

DOI: 10.3901/JME.2010.11.156

# 基于改进非支配排序遗传算法的多目标 柔性作业车间调度\*

张超勇 董 星 王晓娟 李新宇 刘 琼

(华中科技大学数字制造装备与技术国家重点实验室 武汉 430074)

**摘要:** 采用多目标进化算法解决具有工件释放时间、工件目标差异的柔性作业车间调度问题。依据实际制造系统中存在较多的最大完工时间、平均流经时间、总拖期时间、机器总负荷、瓶颈机器负荷和生产成本性能指标,建立多目标柔性作业车间调度模型。针对柔性作业车间调度问题的特点,设计一种扩展的基于工序的编码及其主动调度的解码机制,以及初始解产生机制和有效的交叉、变异操作;针对非支配排序遗传算法(Non-dominated sorting genetic algorithm II, NSGA-II)在非支配排序和精英选择策略方面的不足,设计一种改进的非支配排序遗传算法,应用改进的算法求解柔性作业车间调度问题得到一组 Pareto 解集,并运用层次分析法选出最优妥协解。通过测试基准和模拟实际生产的实例,验证提出算法的可行性和有效性。

**关键词:** 柔性作业车间调度 多目标进化算法 非支配排序遗传算法 层次分析法

**中图分类号:** TP18

## Improved NSGA-II for the Multi-objective Flexible Job-shop Scheduling Problem

ZHANG Chaoyong DONG Xing WANG Xiaojuan LI Xinyu LIU Qiong

(State Key Laboratory of Digital Manufacturing Equipment and Technology,  
Huazhong University of Science and Technology, Wuhan 430074)

**Abstract:** An improved multi-objective evolutionary algorithm is proposed for solving the flexible job-shop scheduling problem (FJSP) with released time and job-oriented multi-objective. The multi-objective FJSP optimization model is put forward, in which the makespan, the mean flow-time, total tardiness, total workload of machines, workload of the bottleneck machine and production cost widely concerned in complex manufacturing system are considered. According to the characteristics of the FJSP, an extended operation-based encoding and an active scheduling decoding mechanism are presented, an initial solution generation mechanism, and two effective crossover and mutation operations are designed for the genetic algorithm. In order to ensure convergence and the diversity of the solutions, an improved non-dominated sorting genetic algorithm (NSGA-II) is proposed. A set of Pareto solutions are obtained by the improved NSGA-II, and the analytic hierarchy process (AHP) approach is used to select the optimal compromise solution. The approach is tested on instances taken from the literature and practical data. The computation results validate the effectiveness of the proposed algorithm.

**Key words:** Flexible job-shop scheduling problem Multi-objective evolutionary algorithm

Non-dominated sorting genetic algorithm II Analytical hierarchy process

## 0 前言

调度是影响制造业生产效率的关键因素,采用

有效的调度优化技术能提高设备利用率、保证按时交货、降低库存及成本等,从而确保企业的经济目标。其中柔性作业车间调度问题(Flexible job-shop scheduling problem, FJSP)突破了资源唯一性限制,每道工序可在多台不同的机床上加工,更符合实际的生产环境,对它的研究具有重要的理论价值和应用意义。

\* 国家高技术研究发展计划(863 计划, 2007AA04Z107、2007AA04Z190)、国家自然科学基金(70772056)和湖北省教育厅中青年基金(Q20092303)资助项目。20090716 收到初稿, 20091215 收到修改稿

FJSP 是经典作业车间调度问题 (Job-shop Scheduling Problem, JSP) 的扩展, 它不仅需要确定工序的加工顺序, 还要给每个工序分配机器, 是更为复杂的 NP-hard 问题<sup>[1]</sup>。由于调度具有很高的计算复杂性, 许多学者主要研究的是单目标的问题。然而, 在实际生产中不仅需要考虑工件最大完工时间、平均流经时间、总拖期时间、机器总负荷和瓶颈机器负荷等调度性能指标优化, 还要考虑生产成本等费用指标, 单一目标很难反映实际生产车间调度问题。此外, 目前大多数文献研究各个工件性能指标均相同的调度问题; 而在实际生产中, 经常会遇到工件调度目标不同的情况, 例如, 车间调度员在安排一批加工任务时, 有些工件必须需满足客户要求要在交货期之前完工, 有些工件要求越早完工越好, 有些工件要求生产成本尽可能的低, 此时必须兼顾各个工件的调度指标<sup>[2]</sup>, 该问题更接近实际问题的需要。这类问题也是本文要研究工件目标差异的多目标调度优化问题。

实际生产中的调度问题通常涉及多个目标的同时优化, 各个优化目标之间可能不一致甚至互相冲突。相对于单目标优化问题, 多目标优化的结果不是一个最优解, 而是符合 Pareto 最优概念的一组解, 称为 Pareto 最优解集或者非支配解集。进化算法是一类模拟生物自然选择和自然进化的随机搜索算法, 具有隐含的并行性和在全局解空间进行搜索的能力。它一次运行便能得到一组折衷解, 非常适合求解多目标优化的问题。目前国内外学者已提出多种典型求解多目标的进化算法 (Multi-objective evolutionary algorithm, MOGA), 如: DEB 等<sup>[3]</sup>提出的非支配排序遗传算法 NSGA-II, Corne 等<sup>[4]</sup>提出的 PESA, KNOWLES 等<sup>[5]</sup>提出的 PAES, ZITZLER 等<sup>[6]</sup>提出的 SPEA2 等。其中, DEB 等提出的 NSGA-II 既有良好的分布性又有较快的收敛速度, 被国内外学者广泛应用。但是随着其应用领域的不断扩大, 其运算复杂度高、精英选择策略影响种群的多样性等不足逐渐显露。

本文首先根据现实生产环境建立具有工件释放时间, 考虑工件最大完工时间、平均流经时间、总拖期时间、机器总负荷、瓶颈机器负荷和生产成本等的多目标柔性作业车间调度模型, 然后提出一种改进的非支配排序的遗传算法解决工件目标差异的多目标柔性作业车间调度优化问题, 并利用层次分析法 (Analytical hierarchy process, AHP) 在 Pareto 解集中选出最优满意解。

