

# 控制与决策

Control and Decision

## 带二次装夹约束的可重构柔性作业车间调度优化研究

杨志杰, 梁穆熙, 李益兵, 汪开普, 梅立文, 郭顺生

引用本文:

杨志杰, 梁穆熙, 李益兵, 等. 带二次装夹约束的可重构柔性作业车间调度优化研究[J]. *控制与决策*, 2026, 41(1): 67-80.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2025.0807>

---

### 您可能感兴趣的其他文章

#### Articles you may be interested in

##### [基于粒子群算法的满载需求可拆分车辆路径规划](#)

Split vehicle route planning with full load demand based on particle swarm optimization

*控制与决策*. 2021, 36(6): 1397-1406 <https://doi.org/10.13195/j.kzyjc.2019.1323>

##### [基于最佳相位匹配的减摇鳍低航速控制策略](#)

[Best phase matching based control strategy for ship stabilizer at low speed](#)

*控制与决策*. 2021, 36(4): 940-946 <https://doi.org/10.13195/j.kzyjc.2019.1028>

##### [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>

##### [顺序依赖的调整时间和拖期的无缝钢管热轧批量调度算法](#)

Hot-rolled batch scheduling algorithm for seamless steel tube with sequence-dependent setup times and tardiness

*控制与决策*. 2021, 36(2): 505-512 <https://doi.org/10.13195/j.kzyjc.2019.0723>

##### [基于机床超低待机状态的流水车间能耗调度](#)

Energy consumption scheduling in flow shop based on ultra-low idle state of numerical control machine tools

*控制与决策*. 2021, 36(1): 143-151 <https://doi.org/10.13195/j.kzyjc.2019.0433>

# 带二次装夹约束的可重构柔性作业车间调度优化研究

杨志杰<sup>1</sup>, 梁穆熙<sup>1</sup>, 李益兵<sup>1,2†</sup>, 汪开普<sup>1,2</sup>, 梅立文<sup>3</sup>, 郭顺生<sup>1,2</sup>

(1. 武汉理工大学机电工程学院, 武汉 430070; 2. 数字制造湖北省重点实验室, 武汉 430070;  
3. 郑州众城润滑科技有限公司, 郑州 450000)

**摘要:** 可重构柔性作业车间凭借其动态适应能力能够快速响应市场变化, 然而在加工部分具备复杂曲面的工件时, 难以一次完成加工, 需拆卸工件后进行二次装夹, 且依赖于人工操作, 而工人资源的约束会限制该操作的并行性进而降低生产系统的柔性. 鉴于此, 以最小化最大完工时间和工人数量为目标, 建立考虑二次装夹的可重构柔性作业车间调度问题的混合整数线性规划模型, 设计一种改进的多目标遗传算法. 为处理复杂约束, 采用三向量编码结构; 为加速种群收敛, 设计一种混合贪婪初始化策略; 为提高算法的局部探索能力, 设计 3 种基于关键操作的邻域搜索策略. 通过 15 个标准算例验证所提出算子的有效性, 并与 4 种算法对比验证所提出算法的优越性. 最后将所提出模型和算法应用到汽车差速器制造过程, 相较于传统调度方案, 所得方案将最大完工时间缩短 56%.

**关键词:** 二次装夹; 工人资源约束; 可重构柔性作业车间调度; 整数线性规划; 多目标遗传算法

中图分类号: TH165; TP18 文献标志码: A

DOI: 10.13195/j.kzyjc.2025.0807

引用格式: 杨志杰, 梁穆熙, 李益兵, 等. 带二次装夹约束的可重构柔性作业车间调度优化研究 [J]. 控制与决策, 2026, 41(1): 67-80.

## Optimization of reconfigurable flexible job shop scheduling considering secondary clamping constraints

YANG Zhi-jie<sup>1</sup>, LIANG Mu-xi<sup>1</sup>, LI Yi-bing<sup>1,2†</sup>, WANG Kai-pu<sup>1,2</sup>, MEI Li-wen<sup>3</sup>, GUO Shun-sheng<sup>1,2</sup>

(1. School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan 430070, China;  
2. Hubei Key Laboratory of Digital Manufacturing, Wuhan 430070, China; 3. Zhengzhou Zhongcheng Lubrication Technology Co., Ltd, Zhengzhou 450000, China)

**Abstract:** Reconfigurable flexible job shops are endowed with dynamic adaptability to respond rapidly to market changes. However, when machining workpieces with complex surfaces, completion of all operations in a single setup is infeasible; workpiece removal and secondary clamping are required, and their dependence on manual operations constrains the parallelism of clamping tasks, thereby reducing system flexibility. To address this, a mixed-integer linear programming model is formulated for the reconfigurable flexible job shop scheduling problem with secondary clamping, with makespan and the number of workers set as optimization objectives. An improved multi-objective genetic algorithm is developed. A three-vector encoding structure is adopted to handle complex constraints effectively. A hybrid greedy initialization strategy is designed to accelerate population convergence. Three neighborhood search strategies based on key operations are introduced to enhance local search capability. Validation on 15 benchmark instances confirms the effectiveness of the proposed operators, and comparison with four advanced algorithms demonstrates superior performance of the developed method. Application to an automotive differential manufacturing process shows that, relative to traditional scheduling schemes, the proposed approach reduces makespan by 56%.

**Keywords:** secondary clamping; worker resource constraints; reconfigurable flexible job shop scheduling; mixed-integer linear programming; multi-objective genetic algorithm

## 0 引言

近年来, 客户对产品的定制化需求日益凸显, 为响应市场需求, 制造业企业的生产模式正加速向多

品种、小批量化方向转型<sup>[1]</sup>. 提高生产的柔性是企业应对动态多变市场需求的有效方式, 因此, 柔性作业车间调度受到了学者们的广泛研究. 其中可重构柔

收稿日期: 2025-08-02; 录用日期: 2025-09-17.

基金项目: 国家自然科学基金项目 (52375510, 52575595).

责任编辑: 李新宇.

†通信作者. E-mail: ahlyb@whut.edu.cn.

性作业车间 (reconfigurable flexible job shop, RFJS) 通过集成可重构机床与可更换辅助模块, 能够显著提升生产系统的动态适应能力, 引起众多学者的关注<sup>[2]</sup>. 在 RFJS 中, 工件加工过程的工艺约束往往呈现出多维协同优化的特点, 如加工顺序、装夹方案与资源调度等方面的决策. 在加工制造高端装备时 (如航空航天零件加工、精密模具制造等)<sup>[3]</sup>, 由于工件几何特征的复杂性和加工精度的严苛要求, 特别是在面对复杂轴类零件时很难一次将所有表面全部加工完成, 需拆卸后通过二次装夹 (secondary clamping, SC) 以完成零件的其他表面加工, 并且通常需要工人操作. 这种现象源于工件加工特征的空间分布特性, 当加工基准面发生转换或需要加工正交特征面时, 必须通过重新定位夹持基准来保证形位公差, 这使得单次装夹无法完成全部加工任务.

可重构制造系统 (reconfigurable manufacturing system, RMS) 作为一种先进的制造模式, 因其动态适应能力逐渐被企业采用<sup>[1]</sup>, 通常包含可重构机器 (reconfigurable manufacturing tools, RMT) 和一组可替换的辅助模块 (auxiliary module, AM)<sup>[4-5]</sup>, AM 可以在不同的 RMT 上进行切换以实现不同需求<sup>[6]</sup>. 而在柔性作业车间调度问题中, AM 通常被认为是一个有限的资源, 这也导致将有限的 AM 分配给 RMT 在生产调度中起着至关重要的作用, 此时调度问题的复杂性将呈指数级增长<sup>[7-8]</sup>. 这种由多个可重构单元组合组成的车间可以被称为 RFJS, 它也是 RMS 的主要形式之一<sup>[9]</sup>, 以此为背景所开展的调度问题研究称为可重构柔性作业车间调度问题 (reconfigurable flexible job shop problem, RFJSP). Mahmoodjanloo 等<sup>[10]</sup> 提出了刻画机器重构过程的配置相关准备时间 (CDST) 概念, 将融合机器重构因素的 FJSP 建模为 FJSP-CDST 问题, 开发了相应的混合整数线性规划模型, 并设计了一种改进的差分进化算法来求解此问题. Fan 等<sup>[7]</sup> 针对考虑辅助模块的柔性作业车间调度问题, 建立了最小化总加权延迟的混合整数规划模型, 并开发了一种改进遗传算法予以求解. 上述研究主要考虑 RFJS 的可重构特性对生产调度所产生的影响, 但忽视了 RFJS 在进行复杂工件加工时的二次装夹现象.

