

# 改进遗传算法求解多柔性作业车间调度问题

魏金生, 李仁旺

浙江理工大学机械工程学院, 浙江 杭州

收稿日期: 2023年10月17日; 录用日期: 2023年11月22日; 发布日期: 2023年11月29日

## 摘要

针对柔性作业车间调度问题, 考虑机器预维护的情况, 建立了最小化最大完工时间、关键机器负载以及机器总负载为目标的调度模型, 并提出一种改进遗传算法进行问题求解。根据问题特点, 设计了基于工序和机器的双层编码, 并采用三种种群初始化策略来生成初始种群。为了将解转化为可行有效的调度方案, 设计了考虑运输时间以及机器预维护的插入式解码。采用不同的个体更新方式, 并引入了变邻域搜索来提高算法的局部搜索能力, 以避免陷入局部最优解。最后, 在对基准算例改造并进行对比实验, 验证了所提算法的可行性与有效性。

## 关键词

柔性作业车间调度, 运输时间, 工件插入, 改进遗传算法, 机器预维护

# Improved Genetic Algorithm for Solving Multiple Flexible Job Shop Scheduling Problems

Jinsheng Wei, Renwang Li

School of Mechanical Engineering, Zhejiang University of Technology, Hangzhou Zhejiang

Received: Oct. 17<sup>th</sup>, 2023; accepted: Nov. 22<sup>nd</sup>, 2023; published: Nov. 29<sup>th</sup>, 2023

## Abstract

A scheduling model was established to minimize the maximum completion time, critical machine load, and total machine load for the flexible job shop scheduling problem, taking into account machine pre maintenance. An improved genetic algorithm was proposed to solve the problem. Based on the characteristics of the problem, a dual layer encoding based on process and machine was designed, and three population initialization strategies were used to generate the initial population. In order to transform the solution into a feasible and effective scheduling scheme, an inser-

tion decoding method was designed that takes into account transportation time and machine pre maintenance. Adopting different individual update methods and introducing variable neighborhood search to improve the algorithm's local search ability to avoid falling into local optima. Finally, the feasibility and effectiveness of the proposed algorithm were verified by modifying the benchmark calculation example and conducting comparative experiments.

## Keywords

Flexible Job Shop Scheduling, Transportation Time, Workpiece Insertion, Improved Genetic Algorithm, Machine Pre Maintenance

Copyright © 2023 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

柔性作业车间调度问题(Flexible Job-shop Scheduling Problem, FJSP)是经典作业车间调度问题的重要分支,相较于普通的车间调度, FJSP 消除了机器的唯一性, 更适合当前的生产环境[1]。FJSP 自从被提出以来, 研究人员提出了多种算法来求解 FJSP 问题的最优解, Calleja 和 Pastor 提出了一种具有一些优先级调度规则的调度算法[2], Li 和 Gao 提出了一种混合禁忌搜索(Tabu Search, TS)与遗传算法的方法来对 FJSP 进行优化[3], 郑捷, 潘大志设计了一种改进萤火虫算法[4], 李轲研究了柔性作业车间机器与 AGV 集成调度方法[5]。

以上算法都从不同角度对 FJSP 进行了一定的优化, 在实际生产过程中, 往往需要考虑很多环境因素, 如机器故障, 机器加工速度, 新工件插入, 运输时间等, Wang 等对考虑预维护活动时间固定和不固定的两种 FJSP 进行了研究[6], 包哲人等以能耗、完工时间、生产质量和质量为目标, 提出了一种改进的离散蝙蝠算法, Karimi 等针对工件运输时间提出了一种基于模拟退火的局部搜索与帝国主义竞争算法相结合的自适应算法进行求解, 张立果研究了工件插入的模糊 FJSP [7]。

综上所述, 本文考虑机器速度以及机器预维护的情况, 建立了最小化最大完工时间、关键机器负载以及机器总负载为目标的调度模型, 并提出一种改进遗传算法进行求解。根据问题特点, 设计了基于工序、机器的双层编码, 并采用三种种群初始化策略来生成初始种群。

