

# 控制与决策

Control and Decision

双种群混合遗传算法求解具有预防性维护的分布式柔性作业车间调度问题

李佳磊, 顾幸生

引用本文:

李佳磊, 顾幸生. 双种群混合遗传算法求解具有预防性维护的分布式柔性作业车间调度问题[J]. *控制与决策*, 2023, 38(2): 475–482.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2021.1152>

---

您可能感兴趣的其他文章

Articles you may be interested in

[带工序跳跃的绿色混合流水车间机器与AGV联合调度](#)

Joint scheduling of machines and AGVs in green hybrid flow shop with missing operations

*控制与决策*. 2022, 37(10): 2723–2732 <https://doi.org/10.13195/j.kzyjc.2021.0318>

[区间数可重入混合流水车间调度与预维护协同优化](#)

Collaborative optimization of interval number reentrant hybrid flow shop scheduling and preventive maintenance

*控制与决策*. 2021, 36(11): 2599–2608 <https://doi.org/10.13195/j.kzyjc.2020.0973>

[基于改进蛙跳算法的分布式两阶段混合流水车间调度](#)

An improved shuffled frog leaping algorithm for the distributed two-stage hybrid flow shop scheduling

*控制与决策*. 2021, 36(1): 241–248 <https://doi.org/10.13195/j.kzyjc.2019.0472>

[基于多班教学优化的多目标分布式混合流水车间调度](#)

Multi-class teaching-learning-based optimization for multi-objective distributed hybrid flow shop scheduling

*控制与决策*. 2021, 36(2): 303–313 <https://doi.org/10.13195/j.kzyjc.2020.0549>

[基于预防维护的单机调度问题](#)

Single-machine scheduling problem with preventative maintenance activities

*控制与决策*. 2021, 36(2): 395–402 <https://doi.org/10.13195/j.kzyjc.2019.0626>

# 双种群混合遗传算法求解具有预防性维护的 分布式柔性作业车间调度问题

李佳磊, 顾幸生<sup>†</sup>

(华东理工大学 能源化工过程智能制造教育部重点实验室, 上海 200030)

**摘要:** 在实际生产过程中, 生产调度和设备维护相互影响, 因此两者应该统筹优化. 为研究具有预防性维护的分布式柔性作业车间调度问题, 以最小化最大完工时间为目标, 提出一种双种群混合遗传算法. 结合问题特性, 设计三维编码以及对应的机器解码方案, 采用不同的策略初始化种群以均衡一部分工厂负载, 为双种群设计不同的交叉变异算子提高算法的多样性, 并利用交换精英解的方法实现两个种群的协作优化, 同时针对关键工厂和预防性维护操作设计相应的局部搜索. 最后对比现有算法, 在同构和异构工厂的算例上进行实验, 使用正交试验法优化算法参数设置. 实验结果验证了局部搜索以及种群协作的有效性和双种群混合遗传算法求解具有预防性维护的分布式柔性作业车间调度问题的优越性.

**关键词:** 生产调度; 预防性维护; 分布式; 柔性作业车间; 双种群; 遗传算法

中图分类号: TP18

文献标志码: A

DOI: 10.13195/j.kzyjc.2021.1152

**引用格式:** 李佳磊, 顾幸生. 双种群混合遗传算法求解具有预防性维护的分布式柔性作业车间调度问题[J]. 控制与决策, 2023, 38(2): 475-482.

## Two-population hybrid genetic algorithm for distributed flexible job-shop scheduling problem with preventive maintenance

LI Jia-lei, GU Xing-sheng<sup>†</sup>

(Key Laboratory of Smart Manufacturing in Energy Chemical Process of Ministry of Education, East China University of Science and Technology, Shanghai 200030, China)

**Abstract:** In actual production, production scheduling and equipment maintenance affect each other, and they should be optimized together. In order to study the distributed flexible job-shop scheduling problem with preventive maintenance, a hybrid genetic algorithm with two-population optimization mechanism is proposed to minimize the maximum completion time. Combined with the characteristics of the problem, the 3-D encoding and the corresponding decoding scheme are designed. Different strategies are used to initialize the population to balance part of the factory load. Different crossover and mutation operators are designed for the two populations to improve the diversity of the algorithm. The cooperative optimization of two populations is realized using the method of exchanging elite solutions. Local searches are designed for critical factories and preventive maintenance operations. Finally, compared with the existing algorithms, experiments on homogeneous and heterogeneous factories are carried out, and orthogonal test method is used to optimize the parameter setting of the algorithm. The simulation results verify the effectiveness of local search and population cooperation and the superiority of the two-population hybrid genetic algorithm to solve the distributed flexible job shop scheduling problem with preventive maintenance.

**Keywords:** production scheduling; preventive maintenance; distributed; flexible job shop; two-population; genetic algorithm

## 0 引言

高效的生产调度和设备维护技术对提高设备利用率、降低生产成本、提高企业竞争力等具有重要

的作用<sup>[1]</sup>. 分布式柔性作业车间调度问题(distributed flexible job-shop scheduling problem, DFJSP)是柔性作业车间调度问题(flexible job-shop scheduling

收稿日期: 2021-07-02; 录用日期: 2021-09-22.

基金项目: 国家自然科学基金项目(61973120, 62076095, 61673175, 61573144).

责任编辑: 王凌.

