

# 基于改进 NSGA-II 算法的多目标生产智能调度

齐琦, 毋涛

(西安工程大学 计算机科学学院, 陕西 西安 710048)

**摘要:**在制造业自动化、智能化生产模式的需求日益增加的趋势下,针对生产制造过程中生产工序安排不合理造成的生产效率低、资源浪费严重等问题,构建以最大完成时间和最大生产成本的智能优化调度模型,使用一种改进的 NSGA-II 算法进行研究。通过 MSOS 染色体编码方案,将个体基因分成机器和工序两部分分别编码。种群初始化通过适当扩大种群的方式,提高算法的全局搜索能力。采用动态拥挤度计算和精英保留的方式选择个体,保证解得多样性的同时保留优秀个体。采用动态混合交叉算法,确保种群更快地向最优方向进化。利用动态自适应变异概率来提高种群后期的多样性。通过对西装上衣缝制环节生产环节编排调度的仿真实验,验证了改进算法在多目标生产智能优化调度中的可行性和有效性。

**关键词:**NSGA-II 算法;柔性车间调度;多目标优化;动态拥挤度;精英保留

**中图分类号:**TP39

**文献标识码:**A

**文章编号:**1673-629X(2021)08-0162-07

**doi:**10.3969/j.issn.1673-629X.2021.08.028

## Multi-objective Intelligent Production Optimal Scheduling Based on Improved NSGA-II Algorithm

QI Qi, WU Tao

(School of Computer Science, Xi'an Polytechnic University, Xi'an 710048, China)

**Abstract:**With the increasing demand for automated and intelligent production models in the manufacturing industry, in view of the low production efficiency and serious waste of resources caused by unreasonable production process arrangements in the manufacturing process, an intelligent optimization scheduling model with the maximum completion time and maximum production cost is constructed. An improved NSGA-II algorithm is used for research. Through the MSOS chromosome coding scheme, the individual gene is divided into the machine and the process to code respectively. The population initialization improves the global search ability of the algorithm by appropriately expanding the population. The dynamic crowding calculation and elite retention are used to select individuals to ensure the diversity of solutions while retaining outstanding individuals. The dynamic hybrid crossover algorithm is adopted to ensure that the population evolves to the optimal direction faster. The dynamic adaptive mutation probability is to increase the diversity of the population in the later stage. Through the simulation experiment of the scheduling and scheduling of the production link in the sewing of suits, the feasibility and effectiveness of the improved algorithm in the intelligent scheduling of multi-objective production are verified.

**Key words:**NSGA-II algorithm; flexible job shop scheduling; multi-objective optimization; dynamic crowding; elite retention

### 1 概述

随着制造业的快速发展,生产过程中生产时间和成本的冲突变得更加明显。长期以来,生产过程中的平衡问题很大程度上依赖于生产技术人员个人经验<sup>[1]</sup>,致使生产方案很不理想。通过综合考虑时间、生产成本等多个因数的合理调度策略可以有效提高生产的执行效率。

随着智能算法的发展,许多学者将智能算法应用到机械制造、电子工业领域,以便得到更加有效的解决

方案。最开始出现,有基于遗传算法、变邻域搜索<sup>[2]</sup>、混合人工蜂群<sup>[3]</sup>、多种群遗传算法<sup>[4]</sup>、改帝国竞争算法<sup>[5]</sup>求解最大完工时间的柔性车间调度问题,这种仅以最大完工时间为目标的研究存在实用性较差的问题。后来,文献[6-7]提出最大完工时间、提前/拖期惩罚函数、生产总成本的多目标优化问题。这些多目标问题研究更加符合实际生产需求,但现有的多目标优化调度研究多是为不同的优化目标函数设定不同的权重值,把多目标优化问题转化为一个单目标优化函

收稿日期:2020-10-08

修回日期:2021-02-08

基金项目:陕西省科技研究成果转移与推广计划项目(2019CGXNG-018)

作者简介:齐琦(1993-),男,硕士生,研究方向为信息系统设计与软件开发;毋涛,副教授,研究方向为纺织服装智能与信息技术。

数来求解<sup>[8-9]</sup>。这种方法存在的缺点是各权重难以确定、优化目标之间的非线性相关性导致求解很难得到全局最优<sup>[10]</sup>,且对于多目标优化问题几乎不存在单个全局最优解,而使用多目标函数直接求解得到最优解集的方案将会克服以上问题,并且文献[11-13]研究表明 NSGA-II 算法在求解多目标优化问题上有着良好的求解能力。

通过对 NSGA-II 算法的研究,在种群初始化、拥挤度计算、交叉算子选择环节进行优化改进,提高求解能力及解的质量。并将改进的算法用于西装上衣缝制的工序编排中进行测试,为实际生产提供了多种有效解决方案,验证了算法的有效性。

