

# 动态柔性作业车间优化调度与决策方法

王艳, 丁宇

(江南大学 物联网技术应用教育部工程研究中心, 江苏 无锡 214122)

**摘要:** 针对多目标动态柔性作业车间调度问题, 提出一种改进的多目标差分进化算法进行求解。在差分进化算法中引入自适应交叉变异算子, 提高算法的全局搜索能力; 在选择排序时引入基于免疫学原理的快速非支配排序法, 提高解集的质量。提出改进的 TOPSIS (Technique for Order Preference by Similarity to an Ideal Solution)—G1—熵权综合决策方法。通过 Nash 均衡理论计算得出 G1—熵权法的综合权重; 将综合权重与 TOPSIS 评价体系组合对各调度方案进行评价。通过实验仿真验证了优化调度算法在寻优能力上的优越性以及综合决策方法的有效性。

**关键词:** 柔性作业车间; 动态调度; 多目标差分进化算法; TOPSIS-G1-熵权法; 综合决策

中图分类号: TP391.9      文献标识码: A      文章编号: 1004-731X (2020) 11-2073-11

DOI: 10.16182/j.issn1004731x.joss.20-0732

## Optimal Scheduling and Decision Making Method for Dynamic Flexible Job Shop

Wang Yan, Ding Yu

(Engineering Research Center of Internet of Things Technology Applications Ministry of Education, Wuxi 214122, China)

**Abstract:** For multi-objective dynamic flexible job-shop scheduling, an improved multi-objective differential evolution algorithm is proposed. The adaptive cross-mutation operator is introduced into the differential evolution algorithm to improve its global search capability. The fast non-dominated sorting method based on immunological principles is introduced to improve the quality of the solution set in the selection and sorting. An improved TOPSIS-G1-EVM comprehensive decision-making method is proposed. The comprehensive weight of G1-EVM is calculated by Nash equilibrium theory. The comprehensive weight and TOPSIS evaluation system are combined to evaluate each dispatching scheme. The experimental results show that the optimal scheduling algorithm is superior in the optimization ability and the effectiveness of the comprehensive decision-making method.

**Keywords:** flexible workshop; dynamic scheduling; multi objective differential evolution algorithm; TOPSIS-G1-EVM; comprehensive decision making

## 引言

在实际的车间调度应用中, 动态调度问题与实际生产环境更相近。柔性车间动态调度问题是在柔

性作业车间调度问题和动态作业车间调度问题上发展起来的, 延伸了传统作业车间调度问题的范畴, 同时还能应对实际生产环境下的各种突发的车间干扰情况, 如机器故障、紧急订单插入、交货期变更、订单取消、加工时间变化等, 所以是一种更加广泛、更加契合实际生产情况的调度问题, 已成为调度研究领域的热点之一。因此, 如何更好地解决柔性作业车间的动态调度优化问题以实现企业



收稿日期: 2020-09-22      修回日期: 2020-10-18;  
基金项目: 国家自然科学基金(61973138);  
作者简介: 王艳(1978-), 女, 江苏无锡, 博士, 教授, 长江学者, 研究方向为制造系统能效优化; 丁宇(1996-), 男, 江苏盐城, 硕士, 研究方向为车间多目标优化调度。

车间生产更加高效平稳的运行具有重要的理论意义和工程价值。

国内外学者对于动态车间调度问题展开了研究, Nouri 等<sup>[1]</sup>以能耗及完工时间为目标, 采用 PSO(Particle Swarm Optimization)算法求解机器故障的动态柔性车间调度问题; Zhou 等<sup>[2]</sup>提出了 4 种基于多目标遗传规划的超启发式算法, 求解以平均加权拖延度、最大拖延度和平均流经时间为目标的多目标动态柔性作业车间调度问题; Shen 等<sup>[3]</sup>建立以稳定性为优化目标的动态柔性作业车间调度模型, 提出一种集合启发式动态优化策略的 MOEA(Multi Objective Evolutionary Algorithm)算法进行求解; 陈超等<sup>[4]</sup>设计了一种 GASA(Genetic and Simulated Annealing Algorithm)混合算法, 来求解机器故障下的以平均流经时间和能耗为目标的动态柔性作业车间调度模型; 朱传军等<sup>[5]</sup>以调度稳定性和鲁棒性为优化目标, 采用一种多目标差分进化算法求解多目标柔性作业车间动态调度模型; 刘爱军等<sup>[6]</sup>通过自适应遗传算法, 求解以最小工件拖期惩罚和工件完工时间为目标的多目标柔性作业车间动态调度问题。

虽然国内外学者对于动态柔性作业车间问题(Dynamic Flexible Job Shop Problem, DFJSP)的研究取得了较多的成果, 但往往仅考虑机器故障等动态因素, 且很少同时兼顾能耗以及调度稳定性目标, 针对此问题, 本文考虑机器故障和紧急订单插入动态因素, 并以能耗、完工时间和稳定性为目标构建多目标动态柔性作业车间调度模型, 提出一种自适应的多目标差分进化算法, 并引入基于免疫学原理的快速非支配排序法以及精英保留策略来提高算法的寻优能力。同时, 在柔性车间多目标调度问题求解后往往存在多组最优解, 目前对于此问题的解决方法尚缺, 于是本文提出一种改进的 TOPSIS—G1—熵权综合决策方法, 为不同动态因素影响下的实际车间生产情况筛选出最合理的调度方案。

## 1 动态柔性作业车间问题描述及模型建立

### 1.1 问题描述

本文主要考虑机器故障和紧急订单插入 2 类动态问题对生产车间的影响。当求解静态调度问题获取初始调度方案后, 车间开始正常生产, 在某个时刻某台机器突然故障或紧急订单插入, 生产制造系统能够及时地对已有的生产调度方案在动态问题突发时刻保留已经完成部分的调度方案对剩余部分做出重调度优化, 重新确定剩下未加工的工件工序的加工顺序与加工设备的选择, 尽可能地减少设备故障对车间生产的影响。