## 2. 问题描述及数学模型

### 2.1. 符号定义

本文所涉及的变量符号及其含义如表 1 所示。

**Table 1.** Variable symbol definition

**表 1.** 变量符号定义

符号	含义
$N$	工件数量
$M$	机器数量
$j, k$	工件索引

## Continued

$j, h$	工序索引
$i, e$	机器索引
$e$	速度集索引
$h_j$	工件 $j$ 的工序数量
$O_{jh}$	工件 $j$ 的第 $h$ 道工序
$Q_i$	机器 $i$ 的预维护次数
$p_{ijh}$	$O_{jh}$ 在机器 $i$ 上的加工时间
$s_{ijh}$	$O_{jh}$ 在机器 $i$ 上的加工开始时间
$c_{ijh}$	$O_{jh}$ 在机器 $i$ 上的加工结束时间
$C_j$	工件 $j$ 的完工时间
$st_{iq}$	机器 $i$ 第 $q$ 次维护的开始时间
$ft_{iq}$	机器 $i$ 第 $q$ 次维护的结束时间
$DT_{iq}$	机器 $i$ 第 $q$ 次维护的持续时间
$ET_{iq}$	机器 $i$ 第 $q$ 次维护的最早开始时间
$LT_{iq}$	机器 $i$ 第 $q$ 次维护的最晚开始时间
$PN_i$	机器 $i$ 单位时间空载能耗
$TS_i$	机器 $i$ 的开机时间
$TC_i$	机器 $i$ 的关机时间
$x_{ijh}$	$\{0, 1\}$ , 如果工序 $O_{jh}$ 在机器 $i$ 上加工为 1 否则为 0

## 2.2. 问题描述

本文研究的问题描述为：车间内有  $n$  个工件以及  $m$  台机器，每个工件有固定的工序等待加工。每台机器有一系列已知的维护计划，当机器需要进行维护时，将无法执行加工操作。调度目标为最小化最大完工时间、关键机器负载以及机器总负载。假设如下：

- 1) 工件和机器在零时刻可用；
- 2) 同一机器单次只能加工一个工件；
- 3) 同一工件不能同时被多个机器加工；
- 4) 工序一经开始不能中断；
- 5) 工件工序之间存在顺序约束；

## 2.3. 模型建立

### 2.3.1. 优化目标

基于以上假设, 建立数学模型如下:

$$f_1 = \min \left( \max_{1 \leq j \leq n} (C_j) \right) \quad (1)$$

$$f_2 = \min \left( \max_{1 \leq j \leq m} \sum_{i=1}^n \sum_{h=1}^{h_j} P_{ijh} X_{ijke} \right) \quad (2)$$

$$f_3 = \min \left( \sum_{i=1}^m \sum_{j=1}^n \sum_{h=1}^{h_j} P_{ijh} X_{ijke} \right) \quad (3)$$

式(1)、(2)、(3)为优化目标, 分别表示最小化最大完工时间、关键机器负载以及机器总负载。

### 2.3.2. 约束条件

1) 一个工件的工序先后顺序约束。

$$s_{ijh} + x_{ijh} \times P_{ijh} \leq c_{jh} \quad (4)$$

式中:  $i=1,2,3,\dots,m$ ;  $j=1,2,3,\dots,n$ ;  $h=1,2,3,\dots,h_j$ 。

$$c_{jh} \leq s_{j(h+1)} \quad (5)$$

式中:  $j=1,2,3,\dots,n$ ;  $h=1,2,3,\dots,h_j-1$ 。

2) 工件完工时间约束。

$$c_{jh_j} \leq C_{\max} \quad (6)$$

式中:  $j=1,2,3,\dots,n$ 。

3) 同一时刻同一台机器只能加工一道工序。

$$s_{jh} + P_{ijh} \leq s_{kl} + L(1 - y_{ijhkl}) \quad (7)$$

式中:  $j=1,2,3,\dots,n$ ;  $k=1,2,3,\dots,n$ ;  $h=1,2,3,\dots,h_j$ ;  $l=1,2,3,\dots,h_k$ ;  $i=1,2,3,\dots,m$ 。

$$c_{jh} \leq s_{j(h+1)} + L(1 - y_{ikl(j=h)}) \quad (8)$$

式中:  $j=1,2,3,\dots,n$ ;  $k=1,2,3,\dots,n$ ;  $h=1,2,3,\dots,h_j-1$ ;  $l=1,2,3,\dots,h_k$ ;  $i=1,2,3,\dots,m$ 。