### 1.1 问题描述

在服装缝制生产环节,一件衣服可分成  $n$  个不同的部件,每个部件又包含多道工序,每道工序可以在  $m$  台机器中的一台或多台上完成,且各工序有先后顺序,前一道工序做完后一道工序才可以开始,各工序在每台机器上的加工时间已知,各衣服部件在进行组合前互相不影响。以所有部件完成的加工时间和加工成本为优化目标,在不考虑人为因素的前提下,确定所有工序的加工顺序以及使用的机器,得到有效的生产方案。

### 1.2 数学建模

#### (1) 最小化最大完成时间。

最大完成时间是指所有部件同时进行加工,最后一个部件完成时所花费的时间,时间越短表明方案越好。最小化完成时间的目标函数如式(1)所示。

$$f(1) = \min C_{\max} = \min \{ \max C_j \mid j \in n \} \quad (1)$$

其中,  $C_{\max}$  表示所有部件加工完成的最大时间;  $n$  表示所有的部件总个数;  $C_j$  表示部件  $j$  的完成时间。

#### (2) 最小化最大生产成本。

机器在加工和待机两种状态都会产生能耗,但其单位能耗值不相同。该文将两种状态下的能耗作为生产成本进行考虑。用  $\text{runtime}_i$  和  $\text{waittime}_i$  分别表示机器  $i$  的加工时间和待机时间,  $\text{PR}_i$  表示机器  $i$  加工状态的单位能耗;  $\text{PW}_i$  表示机器  $i$  待机状态时的单位能耗;  $m$  表示机器总个数;得到机器  $i$  总的使用时间和生产成本  $W_i$ , 分别如式(2)、式(3)所示,最小化生产成本的目标函数如式(4)所示:

$$j_j = \text{runtime}_i + \text{waittime}_i \quad (2)$$

$$W_i = \text{runtime}_i * \text{PR}_i + \text{waittime}_i * \text{PW}_i \quad (3)$$

$$f(2) = \min \sum_{i=1}^m W_i \quad (4)$$

## 2 算法设计

### 2.1 编码

针对柔性车间调度多目标优化的复杂性,该文采

用 MSOS 染色体编码方案,个体基因的编码由两部分组成:机器选择部分(前半部分)、工序排序部分(后半部分)。工序排序部分基因位的值为工件号;工件号第几次出现即表示该工件的第几道工序,机器选择部分的基因位根据工序的排列产生,基因位的编码表示该工序选择的加工机器在可选择机器集中的序号。编码规则如图1所示。

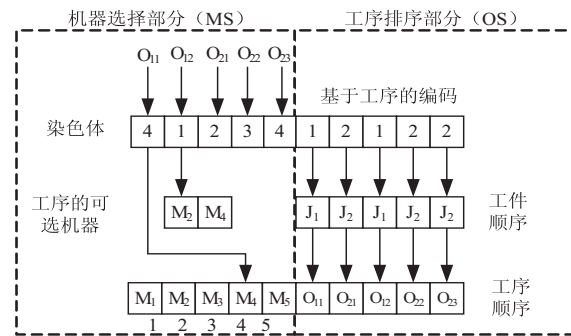


图1 染色体编码示例图

### 2.2 解码

对工序排序部分和机器选择部分分别进行解码。

工序排序部分解码方式:创建一个矩阵  $S[1, n]$ , 存放每个工件的出现次数,其中  $n$  表示工件个数,如工件2第3次在编码中出现,就将矩阵的  $S[2]$  值变成3;创建一个矩阵  $H[1, m]$ , 用于存放工序解码后的信息,其中  $m$  表示工序总个数。开始解码:取编码的工序部分基因,从第一位开始遍历:得到基因位基因编码  $i$ , 给  $S[i]$  的  $i$  号工件出现次数加1,用工件号乘以100再加上工件已出现的次数,得到对应的解码结果记录到  $H$  中。

机器选择部分解码方式:遍历机器编码基因,依次将  $H$  中工序解码结果除以100取余数,便得到工序编号,将编码结果减去解出的工序编号,除以100得到工件编号,用工件编号、工序编号以及机器编码位基因值,在对应的加工时间矩阵中找到该工序的加工时间。将该工序的开始时间和上一道工序的完成时间进行比较,如果前一道工序完成时间小于本工序开始时间就取本工序的开始时间,否则取前道工序的结束时间作为本工序的开始时间,将开始时间加上该工序的加工时间得到该工序的结束时间,将开始时间和结束时间存入到一个矩阵  $G[2, l]$  中,其中  $l$  表示工序总个数。

经过上述的两步解码过程,便得到对应编码所表示的工序加工顺序矩阵  $H$  以及工序加工时间矩阵  $G$ , 完成解码过程。

### 2.3 种群初始化

NSGA-II 算法中,采用随机选择的方式产生初始种群,第一次迭代时相当于一次随机搜索,由于随机搜索时种群越大找到最优解的概率就越大,但考虑到算

法的复杂度,种群数量也不能无限加大<sup>[14]</sup>。文中设计在不影响算法复杂度的前提下,在种群初始化阶段将初始种群数量设置为种群数量的 1.5 倍,经过适应度计算和拥挤度排序处理得到种群规模的个体进行后续的迭代,来提高初始种群丰富度,减少随机性,加大 NSGA\_II 的求得最优的概率。

