 2, L, K, p = 1, 2, L, d_k \quad (9)$$

$$\sum_{i=1}^N \sum_{j=1}^{n_i} Y_{ijkp} = 1, k = 1, 2, L, K, p = 1, 2, L, d_k \quad (10)$$

$$\sum_{p=1}^{d_k} Y_{ijkp} = X_{ijk}, i = 1, 2, L, N, j = 1, 2, L, n_i, k = 1, 2, L, K \quad (11)$$

$$C_i \leq G_i, i = 1, 2, L, N \quad (12)$$

约束(3)表示任何一个工序在任意时刻都只能分配给一台机器进行加工; 约束(4)表示机器 k 上第 $(p+1)$ 个工序操作只有在该机器上第 p 个工序操作完成后才能开始; 约束(5)表示订单任务 i 的第 $(j+1)$ 道工序只有在该订单任务的第 j 道工序完成后才能开始; 约束(6)和(7)衔接每个工序操作相对应的开始时间; 约束(8)和(9)衔接每个工序操作相对应的完成时间; 约束(10)保证机器 k 在任何时刻都只能进行一个工序操作; 约束(11)保证所有工序都被分配到相对应的机器上。约束(12)保证各项订单任务在交货期内完成。

2.3 基于松弛时间的插入重调度策略

针对虚拟单元生产过程中存在新订单任务插入的情形, 研究了考虑生产稳定性, 即尽量维持原生产作业计划安排的虚拟单元重调度问题。为尽可能降低重调度对生产稳定性的影响和保证订单交货期的同时, 本文提出了一种基于松弛时间的插入重调度策略, 将松弛时间细分为空闲时间、部分松弛时间和总松弛时间。

该策略考虑了各工序操作的空闲时间以及部分松弛时间和总松弛时间。工序操作的空闲时间是指处理该工序操作的机器在开始该工序操作前的空闲时间; 工序操作的部分松弛时间是指在不影响后续操作的前提下, 可延迟该工序操作开始时间的最大时间间隔; 工序操作的总松弛时间是指在不增加计划完工时间的情况下, 可通过延迟该工序及其后续工序操作开始时间的最大总时间间隔。两个松弛时间的区别在于, 部分松弛时间只考虑单个工序操作的右移, 而总松弛时间考虑了该工序及其后续工序操作的右移。各工序操作的部分和总松弛时间是相对于参考完工时间值计算的, 该值对应于仅考虑可用空闲时间间隔进行调度获得的最大完工时间。

该策略的具体流程如下:

①新订单到达后, 依据新订单任务的工艺要求, 将各道工序的加工任务临时分配至相应有能力的机器上;

②计算在新订单到达时刻后的各道工序开始前的可用空闲时间、部分松弛时间和总松弛时间

③从最早时刻的空闲时间间隔开始，检查新订单任务的第一道工序是否可以插入到该空闲时间间隔中；

④如果新工序的处理时间小于空闲时间，则可将工序分配给相应机器进行加工，并转至步骤 8；

⑤如果新工序的处理时间小于空闲时间与部分松弛时间之和，则通过右移原工序来实现新工序的插入，并转至步骤 8；

⑥如果新工序的处理时间小于空闲时间与总松弛时间之和，则通过右移原工序及其作业、机器后续工序，来实现新工序的插入，并转至步骤 8；

⑦如果新工序的处理时间大于空闲时间与总松弛时间之和，则将新工序分配至原生产作业计划完成之后；否则，重新选择新的机器，转至步骤 3；

⑧如果新订单任务中所有工序都被分配完成，则停止；否则，继续分配新订单任务的下一道工序，转至步骤 2。

为便于理解，下面给出一个示例：

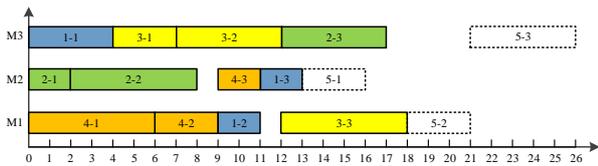


图 1 重调度前甘特图

图 1 中的虚线框代表新订单任务中各道工序的临时分配，其他为初始调度方案下的工序任务安排。由于没有合适的空闲时间间隔，新订单任务的各道工序被临时安排在相应机器上的最后一道工序操作之后。将图 1 中的完工时间（此处为 26）作为参考完工时间，进而计算各道工序的部分松弛时间和总松弛时间。假设新订单任务在时刻 8 到达，表 1 列出了新订单任务到达时未开始处理的工序及其对应的空闲时间和松弛时间。

表 1 重调度前各工序的空闲时间及其松弛时间

工序	前一空闲时间的开始时间	空闲时间间隔	部分松弛时间	总松弛时间
O_{43}	8	1	0	13
O_{12}	9	0	0	9
O_{13}	11	0	13	13
O_{33}	11	1	8	8
O_{23}	12	0	9	9