动态柔性作业车间调度问题描述为:  $m$  台机器加工  $n$  个工件, 每个工件  $J_i$  由  $h_i$  个工序组成, 工件的每道工序可被多台设备所加工, 且加工时间和能耗会随所选机器的不同而不同。同时, 在加工过程中需要满足以下约束: 零时刻可以加工任何工件; 零时刻任意一台加工设备可以进行加工; 一台加工设备同一时间只能加工一个工件; 同一个工件之间的不同工序之间存在先后顺序约束, 不同的工件之间没有加工顺序约束; 工序在设备上的加工是不可中断的, 除非设备在加工过程中出现故障; 工序在加工过程设备出现故障, 该道工序加工时间作废, 即该道工序重新加工; 不同工件根据实际加工情况加工优先级不同。

### 1.2 模型建立

#### 1.2.1 目标函数

$$f_1 = \min(\sum_{i=1}^n \sum_{j=1}^{h_i} \sum_{k=1}^m UME_{ijk} \cdot t_{ijk} \cdot T_{ijk} + \sum_{i=1}^m UIE_w \cdot T_k) \quad (1)$$

式中:  $f_1$  为总能耗最小化目标函数;  $UME_{ijk}$  为工序  $o_{ij}$  在设备上  $m_k$  的单位时间加工能耗;  $t_{ijk}$  为工序  $o_{ij}$  在设备  $m_k$  上的加工时间;  $UIE_w$  为工件设备  $m_k$  上的单位时间空载能耗;  $T_k$  为设备  $m_k$  上的空载时间;  $T_{ijk}$  为决策变量。

$$f_2 = \min \left( \sum_{i=1}^n \sum_{j=1}^{n_i} p_j |t_{ijk} - t_{ijk'}| + \sum_{k=1}^m p_k \cdot g_k \right) \quad (2)$$

式中:  $f_2$  为调度稳定性最小化目标函数;  $t_{ijk}$  为初始调度方案工序  $o_{ij}$  在设备  $m_k$  上的开始加工时间;  $t_{ijk'}$  为重调度方案工序  $o_{ij}$  在设备  $m_{k'}$  上的开始加工时间;  $p_j$  为重调度使工序开始加工时间变化而引起的惩罚系数;  $p_k$  为初始调度中工序在重调度时刻发生变化引起的机器惩罚系数;  $g_k$  为设备  $m_k$  上从初始调度方案增加和减少的总工序数, 一般取  $p_k=1$ ,  $g_k=1$ 。

$$f_3 = \min \{ \max(t_1, t_2, \dots, t_n) \} \quad (3)$$

式中:  $f_3$  为完工时间最小化目标函数;  $t_i$  为工件  $i$  的最后一道工序完成时刻。

### 1.2.2 约束条件

设备约束

$$h_{ijk} - c_{pqk} > 0 \quad (4)$$

式中:  $h_{ijk}$  为工序  $o_{ij}$  在设备  $m_k$  上的加工开始时间;  $c_{pqk}$  为工序  $o_{pq}$  在设备  $m_k$  上的加工结束时间, 表示一台设备在同一时刻只能对一件工件进行加工。

过程约束

$$c_{ijk} - h_{ijk} = t_{ijk}, T_{ijk} = 1 \quad (5)$$

式中:  $c_{ijk}$  为工序  $o_{ij}$  在  $m_k$  上的结束加工时间;  $h_{ijk}$  为工序  $o_{ij}$  在设备  $m_k$  上的开始加工时间, 表示任意工件的某道工序一旦开始加工, 不可以中断。

工序约束

$$h_{i(j+1)k} \geq c_{ijk}, T_{ijk} = 1 \quad (6)$$

式中:  $h_{i(j+1)k}$  为工序  $o_{i(j+1)k}$  在设备  $m_k$  开始加工时间, 表示同一个工件工序的加工过程是有序的。

## 2 预反应重调度策略

### 2.1 滚动调度窗口

在实际的生产中扰动是不可预测的。所以将整个动态优化调度过程分解为连续的几个静态调度优化区间, 再使用静态调度的方法完成各区间的调度优化, 最终解决整体的动态优化调度问题。根据

实际车间生产情况, 将所有工序定义为 5 类: (1) 已完工加工工序; (2) 正在加工工序; (3) 等待加工工序; (4) 未进入加工队列工序; (5) 返工工序。同时, 定义 3 种工序窗口: (1) 已完工窗口; (2) 调度加工窗口; (3) 等待加工窗口。其中, 已完工窗口存放已完工加工工序, 调度加工窗口存放正在加工工序和等待加工工序, 等待加工窗口存放未进入加工队列工序和返工工序。

### 2.2 工件优先级

本文在周期性和事件驱动混合重调度方法的驱动下, 对加工窗口和等待窗口的所有工件进行安排, 并对工序进行优先级排列。

#### (1) 机器故障

重调度时刻, 正在机器上加的工件继续在该机器上加工, 其优先级最高, 若该机器发生故障, 将其选择正常机器重新加工, 优先级依旧最高; 优先级其次是返工工序, 剩余工序优先级相同。

#### (2) 紧急订单插入

与(1)基本相同, 只是在正在加工工序和返工工序优先级中间插入紧急订单, 所以此时优先级第二的是紧急订单。

### 2.3 DFJSP 重调度步骤

step1: 根据车间初始时刻机器和工件信息, 以能耗及完工时间为目标建立调度优化模型, 利用改进的多目标差分进化算法以及改进的 TOPSIS—G1—熵权法相结合的综合决策方法筛选出最优初始方案。

step 2: 判断是否启动重调度, 若是, 则执行 step 3, 否则继续执行初始调度方案。

step 3: 判断驱动重调度的原因, 若是周期性驱动, 则按照初始调度继续执行, 若是动态事件驱动, 则确定重调度对象, 根据重调度策略建立新的调度优化模型。

step 4: 依据重调度策略建立以能耗、调度稳定性和完工时间为目标的调度优化模型, 利用

step 1 的算法的获取最优重调度方案, 并执行重调度方案。

step 5: 判断是否所有工件都加工完成, 是则退出, 否则转到 step 2 继续执行。

### 3 基于 IMODE 的 DFJSP 多目标优化调度

#### 3.1 多目标差分进化算法

差分进化算法(Differential Evolution Algorithm, DE)是由 storn 等于 1997 年提出的一种具有快速收敛能力的优化算法。本文基于 DE 算法提出一种改进的多目标差分进化算法(Improved Multi-objective Differential Evolution Algorithm, IMODE), 相较于传统的 MODE 算法, 本文算法在初始化时引入 Logistic 混沌迭代方程, 并在交叉变异过程中引入自适应参数来保证算法的全局搜索能力。同时, 在个体筛选过程中采用引入免疫学原理的 NSGA-II 快速非支配排序法, 来提高算法的局部搜索能力, 采用外部存档保存进化过程中的非支配解集。