4) 同一时刻同一台机器只能加工一道工序。

$$\sum_{i=1}^{m_{jh}} x_{ijh} = 1 \quad (9)$$

式中:  $j=1,2,3,\dots,n$ ;  $h=1,2,3,\dots,h_j$ 。

5) 机器维护时间之间的关系。

$$ET_{iq} \leq st_{iq} \leq LT_{iq} \quad (10)$$

$$ft_{iq} \geq st_{iq} + DT_{kq} \quad (11)$$

## 3. 改进遗传算法求解 FJSP

遗传算法是一种优秀的全局并行求优搜索算法, 但其局部搜索能较差, 往往会陷入局部最优解。针

对这一问题本章将具有全局搜索能力的遗传算法与具有局部搜索能力的变邻域搜索算法进行优势组合, 同时结合改进的基于 Pareto 多目标优化方法, 设计一种基于 Pareto 多目标混合算法。算法步骤如下图 1:

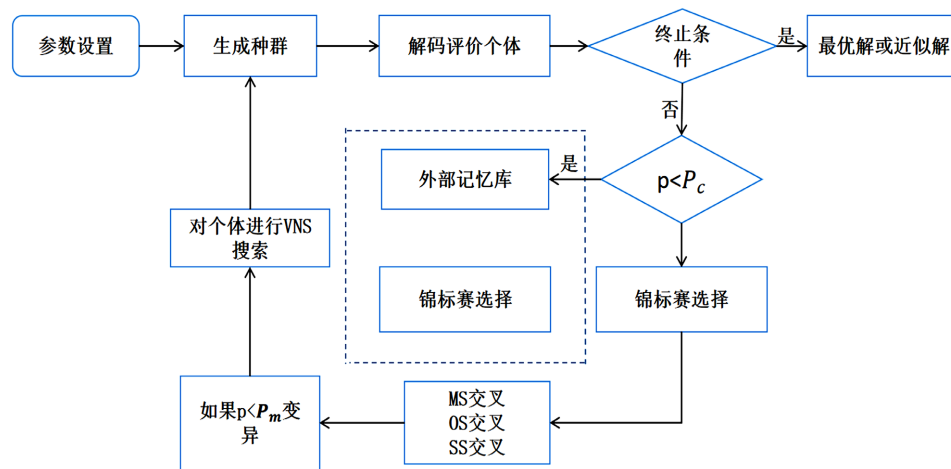


Figure 1. Algorithm flow chart  
图 1. 算法流程图

### 3.1. 编码与解码

本文采用工序排序(Operation Sequence, OS)和机器分配(Machine Assign, MA)双层编方式对解进行编码。OS 编码层由工件序号组成, 假设 OS 编码中某个位置编码值为 2, 且 2 在 OS 中第三次出现, 则表示工件 2 的第三道工序。MA 层编码由工件可用机器的索引组成, 并按照工件顺序依次编码[8]。以图 2 为例, 上层 OS 编码表示的加工顺序为  $O_{21}$ - $O_{11}$ - $O_{12}$ - $O_{31}$ - $O_{32}$ - $O_{22}$ ; 下层表示的是选择加工的机器在可选机器集中的序号, 例如工序  $O_{11}$  可以在机器  $M_1$  和  $M_2$  上加工, 染色体上的 1 表示选择可选机器集中的第一个, 也就是  $M_1$  进行加工。采用这种方式进行下层染色体的编码, 可以在编码时避免某些情况产生的“非法染色体”出现导致的求解失误。

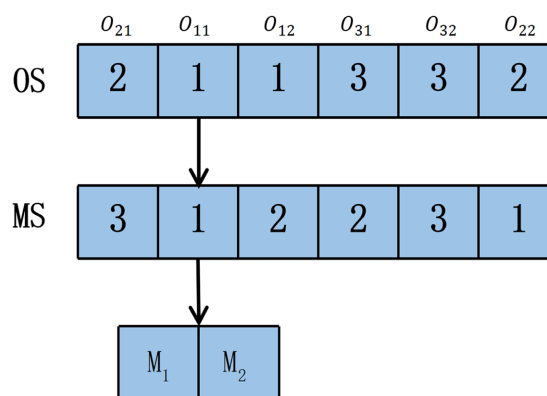


Figure 2. Encoding example  
图 2. 编码示例

针对问题的特点, 前文提到的编码方式不包含设备维护活动的信息, 因此我们在解码过程中考虑了设备维护的安排。本文在考虑运输时间的基础上引入了贪婪机制, 将工件插入到机器的空闲时段内进行加工, 以提高机器的利用率, 并进而提高生产效率和降低能源消耗。