1 多目标 FJSP 调度优化模型

1.1 问题描述

成本对企业管理决策是至关重要, 它与时间是衡量现实生产调度优化效果的两个重要指标。考虑加工成本和时间因素的 FJSP 可描述为: 有  $n$  个工件 ( $J_i, i \in \{1, 2, \dots, n\}$ ) 在  $m$  台机器 ( $M_k, k \in \{1, 2, \dots, m\}$ ) 上加工, 每个工件  $J_i$  依据给定工艺路线的一道或多道工序 ( $O_{ij}, j \in \{1, 2, \dots, n_i\}, n_i$  表示加工工件  $J_i$  的工序总数) 加工。工件的每道工序  $O_{ij}$  可在多台不同的机床中一台加工,  $M_{ij}$  表示工件  $J_i$  的第  $j$  道工序可用机器集合,  $M_{ij} \subseteq \{1, 2, \dots, m\}$ 。工件  $J_i$  的第  $j$  道工序在机床  $k$  上的加工时间为  $p_{ijk}$ , 每台机器  $M_k$  单位时间加工费用为  $C_{Mk}$ 。生产环境是动态的, 每个工件  $J_i$  的释放时刻  $R_i$  不一致, 部分工件有交货期  $D_i$  约束, 所有工件的性能指标不一定相同。调度目标是为确定每道工序的加工机器及每台机器上各工件的加工顺序, 最优化各个工件预定的性能指标。此外, 在加工过程中还需要满足以下约束条件。

同一时刻同一台机器只能加工一个零件; 工件的加工不允许中断; 不同工件的工序之间没有先后约束; 不同工件具有相同的优先级; 所有的机器在 0 时刻均可用。

一个 3 个工件在 5 台机器上加工考虑成本和时间的柔性作业车间调度问题如表所示。在释放时刻  $R_i$  列的数字表示工件的释放(或)到达时间; 加工时间列中的数字表示加工时间  $p_{ijk}$ , 符号 “-” 表示  $M_k$  不能加工工序  $O_{ij}$ 。交货期  $D_i$  列中的数字表示该工件客户有交货期要求, 符号 “-” 表示工件要求尽早完工, 没有交货期要求。最后一行表示每台机器  $M_k$  单位时间加工工件的费用。

表 1 柔性作业车间调度问题的加工时间表

工件	工序	释放时刻 $R_i/h$	加工时间 $t/h$					交货期 $D_i/h$
			$M_1$	$M_2$	$M_3$	$M_4$	$M_5$	
$J_1$	$O_{11}$	6	12	18	—	18	—	—
	$O_{12}$		—	8	6	20	20	
	$O_{13}$		—	8	—	—	18	
$J_2$	$O_{21}$	2	18	—	8	20	14	50
	$O_{22}$		—	18	12	9	—	
	$O_{23}$		7	—	20	12	18	
$J_3$	$O_{31}$	2	18	20	8	7	6	—
	$O_{32}$		—	7	8	—	20	
单位时间加工费用 $a/(元 \cdot h^{-1})$			6	8	7	4	5	

1.2 优化目标函数

制造车间处于企业的中心环节, 制造部(企业高

层)、销售部和生产车间分别从自己利益出发会对车间调度决策寄予不同的期望,例如,销售部希望能准时交货,制造部希望降低成本,生产车间希望提高生产效率,因此寻求各方利益合理的折衷成为生产调度决策的关键。本文针对柔性作业车间的需求,给出了以下有意义的目标:缩短最大完工时间和平均流经时间以增加生产效率,减少总拖期时间以提高客户满意度,降低机器总负荷和瓶颈机器负荷以提高设备利用率,以及降低生产成本满足企业发展的需求。实际生产中还可依据需要对上述指标进行调整。本文需要同时进行优化的目标函数如下:最大完工时间函数

$$f_1 = \max \{C_i | i=1,2,\dots,n\}$$

平均流经时间函数

$$f_2 = \frac{1}{n} \times \sum_{i=1}^n (C_i - R_i)$$

总拖期时间函数

$$f_3 = \sum_{i=1}^{n_l} \max \{C_i - D_i, 0\}$$

机器总负荷函数

$$f_4 = \sum_{i=1}^n \sum_{j=1}^{n_i} \sum_{k=1}^m p_{ijk} x_{ijk}$$

瓶颈机器负荷

$$f_5 = \max \sum_{i=1}^n \sum_{j=1}^{n_i} p_{ijk} x_{ijk} \quad k=1,2,\dots,m$$

生产成本

$$f_6 = \sum_{i=1}^n \left( MC_{J_i} + \sum_{j=1}^{n_i} \sum_{k=1}^m C_{Mk} p_{ijk} x_{ijk} \right)$$

式中,  $n$  表示待加工的工件总数;  $m$  表示加工工件的机器数量;  $C_i$  为工件  $J_i$  的完工时间;  $R_i$  为工件  $J_i$  的释放时间;  $D_i$  为工件  $J_i$  的交货期,  $l \in \{1,2,\dots,n_l\}$ ,  $n_l$  表示有交货期约束工件的序号;  $x_{ijk}$  为决策变量, 当工件  $J_i$  的第  $j$  道工序选择在机床  $k$  上加工时,  $x_{ijk}=1$ , 否则为 0。  $MC_{J_i}$  为工件  $J_i$  的原材料成本;  $C_{Mk}$  为机床  $M_k$  单位时间的加工费用。

## 2 基于改进 NSGA-II 的多目标柔性作业车间调度算法

实际车间调度优化通常会遇到以下两类主要问题: ① 多个相互冲突的目标; ② 高维复杂的搜索空间。例如, 本文调度优化模型包括 6 个性能指标, 各个目标之间相互制约和冲突, 搜索空间极为