在现代工业制造中, 针对复杂几何与高精度要求零件 (尤其是轴类零件), 二次装夹是保证多面加工定位精度的必要环节, 直接影响加工一致性与良品率<sup>[11]</sup>. 在二次装夹的实际操作过程中, 操作流程为“取下、翻转工件、重新装夹、定位确认”, 整个步骤通常由工人配合机床测头测量后手动进行<sup>[12]</sup>, 且依

赖人工操作完成返夹与对齐<sup>[13]</sup>. 这种人工依赖性在模型中表现为: 二次装夹需要工人操作, 且该工人同时只能执行单一装夹任务而车间工人总数固定. 这些约束直接导致装夹任务排队等待, 显著制约了系统柔性, 成为现有研究中尚未充分解决的瓶颈问题.

工人作为重要的生产资源, 在实际的生产制造中很常见, 并且对生产调度有着巨大影响<sup>[14]</sup>. 近年来, RMS 与工人结合的调度问题正吸引越来越多研究者的关注. 王凌等<sup>[15]</sup> 研究人机协同能效车间调度问题, 在传统的人机独立生产模式中增加人机资源配置, 并提出了一种群智能协同优化算法进行求解. 胡金昌等<sup>[16]</sup> 研究单人作业车间调度问题, 提出一种基于贪婪的模拟退火算法进行求解, 并在该问题中考虑了工人重复操作产生的学习效应. 张佳朋等<sup>[17]</sup> 研究考虑工人多技能的可重构装配车间调度问题, 使用超启发式算法对问题求解, 在某航天企业卫星装配车间的实例应用验证了算法的优越性. Li 等<sup>[18]</sup> 研究设备有限、工艺复杂及动态事件耦合的动态可重构车间调度问题, 并提出了一种基于多智能体深度强化学习的动态调度方法进行求解. 综上, 工人资源在 RMS 调度中的集成研究已开展较多, 学者们通过引入工人技能差异、工人学习遗忘曲线、人机协作机制、动态事件响应等维度, 显著提升了调度模型的实践性.

多目标的车间调度问题相较于基础的单目标车间调度问题更符合实际的生产情况, 也具有更高的复杂度<sup>[19]</sup>, 这时传统的调度方法难以同时满足多目标协同优化的要求. 刘超等<sup>[20]</sup> 研究了以最大完工时间和总能耗为目标具有自动导引车的柔性作业车间调度问题, 并使用多目标协同进化算法进行求解. 苏建涛等<sup>[21]</sup> 研究以最小化最大完工时间和机器负载为目标的混合流水车间重调度问题, 并使用 NSGA-II 算法进行求解, 该算法使用启发式规则和随机化的方法生成初始解, 加快了收敛种群速度. 赵仕存等<sup>[22]</sup> 设计了以强化学习驱动进化的模因算法, 求解以最小化完工时间、能量消耗和总生产成本为目标的分分布式柔性作业车间调度问题. 张梓琪等<sup>[23]</sup> 针对低碳分布式装配流水车间问题, 提出了以最小化总能耗和总完工时间为优化目标的多维分布估计算法 (MEDA). 付威等<sup>[24]</sup> 提出了一种新的禁忌搜索与快速非支配排序遗传融合算法 TSNSGA-II, 可以高效求解以最小化最大完工时间、总加工成本和总加工能耗为目标的多车间作业任务协同调度问题. 可以发现, 多目标遗传算法配合有效的初始化规则和邻域搜索, 能够加快种群的收敛速度, 扩大解空间的范

围,是一种有效的求解策略.

可以看出,尽管工人资源与可重构制造系统的协同优化已在调度研究中取得丰富成果,但二次装夹对工人的依赖性及其在可重构柔性作业车间调度中的影响尚未得到系统研究.基于此,本文研究具有二次装夹约束的可重构柔性作业车间调度问题(reconfigurable flexible job shop scheduling problem with secondary clamping, RFJSP-SC),以最小化最大完工时间和工人数量为目标,建立其混合整数线性规划模型.针对问题的特性,设计一种改进的多目标遗传算法用于该问题的求解,设计一种混合贪婪初始化策略加快种群收敛速度,并使用基于关键操作的邻域结构提高算法的局部搜索能力.最后通过实验验证所提出方法的有效性.

### 1 问题描述及数学模型

#### 1.1 问题描述

本文所讨论的 RFJSP-SC 模型描述如下:车间中有一组  $m$  台机器  $M = \{M_1, M_2, \dots, M_m\}$ ,  $l$  个辅助模块  $A = \{A_1, A_2, \dots, A_l\}$ , 现需对  $n$  个工件进行生产, 每个工件包含  $h_i$  个工序  $J = \{1, 2, \dots, h_i\}$ , 需按照给定的顺序进行加工, 加工每道工序  $O_{ij}$  ( $i = 1, 2, \dots, n, j = 1, 2, \dots, h_i$ ) 的可选机器集合为  $M_i \subseteq M$ , 以给定的顺序一个接一个执行, 操作  $O_{ij}$  可以由一组具有可替换辅助模块  $A_i$  的机器  $M_i \subseteq M$  处理, 但部分工件的工序必须要辅助模块进行加工处理, 并且处理时间根据不同的机器配置而变化. 操作完成后, 用于处理此操作的 AM 可以从机器上分离, 并可用于其他操作. 有  $d$  名工人  $W = \{W_1, W_2, \dots, W_d\}$ , 需要对部分工件的部分工序进

行二次装夹, 二次装夹的开始时间和持续时间由工件的加工工艺决定.

按照车间实际工作情况, 还要满足以下条件:

- 1) 各种作业没有重要性层次;
- 2) 同一作业各种操作之间存在优先级约束, 但不同作业操作之间不存在优先级;
- 3) 每个工人一次只对一个工件进行二次装夹;
- 4) 二次装夹操作需要在机器上进行;
- 5) 每个操作只能在一台机器上进行;
- 6) 每个操作过程都不能被中断;
- 7) 从 0 时刻开始所有机器和辅助模块都可用;
- 8) 每个辅助模块只能同时安装到一台机器上.

图 1 表现了二次装夹和辅助模块在柔性作业车间调度问题的甘特图中所产生的影响. 图 1 中,  $O_{11}$ ,  $O_{12}$ ,  $O_{32}$  和  $O_{22}$  都需要进行二次装夹, 例如  $M_3$  的工序  $O_{32}$  加工过程中, 需要工人  $W_1$  进行二次装夹后才能继续加工其他表面. 由于同一工人同一时间只能对一个工件进行二次装夹,  $O_{32}$  需要等待  $M_2$  上工人  $W_1$  完成工序  $O_{12}$ , 同理,  $M_1$  的工序  $O_{22}$  需要等  $O_{32}$  的工人完成二次装夹才能进行.  $O_{13}$  在  $O_{32}$  工序后加工是因为两道工序都需要使用辅助模块  $A_1$ ,  $O_{13}$  需要等待  $O_{32}$  的辅助模块使用完成后才能进行.