## 2.4 选择操作

根据基因编码得到所有工序的开始时间和完成时间以及每个机器的加工时间,使用 1.2 节的最大完成时间和最大生产成本目标函数计算适应度函数值。由于机器待机时间等于使用时间减去加工时间,通过加工功率和待机功率乘以对应的加工时间和待机时间可以求得生产成本。在对子代、父代合并后的种群进行快速非支配排序后,改变 NSGA-II 算法中固定拥挤度计算方式,采用动态计算拥挤度值并按照精英保留策略选择个体组成新父种群。

由于 NSGA-II 算法的求解特点,在迭代后期会出现 Pareto 层次较少,采用分层序号以及拥挤度作为比较条件的方式,将变成以个体拥挤度作为主要的影响因素的选择方式<sup>[15]</sup>。例如,以求解标准测试函数 ZDT1 为例,某一次迭代后得到的 Pareto 面上各粒子的分布如图 2 所示;如果按照固定拥挤度排序,当个体聚集出现时,可能会造成该区域全部个体被淘汰的情况<sup>[16]</sup>,使得 Pareto 解集多样性降低,解集的分布性较差,筛选后的个体分布如图 3 所示。实际上,当某一个体被淘汰后,相邻个体的拥挤度将会发生变化,不考虑这种拥挤度变化的排序方式不能够很好地表现解集的分布情况。采用文中的动态拥挤度排序方式选择个体的结果如图 4 所示。

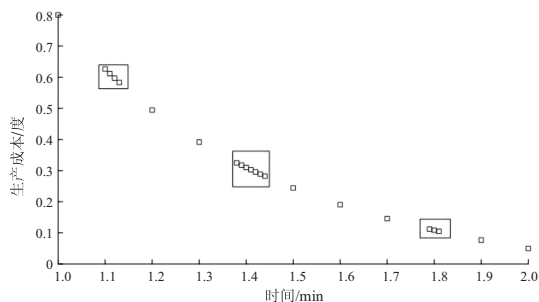


图 2 Pareto 最优个体集筛选前分布

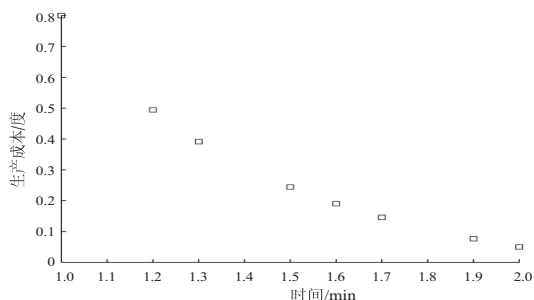


图 3 采用固定拥挤度排序方式保留

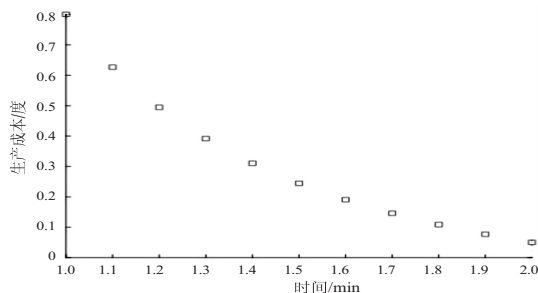


图 4 采用动态拥挤度排序方式保留

文中采用的动态计算个体拥挤度的方式执行流程如图 5 所示。首先根据个体的拥挤度,删除其中拥挤度系数最小的粒子  $i$ ,如果没有达到种群数量,则根据删除个体  $i$  的位置判断需要重新计算拥挤度的个体,并重新计算该个体的最新拥挤度,更新完成后比较所有个体的拥挤度,删除整个解集中拥挤度系数最小的个体,直到得到种群数量规模。

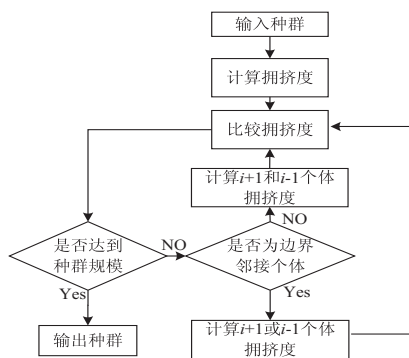


图 5 动态拥挤度计算流程

## 2.5 交叉操作

NSGA-II 算法采用的是模拟二进制交叉 SBX (simulated binary crossover) 对实数编码,优点是实现简单,但是搜索的范围相对较小,容易陷入局部最优解。而 NDX (normal distribution crossover) 算子比 SBX 算子的搜索空间要大<sup>[13]</sup>。NSGA-II 算法在迭代初期是主要处于探索阶段,此时个体适应度偏低、种群多样性较高,应促进交叉产生新的基因;在算法后期搜索范围逐渐变小,应保护优良基因不被破坏。