#### 3.2 编码

柔性车间问题的编码由两部分构成: 工序排序(Operation Sequence, OS)和机器选择(Machines Selection, MS)。其中, OS 与 MS 的长度等于所有工件的工序总数, OS 与 MS 部分的编码是一一对应的。由于本文求解动态车间问题, 所以在重调度时刻编码过程中需要考虑机器是否可用以及工序优先级问题。

如图 1 所示, 初始调度编码中 OS 和 MS 确定。当发生机器故障以及紧急插入动态事件时, OS 和 MS 编码会根据情况发生变化, 机器 2 发生故障阶段, 机器 3 在这一阶段不可用, 故机器加工工序顺序会发生变化; 紧急工件 3 插入时, 此时工序优先级会发生变化, 优化加工工件 3, 工件在机器上的加工顺序因此发生变化。

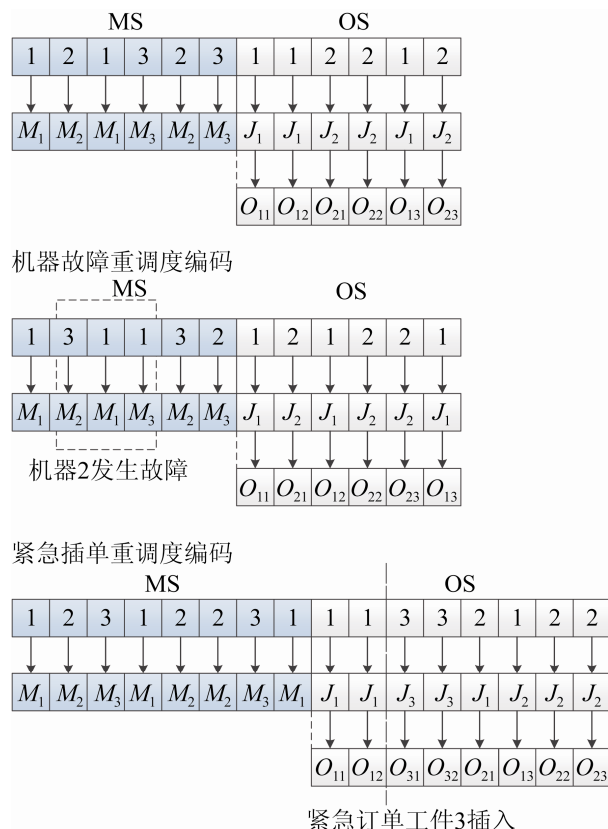


图 1 编码示例  
Fig. 1 Coding examples

#### 3.3 初始化

在初始化时采用随机方式, 可能导致种群出现分布不均问题, 过早陷入局部最优, 所以引入 Logistic 混沌映射迭代方程。

Logistic 混沌映射迭代方程为:

$$z_{t+1} = \mu z_t (1 - z_t) \quad t = 0, 1, \dots, n \quad (7)$$

式中:  $z_t \in (0, 1)$ ;  $\mu$  为控制参数,  $\mu \in (0, 4)$ 。

定义个体维数为  $m$ , 种群规模为  $N$ 。

$$x_{ij} = u_j + z_{ij} (v_i - u_i) \quad j = 0, 1, \dots, M \quad (8)$$

式中:  $v_j$  和  $u_j$  分别为个体第  $j$  维变量取值的上界和下界;  $M$  为每个个体的维数。

随机生成一个  $m$  维的初始序列  $z_1 = (z_{11}, z_{12}, \dots, z_{1m})$  通过公式(7)将  $z_1$  中的每个分量迭代  $N$  次, 得到初始混沌序列矩阵  $z_{ij}$ ; 然后将初始混沌序列的各分量按公式(8)变换到符合所求解问题的变量区间, 完成混沌映射。

### 3.4 自适应策略引入改进

引入自适应策略调整缩放因子  $F$ , 算法初期  $F$  值较大, 有利于种群的多样性以及提高算法的全局搜索能力, 随着迭代次数的增加,  $F$  值逐渐减小, 有利于提高算法后期的收敛速度。同理, 对交叉因子  $CR$  采用自适应调整策略, 初始  $CR$  值较大, 种群交叉程度较大有利于种群的多样性, 随着迭代次数的增加,  $CR$  值逐渐减小, 有利于种群的收敛。

$$F = (F_{\max} - F_{\min}) \cdot \frac{T-t}{T} + F_{\min} \quad (9)$$

式中:  $F_{\max}$  为缩放因子  $F$  的最大值;  $F_{\min}$  为缩放因子  $F$  的最小值;  $T$  为最大迭代次数;  $t$  为当前迭代次数。

$$CR = (CR_{\max} - CR_{\min}) \cdot \frac{t}{T} + CR_{\min} \quad (10)$$

式中:  $CR_{\max}$  为交叉因子  $CR$  的最大值;  $CR_{\min}$  为交叉因子  $CR$  的最小值;  $T$  为最大迭代次数;  $t$  为当前迭代次数。

### 3.5 差分进化选择排序与精英保留策略

#### 3.5.1 选择排序

采用 NSGA-II 算法的快速非支配排序法, 优先选择快速非支配排序中等级小的个体, 当等级相同, 选择拥挤度距离较小的个体。但会出现大部分非支配解出现在等级为 1 的曲面上, 且出现拥挤度为 0 的情况, 于是引入免疫学原理, 当个体拥挤度相同时, 通过计算抗体之间的亲和度, 选择抗体浓度较大的个体, 由此可以增强种群个体间的竞争, 提高算法的局部搜索能力。