复杂, 传统的数学规划方法, 如多目标加权法, 约束法等, 不可能收敛到 Pareto 最优前沿面上的非凸区域。

非支配排序遗传算法 NSGA 是由 SRINIVAS 和 DEB 在 1994 年提出, 它采用分级选择方法突出好的个体, 以及小生境方法维持优秀个体子种群的稳定<sup>[7]</sup>。DEB 等进一步改进 NSGA 算法提出了 NSGA-II, 在 NSGA-II 中使用了新的非支配解排序方法、精英选择策略以及拥挤距离, 计算复杂度也从原来的  $O(MN^3)$  减少到  $O(MN^2)$ , 其中  $M$  代表目标的数量,  $N$  代表种群的大小<sup>[3]</sup>。但是 NSGA-II 算法也存在一些不足, 比如精英选择策略影响种群的多样性。本文改进 NSGA-II 算法求解多目标 FJSP。

### 2.1 初始解

初始解的质量对于遗传算法求解 FJSP 的效果有较大影响, 随机产生的初始群体通常获得最终解的质量较差。针对决策者对于不同性能指标的偏好, 本文提出了一种简单的初始解产生机制。设本文决策者更关注的是加工时间的性能指标, 则按照一给定概率优先选择加工某工序加工时间较小的机器, 那么初始解产生机制为随机产生工序的加工顺序, 然后在每个工序的机器集中随机选取两台机器, 如果随机产生的 0~1 之间随机值小于 0.8, 则选取加工时间短的机器, 否则选择加工时间长的机器。

### 2.2 编码和解码

编码是遗传算法成功实施优化的首要和关键问题。对于传统的 JSP, 目前应用最多的是基于工序的编码, 在该编码染色体中每个工件的工序都用相应的工件序号表示, 从左到右扫描染色体, 第  $k$  次出现的工件序号, 表示该工件的第  $k$  道工序。它具有任意置换染色体后总能得到可行调度、避免死锁、对解空间表征的完全性等等优点。

但是 FJSP 问题不仅要确定工序的加工顺序, 还需为每道工序选择一台合适的机器, 仅采用基于工序的编码方法不能得到问题的解。本文设计一种扩展的基于工序的编码, 该编码由两部分组成, 第一部分为基于工序的编码, 用来确定工序的加工先后顺序; 第二部分为基于机器的编码, 用来选择每道工序的加工机器。融合这两种编码方法, 可得到柔性作业车间调度问题的一个可行解。

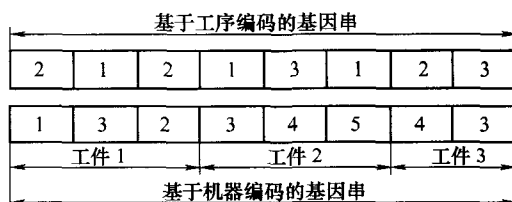


图1 扩展基于工序的编码

例如，图 1 显示了扩展的基于工序编码的染色体，该染色体由基于工序编码的基因串和基于机器编码的基因串组成。其中基于工序编码的基因串确定工序的加工先后顺序，基于机器编码的基因串为每个工件所有工序的加工机器。因此，图 1 中的染色体表示的工序和机器序列为：(O<sub>21</sub>, M<sub>3</sub>), (O<sub>11</sub>, M<sub>1</sub>), (O<sub>22</sub>, M<sub>4</sub>), (O<sub>12</sub>, M<sub>3</sub>), (O<sub>31</sub>, M<sub>4</sub>), (O<sub>13</sub>, M<sub>2</sub>), (O<sub>23</sub>, M<sub>5</sub>), (O<sub>32</sub>, M<sub>2</sub>)，加工时间序列为[8 12 9 6 7 8 18 8]。

设工件  $J_i$  的第  $j$  道工序  $O_{ij}$  的完工时间分别为  $t_{ij}$ ，工序的加工时间为  $p_{ijk}$ ，其中  $k$  为工序  $O_{ij}$  的加工机器。对于本文具有工件释放时间的 FJSP，解码计算时需考虑工序  $O_{ij}$  是否是工件的第一道工序。工序  $O_{ij}$  完工时间的计算公式为

$$t_{ij} = \begin{cases} \max\{R_i, t_{hp}\} + p_{ijk} & j = 1 \\ \max\{t_{i(j-1)}, t_{hp}\} + p_{ijk} & j > 1 \end{cases} \quad (1)$$

式中， $R_i$  表示工件  $J_i$  的释放时刻； $t_{hp}$  表示工序  $O_{ij}$  在加工机器  $M_k$  上紧前工序的完工时间(如果工序  $O_{ij}$  没有机器紧前工序，则  $t_{hp}=0$ )； $t_{i(j-1)}$  表示工序  $O_{ij}$  的工件紧前工序  $O_{i(j-1)}$  的完工时间； $p_{ijk}$  为工序  $O_{ij}$  在机器  $M_k$  上的加工时间。

从式(1)可以看出，工件  $J_i$  第一道工序的开始时间是由工件的释放时刻  $R_i$  和该工序在加工机器上紧前工序的完工时间中最大值决定；工件  $J_i$  的非第一道工序  $O_{ij}$  的开始时间由它在加工机器上的紧前工序和工件的紧前工序中的最大值决定。

根据式(1)进行解码，可得到该染色体对应的半主动调度解。本文采用一种插入式贪婪解码算法运用到 FJSP，将染色体经过解码后产生主动调度。该解码方法的描述如下：把染色体看作工序的有序序列，根据工序在该序列上的顺序进行解码，将每道工序插入到该工序可选加工机器上的最佳可行的加工时刻安排加工，直到序列上所有工序都安排完为止。应用插入式贪婪解码算法将图 1 中染色体解码得到的主动调度解的机器甘特图如图 2 所示。