文中提出一种混合交叉算子,通过迭代次数的变化来改变交叉算子使用比例,在迭代初期为了保证种群的多样性,使用较多的 NDX 算子进行交叉以扩大搜索范围。在迭代后期,由于种群的解集已经趋于一个稳定的区域,可以通过加大 SBX 算子的使用比例,提高局部搜索性能,保存优秀基因,加快收敛。通过引入基于迭代次数的混合交叉算子,可以保证种群搜索空间的多样性与解集的分布均匀性。交叉方式的选择如式(5)所示。

$$\begin{cases} P_n = \cos\left(\frac{(N-1)\pi}{(\text{MAX}_{\text{Gen}}-1)}\right) \\ P_b = 1 - P_n \end{cases} \quad (5)$$

其中,  $P_n$  表示 NDX 交叉算子的使用概率;  $P_b$  表示 SBX 算子的使用概率;  $\text{MAX}_{\text{Gen}}$  表示最大迭代次数;  $N$  表示当前迭代次数。

SBX 算子的计算公式如式(6)所示。

$$\begin{cases} c_{1,i} = \frac{1}{2}(p_{1,i} + p_{2,i}) + \frac{\beta_{\text{SBX}}}{2}(p_{1,i} - p_{2,i}) \\ c_{2,i} = \frac{1}{2}(p_{1,i} + p_{2,i}) - \frac{\beta_{\text{SBX}}}{2}(p_{1,i} - p_{2,i}) \end{cases} \quad (6)$$

其中,  $p_1, p_2$  表示父代个体,  $c_1, c_2$  表示子代个体,  $i$  表示代数;  $\beta_{\text{SBX}}$  是由式(7)产生的随机变量, 式中令  $\mu$  为在区间(0,1)上产生的均匀随机数,  $\gamma$  是自己定义的一个非负数; 而 NDX 算子的  $\beta_{\text{NDX}}$  生成函数公式见式(8)。

$$\beta_{\text{SBX}} = \begin{cases} (2\mu)^{\frac{1}{\gamma+1}}, & \mu \leq 0.5 \\ \left[\frac{1}{2(1-\mu)}\right]^{\frac{1}{\gamma+1}}, & \mu > 0.5 \end{cases} \quad (7)$$

$$\beta_{\text{NDX}} = \begin{cases} 1.481(p_{1,i} - p_{2,i}) \mid N(0,1), & \mu \leq 0.5 \\ -1.481(p_{1,i} - p_{2,i}) \mid N(0,1), & \mu > 0.5 \end{cases} \quad (8)$$

综合式(5)~式(8)得到混合算子的计算公式, 如公式(9)所示:

$$\begin{cases} c_{1,i} = \frac{1}{2}(1 - p_n)(Q + \frac{\alpha}{2}M) + \frac{1}{2}p_n(Q + \frac{1.481M\lambda}{2}) \\ c_{2,i} = \frac{1}{2}(1 - p_n)(Q - \frac{\alpha}{2}p_nM) + \frac{1}{2}p_n(Q - \frac{1.481M\lambda}{2}) \\ \alpha = \begin{cases} (2\mu)^{\frac{1}{\gamma+1}}, & \mu \leq 0.5 \\ \left[\frac{1}{2(1-\mu)}\right]^{\frac{1}{\gamma+1}}, & \mu > 0.5 \end{cases} \\ Q = p_{1,i} + p_{2,i} \\ M = p_{1,i} - p_{2,i} \end{cases} \quad (9)$$

## 2.6 变异操作

变异操作可以起到扩大随机性的作用, 增加算法的搜索能力, 在迭代后期可以通过适当增加变异概率的方式来产生多样性, 所以在迭代过程中使用随迭代次数递增的线性自适应变异概率可以使变异概率随迭代次数的增加而增大, 适当增加迭代后期的多样性。设置初始变异概率值为  $P_0$ , 最大变化概率为  $\rho$ , 最大迭代次数为  $\text{MAX}_{\text{Gen}}$ ; 当前迭代次数为  $N$ , 变异概率计算公式见式(10)

$$P_N = P_0 + \rho \left( \frac{N}{\text{MAX}_{\text{Gen}}} \right) \quad (10)$$

## 2.7 算法流程

算法实现步骤如下:

步骤 1: 确定种群规模、最大迭代次数;

步骤 2: 初始化种群, 设置参数生成种群数量 1.5 倍的个体组成的初始种群, 对个体求解适应度, 进行非支配排序, 如果非支配等级相同则选择拥挤度系数大的, 选出种群数量的优秀个体进行后续迭代;

步骤 3: 将新产生的父代和子代合并成中间种群;

步骤 4: 选择父代。使用快速非支配排序对种群进行分层, 结合动态计算拥挤距离及精英保留方式选择优秀的个体进入下一代种群;

步骤 5: 生成子代。采用 BSX 算子和 NDX 算子的混合交叉算子进行交叉, 采用多项式变异产生新的子代;