### 3.2. 适应度计算

在多目标柔性作业车间调度问题的优化过程中, 适应值分配(fitness assignment)是一个关键问题。对于单目标问题, 解的适应度函数与目标函数是一致的, 可以方便地对解进行排序比较。然而, 在多目标优化中, 多个目标之间存在相互冲突的情况, 简单的排序评估无法有效衡量解的优劣, 因此需要综合考虑多个目标[9]。为了解决这个问题, 本文采用了一种快速非支配排序方法来对种群  $P$  进行分类, 将其划分为互不相交且具有支配关系的子群体  $P_1 \prec P_2 \prec \dots \prec P_r$ 。其中,  $S_p$  表示受个体  $p$  支配的集合,  $n_p$  表示支配个体  $p$  的个体数。具体执行步骤如下:

- 1) 创建两个空集合  $F$  和  $S$ ,  $F$  集合用来存放非支配个体,  $S$  集合存放每个个体所支配个体的集合。
- 2) 初始化  $n_p$  为 0, 初始化  $S_p$  为空集合, 前者表示  $p$  被支配的个体数, 后者用来存放  $p$  所支配的个体。
- 3) 对种群中  $P$  中的个体进行比较, 如果  $p$  支配  $q$ , 则将  $q$  添加到  $S_p$  集合,  $n_q$  随之增加 1, 反之, 若  $q$  支配  $p$ , 则将  $n_p$  加 1。
- 4) 对种群  $P$  中每个个体  $p$  进行判断, 如果  $n_p = 0$ , 则将  $p$  加入到  $F$  中, 否则加入到  $S$  中。
- 5) 重复步骤 3 和 4, 直到种群  $P$  中所有个体处理完毕。

### 3.3. 交叉原则

交叉操作在模拟生物进化过程中起到关键作用, 它是通过两个染色体进行交配和重组产生新染色体的过程。在遗传算法中, 交叉操作是一项核心技术。常用的交叉方法包括 GPPX、PBX、SPX、MPX、POX、LOX 等[10]。本文以 Brandimarte 所提出的 10 个案例中的 Mk10 案例进行测试, 以最大完工时间最小为目标函数, 最大进化次数为 200。分别选用以下不同的交叉算子进行测试 10 次, 测试结果如表 2 所示:

**Table 2.** Calculation results of different crossover methods in the Mk10 case  
**表 2.** 不同交叉方法在 Mk10 案例计算结果

序号	交叉方法	平均解
1	POX	218
2	MPX	218
3	SPX	220
4	GPPX	223
5	PBX	223
6	LOX	219

根据所采用的编码的特点, 本文使用两种不同的交叉操作。首先是基于工序编码的 POX 交叉操作。POX 交叉首先从父代  $P_1$  中选择一个或多个序号, 并将它们放置在子代  $C_1$  相同的位置。然后按照顺序将父代  $P_2$  中未被选择的序号依次填补到子代  $C_1$  中, 以满足特定条件。同样地, 基于父代  $P_2$  中被选中的序号生成子代  $C_2$ , 交叉过程如图 3 所示。第二种交叉操作是基于机器编码的 MPX 交叉操作 MPX 交叉操作的过程如下[11]: 首先随机生成一个与染色体长度相等的由 0 和 1 组成的集合  $R$ , 然后按照  $R$  中的 1 的位置顺序从  $P_1$  和  $P_2$  中选出相应的工序, 并交换它们的机器分配。 $P_1$  和  $P_2$  中的其他机器保留在子代中, 从而产生子代  $C_1$  和  $C_2$ , 交叉过程如图 3 所示。