<sup>†</sup>通讯作者. E-mail: xsgu@ecust.edu.cn.

problem, FJSP)的一个扩展. 相比FJSP, DFJSP涉及多个工厂协同调度, 具有更高的复杂性. 而在加工过程中, 工件生产会增加设备的役龄, 需要对设备进行包括润滑、清洁和更换等预防性的维护任务 (preventive maintenance, PM)<sup>[2]</sup>, 这会使得设备一段时间内不可用, 从而影响整个调度计划. 因此, 将DFJSP和PM看作一个整体, 进行统筹优化提高生产效益是十分必要的.

国内外已有学者在生产调度和预防性维护集成优化方面进行了相关研究, 其中预防性维护大致可以分为周期性和基于设备状态或可靠性两大类, 针对前者已有很多学者进行了研究<sup>[3-8]</sup>, 近年来也出现不少对于后者的研究. 针对流水车间, Miyata等<sup>[9]</sup>和Jomaa等<sup>[10]</sup>分别采用启发式算法和变邻域搜索算法进行求解. 针对作业车间, Berrichi等<sup>[11]</sup>设计了基于蚁群优化和预防性维护联合的调度算法求解多目标FJSP; Zandieh等<sup>[12]</sup>结合改进帝国主义竞争和模拟退火算法求解具有设备维护的FJSP. 针对分布式柔性作业车间和预防性维护计划的联合调度问题, 目前只有少数学者进行了研究, Chan等<sup>[13]</sup>提出了一种带有显性基因的遗传算法 (genetic algorithm with dominant genes, GADG) 解决该问题; Chung等<sup>[14]</sup>在文献[13]算法基础上增加局部搜索机制, 设计了对应的改进遗传算法 (modified genetic algorithm with dominant genes, MGADG); Khalid等<sup>[15]</sup>采用改进免疫算法 (improved immune algorithm, IIA) 解决该问题; Lin等<sup>[16]</sup>设计了具有不完全染色体和阴影染色体的遗传算法, 称为GA-S2. 与上述3种算法相比, 利用两种启发式规则进行工厂和机器解码能够在实验算例上得到目前的最优值.

上述预防性维护方法相较第1种方法更加贴近实际, 计划的制定也更复杂, 因此采用该方法研究具有预防性维护的分布式柔性作业车间调度问题 (distributed flexible job-shop scheduling problem with preventive maintenance, DFJSP-PM). 已知的DFJSP-PM研究仅考虑了同构工厂下的情况, 即每个工厂机器配置一样, 故本文旨在研究同构和异构工厂环境下的DFJSP-PM. 遗传算法作为求解组合优化问题的经典算法, 已在DFJSP上取得了一定的成果<sup>[17-19]</sup>. 标准的遗传算法模拟人类进化过程, 由于演化、环境等局限性, 长时间后种群特性不会发生很大的变化. 而双种群遗传算法能够在两个种群进行不同进化的同时交换种群中的个体来打破种群内的状态跳出局部最优. 由此, 本文以最小化最大完工时间为目标, 提出一

种双种群混合遗传算法求解DFJSP-PM, 并通过对比实验验证算法的有效性.

## 1 模型建立

### 1.1 符号定义

本文相关符号定义如下:

$f$ : 工厂索引号,  $f = 1, 2, \dots, F$ ,  $F$ 为工厂总数;

$i$ : 工件索引号,  $i = 1, 2, \dots, I$ ,  $I$ 为工件总数;

$j$ : 工序索引,  $j = 1, 2, \dots, N_i$ ,  $N_i$ 为工件 $i$ 的总工序数;

$h$ : 加工机器索引号,  $h = 1, 2, \dots, H_f$ ,  $H_f$ 为工厂 $f$ 内的机器数;

$J_i$ : 工件 $i$ ;

$U_f$ : 工厂 $f$ ;

$M$ : 最大机器年龄;

$C_i$ : 工件 $i$ 的完工时间;

$O_{ij}$ : 工件 $i$ 的第 $j$ 道工序;

$S_{ij}$ :  $O_{ij}$ 的起始加工时间;

$E_{ij}$ :  $O_{ij}$ 的结束时间;

$M_{fh}$ : 工厂 $f$ 的第 $h$ 台机器;

$D_f$ : 工厂 $f$ 到共同目的地的运输时间;

$T_{ijfh}$ :  $O_{ij}$ 在工厂 $f$ 机器 $h$ 上的加工时间;

$Eq_{fh}$ : 工厂 $f$ 中第 $h$ 台机器的当前机器年龄;

$P_{ijfh}$ : 工厂 $f$ 中机器 $h$ 在 $O_{ij}$ 操作后进行维护的时间;

$\delta_{ijfh}$ :  $O_{ij}$ 在工厂 $f$ 中机器 $h$ 上加工则为1, 否则为0;

$\gamma_{ijfh}$ : 工厂 $f$ 中的机器 $h$ 在 $O_{ij}$ 操作后进行维护则为1, 否则为0.

### 1.2 问题描述

DFJSP-PM可以这样描述:  $I$ 个包含 $N_i$ 道工序的工件需要被分配到 $F$ 个不同的工厂加工. 加工完的工件从每个工厂移动到一个共同的目的地需要不同的运输时间. 每个工厂有 $H_f$ 台不同的加工机器, 每台机器都有一个最大机器年龄 $M$ , 定义为机器的累计加工时间. 如果当前操作后机器年龄达到 $M$ , 则必须在该操作完成后立即进行维护, 维护时间的长短与机器年龄有关. 因此, DFJSP-PM包含4个子问题: 工件-工厂的分配、工序排序、工序-机器的分配和PM的分配.