#### 1.2 数学模型

本节建立的混合整数线性规划数学模型, 其中符号定义如表 1 所示, 优化目标为最小化最大完工时间以及工人数量, 分别为

$$\min f_1 = C_{\max}, \tag{1}$$

$$\min f_2 = d. \tag{2}$$

数学模型约束条件描述如下:

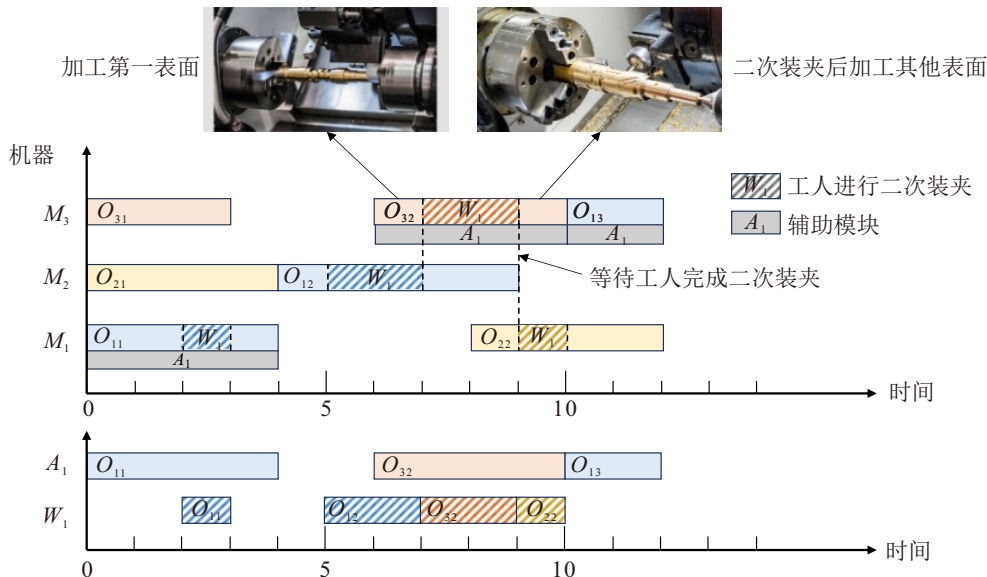


图1 二次装夹示意图

表1 MILP 模型符号定义

| 符号                       | 定义  |
|--------------------------|---|
| $i, i'$                  | 工件, $i = 1, 2, \dots, n$  |
| $I$                      | 工件集, $i \in I$ , 工件数为 $ I $   |
| $j, j'$                  | 工序, $j = 1, 2, \dots, h_i$  |
| $J_i$                    | 工件 $i$ 的工序集, $j \in J_i$ , 工件 $i$ 的工序数为 $ J_i $ ,<br>$J$ 表示最大的工序数                         |
| $k, k'$                  | 机器, $k = 1, 2, \dots, m$  |
| $a, a'$                  | 辅助模块, $a = 0, 1, \dots, l$ , $a = 0$ 表示机器直接加工   |
| $A$                      | 辅助模块集, $a \in A$  |
| $w, w'$                  | 工人, $w = 0, 1, \dots, d$ , $w = 0$ 表示机器直接加工   |
| $W$                      | 工人集, 工人总数为 $d$  |
| $t_{ij}^w$<br>$t_{ij}^s$ | 工人二次装夹工件 $i$ 第 $j$ 道工序的起始时间和装夹时间集合<br>( $s = 1$ 或 $2$ , $s = 1$ 表示起始时间, $s = 2$ 表示二次装夹时间) |
| $Z_{ij}$                 | 需要进行二次装夹的工序集  |
| $M_{ijka}$               | 工件 $i$ 的工序 $j$ 的可加工机器集(包含辅助模块)  |
| $t_{ijka}$               | 工件 $i$ 的工序 $j$ 在机器 $k$ 上使用辅助模块 $a$ 的加工时间<br>( $a=1$ 时表示直接使用机器加工)                          |
| $\psi$                   | 极大数   |
| 决策变量                     | 定义  |
| $x_{ijk}$                | 0-1变量, 若工件 $i$ 的工序 $j$ 在机器 $k$ 上加工, 则为1; 否则为0   |
| $y_{ijkaw}$              | 0-1变量, 若工件 $i$ 的工序 $j$ 在机器 $k$ 上使用辅助模块 $a$ 加工,<br>并由工人 $w$ 进行二次装夹, 则为1; 否则为0              |
| $\eta_{ij'i'j'}$         | 0-1变量, 在某个机器上, 若工件 $i$ 的工序 $j$ 在工件 $i'$ 的<br>工序 $j'$ 之后加工, 则为1; 否则为0                      |
| $\zeta_{ij'i'j'}$        | 0-1变量, 使用同一个辅助模块, 若工件 $i$ 的工序 $j$ 在<br>工件 $i'$ 的工序 $j'$ 之后加工, 则为1; 否则为0                   |
| $\theta_{ij'i'j'}$       | 0-1变量, 同一个工人操作加工, 若工件 $i$ 的工序 $j$ 在<br>工件 $i'$ 的工序 $j'$ 之后加工, 则为1; 否则为0                   |
| $C_{ij}$                 | 连续变量, 工件 $i$ 的工序 $j$ 的结束加工时刻  |
| $C_{\max}$               | 连续变量, 最大完工时间  |

1) 保证每道工序被分配到一台可用的机器, 且仅被分配一次, 有

$$\sum_{k=1}^m x_{ijk} = 1, \forall i, j; \quad (3)$$

$$x_{ijk} \leq \sum_{a=0}^l M_{ijka}, \forall i, j, k. \quad (4)$$

2) 每道工序在可用机器上选择合适的辅助模块, 有

$$x_{ijk} = \sum_{a=0}^l \sum_{w=0}^d y_{ijkaw}, \forall i, j, k; \quad (5)$$

$$\sum_{w=0}^d y_{ijkaw} \leq M_{ijka}, \forall i, j, k, a. \quad (6)$$

3) 限制工件第一道工序的开工, 时间不能早于释放时间, 有

$$C_{i1} \geq \sum_{k=1}^m \sum_{a=0}^l \sum_{w=0}^d t_{i1ka} y_{i1kaw}, \forall i. \quad (7)$$

4) 每个工件只能同时有一道工序被加工, 有

$$C_{ij} \geq C_{i,j-1} + \sum_{k=1}^m \sum_{a=0}^l \sum_{w=0}^d t_{ijka} y_{ijkaw}, \forall i, j > 1. \quad (8)$$

5) 每台机器只能同时加工一道工序, 有

$$C_{ij} \geq C_{i'j'} + t_{ijka} y_{ijkaw} - \psi(3 - y_{ijkaw} - y_{i'j'ka'w'} - \eta_{ij'i'j'}), \quad (9)$$

$$\forall i, i', j, j', k, a, a', w, w', O_{i,j} \neq O_{i',j'};$$

$$C_{i'j'} \geq C_{ij} + t_{i'j'ka'a'} y_{i'j'ka'a'w'} - \psi(2 - y_{ijkaw} - y_{i'j'ka'a'w'} + \eta_{ij'i'j'}), \quad (10)$$

$$\forall i, i', j, j', k, a, a', w, w', O_{i,j} \neq O_{i',j'}.$$

6) 每个 AM 只能同时被装配到一台机器上, 有

$$C_{ij} \geq C_{i'j'} + t_{ijka} y_{ijkaw} - \psi(3 - y_{ijkaw} - y_{i'j'k'a'w'} - \zeta_{ij'i'j'}), \quad (11)$$

$$\forall i, i', j, j', k, k', a > 1, w, w', O_{i,j} \neq O_{i',j'};$$

$$C_{i'j'} \geq C_{ij} + t_{i'j'k'a'a'} y_{i'j'k'a'a'w'} - \psi(2 - y_{i,j,k,a,w} - y_{i'j'k'a'w'} + \zeta_{ij'i'j'}), \quad (12)$$

$$\forall i, i', j, j', k, k', a > 1, w, w', O_{i,j} \neq O_{i',j'}.$$

7) 每个工人只能同时对一个工件进行二次装夹, 有

$$C_{ij} - t_{ijka} y_{ijkaw} + t_{ij1}^w \geq C_{i'j'} - t_{i'j'k'a'a'} y_{i'j'k'a'a'w'} + t_{i'j'1}^w + t_{i'j'2}^w - \psi(3 - y_{ijkaw} - y_{i'j'k'a'a'w'} - \theta_{ij'i'j'}), \quad (13)$$

$$\forall i, i', j, j', k, k', w, a, a', O_{i,j} \neq O_{i',j'};$$

$$C_{i'j'} - t_{i'j'k'a'a'} y_{i'j'k'a'a'w'} + t_{i'j'1}^w \geq C_{ij} - t_{ijka} y_{ijkaw} + t_{ij1}^w + t_{ij2}^w - \psi(2 - y_{ijkaw} - y_{i'j'k'a'a'w'} + \theta_{ij'i'j'}), \quad (14)$$

$$\forall i, i', j, j', k, k', w, a, a', O_{i,j} \neq O_{i',j'}.$$

## 2 带二次装夹约束的柔性作业车间调度优化方法

### 2.1 算法框架

遗传算法是调度优化的经典算法, 本文基于遗传算法提出调度问题的求解框架, 如图2所示. 针对问题的复杂约束, 采用三向量编码结构. 为缓解传统遗传算法易陷入局部最优的缺陷, 结合问题特征设计3种邻域结构, 进一步扩大解空间的范围, 减少陷入局部最优的可能. 在迭代过程中, 对种群个体适应度进行非支配排序, 并对其中的优秀个体进行迭代得到最优解程.