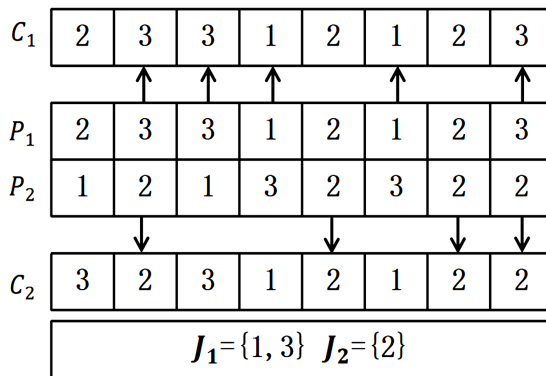


Figure 3. Schematic diagram of POX intersection  
图 3. POX 交叉示意图

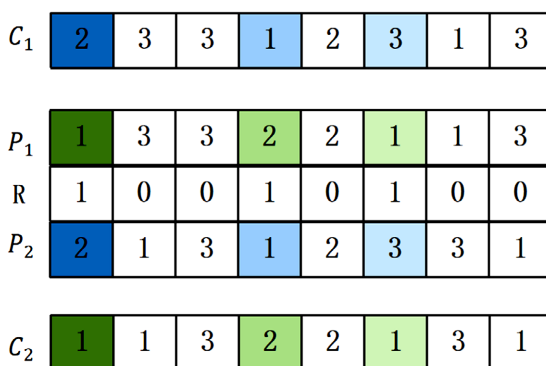


Figure 4. MPX cross schematic diagram  
图 4. MPX 交叉示意图

### 3.4. 种群多样性保持策略

多目标优化算法在进化过程中经常会收敛至单个解, 因此应尽可能得到一组分布均匀的非劣解集。本文主要采用两种方法来保证种群多样性, 动态交叉概率以及拥挤距离[12]。动态交叉是指在选择交叉个体时, 一部分是两个个体通过锦标赛选择得到, 另一部分来自记忆库和锦标赛选择。拥挤距离计算方法如下:

- 1) 初始化集合个体中每个个体的拥挤距离为 0。
- 2) 对于每个目标函数维度, 首先根据该维度对个体集合进行排序。
- 3) 对排序后的个体集合, 去除最大和最小个体, 计算其余个体在当前维度的拥挤距离, 设当前维度为  $k$ , 个体  $P_i$  的拥挤距离计算方法如下:

$$D(P_i) = D(P_i) + \frac{k(P_{i+1} - P_{i-1})}{k^{\max} - k^{\min}} \quad (12)$$

- 4) 归一化处理所有个体的拥挤距离, 除以集合的规模, 确保拥挤距离在合理的范围内。

### 3.5. 精英保留策略

目前, 多目标遗传算法迭代过程中精英个体的保留方式主要有两种, 一种是将新旧种群合并进而选择个体, 另一种方法是将算法进化过程中搜索到的非劣解保存到外部记忆库。在进化过程中合理的利用非劣解可以加快算法的收敛速度。本文采用第二种方法。对每一次产生的新种群进行评价排序, 然后对外部记忆库进行更新。在每次更新过程中, 进行如下操作来处理新解和记忆库中的解[13]:

- 1) 针对新解, 检查是否有记忆库中的某个解支配它。如果有, 则丢弃该新解。
- 2) 如果新解支配了记忆库中的某个解, 则用新解替换掉记忆库中被支配的解。
- 3) 如果新解与记忆库中的解互相之间没有支配关系, 即它们是非支配的, 则将新解加入记忆库。

## 4. 算法测试与结果分析

### 4.1. 对比实验

为了验证算法的性能, 采用一些数据集进行测试, 采用的基准算例包括  $8 \times 8$ 、 $10 \times 10$  及  $15 \times 10$  三个测试算例。运行参数为: 种群规模  $P = 200$ , GS 比例为 0.5, LS 比例为 0.3, RS 比例为 0.2。种群的最大迭代次数  $G$  为 200, 变异概率  $P_m = 0.02$ 。结果对比如下所示。

在  $8 \times 8$  问题中, 涉及到 8 台机器和 8 个工件, 总共涉及 27 道工序。平均而言, 每道工序有多台机器可供选择加工, 实验数据如表 3 所示。