一个具有3工件和2工厂的DFJSP-PM例子如表1所示. 数字表示对应工序在对应机器上的操作时间, “-”表示该机器不能加工该工序. 采用Chan等<sup>[13]</sup>提出的机器年龄与维护时间关系函数模型, 设定当 $0 < Eq_{fh} \leq 60$ 时, 维护时间为150; 当 $60 < Eq_{fh} \leq$

140时,维护时间为  $150 + Eq_{fh} + 4 - 60$ ;当  $140 < Eq_{fh} \leq 200$ 时,维护时间为  $300 + Eq_{fh} - 140$ ;当  $Eq_{fh} > 200$ 时,维护时间为720,最大机器年龄为200.

表1 3工件和2工厂的DFJSP-PM

	工厂1			工厂2		
	$D_1$	$M_{11}$	$M_{12}$	$D_2$	$M_{21}$	$M_{22}$
$J_1$	$O_{11}$	80	70	50	80	
	$O_{12}$	60	30	70	50	
	$O_{13}$	-	90	60	50	
$J_2$	$O_{21}$	10	70	50	20	-
	$O_{22}$	40	80	40	80	
$J_3$	$O_{31}$	30	-	60	40	
	$O_{32}$	40	50	70	-	
	$O_{33}$	50	60	70	30	

1.3 模型建立

大多数企业在生产调度环节中最希望能够尽快完成生产任务,因此本文以最小化最大完工时间(makespan)为目标即适应度函数,作出如下假设.

假设1 在0时刻所有工厂中的机器均可用,所有工件均可被加工;

假设2 在同一工厂中完成的工件必须一起运输到目的地;

假设3 一台机器一次只能处理一个作业,且不能中断作业的处理;

假设4 每台机器加工同一道工序的加工时间不同,且不是所有机器都能处理每道工序;

假设5 在每次PM操作后,机器年龄将重置为0,且每台机器的初始机器年龄均为0;

假设6 PM操作只能在机器完成一个工序后立即执行.

基于上述假设,以最小化 makespan 为优化目标的DFJSP-PM模型如下:

$$\min\{\max_{i=1}^N C_i\}; \tag{1}$$

$$C_i = E_{iN_i} + D_f, \tag{2}$$

$$E_{ij} = S_{ij} + \sum_{f=1}^F \sum_{h=1}^{H_f} \delta_{ijfh} T_{ijfh} \sum_{f=1}^F \sum_{h=1}^{H_f} \gamma_{ijfh} P_{ijfh}, \tag{3}$$

$$S_{ij} \geq E_{i(j-1)}, \tag{4}$$

$$\sum_{j=1}^{N_i} \sum_{h=1}^{H_f} \delta_{ijfh} \in \{0, N_i\}, \tag{5}$$

$$\sum_{f=1}^F \sum_{h=1}^{H_f} \delta_{ijfh} = 1. \tag{6}$$

式(1)为目标函数,最大完工时间是所有工件完工时间的最大值;式(2)表示工件的完工时间是最后一道工序完工时间与运输时间之和;式(3)表示工序的完工时间等于开始加工时间加上实际加工时间和PM操作时间;式(4)表示同一工件内的工序必须按顺序加工;式(5)表示每个工件只能分配给一个工厂;式(6)表示每个工序只能选择一个工厂里的一台机器加工.

2 双种群混合遗传算法求解DFJSP-PM

2.1 编码解码

本文设计的三维编码分别由工序排序向量(operation sequence vector, OV)、工厂选择向量(factory selection vector, FV)和PM选择向量(preventive maintenance selection vector, PV)组成,3个向量的长度均等于所有工件的工序数之和.对应表1示例,其中一个可行解的编码如图1所示.

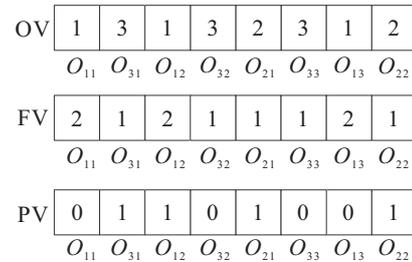


图1 编码表示

OV由该工序对应工件索引号表示,工件索引号第几次出现就代表该工件的第几道工序;FV代表对应OV工序加工的工厂索引号,图1中FV的第1个元素为2,与之对应的OV为1,表示 $O_{11}$ 在 $U_2$ 进行加工.由于需要确保同一工件的所有工序都在同一工厂加工,OV中相同工件索引号对应的FV工厂索引号要一致;PV由0和1组成,同样也与OV的元素位一一对应.0代表该工序完成后不需要进行PM操作,1代表该工序完成后立即进行PM操作.

工序-机器分配通过启发式规则解码得到,以表1例子和图1编码为例,具体操作如下.

step 1: 由FV可知, $U_1$ 中OV编码为33232,PV编码为10101; $U_2$ 中OV编码为111,PV编码为010.

step 2: 对每个工厂分别进行解码.在 $U_1$ 中,第1个加工的 $O_{31}$ 只能在 $M_{11}$ 上进行加工,而 $O_{31}$ 对应的PV为1,故 $O_{31}$ 加工完成后要进行机器维护. $M_{11}$ 当前机器年龄为30,根据机器年龄与维护时间关系的函数模型关系式可知, $M_{11}$ 上的PM虚拟操作的时间为150.更新每台机器可以开始的加工时间,并将 $Eq_{11}$ 置0. $O_{32}$ 可以在 $M_{11}$ 和 $M_{12}$ 上处理,选择 $M_{12}$ 是

因为各机器加上  $O_{32}$  的操作时间后有更小的机器负载。 $O_{32}$  对应的PV为0,所以该操作后不进行PM操作,更新每台机器的当前机器年龄和可以开始加工的时间. 以此类推,最后得到所有工厂的调度结果.