### 2.2 编码与解码

#### 2.2.1 三向量编码

本文模型需要考虑工序排序、机器选择、工人选择和辅助模块选择4个子问题. 考虑到机器选择与

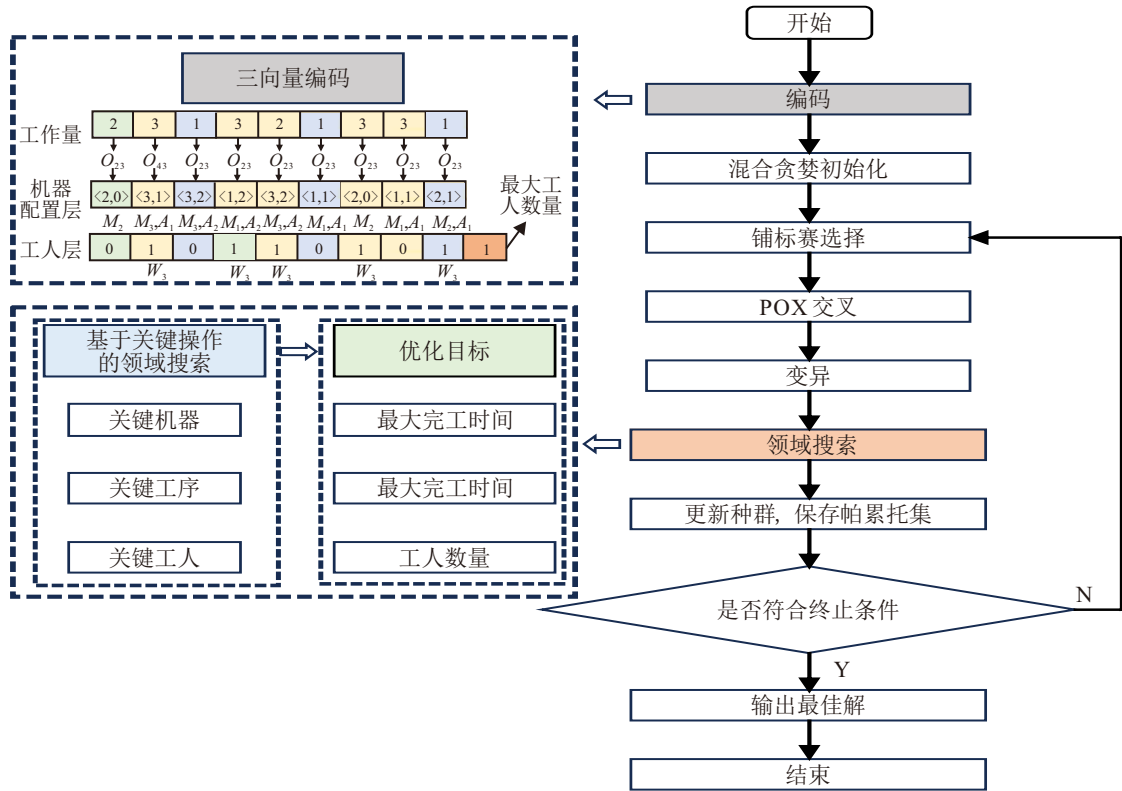


图2 算法框架

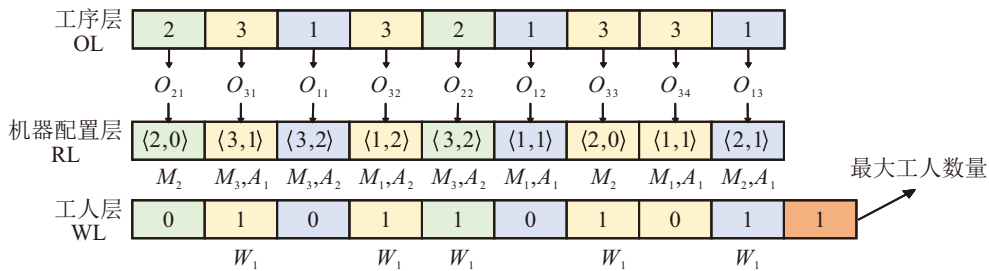


图3 编码示意图

辅助模块选择之间的关联性, 辅助模块需要在机器上才能进行操作, 可以将两个子问题结合到一起. 因此, 本文采用基于工序排序、机器选择和工人选择的3层解空间编码. 图3为编码示意图, 工序层OL为代码第1层, 指定了零件工件的顺序, 各数字表示工件号, 各数字出现的次数表示工序号, 例如第2位的3和第4位的3分别表示工件3的第1道工序和第2道工序, 以此类推. 第2层为机器配置层RL, 表示使用组装辅助模块的机器处理对应操作, 例如,  $\langle 1, 2 \rangle$ 表示机器1组装 $A_2$ 进行处理. 第3层为工人层WL, 表示操作机器的工人安排, 0表示不使用工人进行操作, 例如, 第2位数字1表示工人1进行二次装夹操作.

2.2.2 半主动解码

半主动解码可迅速生成可行的解决方案和完工时间, 再通过右移规则进一步压缩空闲<sup>[25]</sup>. 半主动解码按顺序访问工序层以找到要分配的操作, 然后通

过机器配置层识别加工的机器和辅助模块配置, 最后通过工人层确定二次装夹所需的工人配置. 之后, 计算操作的开始时间和结束时间以获得调度计划, 在计算每道工序时间时, 根据工件计时器 $TJ_i$ 、机器计时器 $TM_m$ 、辅助模块计时器 $TA_a$ 和工人计时器 $TW_p$ 确定本道工序段开始时间 $C_{P_{ij}}$ , 加上工序加工时间 $t_{ij}$ 得到本道工序的完成时间 $C_{ij}$ , 最后输出最佳调度个体的工序时间 $P_{val}$ , 最佳输出的调度基因 $S$ , 种群中每个个体的调度工序时间 $T_{objv}$ . 其中, 工人计时器 $TW_p$ 需要注意工人的加工时间段与工序加工时间段并不一致, 需要考虑本道工序二次装夹的开始时间 $t_{start}^{ws}$ 和持续时间 $t_{clamp}^{ws}$ , 具体计算如下:

$$TW_p = C_{ij} - t_{ij} + t_{start}^{ws} + t_{clamp}^{ws}. \quad (15)$$

半主动解码的策略伪码如表2所示.

2.3 初始化

初始化种群的质量会直接影响算法的收敛速度, 通过启发式规则生成的初始化种群包含这种规则,

表2 半主动解码

|  |
|--|
| 1: input: $I, J_i, A, W, Z, M, t;$                             |
| 2: output: $P_{val}, S, T_{objv}$                              |
| 3: setting $TJ_i, TM_m, TA_a, TW_p$                            |
| 4: for $n = 1$ to $\sum_{i=1}^n N_i$ do                        |
| 5: for $i = OL_p(X)$   |
| 6: $i$ 在OL中出现 $j^{th}$ 次                                       |
| 7: $\rightarrow$ 解码工作空间 $O_{ij}$                               |
| 8: 检查RL和WL中的位置   |
| 9: $\rightarrow$ 选择机器 $m$ , 辅助模块 $am$ , 工人 $p$                 |
| 10: if $am = 0, p = 0$   |
| 11: $TA_a = 0, TW_p = 0$                                       |
| 12: else   |
| 13: $TW_p = C_{ij} - t_{ij} + t_{start}^{ws} + t_{clamp}^{ws}$ |
| 14: end  |
| 15: $C_{P_{ij}} = \max\{TJ_i, TM_m, TA_a, TW_p\}$              |
| 16: 更新 $TJ_i, TM_m, TA_a, TW_p$ , 找到 $O_{ij}$ 加工时间 $t_{ij}$    |
| 17: $C_{ij} = C_{P_{ij}} + t_{ij}$                             |
| 18: end  |
| 19: end  |

通常会优于随机生成的解决方案,且能提高算法收敛性<sup>[26]</sup>,而贪婪策略配合其他机制协同,可以取得更有效的调度解<sup>[27]</sup>.因此问题的种群初始化使用混合贪婪策略,将作业分配给工作量最少的工人和机器.即首先随机生成最大工人数量,最大工人数量不能大于机器数量;然后根据约束条件随机生成工序层编码;接着根据工序层确定可用的机器集合,从中选择时间最少的机器和辅助模块进行加工;最后遍历之前的工人选择,从中选择工人负载最小的工人.