定义种群  $G$  由  $N$  个长度为  $m$  的字符串够成, 则亲和度评价算子计算公式为:

种群中基因座为  $j$  的信息熵:

$$H_j(G, N) = -\sum_{i=1}^m p_{ij} \log p_{ij} \quad (11)$$

式中:  $p_{ij}$  为  $m$  中第  $i$  个符号在基因座  $j$  上出现的概率。

种群  $G$  的平均信息熵:

$$H(G, N) = \frac{1}{n} \sum_{i=1}^n H_i(G, N) \quad (12)$$

抗体浓度评价算子计算公式为:

$$aff(i, j) = H\{(i, j), 2\} \quad (13)$$

抗体  $i$  和  $j$  的亲和度为:

$$den(i) = \frac{1}{N} \sum_{j=0}^{N-1} aff(i, j) \quad (14)$$

式中:  $N$  为种群规模;  $aff(i, j)$  为抗体  $i$  和  $j$  的亲和度。

#### 3.5.2 精英保留策略

采用外部存档(External Archive, EA)的精英保留策略, 存放 IMODE 算法全局搜索过程中的非支配解。然后将一个新个体  $p$  和 EA 中的个体进行比较, 若其被 EA 中的一个个体所支配, 则不让其进入外部存档中; 若其和 EA 中的个体互不支配, 则将其加入到外部存档中; 若其支配 EA 中的部分个体, 则将其加入到外部存档中并将其支配的个体去除。

## 4 基于 TOPSIS—G1—熵权法的 DFJSP 多目标最优解的决策体系

主观赋权 G1 法<sup>[7]</sup>和客观赋权熵权法<sup>[8]</sup>各有利弊, 所以将 2 种方法结合, 既能反应主观决策, 又能包含数据本身信息。但传统的综合赋权法—加权平均合理性较差, 于是将博弈论中 NASH 均衡模型引入到综合权重协调中, 在不同权重之间寻找一致或妥协, 使得最终权重与各指标权重之间偏差最小, 最终得到一个均衡的综合权重博弈论; 并利用改进的 TOPSIS 决策体系, 计算各调度方案解与理想最优解及最劣解的之间距离, 最后通过理想贴近度的大小来排序, 为生产车间筛选出一个最合理的调度方案。

### 4.1 主、客观权重法赋权

由于在求解实际柔性作业车间多目标调度问题时, 优化调度结果往往是多组 pareto 解。而针对不同实际生产情况, 各指标的重要程度不尽相同, 当权衡各指标对整体的贡献程度时, 就需要对每个指标合理赋权。

主观赋权采用 G1 法, G1 法是对层析分析法

改进的一种方法,不需要构造判断矩阵,计算量明显减少;同时,其不需要进行一致性检验,更加直观、简洁。

客观赋权采用熵权法,熵权法是根据各指标的变异程度,利用信息熵计算各指标的熵权,再通过熵权对各指标的权重进行修正,从而得出客观的指标权重。

## 4.2 综合赋权法—Nash 均衡法

使用  $n$  种方法对评价体系中各指标进行赋权,由此得到一个基本权重集  $w=\{w_1, w_2, \dots, w_n\}$ , 将其中  $n$  个向量任意线性组合构造一个综合权重集:

$$w = \sum_{k=1}^n \alpha_k w_k^T \quad (\alpha_k > 0) \quad (15)$$

从权重集中挑选出一个最满意的权重  $w^*$ , 采用博弈论模型, 对式中  $n$  个线性组合系数  $\alpha_k$  进行优化, 优化目标是使  $w$  和各个  $w_k$  离差极小化, 因此, 得出如下优化模型:

$$\min \left\| \sum_{j=1}^n \alpha_j w_j^T - w_i^T \right\|_2, i, j = 1, 2, \dots, n \quad (16)$$

根据矩阵的微分性质可以得出式最优化的一阶导数条件为:

$$\sum_{j=1}^n \alpha_j \cdot w_i \cdot w_j^T = w_i \cdot w_i^T, i, j = 1, 2, \dots, n \quad (17)$$

其对应的线性方程组为:

$$\begin{bmatrix} w_1 \cdot w_1^T & w_1 \cdot w_2^T & \cdots & w_1 \cdot w_n^T \\ w_2 \cdot w_1^T & w_2 \cdot w_2^T & \cdots & w_2 \cdot w_n^T \\ \vdots & \vdots & \ddots & \vdots \\ w_n \cdot w_1^T & w_n \cdot w_2^T & \cdots & w_n \cdot w_n^T \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} w_1 \cdot w_1^T \\ w_2 \cdot w_2^T \\ \vdots \\ w_n \cdot w_n^T \end{bmatrix}$$

计算求得  $\{\alpha_1, \alpha_2, \dots, \alpha_n\}$ , 然后对其进行归一化处理

$$\alpha_k^* = \frac{\alpha_k}{\sum_{k=1}^n \alpha_k} \quad (18)$$

求得最终均衡综合权重为:

$$w_k^* = \alpha_k^* \cdot w_k^T \quad (19)$$

## 4.3 改进的 TOPSIS 决策体系

传统的综合评价体系计算过程复杂且不能充分利用客观数据, 本文采用 TOPSIS 法<sup>[9]</sup>来进行评价计算, TOPSIS 法是通过计算每个方案的距离最佳和最差方案的距离, 将理想解的相对近似程度作为综合评价的标准。同时, 引入绝对理想点和投影法, 来解决传统 TOPSIS 法在理想方案的选取和距离计算上存在不足的问题。

step 1: 假设有  $m$  个评价方案,  $n$  个指标, 构造标准化矩阵  $(a_{ij})_{m \times n}$ ,  $(i=1, 2, \dots, m; j=1, 2, \dots, n)$ 。

$$\text{效益指标: } r_{ij} = a_{ij} / \max_i(a_{ij}) \quad (20)$$

$$\text{成本指标: } r_{ij} = \min_i(a_{ij}) / a_{ij} \quad (21)$$

step 2: 构建加权规范矩阵。

$$y_{ij} = r_{ij} \cdot w_j \quad (22)$$

step 3: 确定正负理想点。

$$y^+ = [1, 1, \dots, 1]_n^T, \quad y^- = [0, 0, \dots, 0]_n^T \quad (23)$$

step 4: 计算各方案与正理想点的投影。

$$\text{Pro}(y_i) = \frac{\sum_{j=1}^n y_j^+ y_{ij}}{\sqrt{\sum_{j=1}^n (y_j^+)^2}} \quad (24)$$

step 5: 按照投影值  $\text{Pro}(y_i)$  的大小对方案集进行排序。

## 5 实例验证与分析

为测试 IMODE 算法在多目标柔性作业车间问题(Multi-Objective Flexible Job Shop Problem, MOFJSP)上的求解性能, 进行实验验证。所有实验均由 Matlab2015b 编程实现, 并基于平台: Windows 10, Intel(R) Core(TM) i7-6700HQ CPU 2.60 GHz, 8 G 内存。