### 2.2 初始化

初始化种群质量的好坏对遗传算法的求解质量有着较大影响,本文提出一种初始化方法,在保证初始化充分随机性的基础上均衡一部分工厂负载. 分别对两个种群进行相同的初始化操作,采用工序随机排序方法生成OV,随机生成01序列组成PV. FV采用如下策略生成:50%FV通过随机为工件分配工厂生成. 50%FV通过规则生成,规则具体操作如下.

step 1: 根据OV中每个工件第1道工序出现的顺序得出工件的加工顺序.

step 2: 计算每个工件在每个工厂中的平均加工时间.

step 3: 根据加工顺序依次为工件分配工厂,始终选择加上当前工件加工时间后负载最小的工厂. 若存在多选时,则在待选工厂内随机选择. 一旦确定加工该工件的工厂,则FV编码也随之确定.

### 2.3 双种群交叉变异算子

为便于表述,将两个种群分别记为种群1和种群2,种群1的交叉变异记为交叉1、变异1,种群2的交叉变异记为交叉2、变异2. 交叉变异是产生新个体的操作,为提高算法的多样性和搜索效率,针对两个种群设计不同的交叉和变异算子.

#### 2.3.1 交叉1

交叉1采用POX交叉,父代  $P_1$ 、 $P_2$  生成子代  $C_1$ 、 $C_2$ ,具体操作如下.

step 1: 将工件  $J_1, J_2, \dots, J_I$  随机分成两个集合  $Job_1$  和  $Job_2$ .

step 2: 将  $P_1$ 、 $P_2$  的OV中属于  $Job_1$  的元素直接复制到  $C_1$ 、 $C_2$  的相应位置,FV和PV按照对应的OV位置一一对应变化.

step 3: 将  $P_2$ 、 $P_1$  的OV中属于  $Job_2$  的元素按照  $P_2$ 、 $P_1$  原来的顺序依次填入  $C_1$ 、 $C_2$ ,FV和PV按照对应的OV位置一一对应变化.

step 4: 选择  $P_1$ 、 $P_2$ 、 $C_1$ 、 $C_2$  中适应度值最小的两个数据作为下一步操作的父代  $P_{11}$ 、 $P_{22}$ .

以表1数据为例,交叉1的可能操作过程如图2所示.

#### 2.3.2 变异1

变异1保持OV和FV不变,仅对PV进行操作,具体步骤如下.

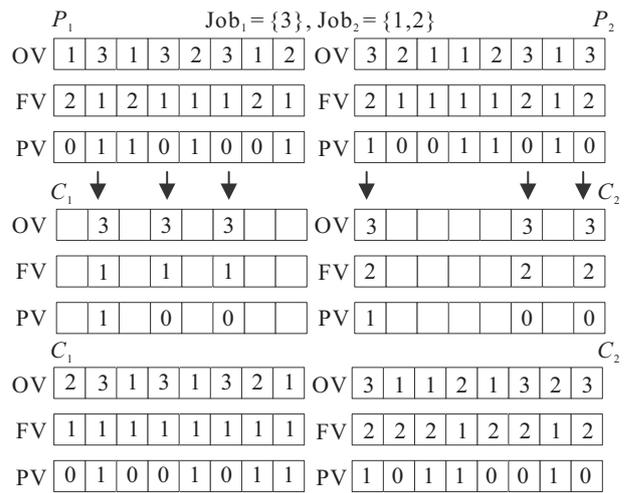


图2 交叉1的操作

step 1: 随机选择  $P_{11}$ PV 编码的一个基因位改变其PM标志生成  $C_{11}$ ,  $P_{22}$  进行同样操作生成  $C_{22}$ .

step 2: 选择  $P_{11}$ 、 $P_{22}$ 、 $C_{11}$ 、 $C_{22}$  中目标函数值最小的两个数据作为新的父代.

#### 2.3.3 交叉2

交叉2设计类似两点交叉的方式由父代  $P_1$ 、 $P_2$  生成子代  $C_1$ 、 $C_2$ ,具体步骤如下.

step 1: 在  $P_1$ 、 $P_2$  上随机选择两点  $r_1$ 、 $r_2$ .

step 2: 将  $P_1$ 、 $P_2$  中属于  $[r_1, r_2]$  范围内的基因位直接保留到  $C_2$ 、 $C_1$ .

step 3: 将  $P_1$ 、 $P_2$  中不属于  $[r_1, r_2]$  范围内的基因位按顺序填入  $C_1$ 、 $C_2$  的空白处.

step 4: 对上述步骤生成的子代OV和FV编码进行非法值检查. 若OV中数字出现的次数超出对应工

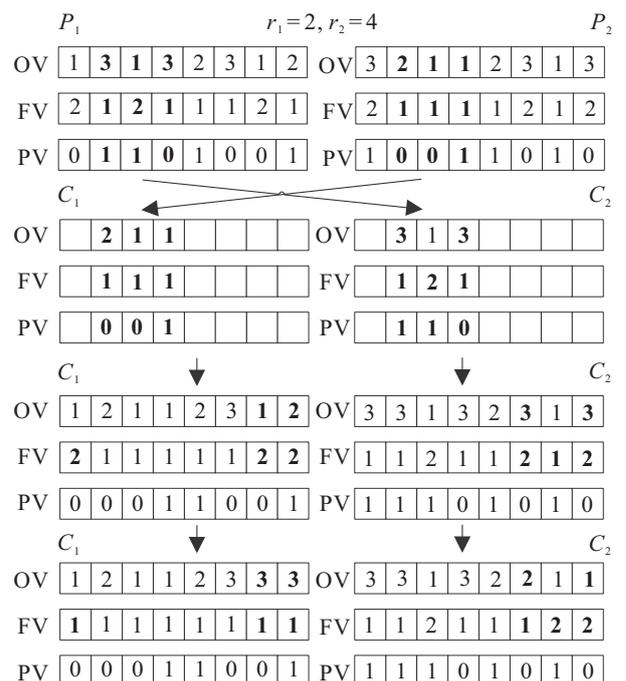


图3 交叉2的操作

件的工序数则从剩余可加工的工序中随机选择进行替换; FV需保证 $C_1, C_2$ 在 $[r_1, r_2]$ 位置范围内的OV和FV对应关系与 $P_2, P_1$ 相同, 其余位置的OV和FV对应关系与 $P_1, P_2$ 相同.