## 2.4 选择、交叉与变异

### 2.4.1 选择

本文采用锦标赛选择策略.锦标赛选择会在当前种群中随机选择一定数量的个体,比较这些个体的适应度并将其中的最优个体保留到下一种群中,使用 Pareto 前沿的方法计算每个个体的适应度,即根据个体前沿层级和拥挤距离生成适应度值.

### 2.4.2 交叉和变异

本文参考使用 Cheng 等<sup>[28]</sup>提出的 POX 交叉方法,该方法在父代进行交叉时可有效避免非法的出现.根据本研究问题的特性,只对工序层 OL 进行交叉,机器层 RL 和工人层 WL 通过索引层 OL 的交叉同步进行. POX 交叉具体操作如图 4 所示.

变异操作是对工人层 WL 进行变异,通过破译 OL 层确定本道工序工作空间 $O_{ij}$ ,如果 $O_{ij}$ 有多个机

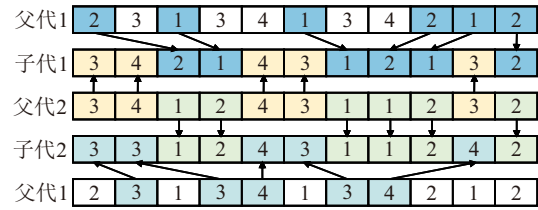


图4 POX 交叉

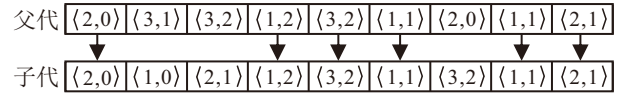


图5 变异

器选择,则向其他机器变异,具体操作如图 5 所示.

## 2.5 邻域结构

本文设计了一种基于关键操作的邻域结构,关键操作分别为关键机器、关键路径中的工件块和关键工人.前两种关键操作针对减少最大完工时间设计,第 3 种针对工人数量设计.

### 2.5.1 关键机器邻域结构

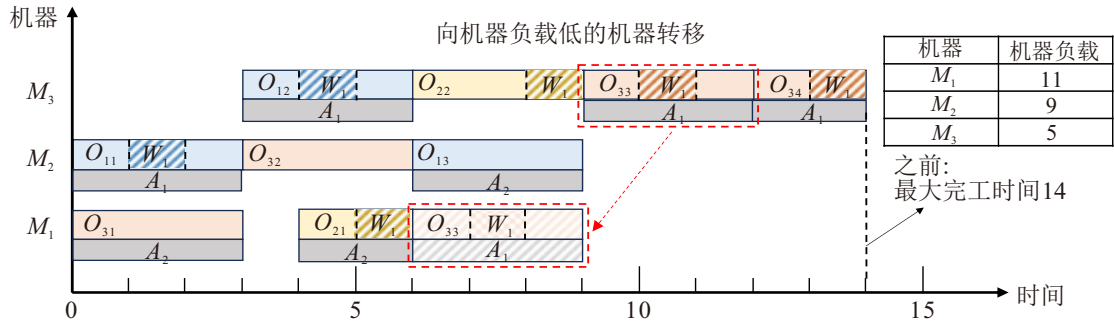
关键机器的邻域结构具体操作为:首先,记录每个机器的负载情况,找出其中负载最高和最低的机器,即关键机器;然后,找出在负载最高机器上加工的工序,使其向机器负载最低的机器转移.若此时最低负载机器无法加工该工序,则将其分配给负载次低的机器.该方法相较于传统方法减少了机器替换的数量,只对机器负载最高的机器进行检索,能够有效缩短邻域搜索的时间,具体操作如图 6 所示.

### 2.5.2 N6 邻域搜索

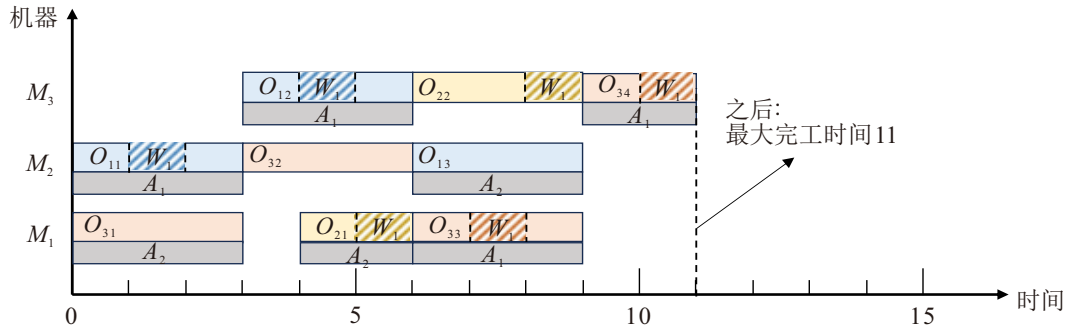
在 FJSP 中,从加工开始到加工结束过程中的最长路径称为关键路径,其长度即为调度方案的完工时间. N6 邻域的操作如下:首先找到关键路径,关键路径中,定义在同一机器上加工的相邻工序为工序块,将块内工序插入到块首工序之前,或将块内工序插入到块尾工序之后(使用随机数 PN6 进行判断).使用 N6 邻域可以有效扩大邻域空间,增加找到更优解的可能性. N6 邻域搜索的伪码如表 3 所示,具体操作如图 7 所示.

### 2.5.3 关键工人邻域结构

关键工人邻域结构的具体操作是通过记录每位工人的加工时间,找出其中加工时间最短的工人,即关键工人.对该工人所加工的工序进行优化,将其加工的所有工序转移到加工时间第二短的工人上,这样可以直接减少一名工人的数量,达到优化工人数量的目的.若此时加工时间最短的工人正在装夹其他工件,则将其分配给加工时间次短的工人.具体操作如图 8 所示.



(a) 使用关键机器邻域结构前



(b) 使用关键机器邻域结构后

图6 关键机器邻域结构

表3 N6 邻域搜索伪码

- 1: input:  $I, J_i, A, W, Z, M, t$ ;
- 2: output: CBSset, CBlock.
- 3: setting  $NJ_b, NM_m, NA_a, NW_w$
- 4: while 当前操作  $O_d > 0$  do
- 5:  $C_{PM(d)} = O_d$  机器前序操作  $PM(d)$  的完工时间
- 6:  $C_{JM(d)} = O_d$  工件前序操作  $JM(d)$  的完工时间
- 7:  $C_{AM(d)} = O_d$  辅助模块前序操作  $AM(d)$  的完工时间
- 8:  $C_{WM(d)} = O_d$  工人前序操作  $WM(d)$  的完工时间
- 9:  $C_{key(d)} = \max(C_{d-1}, C_{PM(d)}, C_{AM(d)}, C_{WM(d)})$
- 10:  $OP_{key} = C_{key(d)}, COP \leftarrow OP_{key}, O_d = OP_{key}$
- 11: end
- 12: 确定COP中包含1个以上任务的关键块CBSset
- 13: 从CBSset中随机选择一个过程CBlock
- 14: if  $PN6 > 0.5$
- 15: 向前插入 CBlock
- 16: else
- 17: 向后插入CBlock
- 18: end

### 3 实验研究与结果分析

实验在 12th Gen Intel(R) Core(TM) i7-12800HX 处理器 (主频 2.00 GHz), 16 G 运行内存, Windows 11 操作系统的个人电脑上进行, 基于 Matlab 2024a 实现.