步骤 6: 判断算法是否满足终止条件。满足则结束算法, 否则返回步骤 3。

算法流程如图 6 所示。

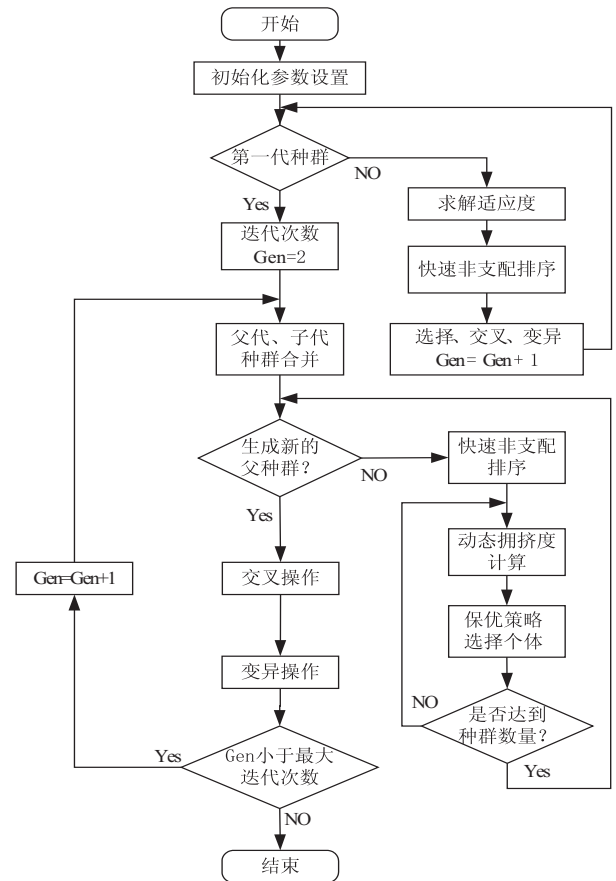


图 6 改进 NSGA-II 算法流程

## 2.8 算法验证

文中将改进的 NSGA-II 算法与 NSGA-II 算法进行对比, 采用文献[11]中的机器加工时间以及加工机器和加工成本信息, 以最小加工时间和最小生产成本作为优化目标进行实验。实验结果如图 7 所示, 其中星号表示 NSGA-II 求解的 Pareto 前沿, 圆点表示改进



NSGA-II 算法的 Pareto 前沿。可以看出,改进的 NSGA-II 算法求得的解质量及分布性都优于原算法,说明改进算法在求解多样性方面性能较好,具有更好的求解能力。

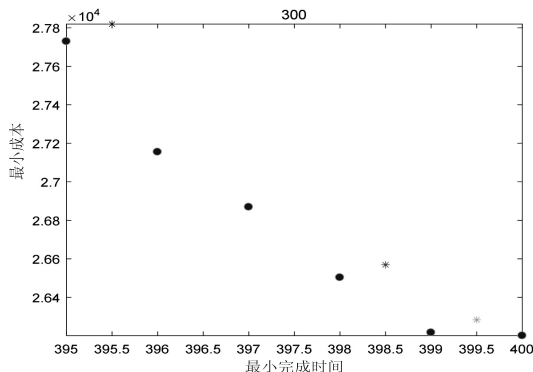


图 7 改进 NSGA-II 算法与 NSGA-II 优化 Pareto 前沿面解分布图

### 3 仿真实例

文中以某西装企业上衣生产缝制环节生产环节为例,对西装各部件的加工工序排列进行研究,验证本算法在缝制环节的有效性。按照工序流程及相邻工序及使用机器的情况分析,对工序进行重新整合,得到西装上衣缝制环节大致工序,如表 1 所示。

表 1 西装上衣生产加工工序

| 工序编号 | 工序              | 工时/s | 设备    |
|------|-----------------|------|-------|
| 1    | 领角领腰处粘带子        | 29   | 烫台    |
| 2    | 领角领腰缝合          | 13   | 平缝机   |
| 3    | 双针装饰线           | 18   | 双针平缝机 |
| 4    | 缝合领面领里\修缝       | 52   | 平缝机   |
| 5    | 烫领(翻、烫、修)       | 53   | 烫台    |
| 6    | 折烫领腰            | 20   | 烫台    |
| 7    | 粘底边             | 20   | 烫台    |
| 8    | 左右后背合拢          | 5    | 平缝机   |
| 9    | 套叉角/修叉角         | 18   | 平缝机   |
| 10   | 缝牵条             | 28   | 平缝机   |
| 11   | 中缝包边            | 10   | 平缝机   |
| 12   | 缝后背缝            | 30   | 平缝机   |
| 13   | 劈烫后背缝/外叉/边/粘领圈粘 | 38   | 烫台    |
| 14   | 整里子叉            | 38   | 平缝机   |