step 5: 选择 $P_1, P_2, C_1, C_2$ 中适应度值最小的两个作为下一步操作的父代 $P_{11}, P_{22}$ .

以表1为例, 交叉2的可能操作过程如图3所示.

### 2.3.4 变异2

变异2保持OV和PV不变, 仅对FV进行操作, 具体步骤如下.

step 1: 重新生成 $P_{11}$ 的FV编码, 即再次为工件随机分配工厂, 由此生成子代 $C_{11}$ . 父代 $P_{22}$ 进行同样操作生成子代 $C_{22}$ .

step 2: 选择 $P_{11}, P_{22}, C_{11}, C_{22}$ 中目标函数值最小的两个数据作为新的父代.

## 2.4 局部搜索

为提高算法的局部搜索能力, 基于关键工厂和PM操作提出两种局部搜索方法. 由于局部搜索在提高算法寻优性能的同时会增加算法的运行时间, 因此仅针对每个种群适应度前 $\text{Popsizel}2*\text{Ph}$ 个体进行搜索, 其中种群数量 $\text{Popsizel}$ 和寻优系数 $\text{Ph}$ 的取值将在第3.2节讨论.

### 2.4.1 基于关键工厂的局部搜索

从解码方法可以看出, 整个生产过程的最大完工时间取决于完工时间最大的工厂, 称为关键工厂. 显然, 只有关键工厂的加工时间提前, 调度方案性能才能得到改善, 因此基于关键工厂设计局部搜索. 具体操作如下.

step 1: 计算每个工厂的完工时间, 得出关键工厂以及关键工厂中的加工顺序.

step 2: 交换关键工厂中工件号相异的工序加工顺序, FV和PV根据OV的变换相应进行交换可得 $C_1$ . 若 $C_1$ 的完工时间小于 $P_1$ , 则用 $C_1$ 替换 $P_1$ , 否则保持 $P_1$ 不变.

step 3: 依次交换关键工厂中所有工件号相异的工序加工顺序, 按step 2的过程替换或保留原有结果.

### 2.4.2 基于PM的局部搜索

DFJSP-PM不仅涉及到DFJSP, PM的操作对于调度结果影响也很大, 越是后面加工的工序PM操作就越非必要, 但一味地减少PM操作又会使得机器役龄超过最大机器年龄而造成PM操作时间过长. 由此, 本文以一定的概率减少非必要的PM操作以缩短完工时间, 具体操作如下.

step 1: 以每个工厂为单位, 提取OV、FV和PV对

应的编码, 保持元素前后顺序不变, 并计算出相应子编码的长度 $\text{len}_f$ .

step 2: 针对每个PV子编码中的非0位, 依次将 $n/\text{len}_f$ 的概率置0, 其中 $n$ 表示该元素位在子编码中的第 $n$ 位.

step 3: 比较改变非0位前后的结果, 若改变后的完工时间优于改变前则进行替换, 否则保留原编码不变.

## 2.5 种群协作

交换种群之间优秀个体所携带的遗传信息可以打破单个种群进化的局限性, 从而提高种群解的质量, 故设计交换精英个体的操作用于种群交流. 具体步骤如下: 在种群1和种群2各自进行交叉、变异以及局部搜索后, 对种群1和种群2中的个体按适应度排序, 分别从两个种群中选择前 $\text{Popsizel}2*\text{Ph}$ 个体进行交换, 形成新的种群1和新的种群2, 进行下一次迭代.

## 2.6 算法整体框架

本文提出用于求解DFJSP-PM的双种群混合遗传算法称为HGA-PM. 两种群采用不同的交叉变异算子各自探索, 并在每次迭代的最后分别依次进行基于关键工厂的局部搜索和基于PM的局部搜索后再进行种群协作. 算法具体流程如图4所示.