### 5.1 IMODE 算法性能测试

为测试 IMODE 算法对多目标排列优化性能, 选取 MK01-MK07<sup>[10]</sup> (7 个标准 FJSP 算例) 作为测试

实例, 每个算例的各道工序在机器上的单位时间加工能耗在(0.5, 1.0)内随机取值。众多研究成果表明, IMODE 的自适应缩放因子  $F$  的最优取值范围为[0.3, 0.6], 自适应交叉因子  $CR$  的最优取值范围为[0.6, 0.9]; 按照往例, 设置 NSGA-II 的交叉概率为 0.9、变异概率为 0.02、模拟二进制交叉系数以及多项式变异系数均为 20; MOGA<sup>[11]</sup>参数按原文配置; 考虑算法运行时间以及一致性问题, 将 3 种算法初始种群规模均设为 100, 最大代数均为 400。本文选择  $\rho_l$ ,  $nd_l$  和  $\ell$  三个指标评估算法性能。

$\rho_l$  为  $nd_l$  的大小与  $|\Delta^*|$  的比值,  $nd_l$  为一种算法的非支配解数量  $\{x \in \Delta_l \cap x \in \Delta^*\}$ , 式中  $\Delta_l$  为一种算法的解集,  $\Delta^*$  为所有算法整合的非支解集。 $\rho_l$  越大说明算法的求解性能更好。

$\ell$  的定义为:

$$\ell(A, B) = \left| \{b \in B \mid \exists a \in A: a \succ b\} \right| / |B| \quad (25)$$

式中:  $||$  为算法解集个数;  $\succ$  为支配;  $a$  与  $b$  为算法  $A$  和  $B$  解集中一个解, 并将  $\ell(A, B)$  和  $\ell(B, A)$  大小进行比较,  $\ell(A, B) > \ell(B, A)$  说明算法  $A$  的求解性能更好。

由表 1~2 结果可知, IMODE 对 7 个算例所得的非劣解的数量更多于对比算法以及对比算法所得的大部分非劣解受 IMODE 的非劣解支配。如图 2~4 所示, 取 MK01, MK04, MK07 的各个算法的 Pareto 解值做直观分析, 可见 IMODE 的大部分的非劣解支配其余算法的非劣解。综上所述, IMODE 算法的寻优能力更加优于算法 NSGA-II 以及 MOGA。

表 1 3 种算法的指标  $\rho_l$  和  $nd_l$  测试结果  
Tab. 1 Test results of 3 algorithms on metrics  $\rho_l$  and  $nd_l$

算例	IMODE	NSGA-II	MOGA
MK01	4, 0.500	1, 0.125	3, 0.375
MK02	6, 0.750	1, 0.125	1, 0.125
MK03	7, 1.00	0, 0.00	0, 0.00
MK04	9, 0.692	2, 0.154	2, 0.154
MK05	8, 0.727	1, 0.091	2, 0.182
MK06	9, 0.900	0, 0.00	1, 0.100
MK07	8, 0.800	0, 0.000	2, 0.200

表 2 3 种算法的指标  $\ell$  测试结果  
Tab. 2 Test results of 3 algorithms on metrics  $\ell$

算例	IMODE (A) $\ell(A, B)$	NSGA-II (B) $\ell(B, A)$	IMODE (A) $\ell(A, C)$	MOGA (C) $\ell(C, A)$
MK01	0.714	0.167	0.429	0.167
MK02	0.875	0.00	0.667	0.00
MK03	1.00	0.00	1.00	0.00
MK04	0.750	0.00	0.714	0.00
MK05	0.800	0.125	0.889	0.125
MK06	1.00	0.00	0.833	0.111
MK07	1.00	0.00	0.857	0.00