**Table 3.**  $8 \times 8$  Experimental data

**表 3.**  $8 \times 8$  实验数据

问题	目标函数	AL + CGA		POS + SA		MOGV			
$8 \times 8$	$f_1$	13	15	15	16	15	14	12	10
	$f_2$	/	/	13	13	10	11	10	12
	$f_3$	78	77	75	74	73	72	74	75

在  $10 \times 10$  问题中, 涉及到 10 台机器和 10 个工件, 共涉及 30 道工序。每道工序都有 10 台机器可供选择加工, 实验数据如表 4 所示。

**Table 4.**  $10 \times 10$  Experimental data

**表 4.**  $10 \times 10$  实验数据

问题	目标函数	AL + CGA		POS + SA			MOGV				
$8 \times 8$	$f_1$	9	8	8	7	7	9	7	6	6	5
	$f_2$	5	5	7	6	5	6	6	5	6	7
	$f_3$	44	45	40	41	43	42	40	41	41	40

在  $15 \times 10$  问题中, 涉及到 15 台机器和 10 个工件, 总共涉及 56 道工序。每道工序都有 10 台机器可供选择加工, 实验数据如表 5 所示。

**Table 5.**  $12 \times 10$  Experimental data

**表 5.**  $12 \times 10$  实验数据

问题	目标函数	AL + CGA		POS + SA		MOGV		
$15 \times 10$	$f_1$	25		11		11		10
	$f_2$	10		12		10		11
	$f_3$	90		91		90		90

由以上结果可知, 本文改进遗传算法在算法稳定性, 搜索能力上均有较好的表现。



## 4.2. 结集分布性评价

解集的分布性是评价多目标优化算法的重要指标, 通常用分布度  $D$  与变化率  $R$  来表示。为验证本文算法的解集分布性, 用本文算法对 Dpdata 实例进行求解, 并与 Kacem 的求解结果进行对比[13]。在 Dpdata 实例中, 有 18 个问题, 这些问题涉及工件数量从 10 到 20, 机器数量从 5 到 10 的不同组合。每组问题中, 工序数在 5 到 25 之间变化。程序会连续运行 20 次。平均结果如表 6 所示。

$$D = \sum_{i=1}^P \frac{|d - \bar{d}|}{P} \quad (13)$$

$$R = \sum_{i=1}^N \frac{|D_i - \bar{D}|}{N} \quad (14)$$

式中:  $d$  表示 Pareto 非支配解集中相邻两个个体之间的欧几里得距离,  $\bar{d}$  代表  $d$  的平均值,  $D_i$  为每次求得的分度  $D$ ,  $\bar{D}$  代表  $D_i$  的平均值,  $P$  表示非支配解集中解的个数,  $N$  为运行次数。

**Table 6.** Comparison of algorithm distribution

**表 6.** 算法分布性对比

Kacem			MOGV		
非劣解个数	$D$	$R$	非劣解个数	$D$	$R$
10	0.745	0.0352	9	0.523	0.0221
9	0.821	0.0324	8	0.541	0.0234
9	0.756	0.0321	10	0.546	0.0241
11	0.744	0.0423	8	0.581	0.0232
13	0.732	0.0416	9	0.601	0.0254
15	0.830	0.0396	10	0.611	0.0236
13	0.765	0.0354	9	0.526	0.0245
12	0.761	0.0342	9	0.543	0.0256
12	0.842	0.0351	10	0.623	0.0284
15	0.753	0.0362	8	0.514	0.0261
13	0.841	0.0345	10	0.533	0.0265
14	0.764	0.0413	10	0.623	0.0274
14	0.752	0.0332	8	0.545	0.0263
16	0.731	0.0314	9	0.534	0.0245
13	0.696	0.0352	8	0.562	0.0273
15	0.736	0.0341	10	0.571	0.0236
12	0.843	0.0403	9	0.569	0.0246
12	0.725	0.0318	10	0.582	0.0251