#### 3.1 测试案例生成

实验基于 BRdata<sup>[29]</sup> 生成的 mk1 ~ mk15 案例

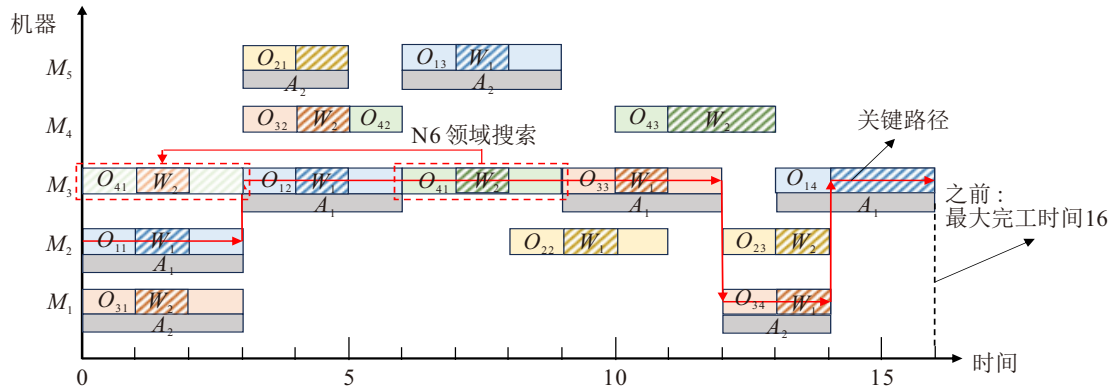
进行, 由于其为针对 FJSP 问题的案例, 并没有涉及辅助模块和工人的参数设置, 辅助模块的设置参见文献 [7], 测试案例如表 4 所示. 其中:  $n$  为工件数,  $m$  为机器数,  $j_{op}$  为每个工序的最小和最大操作次数,  $m_{eq}$  为每次操作的最大等效机器数量,  $t_{proc}$  为每次操作的最小和最大处理时间,  $n_{am}$  为辅助模块数量.

表4 测试案例的参数设置范围

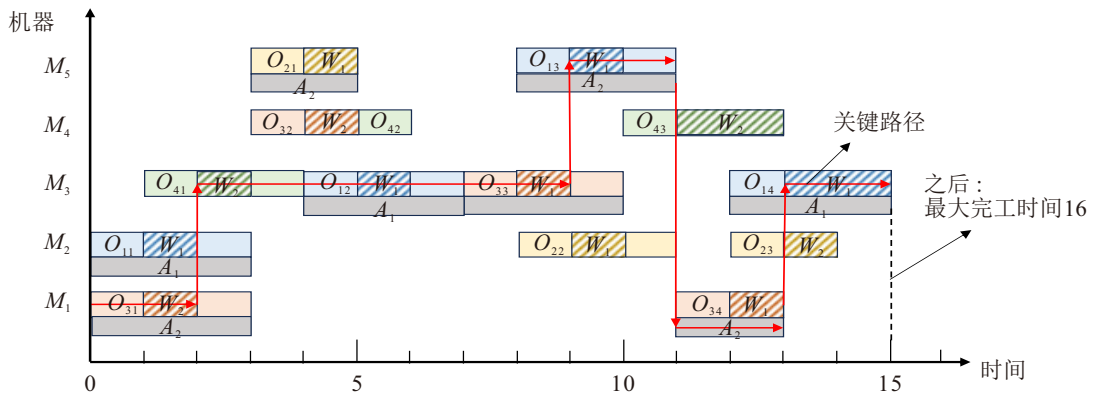
| 案例   | $n$ | $m$ | $j_{op}$ | $m_{eq}$ | $t_{proc}$ | $n_{am}$ |
|------|-----|-----|----------|----------|------------|----------|
| mk1  | 10  | 6   | 5 ~ 7    | 3        | 1 ~ 7      | 3        |
| mk2  | 10  | 6   | 5 ~ 7    | 6        | 1 ~ 7      | 3        |
| mk3  | 15  | 8   | 10 ~ 10  | 5        | 1 ~ 20     | 4        |
| mk4  | 15  | 8   | 3 ~ 10   | 3        | 1 ~ 10     | 4        |
| mk5  | 15  | 4   | 5 ~ 10   | 2        | 5 ~ 10     | 2        |
| mk6  | 10  | 15  | 15 ~ 15  | 5        | 1 ~ 10     | 8        |
| mk7  | 20  | 5   | 5 ~ 5    | 5        | 1 ~ 20     | 3        |
| mk8  | 20  | 10  | 5 ~ 10   | 2        | 5 ~ 20     | 5        |
| mk9  | 20  | 10  | 10 ~ 15  | 5        | 5 ~ 20     | 5        |
| mk10 | 20  | 15  | 10 ~ 15  | 5        | 5 ~ 20     | 8        |
| mk11 | 30  | 5   | 5 ~ 8    | 2        | 10 ~ 30    | 3        |
| mk12 | 30  | 10  | 5 ~ 10   | 2        | 10 ~ 30    | 5        |
| mk13 | 30  | 10  | 5 ~ 10   | 5        | 10 ~ 30    | 5        |
| mk14 | 30  | 15  | 8 ~ 12   | 2        | 10 ~ 30    | 8        |
| mk15 | 30  | 15  | 8 ~ 12   | 5        | 10 ~ 30    | 8        |

#### 3.2 算法参数设置

算法需要确定的主要参数有种群数量 Pop, 交叉率 Cr, 变异率 Mr 和锦标赛选择规模 tSize. 参数水平设置为  $Pop = [100, 300, 500, 700]$ ,  $Cr = [0.75, 0.8]$ ,

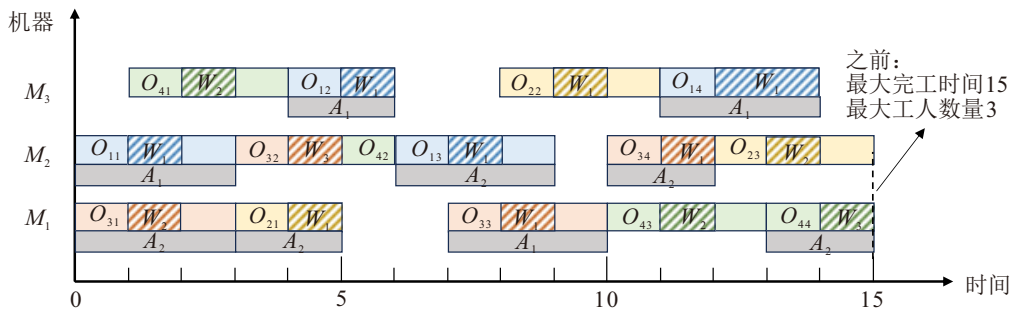


(a) 使用N6领域结构前

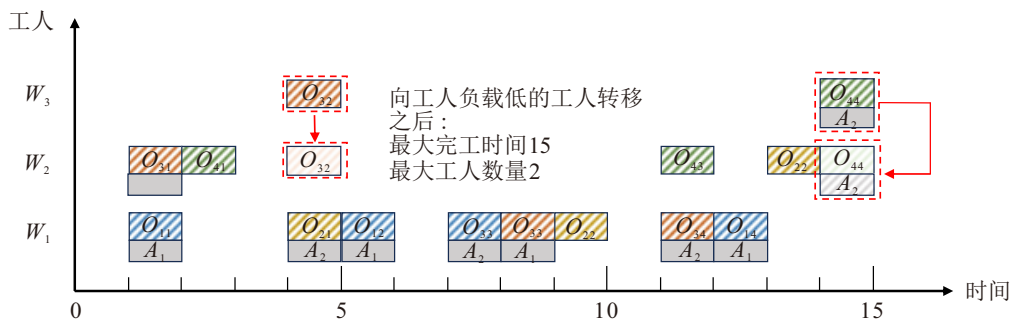


(b) 使用N6领域结构后

图7 N6邻域搜索结构



(a) 关键工人领域结构甘特图示意图



(b) 关键工人领域结构工人负载示意图

图8 关键工人邻域搜索

0.85, 0.9],  $M_r = [0.05, 0.1, 0.15, 0.2]$ ,  $tSize = [2, 3, 4, 5]$ . 实验采用四水平四参数  $L16(4^4)$  田口设计法确定相关参数, 终止标准为  $m \times n \times 5 s$ , 每个案例独立运行 5 次, 使用逆时代距离 (IGD) 作为评价标准.