续表 1

| 工序编号 | 工序              | 工时/s | 设备    |
|------|-----------------|------|-------|
| 15   | 包边拔烫            | 4    | 烫台    |
| 16   | 观音里子哥伦比亚        | 40   | 装饰线机  |
| 17   | 后又包边            | 7    | 平缝机   |
| 18   | 烫袖口粘            | 20   | 烫台    |
| 19   | 缝大(小)袖缝         | 98   | 平缝机   |
| 20   | 烫大(小)袖缝         | 56   | 整烫机   |
| 21   | 订袖扣             | 28   | 订扣机   |
| 22   | 袖子面子拷边          | 45   | 拷边机   |
| 23   | 内(外)袖缝内整线       | 60   | 平缝机   |
| 24   | 上袖叉布            | 16   | 平缝机   |
| 25   | 袖叉拷边            | 38   | 拷边机   |
| 26   | 袖叉包边(整线)        | 28   | 平缝机   |
| 27   | 缝挂面里子           | 56   | 平缝机   |
| 28   | 哥伦比亚(整挂面全里)     | 49   | 装饰线机  |
| 29   | 开袋(大袋)          | 20   | 开袋机   |
| 30   | 烫内袋开线           | 30   | 烫台    |
| 31   | 订商标             | 56   | 曲折缝机  |
| 32   | 包袋整线            | 30   | 平缝机   |
| 33   | 上袋布、封袋口         | 40   | 平缝机   |
| 34   | 内袋口套结           | 14   | 套结机   |
| 35   | 挂面撬带子           | 22   | 撬边机   |
| 36   | 袋布拷边            | 40   | 拷边机   |
| 37   | 烫门襟粘(小片底边、前片底边) | 54   | 烫台    |
| 38   | 缝小片             | 37   | 平缝机   |
| 39   | 底边折烫            | 5    | 烫台    |
| 40   | 缝袖笼肩牵带          | 33   | 平缝机   |
| 41   | 上手巾袋            | 25   | 平缝机   |
| 42   | 袋贴(手巾袋)拷边       | 12   | 平缝机   |
| 43   | 缝袋盖             | 30   | 切边平缝机 |
| 44   | 开袋              | 34   | 开袋机   |
| 45   | 缝袋底             | 22   | 双针平缝机 |
| 46   | 腰袋套结            | 13   | 套结机   |
| 47   | 袋托拷边            | 6    | 拷边机   |
| 48   | 撬驳头带子           | 14   | 撬边机   |
| 49   | 切边              | 29   | 切边机   |
| 50   | 内缝撬边            | 29   | 切边机   |

表 2 各机器单位时间成本

| 状态 | 1  | 2  | 3  | 4  | 5  | 6  | 7 | 8 | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|----|----|----|----|----|----|----|---|---|----|----|----|----|----|----|----|----|----|----|----|----|
| 加工 | 15 | 11 | 10 | 11 | 17 | 22 | 5 | 9 | 15 | 10 | 15 | 11 | 10 | 11 | 17 | 22 | 5  | 9  | 15 | 10 |
| 待机 | 1  | 1  | 2  | 3  | 2  | 1  | 2 | 1 | 1  | 1  | 1  | 1  | 2  | 1  | 3  | 2  | 1  | 2  | 1  | 1  |

通过对表 1 分析整理,将该环节生产问题进行数学建模,按照生产部件划分为:衣领、后背、袖子、挂面、

前片 5 个部件,每个部件划分 10 道工序,机器数量为 20,并对每个机器进行编码,机器名称及编号情况如下

(括号中的数表示机器编号,多台机器使用反斜杠分隔):订扣机(1)、定线机(2)、平缝机(3/4)、开袋机(5)、拷边机(6/17)、撬边机(7)、切边机(8)、切边平缝机(9)、曲边缝机(10/18)、烫台(12/13)、双针平缝机(14/20)、套结机(15)、整烫机(16/19)、装饰线机(11)。各机器在使用和待机期间的成本消耗情况如表 2 所示。每个工序生产可用加工机器信息如表 3 所示,其中第一行第二列的[3,4]表示工件 1 的第 2 道工序可使用的机器编号为 3 和 4。各工序在对应机器上的加工时间的信息如表 4 所示,其中第一行第二列的[13,13]表示工件 1 的第 2 道工序使用机器 3 生产花费的时间为 13 s,使用机器 4 花费的时间为 13 s。

表 3 各工序可用机器

| 工序 | 1       | 2       | 3       | 4       | 5       |
|----|---------|---------|---------|---------|---------|
| 1  | [12,13] | [12,13] | [12,13] | [3,4]   | [12,13] |
| 2  | [3,4]   | [3,4]   | [14,20] | 11      | [3,4]   |
| 3  | [14,20] | 9       | [16,19] | 5       | [12,13] |
| 4  | [3,4]   | [3,4]   | 1       | [12,13] | [10,18] |
| 5  | [12,13] | 9       | [6,17]  | [10,18] | [3,4]   |
| 6  | 9       | [16,19] | [10,18] | [3,4]   | [14,20] |
| 7  | 8       | [16,19] | [3,4]   | [14,20] | 9       |
| 8  | [16,19] | [14,20] | [6,17]  | 15      | 5       |
| 9  | 2       | [12,13] | [3,4]   | 7       | [14,20] |
| 10 | 11      | 11      | 9       | 8       | 15      |