由上述结果可知, 本文算法相对于 Kacem 具有更好的分布性以及稳定性。

## 5. 结语

本文针对传统的柔性作业车间调度问题引入了运输时间和机器预维护这两种实际因素,并在优化完工周期的基础上,进一步考虑了能源消耗目标。研究工作主要集中在考虑运输时间与机器预维护的柔性作业车间绿色调度问题。首先,建立了一个调度模型,旨在优化最小化最大完工时间、关键机器负载以及机器总负载。随后,设计了一种基于 Pareto 排序的多目标遗传算法以解决该问题。在该多目标遗传算法中,采用了多种方法来生成能够平衡加工时间和能耗的初始种群。此外,构建了带有预维护动态调整策略和考虑运输时间的贪婪插入解码方法,并与最早预维护策略以及最晚预维护策略进行了对比,以验证所提出的预维护计划安排策略的有效性。针对不同的解情况,设计了不同的个体更新方式。最后,通过与 AL + CGA、POS + SA 算法在不同规模的测试算例下进行对比实验,验证了本文设计的多目标遗传算法的有效性。需要指出的是,本文所研究的问题模型尚不完善,与实际生产情况存在一定差距,尚未考虑到机器故障和紧急订单等情况。这些方面将在后续的研究中得到进一步完善,以使模型更贴近实际情况。

## 基金项目

浙江省 2023 年度“尖兵”“领雁”研发攻关计划(2022C01SA111123),国家自然科学基金资助项目(51475434)。

## 参考文献

- [1] 周尔民, 马畅, 刘宁. 考虑机器故障的柔性作业车间动态调度[J]. 组合机床与自动化加工技术, 2023(9): 188-192.
- [2] 曲鹏举, 唐向红. 改进粒子群算法在柔性车间调度问题的研究[J/OL]. 机械设计与制造: 1-6. <https://doi.org/10.19356/j.cnki.1001-3997.20230905.006>, 2023-10-17.
- [3] 唐红涛, 沈毅, 张伟, 等. 改进鲸鱼算法求解分布式装配柔性作业车间生产与配送联合调度问题[J]. 计算机应用研究, 2023, 40(7): 1982-1990. <https://doi.org/10.19734/j.issn.1001-3695.2022.12.0780>
- [4] 郑捷, 潘大志. 改进萤火虫算法解多目标柔性作业车间调度[J/OL]. 控制工程: 1-9. <https://doi.org/10.14107/j.cnki.kzgc.20210770>, 2023-10-17.
- [5] 李轲. 柔性作业车间机器与 AGV 集成调度方法研究[D]: [硕士学位论文]. 西安: 西安理工大学, 2023.
- [6] Wang, S. and Yu, J. (2010) An Effective Heuristic for Flexible Job-Shop Scheduling Problem with Maintenance Activities. *Computers & Industrial Engineering*, **59**, 436-447. <https://doi.org/10.1016/j.cie.2010.05.016>
- [7] 张立果. 基于改进遗传算法的柔性作业车间调度优化研究[D]: [硕士学位论文]. 南京: 南京航空航天大学, 2021. <https://doi.org/10.27239/d.cnki.gnhhu.2020.001632>
- [8] 吴日夫. 超启发式算法求解柔性作业车间调度问题研究[D]: [硕士学位论文]. 长春: 吉林大学, 2023. <https://doi.org/10.27162/d.cnki.gilin.2023.005376>
- [9] 王玉芳, 陈凡, 姚彬彬, 等. 改进文化基因算法求解双资源约束柔性作业车间调度问题[J/OL]. 控制工程: 1-15. <https://doi.org/10.14107/j.cnki.kzgc.20230088>, 2023-10-17.
- [10] 李彬. 基于改进遗传算法的多柔性作业车间调度问题研究[D]: [硕士学位论文]. 漳州: 闽南师范大学, 2023.
- [11] 王焯. 基于改进 NSGA-III 的低碳多目标柔性作业车间调度问题研究[D]: [硕士学位论文]. 昆明: 云南大学, 2022. <https://doi.org/10.27456/d.cnki.gyndu.2021.001875>
- [12] 陈逸维, 刘华秋, 陈洪涛, 等. 基于双种群遗传算法的模具车间作业调度研究[J]. 组合机床与自动化加工技术, 2023(8): 183-187. <https://doi.org/10.13462/j.cnki.mmtamt.2023.08.039>
- [13] 高亮, 张国辉, 王晓娟. 柔性作业车间调度智能算法及其应用[M]. 武汉: 华中科技大学出版社, 2012.