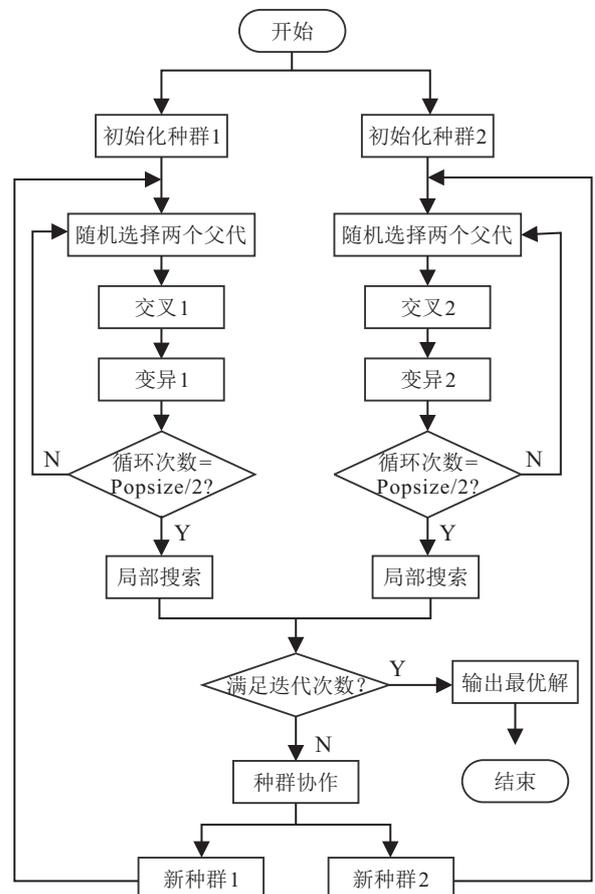


图4 HGA-PM的算法流程

### 3 实验结果与分析

为验证HGA-PM求解DFJSP-PM的性能,进行相关对比实验.首先介绍实验1和实验2算例;然后采用正交实验法对HGA-PM中的参数进行优化配置;最后将目前已有算法作为对比验证HGA-PM在不同工厂配置下的求解性能.此外,将缺少关键工厂局部搜索的算法命名为HGA11-PM,缺少PM局部搜索的算法命名为HGA12-PM,用于对比局部搜索的有效性;缺少种群协作的算法命名为HGA2-PM,用于对比种群协作的有效性.实验的运行环境为: Intel Core i7-8550U CPU 1.8 GHz, RAM 16 GB, Windows 10, 64位操作系统,编程语言为C.

#### 3.1 算例介绍

实验1为Chan等<sup>[13]</sup>提出的10工件2工厂同构算例,对如下4个算法进行对比分析: GADG<sup>[13]</sup>、MGADG<sup>[14]</sup>、IIA<sup>[15]</sup>、GA-S2<sup>[16]</sup>.由于IIA原文采用的机器年龄和维护时间函数与其他算法相异,为保证公平性,在其他参数设置与原文相同的情况下,采用Chan等<sup>[13]</sup>提出的函数关系复现IIA.

目前没有DFJSP-PM异构工厂的标准算例,本文生成2个工厂下10~30工件的新测试算例供使用.新生成算例命名为“P工件数”.每个工件的工序数都是[5,7]的随机数,工厂内机器数是[5,10]的随机数,工序加工时间为[10,30]的随机数.各工厂具备加工所有工件的能力,但并不是每个机器都能够加工所有工序.工厂1和工厂2到公共目的地的运输时间分别为10和20.

实验2仍然选用Chan等<sup>[13]</sup>提出的机器年龄和维护时间函数模型,以实验1中表现较好的IIA<sup>[15]</sup>和GA-S2<sup>[16]</sup>作为对比算法进行实验.

#### 3.2 参数设置

HGA-PM中包括种群数量Popszie、迭代次数Generations、交叉1概率 $P_{c1}$ 、变异1概率 $P_{m1}$ 、交叉2概率 $P_{c2}$ 、变异2概率 $P_{m2}$ 、寻优概率Ph七个参数.由于实验2均在Popszie = 50, Generations = 500的情况下进行对比,采用正交实验法对 $P_{c1}$ 、 $P_{m1}$ 、 $P_{c2}$ 、 $P_{m2}$ 、Ph五个参数进行优化配置,每个参数定为4个水平.

以P20为例,设定Popszie = 50, Generations = 500进行正交实验.为避免结果的随机性,每组实验运行50次,将50次运行结果的平均值作为最终结果列于表2,表2最后4行表示参数在不同水平下的实验均值.由表2可知, $P_{c1}$ 、 $P_{m2}$ 、Ph对结果的影响较大,

其中Ph的值越大算法的性能越好,但计算成本会增加.综合考虑算法的性能效果以及耗费的时间成本,选取HGA-PM参数为 $P_{c1} = 0.9$ ,  $P_{m1} = 0.7$ ,  $P_{c2} = 0.8$ ,  $P_{m2} = 0.8$ ,  $Ph = 0.2$ .

表2 参数设计实验结果

实验号	参数					实验结果
	$P_{c1}$	$P_{m1}$	$P_{c2}$	$P_{m2}$	Ph	
1	0.95	0.90	0.95	0.90	0.08	115.30
2	0.95	0.80	0.90	0.80	0.12	115.42
3	0.95	0.70	0.85	0.70	0.16	115.40
4	0.95	0.60	0.80	0.60	0.20	115.20
5	0.90	0.90	0.90	0.70	0.20	115.42
6	0.90	0.80	0.95	0.60	0.16	115.42
7	0.90	0.70	0.80	0.90	0.12	114.74
8	0.90	0.60	0.85	0.80	0.08	115.64
9	0.85	0.90	0.85	0.60	0.12	116.42
10	0.85	0.80	0.80	0.70	0.08	116.28
11	0.85	0.70	0.95	0.80	0.20	114.26
12	0.85	0.60	0.90	0.90	0.16	115.00
13	0.80	0.90	0.80	0.80	0.16	114.72
14	0.80	0.80	0.85	0.90	0.20	115.04
15	0.80	0.70	0.90	0.60	0.08	116.88
16	0.80	0.60	0.95	0.70	0.12	116.02
Avg	116.75	115.47	115.25	115.02	116.03	
Avg	<b>115.31</b>	115.54	115.68	<b>115.01</b>	115.65	
Avg	115.49	<b>115.32</b>	115.63	115.78	115.14	
Avg	115.67	115.47	<b>115.24</b>	115.98	<b>114.98</b>	

#### 3.3 结果分析

表3为实验1对比结果,表4和表5为实验2对比结果.其中Best为50次运行结果的最优值,Avg为50次运行结果的平均值, $T(s)$ 表示运行一次的平均计算时间, $Dev \% = Deviation/Avg$ 表示相对标准误差,Deviation表示50次运行结果的标准偏差.