表 4 各工序加工时间

| 工序 | 1       | 2       | 3        | 4       | 5       |
|----|---------|---------|----------|---------|---------|
| 1  | [28,30] | [20,21] | [20,20]  | [56,56] | [54,54] |
| 2  | [13,13] | [5,4]   | [98,101] | 49      | [37,36] |
| 3  | [18,17] | 18      | [55,56]  | 20      | [5,6]   |
| 4  | [52,55] | [28,29] | 28       | [30,31] | [33,34] |
| 5  | [53,53] | 10      | [45,47]  | [56,55] | [25,25] |
| 6  | 20      | [30,30] | [60,60]  | [30,31] | [12,13] |
| 7  | 8       | [38,39] | [16,15]  | [40,41] | 30      |
| 8  | [17,17] | [38,20] | [38,39]  | 14      | 34      |
| 9  | 6       | [4,4]   | [28,27]  | 22      | [22,20] |
| 10 | 11      | 40      | 9        | 40      | 13      |

在笔记本配置为 Intel(R) Core(TM) i7-4510U CPU @ 2.00 GHz 上使用 MATLAB R2016b 进行上述实例仿真实验。算法参数配置如下:初始种群设置为 50,迭代次数为 200。实验得到如图 8 所示的 Pareto 前端,以及 Pareto 前端第一个解的甘特图(如图 9 所示)和各工序加工时间表(表 5 所示)。从 Pareto 前端可以看出,综合最小完成时间和最小生产成本两个优化目标得到一个由 6 个非劣解组成的解集,并且各个解均匀分布,在两个目标函数相互影响下,都存在时间短或者成本低的优势,不能区分出哪个解比其他解得到的效果好。在实际生产过程中,决策者可以根据实

际需求选择合适的方案,根据方案得到的生产甘特图以及各工序的加工时间表来指导生产。

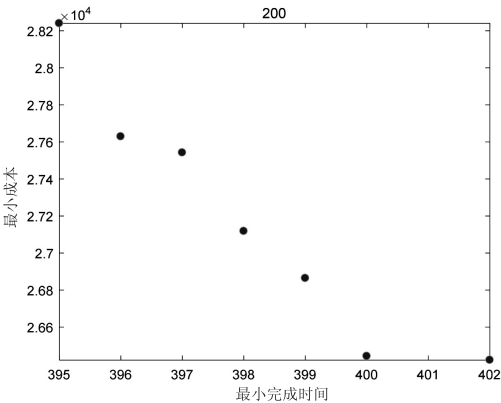


图 8 Pareto 前端解集

通过实验结果生产的甘特图以及加工时间表可以直观地了解各工序的实际开始加工时间和结束时间。在图 9 所示的甘特图中,横坐标表示时间(单位是秒),纵坐标表示加工机器,甘特图中的三位数字编码的含义是:前面(除去后面两位)是部位编号,后两位是该部位所对应的工序编号。如编码 210,表示工件 2 的第 10 道工序。表 5 中用来记录各工序加工的开始时间和完成时间,其中第 1 行的第 2 列[82,95]表示工件 1 的第 2 道工序开始加工时间为 82 s,结束时间为 95 s。

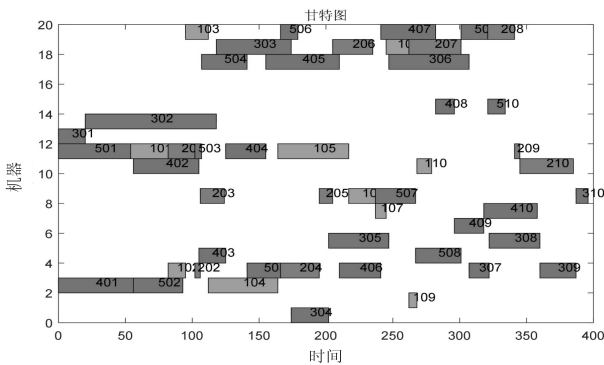


图 9 工序加工甘特图

表 5 各工序加工时间

| 工序 | 1         | 2         | 3         | 4         | 5         |
|----|-----------|-----------|-----------|-----------|-----------|
| 1  | [54,82]   | [82,102]  | [0,20]    | [0,56]    | [0,54]    |
| 2  | [82,95]   | [102,106] | [20,118]  | [56,105]  | [56,93]   |
| 3  | [95,112]  | [106,124] | [118,174] | [105,125] | [102,107] |
| 4  | [112,164] | [166,195] | [174,202] | [125,155] | [107,141] |
| 5  | [164,217] | [195,205] | [202,247] | [155,210] | [141,166] |
| 6  | [217,237] | [205,235] | [247,307] | [210,241] | [166,179] |
| 7  | [237,245] | [262,301] | [307,322] | [241,282] | [237,267] |
| 8  | [245,262] | [321,341] | [322,360] | [282,296] | [267,301] |
| 9  | [262,268] | [341,345] | [360,387] | [296,318] | [301,321] |
| 10 | [268,279] | [345,385] | [387,396] | [318,358] | [321,334] |