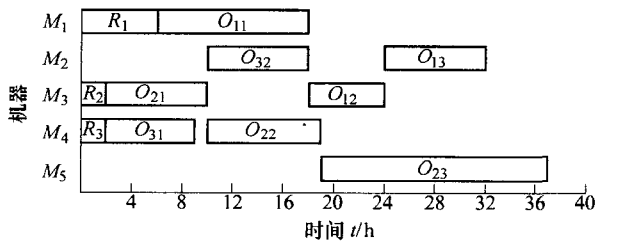


图 2 解码得到主动调度解的机器甘特图

2.3 Pareto 排序

Pareto 排序的具体过程：首先找出当前种群中非支配最优解并分配等级 1；然后将这些个体从种

群中移出，在剩余个体中找出新的非支配解，再分配等级为 2；重复上述过程，直至种群中所有个体都被设定相应的等级，如图 3 所示。

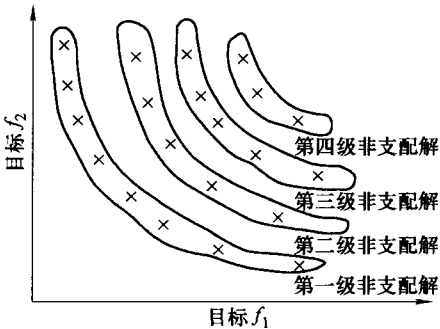


图 3 非支配解分级示意图

传统 NSGA-II 算法虽然在非支配解排序的效率上比 NSGA 有较大提高，但是对于规模大、目标较多的调度问题，NSGA-II 算法仍需要花费较多时间。本文采用郑金华<sup>[8]</sup>提出的一种快速排序方法构造非支配集，该方法在进化个体之间定义一种新的关系： $\forall X, Y \in P$ ，如果  $X \succ Y$  或  $X$  和  $Y$  不相关，则  $X \succ_d Y$ 。快速排序方法每次找一个个体  $X$  作为比较对象，按照关系“ $\succ_d$ ”进行比较判断，经过一趟排序后，以  $X$  为中界将  $P$  中的个体分成两部分，比  $X$  小的一部分肯定是被支配个体；第二部分是比较  $X$  小的个体或与  $X$  不相关的个体，然后对第二部分进行下一轮排序，直至第二部分只有一个个体。快速排序算法的平均时间复杂度小于  $O(n \lg n)$ ，比 NSGA-II 的  $O(rN^2)$  具有更高的运行效率。

2.4 选择操作

采用小生境技术中的拥挤距离来维持群体分布的多样性。一个个体的拥挤距离可以通过计算同级别与其相邻的两个个体在每个子目标上的距离差之和来求取，用  $L[i]_d$  表示。图 4 显示了有两个目标  $f_1$  和  $f_2$ ，个体  $i$  拥挤距离为图 4 中虚线四边形的长和宽之和。拥挤距离大的个体参与繁殖和进化的机会较多，从而维持种群的多样性。

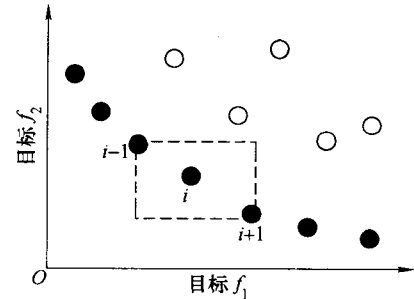


图 4 个体  $i$  的拥挤距离

拥挤距离的计算是为确保算法能收敛到一个

均匀分布的 Pareto 曲面。经过排序和拥挤距离计算, 群体中的每个个体  $i$  都得到两个属性: 非支配排序等级  $i_r$  和拥挤距离  $L[i]_d$ 。定义偏序关系  $>_n$  为  $i >_n j$ , 如果  $i_r < j_r$ , 或者  $i_r = j_r$  且  $L[i]_d > L[j]_d$ 。偏序关系的意义为: 如果两个个体的非支配排序等级不同时, 优先考虑等级低的个体; 如果两个个体等级相同, 优先考虑较不拥挤的个体, NSGA-II 算法构造新群体时将依据这种偏序关系进行操作。

本文选择操作采用二元锦标赛选择。在测试过程中, 作者发现 NSGA-II 算法会有大量个体收敛于第一等级非支配曲面并且拥挤距离为 0 的现象, 此时偏序关系相等。为此本文提出将决策的重要性指标引入到选择操作, 当个体等级和拥挤距离均相同时, 按照给定概率优先选择重要指标较小的个体。以后工作可进一步在选择操作中对群体进行初步拥挤筛选, 以尽量剔除群体中的不合理解。

## 2.5 交叉操作

针对 FJSP 不仅需要确定工序的加工顺序, 还要给工序分配机器的特点, 本文提出了两种交叉操作, 第一种是 IPOX 交叉操作, 用于染色体中工序的加工顺序的交叉; 第二种是多点交叉操作(Multi-point preservative crossover, MPX), 用于染色体工序分配的机器的交叉。下面对这两种交叉操作分别进行介绍。

### 2.5.1 IPOX 交叉操作

IPOX 是基于 POX<sup>[9]</sup>交叉操作的改进, 它仅交叉父代染色体中工序的加工顺序, 保留工件中工序分配的机器到子代。IPOX 交叉操作过程如图 5 所示, 其中  $P_1$  和  $P_2$  为表 1 中调度实例的两个父代染色体, 交叉产生子代  $C_1$  和  $C_2$ 。IPOX 交叉操作的过程为: 把所有的工件机随机分成两个集合  $J_1$  和  $J_2$ , 复制  $P_1$  包含在  $J_1$  的工件到  $C_1$ ,  $P_2$  包含在  $J_2$  的工件到  $C_2$ , 保留它们的位置, 复制  $P_2$  包含在  $J_2$  的工件到  $C_1$ ,  $P_1$  包含在  $J_1$  的工件到  $C_2$ , 保留它们的顺序。

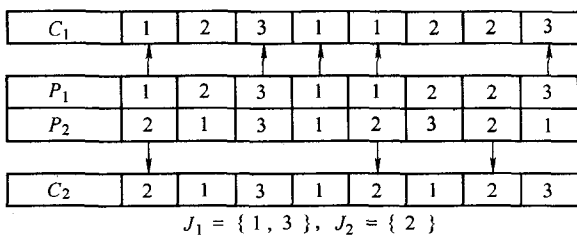


图 5 基于工序编码的 IPOX 交叉操作

### 2.5.2 MPX 交叉操作

MPX 操作仅交叉父代染色体中工序选定的机器, 工序的加工顺序保留到子代。对于表中调度实例, 设父代  $P_1$  和  $P_2$  交叉产生子代  $C_1$  和  $C_2$ 。MPX