在基于松弛时间的插入重调度策略下，重调度后的甘特图如下图 2 所示：

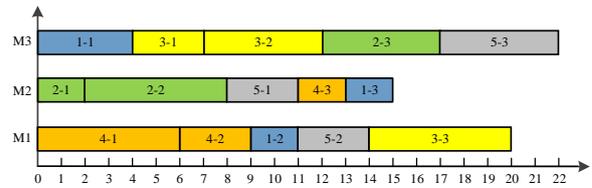


图 2 重调度后甘特图

3. 算法设计

回溯搜索算法（Backtracking Search Algorithm, BSA）是一种用于解决复杂数值优化问题的进化算法，最初由 Civicioglu 于 2013 年开发。它是一种双种群算法，通过比较新老种群为搜索过程提供方向。与其他搜索算法相比，BSA 仅使用单个控制参数，结构相对简单，适用于广泛的优化问题，并能够快速有效的获得最佳结果。

这种算法的优点在于它存在“记忆”。这种“记忆”能够存储上一代种群的信息，用于引导搜索过程。BSA 的“记忆”使其能够利用以前试验中获得的经验，产生新的试验群体。该算法利用独特的交叉和变异操作来提高其探索和开发能力。

考虑到所构模型涉及到多目标优化问题，单纯的回溯搜索算法无法对其进行求解。因此，本文通过引入快速非支配排序和拥挤距离的计算对该算法予以改进，主要过程如下所示。

3.1. 编码和解码

对于工序码，通过优先约束构造操作序列向量，后续操作由各自订单任务编号的重复来表示。在一层工序码中，订单任务编号重复的次数代表其工序号。对于机器码，通过操作序列中相应的机器索引来表示，一个操作只能由少数几个有能力的机器进行处理。编码方案如图 3 所示：

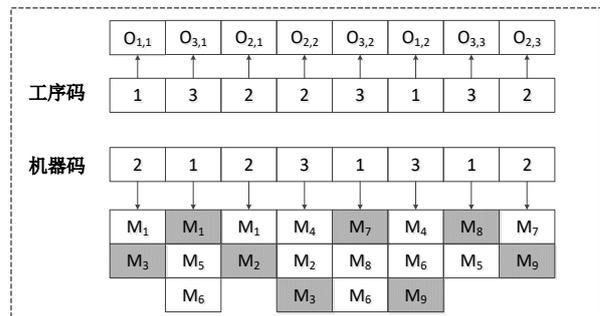


图 3 编码方案

3.2. 变异操作

为提高算法的搜索能力，通过改变搜索幅度因子 F 来动态控制变异。它在每次迭代开始时随机填充 0 和 1。通过确定 F 中 1 的数量 a 来控制变异率。 a 由以下公式确定：

$$a = \lambda_{\min} + (1 - \frac{n}{N_{\max}})(\lambda_{\max} - \lambda_{\min}) \quad (12)$$

其中, $\lambda_{\max}, \lambda_{\min}$ 分别为基于实例大小 F 中所允许 1 的个数的最大值和最小值; n 为当前迭代次数; N_{\max} 为最大迭代次数。

3.3. 交叉操作

交叉操作本质上是通过探索解空间中新的领域来改善解。一个设计良好的交叉操作应当是以有效的方式在后续种群之间交换信息^[13]。本文采用的交叉操作如图 4 所示。

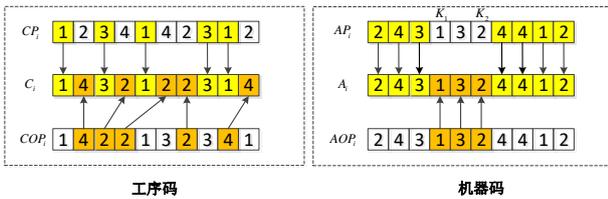


图 4 工序码和机器码的交叉操作

3.4. 转移准则

在改进的回溯搜索算法中, 本文通过一个转移准则来动态控制历史种群 OP 的更新。首先, 生成一个介于 0 和 1 之间的随机数 r , 将其与指数项进行比较。如果 r 的值小于指数项, 则更新历史种群 OP , 否则保留 OP 。随着迭代的进行, 这一准则可增强算法的收敛性。转移准则如下:

$$r < \exp(n / N_{\max} - 1) \quad (13)$$

其中, n 为当前迭代次数, N_{\max} 为最大迭代次数。

4. 算例研究

本文采用的算例数据如下: 已知该车间内存在 8 台机器, 需要完成 8 种类型的订单任务, 每种订单任务均由 3 或 4 道工序组成。具体信息如表 2、3 所示。