表3 实验1对比结果

	Best	Avg
GADG <sup>[13]</sup> (Popszie=100, Generations = 5 000)	1 220	1 280
MGADG <sup>[14]</sup> (Popszie=50, Generations = 5 000)	930	—
IIA (Popszie=50, Generations=300)	922	1 045.6
GA-S2 <sup>[16]</sup> (Popszie=50, Generations=300)	594	<b>665.2</b>
HGA-PM (Popszie=50, Generations=300)	<b>584</b>	685.4

表4 验证局部搜索和种群协作有效性的对比结果

	HGA11-PM				HGA12-PM				HGA2-PM				HGA-PM			
	Best	Avg	Dev%	T(s)	Best	Avg	Dev%	T(s)	Best	Avg	Dev%	T(s)	Best	Avg	Dev%	T(s)
P10	<b>118</b>	120.6	0.0	0.7	<b>118</b>	120.1	0.0	6.9	120	128.1	0.0	9.6	<b>118</b>	118.8	0.0	10.1
P15	112	116.4	0.0	1.3	111	113.3	0.0	41.9	114	124.0	0.1	63.3	<b>109</b>	112.1	0.0	65.8
P20	160	273.1	0.3	1.6	156	256.8	0.3	119.4	174	298.3	0.1	136.5	<b>149</b>	234.0	0.3	139.7
P25	409	468.7	0.1	2.1	374	430.9	0.1	223.5	370	423.0	0.1	249.0	<b>367</b>	387.2	0.1	251.3
P30	388	450.4	0.1	2.6	369	437.4	0.1	438.6	370	433.4	0.1	428.2	<b>365</b>	390.3	0.1	457.4

表5 HGA-PM与其他算法对比结果

	IIA				GA-S2				HGA-PM			
	Best	Avg	Dev%	T(s)	Best	Avg	Dev%	T(s)	Best	Avg	Dev%	T(s)
P10	433	439.1	0.0	5.4	<b>118</b>	121.1	0.0	3.6	<b>118</b>	118.8	0.0	10.1
P15	467	531.2	0.0	14.3	111	116.1	0.0	12.8	<b>109</b>	112.1	0.0	65.8
P20	785	823.1	0.4	16.6	168	311.4	0.7	22.3	<b>149</b>	234.0	0.3	139.7
P25	1090	1174.8	0.0	30.7	404	545.5	0.1	42.2	<b>367</b>	387.2	0.1	251.3
P30	1119	1191.4	0.0	30.6	505	644.9	0.1	64.9	<b>365</b>	390.3	0.1	457.4

由表3可见, HGA-PM能够更新目前已知的最优值, 且能得到比GADG、IIA更好的平均值, 但稳定性略劣于GA-S2. 由此可见, HGA-PM在解决同构工厂的DFJSP-PM方面具有较好的竞争性. 由于MGADG均值在原文中没有给出, 此处不作比较.

由表4的HGA11-PM和HGA-PM对比可见, 除了P10外, HGA-PM都能找到比HGA11-PM更好的最优解, 且HGA-PM的平均值优于HGA11-PM. 故基于关键工厂的局部搜索在增加一些计算时间的情况下, 能够提高算法的寻优性能和稳定性; 而由HGA12-PM和HGA-PM的对比同样可以看出, 除P10外, HGA-PM都能找到比HGA12-PM更好的最优解, HGA-PM的平均值均优于HGA12-PM, 且两者运算时间相差不大, 由此验证了基于PM操作局部搜索的有效性. 由HGA2-PM和HGA-PM对比结果可见, HGA-PM最优解和均值均优于HGA2-PM, 两者Dev%相差不大, 表明缺少种群协作的算法具有一定的稳定性, 但易陷入局部最优, 由此验证了种群协作的有效性.

由表5与其他算法的对比结果可见, IIA虽然算法稳定但易陷入局部最优, 不能搜索到较好的结果. GA-S2在P10能搜索到与HGA-PM一样的最优值, 但平均值略大于所提出算法. 随着算例规模的增大, GA-S2相较于HGA-PM更易陷入局部最优. 图5为P20算例下3种算法最优值收敛曲线, 可以看出500次迭代下, 各个算法都能够收敛, HGA-PM前期收敛速度以及寻优能力相较于其他算法均表现最好. 由此可见, 相较于已有算法, HGA-PM在牺牲一定

计算时间下, 具有较好的性能.

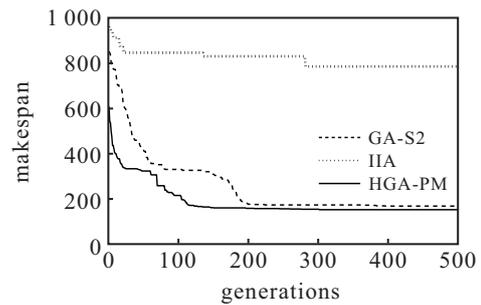


图5 P20算例下对比算法收敛图

综上所述, 实验1验证了HGA-PM在求解同构工厂DFJSP-PM的竞争性, 实验2证实了异构工厂DFJSP-PM下HGA-PM算法中局部搜索和种群协作的有效性以及与其他算法相比的优越性. 因为双种群采用不同的交叉变异算子增强了算法的全局搜索能力, 每次迭代之后的局部搜索使得算法的寻优能力和稳定性得到提高, 而种群协作使得两个种群进行协作优化, 不易陷入局部最优值.