交叉操作的过程为: 首先随机产生一个由整数 0、1 组成与染色体长度相等的集合  $R$ , 依次在  $P_1$  和  $P_2$  中选出与  $R$  中的 1 位置对应相同的工序, 交换它们分配的机器,  $P_1$  和  $P_2$  中其他机器的保留到子代, 这样生子代  $C_1$  和  $C_2$ 。MPX 交叉操作过程如图 6 所示。

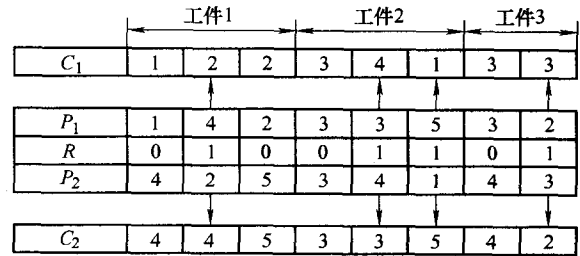


图 6 基于机器编码的 MPX 交叉操作

## 2.6 变异操作

本文采用两种变异操作, 一种是扩展的插入变异操作, 即在基于工序编码的染色体中随机选择一个工件的工序(基因), 将它插入到一个随机的位置, 保持所有工件工序分配机器的不变, 如图 7 所示, 把  $P_1$  染色体位置 5 的工件 3 插入到位置 2 工件之前, 得到可行解的子代  $C_1$ 。第二种分配机器的变异操作, 即随机选择基于机器编码染色体的两个基因, 在该工序的加工机器集中随机选取机器替换该基因, 该变异操作过程如图 8 所示。

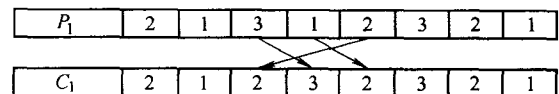


图 7 基于工序编码的插入变异操作

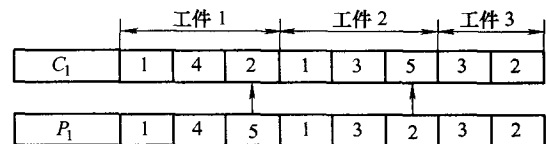


图 8 基于机器编码的变异操作

## 2.7 改进的 NSGA-II 算法

传统的 NSGA-II 算法采用隐性精英解保留策略。首先产生一个组合的种群  $R_t = P_t \cup Q_t$ , 其中  $P_t$  和  $Q_t$  分别为父代和子代种群,  $R_t$  的个数为  $2N$ ; 然后种群  $R_t$  根据偏序关系排序, 直到个体数量达到  $N$  就形成了新的父代种群  $P_{t+1}$ , 在此基础上进行遗传操作产生子代种群  $Q_{t+1}$ , 传统 NSGA-II 执行的流程如图 9 所示。

虽然 NSGA-II 算法的隐性精英解保留策略可以明显地改善遗传算法的收敛性, 同时可以避免丢失进化过程中取得的最优解。但是在这种精英策略也存在着不足, 它可能造成进化过程中大部分非支配解都处于级数为 1 的非支配曲面, 这个曲面可能远

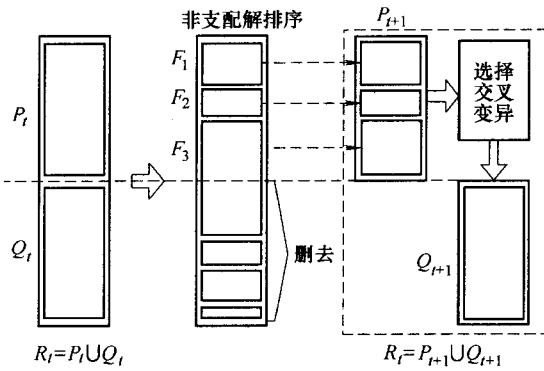


图9 传统NSGA-II算法的流程

离真正的 Pareto 最优曲面, 这样就会造成种群的提前收敛或收敛于局部最优解。在多目标柔性作业车间调度问题优化过程中, 这种情况尤为严重。为了确保搜索方向是朝着真正的 Pareto 最优曲面和得到均匀分布的解, 本文中设计了一个改进的精英策略, 目的是保证种群的多样性和收敛到较优的 Pareto 曲面。该策略通过引入一个分布函数限制父代精英解的数量, 该分布函数为

$$n_i = |F_i| \times r_i \quad (2)$$

式中,  $i$  为非支配等级( $i \geq 1$ );  $n_i$  表示在第  $i$  级非支配曲面  $F_i$  上选取的个体的数量;  $|F_i|$  表示第  $i$  层非支配曲面上个体的数量;  $r_i$  在  $0 \sim 1$  之间。改进的 NSGA-II 算法流程如下所述。

(1) 随机产生初始种群  $P_0$  和  $Q_0$ , 种群规模均为  $N$ , 并设  $t=0$ 。

(2) 合并父代和子代种群  $R_t = P_t \cup Q_t$ , 按照 Pareto 概念构造  $R_t$  非支配等级曲面  $F = \{F_1, F_2, \dots, F_i\}$ 。

(3) 清空  $P_{t+1}$  和设  $i=1$ 。

(4) 如果  $|P_{t+1}| + |F_i| \leq N$ , 则继续步骤(5), 否则跳至步骤(7)。

(5) 计算  $F_i$  集中个体的拥挤距离, 并把  $F_i$  中的个体加入父代种群  $P_{t+1}$  中,  $P_{t+1} = P_{t+1} \cup F_i$ 。

(6)  $i = i+1$ , 返回步骤(4)。

(7) 在  $P_{t+1}$  上执行选择、交叉和变异操作, 产生新的种群  $Q_{t+1}$ 。

(8) 如果  $t < i_{\max}$  ( $i_{\max}$  为最大进化代数), 则  $t=t+1$ , 返回步骤(2); 否则算法终止。

### 3 基于 AHP 的多目标 FJSP 调度决策

基于多目标进化算法的柔性作业车间调度优化得到一组 Pareto 解集, 决策者还面临如何从这组 Pareto 解集选出最满意方案的问题。可以根据决策者主观经验(如依据决策指标重要性的排序)或多指