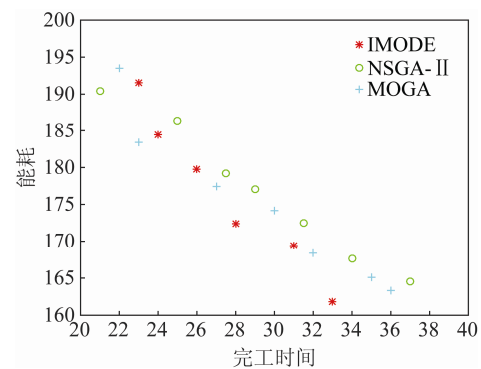


图 2 MK01 3 种算法解集分布  
Fig. 2 Solution set distribution of MK01 3 algorithms

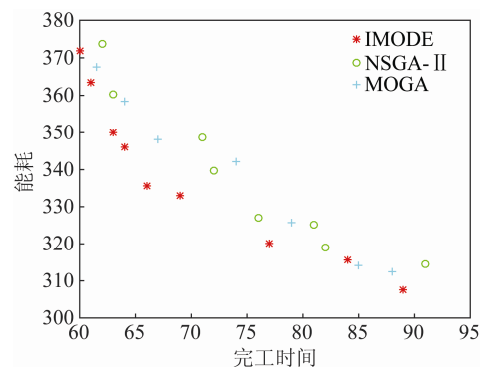


图 3 MK04 3 种算法解集分布  
Fig. 3 Solution set distribution of MK04 3 algorithms

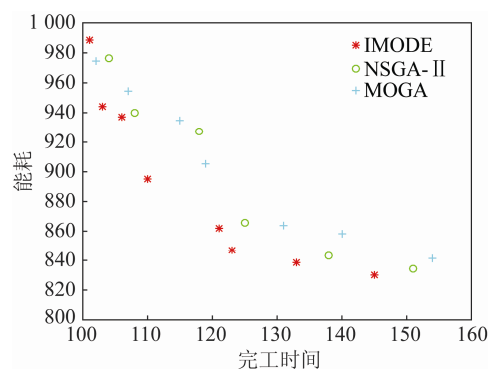


图 4 MK07 3 种算法解集分布  
Fig. 4 Solution set distribution of MK07 3 algorithms



## 5.2 实例测试

本实验的数据从实际生产中获取, 如表 3 所示, 该实例为  $8 \times 10$  即 8 台机器加工 10 个工件, 共

38 道工序。IMODE, NSGA-II 按上文数值进行设置, 文献[12]算法按原文进行设置, 种群规模为 100, 最大迭代次数为 500。

表 3 工件工序加工信息表  
Tab. 3 Processing information table for parts

工件	工序	加工时间/h / 能耗/(kW/h)							
		$M_1$	$M_2$	$M_3$	$M_4$	$M_5$	$M_6$	$M_7$	$M_8$
$J_1$	1	-	4.60/6	-	-	4.89/7	-	6.15/5	-
	2	5.83/8	-	-	6.31/9	-	-	3.95/15	-
	3	-	-	5.51/7	3.57/11	-	5.76/8	-	-
	4	-	4.13/11	-	-	3.42/15	-	-	4.47/10
	5	4.81/11	-	-	3.42/16	-	-	-	6.29/9
$J_2$	1	-	-	4.43/12	5.92/9	4.83/10	-	-	-
	2	3.06/14	-	-	3.09/15	-	-	4.76/7	-
	3	4.95/6	3.19/15	-	-	-	3.69/10	-	-
$J_3$	1	-	-	-	4.24/8	3.80/14	-	-	4.50/15
	2	-	-	4.54/12	5.83/8	-	-	4.50/14	-
	3	3.08/15	-	-	4.50/10	2.93/17	-	-	-
	4	-	4.53/15	-	-	-	2.67/20	-	3.09/18
	5	1.79/12	-	-	2.59/11	-	3.10/9	-	-
$J_4$	1	-	6.05/9	-	-	3.01/17	-	4.44/13	-
	2	-	-	3.25/8	-	3.96/11	-	-	4.73/10
	3	-	4.27/11	-	3.74/15	-	-	5.64/14	-
	4	2.26/16	-	-	-	-	-	4.44/8	3.58/9
$J_5$	1	-	-	-	7.43/18	-	4.39/13	4.89/11	-
	2	-	4.11/7	-	-	3.46/12	-	-	3.60/10
	3	3.37/19	-	4.55/12	-	-	-	-	2.72/17
$J_6$	1	-	4.19/9	-	-	-	3.48/12	3.31/14	-
	2	3.11/14	-	-	-	3.06/15	-	-	3.79/9
	3	-	-	4.07/11	-	5.95/7	-	-	5.98/7
	4	1.92/15	-	-	5.54/10	-	-	2.79/12	-
	5	-	4.47/11	-	-	-	2.96/18	-	2.98/17
	6	6.07/18	-	-	4.08/17	-	4.71/15	-	-
$J_7$	1	4.15/9	-	3.88/14	-	-	-	-	5.91/8
	2	-	7.32/7	-	-	5.22/10	-	6.01/8	-
	3	4.33/11	-	-	2.55/15	-	4.38/12	-	-
$J_8$	1	5.06/8	-	-	8.13/7	-	-	-	6.89/6
	2	-	4.38/10	-	-	3.95/12	-	5.62/8	-
	3	2.66/6	-	-	-	-	-	4.44/14	3.58/9
	4	6.07/19	-	-	4.08/7	-	-	4.71/15	-
$J_9$	1	-	-	-	7.43/22	-	4.39/12	4.89/17	-
	2	4.81/15	-	3.42/8	-	-	-	-	6.29/18
	3	3.81/14	-	-	3.42/8	-	-	-	3.95/11
$J_{10}$	1	-	-	-	7.43/22	-	4.39/12	4.89/17	-
	2	4.81/15	-	3.42/8	-	-	-	-	6.29/18
$J_{11}$	1	3.37/9	-	4.55/12	-	-	-	-	2.72/7
	2	5.83/18	-	-	6.31/19	-	-	3.95/11	-
	3	-	3.89/15	-	-	6.32/10	-	-	5.51/11
	4	-	4.11/18	-	5.88/13	-	7.65/11	-	-
机器空载/(kW/h)		0.35	0.47	0.24	0.28	0.45	0.32	0.28	0.38



初始调度实例测试结果如图 5 所示。

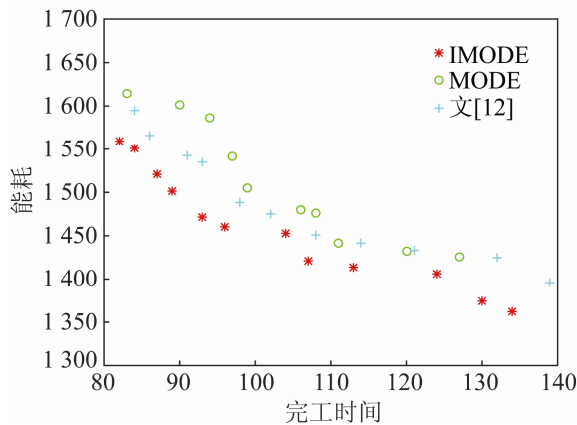


图 5 3 种算法初始化实例测试结果  
Fig. 5 Test results of initialization examples of 3 algorithms

初始调度: 根据 G1 法赋权所得主观权重向量为  $w_1=(0.75, 0.25)$ , 熵权法赋权所得客观权重向量为  $w_2=(0.466, 0.534)$ 。

通过 NASH 均衡求得主、客观权重的综合权重,  $\alpha_1=0.88, \alpha_2=0.15$ , 归一化后  $\alpha_1^*=0.85, \alpha_2^*=0.15$ , 根据式求得综合权重  $w=(0.707, 0.293)$ 。