表 2 初始订单加工任务信息

订单任务	工序	可选机器	加工批量	加工时间 (h)	交货期
P ₁	P ₁₋₁	M ₁ /M ₅ /M ₆	6	6/7/5	180
	P ₁₋₂	M ₂ /M ₃		7/8	

表 4 初始调度方案

订单任务	工序的加工机器 (开始时间-结束时间, 单位: h)			
	工序 1	工序 2	工序 3	工序 4
P ₁	M ₆ (0-30)	M ₃ (30-78)	M ₄ (78-120)	M ₈ (125-167)
P ₂	M ₆ (30-58)	M ₁ (78-86)	M ₇ (86-98)	M ₄ (120-140)
P ₃	M ₁ (0-36)	M ₆ (58-91)	M ₃ (91-115)	M ₅ (132-159)
P ₄	M ₅ (0-75)	M ₅ (75-125)	M ₁ (125-160)	/
P ₅	M ₃ (0-18)	M ₆ (91-99)	M ₇ (99-115)	/
P ₆	M ₂ (0-36)	M ₁ (36-78)	M ₅ (78-132)	M ₃ (132-174)

P ₁	P ₁₋₃	M ₄ /M ₇	4	7/6	160
	P ₁₋₄	M ₈		7	
P ₂	P ₂₋₁	M ₂ /M ₆	4	6/7	160
	P ₂₋₂	M ₁ /M ₃		2/1	
	P ₂₋₃	M ₅ /M ₇		4/3	
	P ₂₋₄	M ₄ /M ₈		5/4	
P ₃	P ₃₋₁	M ₁ /M ₂	3	12/11	180
	P ₃₋₂	M ₆ /M ₇		11/10	
	P ₃₋₃	M ₃ /M ₄ /M ₈		8/7/6	
	P ₃₋₄	M ₅		9	
P ₄	P ₄₋₁	M ₅ /M ₆	5	15/14	180
	P ₄₋₂	M ₂ /M ₈		11/10	
	P ₄₋₃	M ₁ /M ₄ /M ₇		7/8/8	
P ₅	P ₅₋₁	M ₁ /M ₂ /M ₃	2	7/8/9	160
	P ₅₋₂	M ₅ /M ₆		5/4	
	P ₅₋₃	M ₄ /M ₇ /M ₈		8/8/9	
P ₆	P ₆₋₁	M ₂ /M ₄	6	6/7	180
	P ₆₋₂	M ₁ /M ₆		7/8	
	P ₆₋₃	M ₅ /M ₈		9/8	
	P ₆₋₄	M ₃ /M ₇		7/8	
P ₇	P ₇₋₁	M ₁ /M ₃ /M ₄	7	6/7/7	160
	P ₇₋₂	M ₂ /M ₅		7/6	
	P ₇₋₃	M ₆ /M ₇ /M ₈		7/8/6	
P ₈	P ₈₋₁	M ₃ /M ₇	3	4/5	160
	P ₈₋₂	M ₁ /M ₄		3/4	
	P ₈₋₃	M ₂ /M ₅		5/4	

表 3 新订单加工任务信息

订单任务	工序	可选机器	加工批量	加工时间 (h)	到达时刻
P ₉	P ₉₋₁	M ₁ /M ₂	3	4/3	30
	P ₉₋₂	M ₆ /M ₇		11/10	
	P ₉₋₃	M ₃ /M ₄ /M ₈		8/7/6	
	P ₉₋₄	M ₅		9	
P ₁₀	P ₁₀₋₁	M ₁ /M ₃ /M ₄	4	6/7/7	80
	P ₁₀₋₂	M ₂ /M ₅		7/6	
	P ₁₀₋₃	M ₆ /M ₇ /M ₈		7/8/6	

针对初始订单中的加工任务, 本文以最小化完工时间为目标, 通过遗传算法生成初始调度方案, 如表 4 所示。而针对新订单插入下的虚拟单元重调度问题, 在基于松弛时间的插入重调度策略指导下, 考虑到订单交货期的约束, 以最小化完工时间、不稳定系数为目标, 通过改进的回溯搜索算法获得最终的重调度方案, 如表 5 所示。