标决策技术进行排序或评估。本文采用层次分析法 AHP 方法选择多目标优化得到的 Pareto 解, 帮助决策者选出最满意的方案。

AHP 法是美国匹兹堡大学教授 SAATY 在 20 世纪 70 年代提出的一种处理有限方案的多目标决策方法, 它将定性分析与定量分析有效地结合, 具有要求信息量很少, 决策过程花费的时间短等优点, 可应用车间多目标调度问题的决策。AHP 法解决问题的基本思路是: 是把复杂问题分解为若干层次, 在最低层次通过两两对比得出各因素的权重, 然后计算出各方案对总目标的权数, 权数最大的方案即为最优方案。

层次分析法多级递阶结构一般可以分成三层, 即总目标层, 分目标层和方案层, 图 10 给出了多目标 FJSP 调度决策的 AHP 递阶结构, 该结构的三个层次为: ① 总目标层是 FJSP 调度决策解决问题的目的; ② 分目标层为针对目标评价各方案时所考虑的各个子目标; ③ 方案层是基于多目标进化算法的 FJSP 调度优化阶段产生的一组 Pareto 解集。

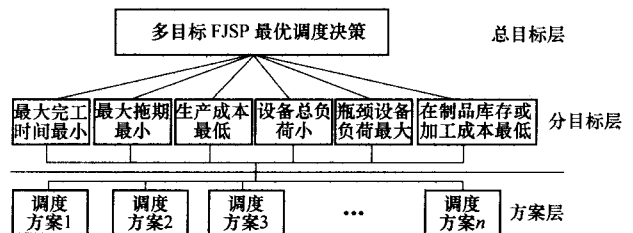


图10 多目标FJSP调度决策的AHP递阶结构

### 4 试验仿真与分析

上述改进的多目标遗传算法用 C++ 编程, 程序运行环境为 P4 CPU, 主频 3.0 GHz, 内存为 512 MB。实例测试分为两部分, 第一部分为了评估和比较提出算法的性能, 对基准问题进行测试; 第二部分通过模拟实际生产车间数据的实例进行测试。

多目标遗传算法运行的参数设置如下: 种群规模  $P=100$ , IPOX 交叉概率 45%, MPX 交叉 45%, 两种变异操作的变异概率各为 2%, 选择概率值  $r_i=0.9$ , 最大进化代数 80。

#### 4.1 基准实例测试

为了验证算法性能, 本文使用文献[10]的基准问题进行测试。该问题包括一个  $8 \times 8$  部分柔性调度实例, 优化性能指标为单一目标, 分别是最大完工时间最小、总负荷最小和机器最大负荷最小。

将本文多目标遗传算法改为单目标算法, 并运用 KACEM 等<sup>[10]</sup>提出的初始化机制产生初始解。对

于 8×8 实例测试结果的比较如表 2 所示。表 2 中分别给出了 SPT、传统 GA、KACEM、PSO+SA 等算法<sup>[10-11]</sup>的结果和本文改进遗传算法得到最好结果的比较。从表 2 中可以看出,对于 8×8 实例的三个性能指标,本文算法得到最好值均优于或等于其他三个算法得到的最好值。由此可见,本文算法对于求解多目标柔性作业车间调度优化问题能取得良好的结果。

表 2 对于 8×8 实例测试结果的比较

目标函数	SPT	GA	AL	KACEM	PSO+SA	本文算法
最大完工时间 $f_1/h$	19	16	16	16	15	15
总负荷时间 $f_4/h$	91	77	75	75	73	73
机器最大负荷时间 $f_5/h$	16	14	—	14	—	14

4.2 实例仿真

通过模拟某模具厂加工车间的实际数据并取整处理后,获得表 3、4 数据。表 3 给出了 10 个工件在 8 台机器上加工 FJSP 实例的工件到达时间、工序加工时间和交货期,并在最后一行给出每台机器单位时间的加工费用;从表 3 可以看出,部分工件对的目标要求不同,如工件  $J_1$ 、 $J_5$  和  $J_{10}$  要求在时刻 60 前交货, $J_4$ 、 $J_6$  和  $J_8$  要求在时刻 80 前交货。表 4 给出了 10 个的工件的原材料成本。多目标遗传算法的执行参数见上。由于 6 个优化目标的 Pareto 超曲面无法通过图直观显示,只在表 5 显示了在一次运行中得到的 60 个 Pareto 解集。现在决策者面临如何从这组 Pareto 解集选出最满意方案。

表 3 具有工件到达时间和交货期的 FJSP 实例

工件	释放时刻 $R_i/h$	工序 $工$	加工时间 $t/h$								交货期 $t_0/h$
			$M_1$	$M_2$	$M_3$	$M_4$	$M_5$	$M_6$	$M_7$	$M_8$	
$J_1$	5	$O_{11}$	—	—	—	12	—	10	9	—	60
		$O_{12}$	17	—	—	—	17	10	—	15	
		$O_{13}$	—	24	—	11	—	—	10	—	
$J_2$	17	$O_{21}$	—	—	16	10	21	14	—	17	—
		$O_{22}$	8	12	—	—	19	11	—	—	
		$O_{23}$	—	—	—	15	—	21	25	—	
		$O_{24}$	—	—	—	—	18	—	9	—	
		$O_{25}$	12	15	—	14	9	—	10	—	
		$O_{26}$	—	9	—	7	10	8	—	—	
$J_3$	16	$O_{31}$	—	14	—	—	—	—	17	—	—
		$O_{32}$	—	23	23	17	—	18	—	—	
		$O_{33}$	—	—	20	9	22	—	—	21	
		$O_{34}$	7	—	10	—	8	11	9	—	
$J_4$	4	$O_{41}$	—	18	—	—	—	17	18	—	80
		$O_{42}$	—	—	—	10	—	—	12	—	
		$O_{43}$	—	8	11	8	—	—	9	20	
$J_5$	17	$O_{51}$	—	—	24	10	16	—	—	—	60
		$O_{52}$	—	—	—	17	—	—	—	8	

(续)