使用改进的 TOPSIS 法对各方案进行评价, 计算得出的各方案投影结果如下, 并给出图 6 最优方案甘特图。

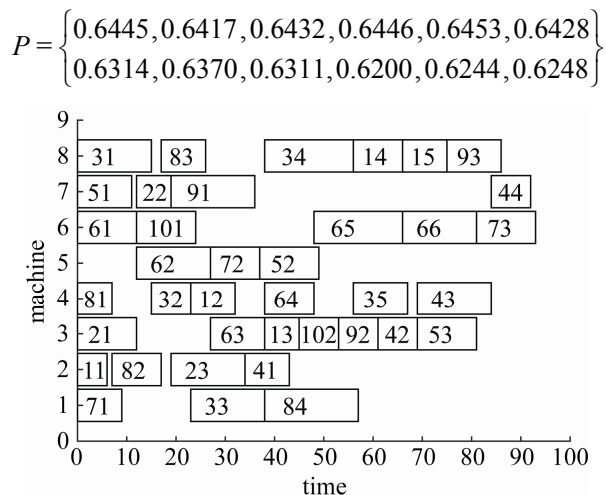


图 6 初始化调度方案最优甘特图  
Fig. 6 Optimal Gantt chart of initial scheduling scheme

机器故障重调度: 实验设置机器 5 于时间 30~60 发生故障。35 时刻, 开始重调度, 重调度实例测试结果如图 7 所示。

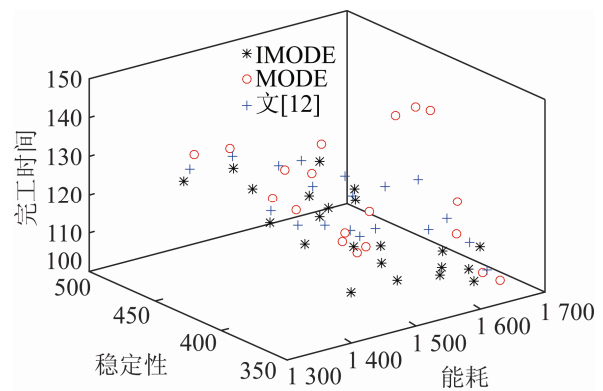


图 7 3 种算法机器故障重调度实例测试结果  
Fig. 7 Test results of 3 algorithms for machine fault rescheduling

根据 G1 法赋权所得主观权重向量为  $w_1=(0.454, 0.346, 0.200)$ , 根据熵权法赋权所得客观权重向量为  $w_2=(0.300, 0.354, 0.346)$ 。

通过 NASH 均衡求得主、客观权重的综合权重,  $\alpha_1=0.845, \alpha_2=0.173$ , 归一化后  $\alpha_1^*=0.83, \alpha_2^*=0.17$ , 根据式求得综合权重  $w=(0.428, 0.347, 0.225)$ 。

使用改进的 TOPSIS 法对各方案进行评价, 计算得出的各方案投影结果如下, 并给出图 8 最优方案甘特图。

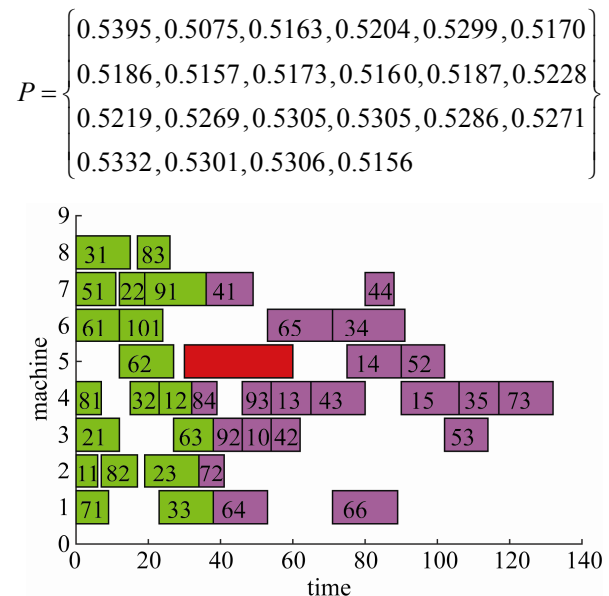


图 8 机器故障重调度方案最优甘特图  
Fig. 8 Optimal Gantt chart for rescheduling of machine failures

紧急插单重调度: 实验设置紧急工件 11 的

4 道工序在时刻 35 插入。35 时刻，开始重调度，重调度实例测试结果如图 9 所示。

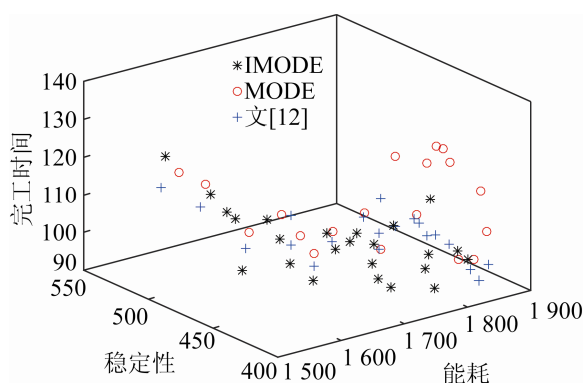


图 9 3 种算法紧急插单重调度实例测试结果

Fig. 9 Test results of 3 algorithms for emergency insertion single scheduling

根据 G1 法赋权所得主观权重向量为  $w_1=(0.477, 0.364, 0.159)$ ，根据熵权法赋权所得客观权重向量为  $w_2=(0.287, 0.381, 0.332)$ 。

通过 NASH 均衡求得主、客观权重的综合权重， $\alpha_1=0.862$ ， $\alpha_2=0.162$ ，归一化后  $\alpha_1^*=0.84$ ， $\alpha_2^*=0.16$ ，根据式求得综合权重  $w=(0.447, 0.367, 0.186)$ 。