P ₇	M ₄ (0-49)	M ₂ (49-98)	M ₆ (99-148)	/
P ₈	M ₇ (0-15)	M ₄ (49-61)	M ₂ (98-113)	/

表 5 新订单插入下的重调度方案

订单任务	工序的加工机器 (开始时间-结束时间, 单位: h)			
	工序 1	工序 2	工序 3	工序 4
P ₁	M ₆ (0-30)	M ₃ (30-78)	M ₄ (78-120)	M ₈ (125-167)
P ₂	M ₆ (30-58)	M ₁ (78-86)	M ₇ (86-98)	M ₄ (120-140)
P ₃	M ₁ (0-36)	M ₆ (58-91)	M₅(103-127)	M ₅ (132-159)
P ₄	M ₅ (0-75)	M ₈ (75-125)	M ₁ (125-160)	/
P ₅	M ₃ (0-18)	M ₆ (91-99)	M ₇ (99-115)	/
P ₆	M ₂ (0-36)	M ₁ (36-78)	M ₅ (78-132)	M ₃ (132-174)
P ₇	M ₄ (0-49)	M ₂ (49-98)	M ₆ (99-148)	/
P ₈	M ₇ (0-15)	M ₄ (49-61)	M ₂ (98-113)	/
P ₉	M ₂ (36-48)	M ₇ (49-79)	M ₃ (79-103)	M ₅ (159-186)
P ₁₀	M ₁ (86-110)	M ₂ (113-141)	M ₆ (148-176)	/

通过比较分析发现, 本文所提方案较临时安排而言, 在保证订单交货期的同时, 在完工时间方面缩短了 22.5%, 机器平均利用率方面提升了 29.03%; 而较完全重调度策略而言, 在不稳定系数方面降低了 94.03%。

5. 结论

本文针对新订单插入下的虚拟单元重调度问题展开研究。考虑到生产稳定性、订单交货期等因素, 构建了以最小化完工时间、不稳定系数最小为目标的非线性整数规划模型, 并通过改进的回溯搜索算法予以求解; 为尽可能降低重调度对原生产作业计划的影响, 本文提出了一种基于松弛时间的插入重调度策略, 并将松弛时间细分为空闲时间、部分松弛时间和总松弛时间。算例研究表明, 该方法在保证生产稳定性以及交货期的同时, 能够有效缩短完工时间、降低不稳定系数、提升机器平均利用率。

REFERENCES

[1] Dhanraj P T, B. E N, S. S M. A novel algorithm for virtual cellular manufacturing considering real life production factors[J]. Int. J. of Services and Operations Management, 2015,20(2).

[2] Han Wenmin, Yu Yi, Gao Longlong, et al. Reconstruction of Inheritance Virtual Unit Driven by New Order and Time Window Hybrid [J]. Computer Integrated Manufacturing Systems, 2018,24(05):1317-1326.

[3] Xixing L, Zhao P, Baigang D, et al. Hybrid artificial bee colony algorithm with a rescheduling strategy for solving flexible job shop scheduling problems[J]. Computers & Industrial Engineering, 2017,113.

[4] Han Wenmin, Zhu Tao, Li Zhengyi, et al. Virtual unit rescheduling drives decision-making when orders arrive one after another [J]. Chinese Management Science, 2017,25(12):126-137.

[5] Nasr A, T. Y E. Robust and stable flexible job shop scheduling with random machine breakdowns using a hybrid genetic algorithm[J]. International Journal of Production Economics, 2011,132(2).

[6] Maroua N, Abdelghani B, Abderrazak J, et al. Two Stage particle swarm optimization to solve the Flexible job shop predictive scheduling problem considering possible machine breakdowns[J]. Computers & Industrial Engineering, 2017.

[7] Wei H, Di-hua S. Scheduling flexible job shop problem subject to machine breakdown with route changing and right-shift strategies[J]. The International Journal of Advanced Manufacturing Technology, 2013,66(1-4).

[8] Yong M W, Hong L Y, Kai D Q. A novel genetic algorithm for flexible job shop scheduling problems with machine disruptions[J]. The International Journal of Advanced Manufacturing Technology, 2013,68(5-8).

[9] Ehsan A, Mostafa Z, Mojtaba F, et al. A multi objective optimization approach for flexible job shop scheduling problem under random machine breakdown by evolutionary algorithms[J]. Computers and Operations Research, 2016,73.

[10] Gao K Z, Suganthan P N, Tasgetiren M F, et al. Effective ensembles of heuristics for scheduling flexible job shop problem with new job insertion[J]. Computers & Industrial Engineering, 2015,90(DEC.):107-117.

- [11] Gao K, Yang F, Zhou M, et al. Flexible Job-Shop Rescheduling for New Job Insertion by Using Discrete Jaya Algorithm[J]. *IEEE transactions on cybernetics*, 2019,49(5):1944-1955.
- [12] Bao Bo, Zhang Lin, Zhang Bo. Flexible job shop rescheduling method under equipment failure conditions [J]. *Computer Engineering and Applications*, 2018,54(19):248-253.
- [13] Caldeira R H, Gnanavelbabu A, Vaidyanathan T. An effective backtracking search algorithm for multi-objective flexible job shop scheduling considering new job arrivals and energy consumption[J]. *Computers & Industrial Engineering*, 2020,149(prepublish).