工件	释放时刻 $R_i/h$	工序	加工时间 $t/h$								交货期 $t_0/h$
			$M_1$	$M_2$	$M_3$	$M_4$	$M_5$	$M_6$	$M_7$	$M_8$	
$J_6$	10	$O_{61}$	—	20	—	—	—	—	22	—	80
		$O_{62}$	—	—	18	—	—	19	—		
		$O_{63}$	19	—	17	16	16	—	—	18	
$J_7$	12	$O_{71}$	—	—	16	—	17	16	—	—	—
		$O_{72}$	22	—	—	—	—	20	22	—	
		$O_{73}$	—	—	—	—	—	18	23	—	
		$O_{74}$	—	—	—	—	—	10	—	24	
$J_8$	0	$O_{81}$	11	—	—	—	—	—	21	21	80
		$O_{82}$	25	11	—	—	—	—	—	—	
		$O_{83}$	24	18	—	—	—	25	—	20	
$J_9$	16	$O_{91}$	—	12	9	—	—	14	—	—	—
		$O_{92}$	24	—	11	21	—	22	25	—	
		$O_{93}$	—	12	—	—	—	16	—	19	
		$O_{94}$	—	6	—	8	7	—	10	—	
		$O_{95}$	9	12	—	6	11	8	10	—	
$J_{10}$	2	$O_{10,1}$	19			21			23	60	
		$O_{10,2}$	18			24			12		
机器单位时间加工费用 $a/(元 \cdot h^{-1})$			8	5	10	9	7	6	9	4	

表 4 工件的原材料成本

工件号	$J_1$	$J_2$	$J_3$	$J_4$	$J_5$	$J_6$	$J_7$	$J_8$	$J_9$	$J_{10}$
原材料成本 $C_m/元$	160	210	440	340	570	300	310	150	390	460

表 5 得到的 Pareto 解

序号	最大完工时间 $f_1/h$	平均流经时间 $f_2/h$	总拖期时间 $f_3/h$	机器总负荷 $f_4/h$	瓶颈机器负荷 $f_5/h$	生产成本 $f_6/元$
1	85	55	0	437	72	6 407
2	85	55	0	438	67	6 426
3	85	57	0	435	75	6 329
4	86	55	0	434	72	6 374
5	88	54	0	432	75	6 318
6	88	57	0	434	74	6 352
7	89	57	3	434	68	6 414
8	89	58	1	432	75	6 313
9	89	58	2	434	68	6 414
10	90	55	0	433	73	6 311
11	90	56	0	432	67	6 389
12	90	57	0	434	80	6 288
13	90	57	0	436	65	6 431
14	90	57	6	433	81	6 289
15	90	57	6	435	81	6 271
16	90	58	0	432	73	6 302
17	90	58	1	430	73	6 288
18	90	58	1	431	80	6 277
19	90	58	1	432	80	6 270
20	90	59	0	433	72	6 362
21	90	60	11	432	81	6 248
22	91	58	1	429	75	6 295
23	92	58	0	430	75	6 320

(续)						
序号	最大完工 时间 $f_1/h$	平均流经 时间 $f_2/h$	总拖期时 间 $f_3/h$	机器总负 荷 $f_4/h$	瓶颈机器 负荷 $f_5/h$	生产成本 $f_6/元$
24	93	58	0	431	73	6 307
25	93	58	6	434	72	6 371
26	94	56	0	430	75	6 336
27	94	60	11	434	81	6 230
28	94	61	13	431	81	6 255
29	95	56	0	431	73	6 329
30	95	58	0	434	72	6 350
31	95	58	3	432	81	6 260
32	95	59	1	429	80	6 259
33	96	62	13	432	81	6 219
34	96	66	20	431	81	6 219
35	97	57	0	432	73	6 302
36	97	58	0	430	93	6 246
37	97	60	1	428	93	6 232
38	97	63	24	431	81	6 219
39	98	63	13	426	83	6 421
40	100	58	0	441	64	6 447
41	100	66	48	428	91	6 328
42	101	57	0	431	93	6 255
43	101	60	0	430	75	6 314
44	101	60	1	427	84	6 277
45	101	60	1	430	93	6 214
46	101	63	12	431	89	6 247
47	103	61	0	430	73	6 288
48	103	62	16	434	71	6 370
49	104	63	19	425	89	6 287
50	108	67	22	426	84	6 306
51	109	77	81	426	99	6 217
52	110	57	0	429	93	6 273
53	110	61	0	428	93	6 264
54	110	64	11	433	89	6 231
55	111	68	18	433	91	6 214
56	113	68	50	423	90	6 280
57	114	80	75	422	91	6 257
58	118	66	1	428	80	6 300
59	121	72	56	429	109	6 169
60	135	60	5	443	63	6 489

4.3 AHP 决策过程

决策者可采用多指标决策技术 AHP 从得到的 Pareto 解集选出满意方案。具体过程如下。

4.3.1 确立判断矩阵

实例决策时重要性依次为总拖期时间、最大完工时间、生产成本、瓶颈机器负荷、机器总负荷和平均流经时间。用 1~9 之间的数据表示它们之间两两重要程度，得到判断矩阵

$$A = \begin{pmatrix} & f_1 & f_2 & f_3 & f_4 & f_5 & f_6 \\ f_1 & 1 & 7 & 1/2 & 6 & 5 & 3 \\ f_2 & 1/7 & 1 & 1/9 & 1/3 & 1/4 & 1/5 \\ f_3 & 2 & 9 & 1 & 7 & 5 & 3 \\ f_4 & 1/6 & 3 & 1/7 & 1 & 1/2 & 1/5 \\ f_5 & 1/5 & 4 & 1/5 & 2 & 1 & 1/3 \\ f_6 & 1/3 & 5 & 1/3 & 5 & 3 & 1 \end{pmatrix}$$

经计算得到权重矢量  $W=(0.2\ 881\ 0.029\ 8\ 0.387\ 2\ 0.052\ 7\ 0.080\ 3\ 0.162\ 0)^T$ 。

4.3.2 计算决策矩阵

多目标决策过程的一个显著问题是各个指标的 量纲不同，没有同一标准，无法比较。因此，在 综合评定之前，需要进行无量纲化处理，即将指标 的特征值同一变换到[0, 1]范围内。本文采用如下方 法来进行统一量化处理。