使用改进的 TOPSIS 法对各方案进行评价，计算得出的各方案投影结果如下，并给出图 10 最优方案甘特图。

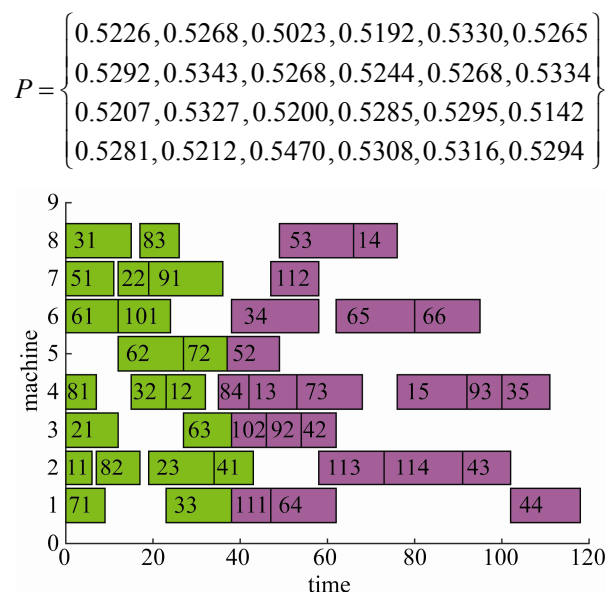


图 10 紧急插单重调度方案最优甘特图

Fig. 10 Optimal Gantt chart for emergency single insertion scheduling scheme

从初始化调度结果、机器故障重调度结果和紧急插单重调度结果可以看出与对比算法 MODE 及文献[12]算法求解所得的非劣解集基本受 IMODE 的非劣解集支配，表明 IMODE 求解多目标柔性作业车间调度问题性能更优。并根据 TOPSIS—G1—熵权法决策方法从众多调度方案中筛选出最合理的调度方案。

## 6 结论

本文针对多目标柔性作业车间动态调度问题提出了一种改进的多目标差分进化算法。在多目标差分进化算法中采用 Logistic 混沌映射初始化种群，并引入自适应变异交叉算子对种群进行更新以及基于免疫学原理的快速非支配排序法对个体进行筛选。运用 DFJSP 规则驱动柔性车间在设备故障及紧急订单插入情况下的重调度优化。提出一种基于 TOPSIS-G1-熵权法的决策方法对初始调度及重调度求解的调度方案进行决策。通过实例对比验证，证明了优化算法 IMODE 求解多目标生产调度问题的有效性以及决策方法 TOPSIS-G1-熵权法的合理性。后续将更多的实际情况考虑进去，进一步完善算法，最终将算法与实际生产结合提高实际问题的解决效率。

## 参考文献:

- [1] Nouri M, Bekrar A, Trentesaux D. Towards Energy Efficient Scheduling and Rescheduling for Dynamic Flexible Job Shop Problem[J]. IFAC Papers On Line (S1474-6670), 2018, 51(11): 1275-1280.
- [2] Yong Z, Jian J Y, Zhuang H. Automatic Design of Scheduling Policies for Dynamic Flexible Job Shop Scheduling via Surrogate-assisted Cooperative Co-evolution Genetic Programming[J]. International Journal of Production Research (S0020-7543), 2020, 58(9): 2561-2580.
- [3] Shen X N, Yao X. Mathematical and Multi-objective Evolutionary Algorithms Applied to Dynamic Flexible Job Shop Scheduling Problems[J]. Information Science (S0020-0255), 2015, 29(8): 198-224.
- [4] 陈超, 王艳, 严大虎, 等. 面向能耗的柔性作业车间动态调度研究[J]. 系统仿真学报, 2017, 29(9): 2168-2174,

2181.  
Chen Chao, Wang Yan, Yan Dahu, et al. Research on Dynamic Flexible Job Shop Scheduling Problem for Energy Consumption[J]. Journal of System Simulation, 2017, 29(9): 2168-2174, 2181.
- [5] 朱传军, 邱文, 张超勇, 等. 多目标柔性作业车间稳健性动态调度研究[J]. 中国机械工程, 2017, 28(2): 173-182.  
Zhu Chuanjun, Qiu Wei, Zhang Chaoyong, et al. Multi-objective Flexible Job Shop Dynamic Scheduling Strategy Aiming at Scheduling Stability and Robustness[J]. China Mechanical Engineering, 2017, 28(2): 173-182.
- [6] 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 System, 2011, 17(12): 2629-2637.
- [7] 任龙, 阳晓剑. 基于 G1 法的高层建筑火灾应急能力评价[J]. 消防科学与技术, 2019, 38(5): 724-727.  
Ren Long, Yang Xiaojian. High-rise Building Fire Emergency Capability Assessment Based on G1[J]. Fire Science and Technology, 2019, 38(5): 724-727.
- [8] 刘润恺, 于龙, 陈德明. 基于 AHP-熵权法的高铁接触网可信度评价研究[J]. 铁道科学与工程学报, 2019, 16(8): 1882-1889.  
Liu Runkai, Yu Long, Chen Deming. Research on Dependability of High-speed Railway Catenary Based on AHP-entropy method[J]. Journal of Railway Science and Engineering, 2019, 16(8): 1882-1889.
- [9] Xia W, Qiu Q D. Improved AHP-TOPSIS Model for the Comprehensive Risk Evaluation of Oil and Gas Pipelines[J]. Petroleum Science (S1672-5107), 2019, 16(6): 1479-1492.
- [10] Brandimarte P. Routing and Scheduling in a Flexible Job Shop by Tabu search[J]. Annals of Operations Research (S0254-5330), 1993, 41(3): 157-183.
- [11] Piroozfard H, Wong K Y, Wong W P. Minimizing Total Carbon Footprint and Total Late Work Criterion in Flexible Job Shop Scheduling by Using an Improved Multi-objective Genetic Algorithm[J]. Resources Conservation and Recycling (S0921-3449), 2018, 12(8): 267-283.
- [12] Yu L G, Xiao X L, Li H L, et al. Optimized of Tube Hydroforming Loading Path Using Multi-objective Differential Evolution[J]. The International Journal of Advanced Manufacturing Technology (S0268-3768), 2017, 88(14): 837-846.