IGD 通过计算与真实 Pareto 前沿的欧几里得距离评价该算法的性能, IGD 值越小表明算法性能越好. 将独立运行 20 次得到的最优解作为真实 Pareto 前沿, 田口实验得到的 IGD 值如表 5 所示, 各主要参数平

表5 田口实验主要参数 IGD 值

| 种群数量 | 交叉率  | 变异率  | 锦标赛规模 | IGD      |
|------|------|------|-------|----------|
| 100  | 0.75 | 0.05 | 2     | 0.240012 |
| 100  | 0.80 | 0.10 | 3     | 0.250042 |
| 100  | 0.85 | 0.15 | 4     | 0.265642 |
| 100  | 0.90 | 0.20 | 5     | 0.219508 |
| 300  | 0.75 | 0.10 | 2     | 0.152532 |
| 300  | 0.80 | 0.05 | 3     | 0.236868 |
| 300  | 0.85 | 0.20 | 4     | 0.23827  |
| 300  | 0.90 | 0.15 | 5     | 0.22584  |
| 500  | 0.75 | 0.15 | 5     | 0.24096  |
| 500  | 0.80 | 0.20 | 4     | 0.19317  |
| 500  | 0.85 | 0.05 | 3     | 0.194374 |
| 500  | 0.90 | 0.10 | 2     | 0.13995  |
| 700  | 0.75 | 0.20 | 3     | 0.224016 |
| 700  | 0.80 | 0.15 | 2     | 0.183884 |
| 700  | 0.85 | 0.10 | 5     | 0.206507 |
| 700  | 0.90 | 0.05 | 4     | 0.170083 |

表7 初始化实验结果

| 案例   | 随机初始化 (IGD) | 混合贪婪初始化 (IGD) | 随机初始化 (HV) | 混合贪婪初始化 (HV)    |
|------|-------------|---------------|------------|-----------------|
| mk1  | 0.0106      | <b>0.0054</b> | 227.20     | <b>798.32</b>   |
| mk2  | 0.0086      | <b>0.0037</b> | 248.43     | <b>697.25</b>   |
| mk3  | 0.1692      | <b>0.1475</b> | 2025.52    | <b>5382.49</b>  |
| mk4  | 0.1526      | <b>0.1332</b> | 1156.70    | <b>1980.47</b>  |
| mk5  | 0.0475      | <b>0.0382</b> | 1118.17    | <b>1784.64</b>  |
| mk6  | 0.2167      | <b>0.1938</b> | 2322.23    | <b>5362.14</b>  |
| mk7  | 0.0679      | <b>0.0578</b> | 1464.76    | <b>2417.99</b>  |
| mk8  | 0.881       | <b>0.8275</b> | 5240.06    | <b>8138.91</b>  |
| mk9  | 0.2404      | <b>0.2138</b> | 7406.01    | <b>11075.59</b> |
| mk10 | 0.3569      | <b>0.3091</b> | 8246.42    | <b>12882.71</b> |
| mk11 | 0.4414      | <b>0.3887</b> | 5690.40    | <b>9967.57</b>  |
| mk12 | 0.2503      | <b>0.2211</b> | 11466.20   | <b>17081.22</b> |
| mk13 | 0.542       | <b>0.4851</b> | 9634.79    | <b>13881.03</b> |
| mk14 | 0.5006      | <b>0.4459</b> | 19099.88   | <b>29951.44</b> |
| mk15 | 0.5494      | <b>0.4909</b> | 19800.61   | <b>28263.56</b> |

均 IGD 值如图 9 所示,对实验记过进行方差分析得到数据如表 6 所示.

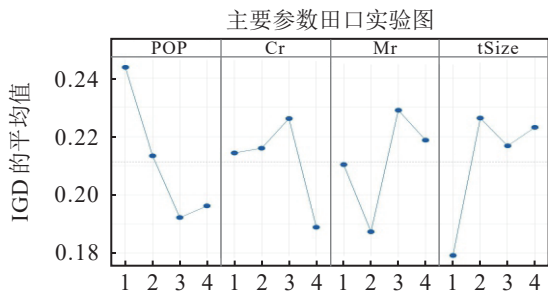


图9 田口实验主要参数 IGD 值

表6 方差分析结果

| 参数    | 平方和    | 均方     | F值    | P值     | 贡献率/% |
|-------|--------|--------|-------|--------|-------|
| Pop   | 0.0067 | 0.0022 | 16.40 | 0.0229 | 33.84 |
| Cr    | 0.0030 | 0.0010 | 7.49  | 0.0661 | 15.45 |
| Mr    | 0.0038 | 0.0013 | 9.39  | 0.0491 | 19.39 |
| tSize | 0.0057 | 0.0019 | 14.18 | 0.0281 | 2.06  |

通过实验可以看出:种群数量Pop = 500,交叉率Cr = 0.9,变异率Mr = 0.1,锦标赛规模tSize = 2时,IGD值明显优于其他指标.因此实验主要参数设置为Pop = 500, Cr = 0.9, Mr = 0.1, tSize = 2.对实验结果进行方差分析可以发现,对于算法影响最大的参数是种群数量.

### 3.3 初始化策略验证

为了验证初始化的有效性,实验将混合贪婪初始化和随机生成的种群独立运行10次,终止标准为 $m \times n \times 5$  s,取其中的最优种群进行比较,使用IGD和HV作为评价标准.超体积指标(HV)通过计算算法获得的非支配解集与参照点围成的目标空间中区域的体积来进行评价,HV值越大表明算法的综合性

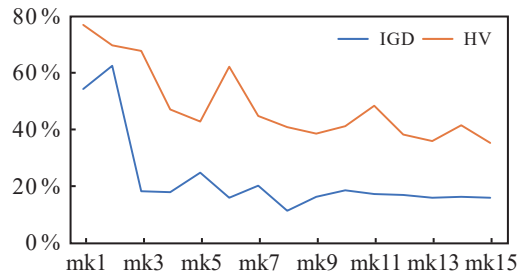


图10 初始化策略相对增益趋势

能越好.实验得到的数据如表7所示,计算后得到的初始化策略相对增益趋势如图10所示.

观察表7中15个案例的对比结果可以发现,本文设计的初始化方法在不同案例背景上均优于随机生成策略.观察图10可以发现:随着案例规模的增大,本文所提出初始化方法相较于随机初始化的相对增益呈递减趋势.这是因为在小规模案例中,混合贪婪初始化可以更好地平均机器负载和工人负载,快速找到更优解;但随着案例规模扩大,解空间的范围也呈几何倍增长,这时仅靠初始化规则很难找到优秀的解决方案,需要使用更好的邻域结构扩大探索空间,对种群进一步优化.综上,本文的混合贪婪初始化策略比传统的随机生成种群更具有优势.

### 3.4 邻域结构策略验证

为验证邻域结构的有效性,本文将不使用邻域结构、使用单邻域结构、使用双邻域结构和使用3邻域结构这4种算法进行对比,每次实验独立运行5次,终止标准为 $m \times n \times 10$  s,计算IGD和HV,结果如表8所示,箱形图如图11所示.