### 4 结论

本文研究了双种群混合遗传算法求解具有预防性维护的分布式柔性作业车间调度问题. 首先, 基于问题特点提出了三维编码以及相应的解码方案; 然后, 设计了包括种群初始化方法、双种群不同的交叉变异操作、局部搜索以及种群协作等操作; 最后在实验对比中验证了种群协作的有效性和所提算法在求解同构、异构工厂环境下DFJSP-PM的优越性.

就目前而言, 关于具有预防性维护的分布式柔性

作业车间调度问题的研究还有待进一步探索,大致包括:1)寻找新的算法或混合算法求解DFJSP-PM;2)考虑多目标的DFJSP-PM.

#### 参考文献(References)

- [1] 马慧民, 叶健飞. 柔性车间调度与设备维护的联合优化研究[J]. 机械设计与制造, 2015(7): 248-250.  
(Ma H M, Ye J F. Joint optimization of production scheduling and maintenance planning in the flexible job-shop problem[J]. Machinery Design & Manufacture, 2015(7): 248-250.)
- [2] Wong C S, Chan F T S, Chung S H. A joint production scheduling approach considering multiple resources and preventive maintenance tasks[J]. International Journal of Production Research, 2013, 51(3): 883-896.
- [3] Xu S L, Dong W Q, Jin M Z, et al. Single-machine scheduling with fixed or flexible maintenance[J]. Computers & Industrial Engineering, 2020, 139: 106203.
- [4] Zhou H M, Tsai Y C, Wu F C, et al. Improved approaches to minimize the makespan on single-machine scheduling with periodic preventive maintenance activities[J]. Mathematical Problems in Engineering, 2020, 2020: 8548463.
- [5] Najat A, Yuan C, Gursel S, et al. Minimizing the number of tardy jobs on identical parallel machines subject to periodic maintenance[J]. Procedia Manufacturing, 2019, 38: 1409-1416.
- [6] Li J Q, Pan Q K, Tasgetiren M F. A discrete artificial bee colony algorithm for the multi-objective flexible job-shop scheduling problem with maintenance activities[J]. Applied Mathematical Modelling, 2014, 38(3): 1111-1132.
- [7] 朱传军, 宋文家, 张超勇, 等. 基于维修时间窗的柔性作业车间调度优化研究[J]. 中国机械工程, 2016, 27(10): 1337-1343.  
(Zhu C J, Song W J, Zhang C Y, et al. Research on optimization of FJSP based on maintenance time window[J]. China Mechanical Engineering, 2016, 27(10): 1337-1343.)
- [8] El Khoukhi F, Boukachour J, El Hilali Alaoui A. The "Dual-Ants Colony": A novel hybrid approach for the flexible job shop scheduling problem with preventive maintenance[J]. Computers & Industrial Engineering, 2017, 106: 236-255.
- [9] Miyata H H, Nagano M S, Gupta J N D. Integrating preventive maintenance activities to the no-wait flow shop scheduling problem with dependent-sequence setup times and makespan minimization[J]. Computers & Industrial Engineering, 2019, 135: 79-104.
- [10] Jomaa W, Eddaly M, Jarbouli B. Variable neighborhood search algorithms for the permutation flowshop scheduling problem with the preventive maintenance[J]. Operational Research, 2021, 21(4): 2525-2542.
- [11] Berrichi A, Yalaoui F, Amodeo L, et al. Bi-objective ant colony optimization approach to optimize production and maintenance scheduling[J]. Computers & Operations Research, 2010, 37(9): 1584-1596.
- [12] 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.
- [13] Chan F T S, Chung S H, Chan L Y, et al. Solving distributed FMS scheduling problems subject to maintenance: Genetic algorithms approach[J]. Robotics and Computer-Integrated Manufacturing, 2006, 22(5/6): 493-504.
- [14] Chung S H, Chan F T S, Chan H K. A modified genetic algorithm approach for scheduling of perfect maintenance in distributed production scheduling[J]. Engineering Applications of Artificial Intelligence, 2009, 22(7): 1005-1014.
- [15] Khalid M N A, Yusof U K. Improved immune algorithm for optimizing distributed production scheduling problem in flexible manufacturing system subject to machine maintenance[J]. International Journal of Mathematics and Computers in Simulation, 2014, 8: 19-30.
- [16] Lin C S, Lee I L, Wu M C. Merits of using chromosome representations and shadow chromosomes in genetic algorithms for solving scheduling problems[J]. Robotics and Computer-Integrated Manufacturing, 2019, 58: 196-207.
- [17] Chang H C, Liu T K. Optimisation of distributed manufacturing flexible job shop scheduling by using hybrid genetic algorithms[J]. Journal of Intelligent Manufacturing, 2017, 28(8): 1973-1986.
- [18] Wu M C, Lin C S, Lin C H, et al. Effects of different chromosome representations in developing genetic algorithms to solve DFJS scheduling problems[J]. Computers & Operations Research, 2017, 80: 101-112.
- [19] Lu P H, Wu M C, Tan H, et al. A genetic algorithm embedded with a concise chromosome representation for distributed and flexible job-shop scheduling problems[J]. Journal of Intelligent Manufacturing, 2018, 29(1): 19-34.

#### 作者简介

李佳磊(1996—),女,硕士生,从事生产调度与智能优化的研究, E-mail: 18752018058@163.com;

顾幸生(1960—),男,教授,博士生导师,从事生产计划与调度、复杂工业过程建模控制与优化、故障检测与诊断等研究, E-mail: xsgu@ecust.edu.cn.

(责任编辑: 郑晓蕾)