设  $b_{ij}$  表示指标  $f_j$  对 Pareto 方案  $X_i$  的量纲一值， 定义以下函数

$$b_{ij} = \frac{a_j^{\max} - a_{ij}}{a_j^{\max} - a_j^{\min}} \tag{3}$$

$$a_j^{\max} = \max\{a_{1j}, a_{2j}, \cdots, a_{pj}\} \tag{4}$$

$$a_j^{\min} = \min\{a_{1j}, a_{2j}, \cdots, a_{pj}\} \tag{5}$$

式中， $i=1, 2, \cdots, p$  是评价方案的数量； $j=1, 2, \cdots, q$  为指标  $f_j$  的数量； $a_{ij}$  是采用方案  $X_i$  提供指标  $f_j$  的值。 由式(3)可得到的方案的决策矩阵  $B=(b_{ij})_{p \times q}$ 。

4.3.3 综合评判

计算  $D=(D_i) = BW^T = \sum_{j=1}^q b_{ij}w_j$ ，可得到评价方 案满意度的矩阵  $(D_i)_p$ 。

取  $D_L = \max\{D_i\} = D_{17}=0.864\ 419$ ，说明方案 17 为最满意方案，其对应的调度解如图 11 所示。

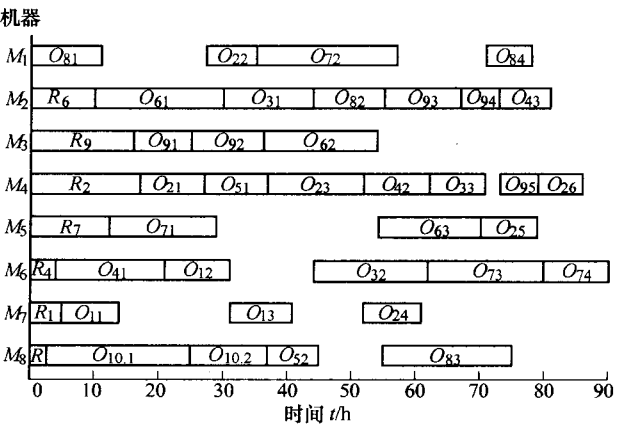


图 11 最满意方案 17 对应的调度解

5 结论

- (1) 根据实际制造车间建立了具有时间和成本性能指标的多目标柔性作业车间调度模型。
- (2) 设计了一种扩展的基于工序的编码，提出了相应初始解产生策略、主动调度解码机制、交叉和变异操作。运用提出的遗传算法解决 FJSP 基准



问题, 获得较好的结果。

(3) 针对 NSGA-II 算法非支配解排序效率低和非支配解易于收敛到第一等级的非支配曲面的缺点, 设计了一种改进的非支配排序遗传算法求解多目标 FJSP, 并采用层次分析法从获得的 Pareto 解集中选出最优满意解。

最后, 运用提出的算法解决实际制造车间中多目标柔性作业车间调度问题, 证实了算法的有效性。以后工作可考虑进一步将决策者的偏好和多目标进化算法结合, 在遗传操作中尽量剔除群体中的不合理解, 更有效地指导算法向决策者偏好的方向搜索。

### 参 考 文 献

- [1] GAREY E L, JOHNSON D S, SETHI R. The complexity of flowshop and job-shop scheduling[J]. *Mathematics of Operations Research*, 1976, 1: 117-129.
- [2] 吴秀丽, 孙树栋, 杨展, 等. 多目标柔性 Job Shop 调度问题的技术现状和发展趋势[J]. *计算机应用研究*, 2007, 24(3): 1-5, 9.  
WU Xiuli, SUN Shudong, YANG Zhan, et al. Survey on multi-objective flexible job shop scheduling problem [J]. *Application Research of Computers*, 2007, 24(3): 1-5, 9.
- [3] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multi-objective genetic algorithm: NSGA-II [J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182-197.
- [4] CRONE D W, KNOWLES J D, OATES M J. The pareto envelope-based selection algorithm for multi-objective optimization [C] //SCHOENAUER M, DEB K, RUDOLPH G, et al. *Proceedings of the Parallel Problem Solving from Nature VI Conference*, Paris, France. *Lecture Notes in Computer Science*: Springer, 2000, 1917: 839-848.
- [5] KNOWLES J, CORNE D. The Pareto archived evolution strategy: a new baseline algorithm for multiobjective optimization[C] //Proceedings of the 1999 Congress on Evolutionary Computation. Piscataway, NJ: IEEE Press, 1999: 98-105.
- [6] ZITZLER E, THIELE L. Comparison of multiobjective evolutionary algorithms: empirical results[J]. *Evolutionary Computation*, 2000, 8(2): 173-195.
- [7] SRINIVAS N, DEB K. Multi-objective function optimization using non-dominated sorting genetic algorithm[J]. *Evolutionary Computation*. 1995, 2(3): 221-248.
- [8] 郑金华. 多目标进化算法及其应用[M]. 北京: 科学出版社, 2007.  
ZHENG Jinhua. Multi-objective evolutionary algorithm and application[M]. Beijing: Science Press, 2007.
- [9] 张超勇, 饶运清, 李培根, 等. 基于 POX 交叉的遗传算法求解 Job-Shop 调度问题 [J]. *中国机械工程*, 2004, 15(23): 2 149-2 153.  
ZHANG Chaoyong, RAO Yunqing, LI Peigen, et al. An improved genetic algorithm for the job shop scheduling problem[J]. *China Mechanical Engineering*, 2004, 15(23): 2 149-2 153.
- [10] KACEM I, HAMMADI S, BORNE P. Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems[J]. *IEEE Transaction Systems, Man, and Cybernetics - Part C*, 2002, 32(1): 1-13.
- [11] XIA Weijun, WU Zhiming. An effective hybrid optimization approach for multi-objective flexible job shop scheduling problems [J]. *Computers & Industrial Engineering*, 2005, 48(2): 409-425.

作者简介: 张超勇, 男, 1972 年出生, 讲师。主要研究方向智能调度算法、制造执行系统、网络化制造和敏捷供应链。

E-mail: zcyhust@sohu.com

刘琼(通信作者), 女, 1965 年出生, 副教授。主要研究方向制造系统集成优化、物流与供应链管理等。

E-mail: qiongliu@mail.hust.edu.cn