#### 4 结束语

文中针对智能生产调度的多目标优化问题,通过将多目标转换成单目标的方式,将多目标问题进行直接求解,减少了前期对各目标函数所占权重的人为制定,充分利用 NSGA-II 算法的全局收敛能力对算法进行改进,通过动态拥挤度以及自适应混合交叉算子保证解的多样性,有效提高解的质量。以西装上衣缝制生产环节作为实例,通过改进的 NSGA-II 算法进行实验研究,最终得到了有效的工序编排方案。

#### 参考文献:

- [1] 张旭靖,王立川,陈雁. 基于遗传算法的服装缝制生产线平衡优化[J]. 纺织学报,2020,41(2):125-129.
- [2] 吴树景,游有鹏,罗福源. 变邻域保优遗传算法求解柔性车间调度问题[J]. 计算机工程与应用,2020,56(22):236-243.
- [3] LI Xixing, PENG Zhao, DU Baigang, et al. Hybrid artificial bee colony algorithm with a rescheduling strategy for solving flexible job shop scheduling problems[J]. Computers & Industrial Engineering, 2017, 113:10-26.
- [4] 王雷,蔡劲草. 基于改进多种群遗传算法的柔性作业车间调度研究[J]. 云南大学学报:自然科学版,2017,39(2):192-199.
- [5] ZANDIEH M, KHATAMI A R, RAHMATI S H A. Flexible job shop scheduling under condition-based maintenance; improved version of imperialist competitive algorithm[J]. Applied Soft Computing, 2017, 58:449-464.
- [6] 杨振泰,黎向锋,左敦稳,等. 融合 Powell 搜索法的遗传算法求解柔性车间调度问题[J]. 现代制造工程,2019(9):41-47.
- [7] 张国辉,胡一凡,孙靖贺. 改进遗传算法求解多时间约束的柔性作业车间调度问题[J]. 工业工程,2020,23(2):19-25.
- [8] 刘小华,林杰,邓可. 基于遗传粒子群混合的可重入生产调度优化[J]. 同济大学学报:自然科学版,2011,39(5):726-730.
- [9] 胡桂武. 基于广义遗传粒子群优化算法的供应链优化求解[J]. 计算机应用,2008,28(11):2840-2843.
- [10] 朱光宇,贺利军. 基于灰熵并行分析优化算法的多目标流水车间调度[J]. 计算机工程,2015,41(10):165-170.
- [11] 景志强,王兆辉,高琦. 混合 NSGA-II 算法求解多目标柔性作业车间调度问题[J]. 组合机床与自动化加工技术,2019(7):138-140.
- [12] HLAL M I, RAMACHANDARAMURTHY V K, SARHAN A, et al. Optimum battery depth of discharge for off-grid solar PV/battery system[J]. Journal of Energy Storage, 2019, 26:100999.
- [13] YANG Shuai, SHAO Yunfeng, ZHANG Kai. An effective method for solving multiple travelling salesman problem based on NSGA-II[J]. Systems Science & Control Engineering, 2019, 7(2):108-116.
- [14] 时思思,张新燕,王志浩. 一种基于 DCD 和 a-tDX 改进的 NSGA-II 算法[J]. 计算机仿真,2019,36(12):257-262.
- [15] 王青松,谢兴生,周光临. 一种改进的非支配排序遗传算法[J]. 信息技术与网络安全,2019,38(5):28-32.
- [16] 董骏峰,王祥,梁昌勇. 基于个体邻域的改进 NSGA-II 算法[J]. 计算机工程与应用,2019,55(5):166-174.
- [17] 丁哲,秦臻,秦志光. 基于差分隐私的不确定数据频繁项集挖掘算法[J]. 计算机应用研究,2018,35(7):1942-1946.
- [18] WU H, CHEN H. Randomized response model versus mixture model in the crossover design for clinical trials[J]. Communications in Stats: Theory & Methods, 2016, 45(15):4611-4627.
- [19] BLUM A, LIGETT K, ROTH A. A learning theory approach to non-interactive database privacy[J]. Journal of the ACM, 2013, 60(2):1-25.
- [20] MCSHERRY F. Privacy integrated queries: an extensible platform for privacy-preserving data analysis[J]. Communications of the ACM, 2010, 53(9):89-97.
- [21] NGUYEN T, XIAO X, YIN Y, et al. Collecting and analyzing Multidimensional data with local differential privacy[J]. arXiv:1907.00782, 2016.
- [22] LI N, QARDAJI W, SU D, et al. PrivBasis: frequent itemset mining with differential privacy[J]. Proceedings of the VLDB Endowment, 2012, 5(11):1340-1351.
- [23] BURGESS J C, SHAKED T, RENSHAW E, et al. Learning to rank using gradient descent[C]//Proceedings of the twenty-second international conference on machine learning (ICML). Bonn, Germany: ACM, 2005:89-96.

(上接第 99 页)

2010:78-80.