观察图11可以看出,使用邻域策略可以显著提升解的质量.使用考虑关键机器的邻域后,算法IGD值迅速下降,然而在使用其他邻域后,IGD降低

表8 邻域化实验结果

| 案例   | 无邻域结构 |          | 单邻域结构       |          | 双邻域结构       |          | 三邻域结构       |                 |
|------|-------|----------|-------------|----------|-------------|----------|-------------|-----------------|
|      | IGD   | HV       | IGD         | HV       | IGD         | HV       | IGD         | HV              |
| mk1  | 0.22  | 1001.87  | <b>0.00</b> | 1102.06  | 0.13        | 1123.45  | 0.04        | <b>1389.57</b>  |
| mk2  | 0.43  | 571.94   | <b>0.07</b> | 629.13   | 0.21        | 679.94   | 0.14        | <b>935.61</b>   |
| mk3  | 0.10  | 6831.98  | 0.25        | 7515.18  | 0.02        | 7355.47  | <b>0.00</b> | <b>9384.52</b>  |
| mk4  | 0.26  | 2673.29  | <b>0.03</b> | 2940.62  | <b>0.03</b> | 3000.40  | 0.15        | <b>3700.29</b>  |
| mk5  | 0.29  | 3012.80  | 0.25        | 3314.08  | 0.16        | 3371.61  | <b>0.02</b> | <b>4173.19</b>  |
| mk6  | 0.14  | 8206.49  | 0.23        | 9027.14  | 0.13        | 9213.38  | <b>0.01</b> | <b>11337.07</b> |
| mk7  | 0.11  | 3306.47  | 0.11        | 3637.12  | 0.16        | 3704.36  | <b>0.06</b> | <b>4564.76</b>  |
| mk8  | 0.22  | 13131.51 | 0.14        | 14444.66 | <b>0.03</b> | 13876.85 | 0.11        | <b>18035.12</b> |
| mk9  | 0.05  | 18931.46 | 0.09        | 20824.61 | 0.08        | 19226.19 | <b>0.01</b> | <b>24446.73</b> |
| mk10 | 0.03  | 18971.57 | 0.10        | 20868.73 | <b>0.00</b> | 22826.63 | 0.01        | <b>28117.23</b> |
| mk11 | 1.01  | 17439.05 | 0.34        | 19182.95 | 0.06        | 19533.69 | <b>0.03</b> | <b>24064.83</b> |
| mk12 | 0.09  | 30814.43 | <b>0.03</b> | 33895.88 | 0.13        | 33519.25 | 0.11        | <b>40991.22</b> |
| mk13 | 0.14  | 24470.21 | 0.02        | 26917.23 | 0.05        | 27412.12 | <b>0.02</b> | <b>32786.12</b> |
| mk14 | 0.11  | 47137.11 | 0.05        | 51850.82 | <b>0.00</b> | 53877.59 | 0.02        | <b>68696.58</b> |
| mk15 | 0.10  | 41499.45 | 0.15        | 45649.39 | 0.03        | 46499.04 | <b>0.02</b> | <b>58084.88</b> |

表9 算法对比实验结果

| 案例   | GA(IGD)     | PSO(IGD)    | SA(IGD)     | ACO(IGD) | IG(IGD) | GA(HV)          | PSO(HV)  | SA(HV)   | ACO(HV)  | IG(HV)          |
|------|-------------|-------------|-------------|----------|---------|-----------------|----------|----------|----------|-----------------|
| mk1  | <b>0.08</b> | 0.42        | 0.82        | 1.70     | 0.17    | <b>848.90</b>   | 761.37   | 591.60   | 548.02   | 870.88          |
| mk2  | 0.20        | <b>0.00</b> | 2.33        | 2.77     | 0.20    | 414.41          | 323.24   | 382.38   | 295.01   | <b>19973.70</b> |
| mk3  | <b>0.09</b> | 0.14        | 1.05        | 0.92     | 0.45    | <b>10714.81</b> | 9048.77  | 8690.21  | 7181.21  | 10535.39        |
| mk4  | <b>0.06</b> | 0.29        | 0.58        | 3.20     | 1.66    | 2065.24         | 1829.99  | 1492.87  | 1210.21  | <b>3974.60</b>  |
| mk5  | 0.08        | 0.53        | <b>0.06</b> | 1.24     | 0.48    | <b>2604.06</b>  | 2434.91  | 2059.89  | 2366.30  | 2576.87         |
| mk6  | <b>0.04</b> | 0.21        | 1.16        | 1.34     | 0.52    | 4485.63         | 3124.33  | 2756.12  | 3548.15  | <b>14553.21</b> |
| mk7  | <b>0.10</b> | 0.67        | 1.86        | 2.52     | 1.23    | <b>4660.61</b>  | 4161.56  | 3794.64  | 3840.73  | 4434.89         |
| mk8  | 0.05        | <b>0.02</b> | 0.18        | 0.74     | 0.40    | 16613.52        | 14896.17 | 13828.21 | 11824.85 | <b>18131.63</b> |
| mk9  | <b>0.07</b> | 0.21        | 0.25        | 1.19     | 0.44    | 11854.63        | 10008.60 | 8559.47  | 8824.18  | <b>25203.21</b> |
| mk10 | 0.14        | 0.11        | <b>0.04</b> | 0.87     | 0.30    | <b>23488.36</b> | 22493.44 | 19371.75 | 20510.26 | 22260.81        |
| mk11 | <b>0.16</b> | 0.29        | <b>0.16</b> | 1.08     | 0.51    | <b>20691.39</b> | 19769.70 | 18172.95 | 18151.14 | 17412.37        |
| mk12 | 0.15        | 0.33        | <b>0.12</b> | 0.50     | 0.35    | <b>31149.58</b> | 25758.60 | 25031.51 | 24542.71 | 19200.60        |
| mk13 | <b>0.08</b> | 0.12        | 0.15        | 0.62     | 0.59    | <b>28083.66</b> | 26867.70 | 24682.68 | 24891.88 | 27015.30        |
| mk14 | <b>0.07</b> | 0.28        | 0.18        | 0.15     | 0.14    | <b>46953.15</b> | 43166.73 | 40185.55 | 39329.83 | 18582.30        |
| mk15 | <b>0.02</b> | 0.19        | 0.14        | 0.88     | 0.40    | <b>50291.68</b> | 46884.71 | 42922.28 | 41772.43 | 33201.90        |

观察图12中IGD的箱形图可以发现,本文所提出方法的中位数最低,箱体最窄,所有点都聚集在一起,表明本文方法不仅能获得最低的平均IGD,而且性能更稳定且波动最小.观察HV的箱形图和实验数据可以发现,虽然本文所提出方法的箱形图并没有与PSO拉开太大差距,但HV的实验数据大多为

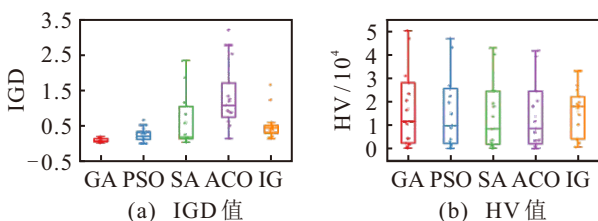


图12 算法对比实验结果

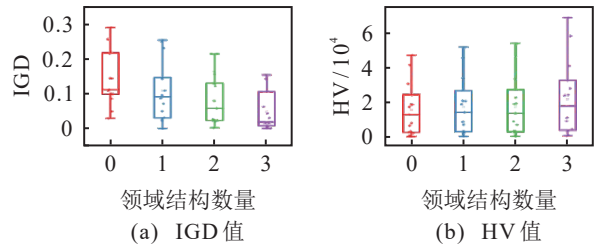


图11 邻域结构实验结果

速度变慢.这是因为使用考虑关键机器的邻域后,算法收敛性得到了显著提升,后续添加其他邻域结构相较于前面的提升并不明显.

### 3.5 算法对比验证

将所提出算法与粒子群算法、模拟退火算法、蚁群算法和近期提出的迭代贪婪算法<sup>[30]</sup>进行对比,每个算法均对15个不同规模的案例独立运行5次,终止标准为 $m \times n \times 10$  s,计算得到的IGD和HV如表9所示,箱形图如图12所示,部分案例的Pareto前沿对比如图13所示.

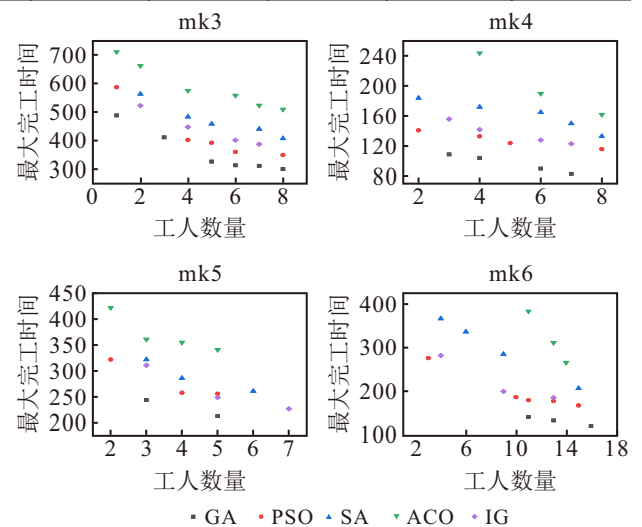


图13 Pareto 前沿对比结果

最优值. 观察图 13 所示的 Pareto 前沿对比图可以发现, 本文所提出方法在 4 个案例中生成的前沿均支配了其他算法. 因此, 本文所提出方法表现的稳定性更强, 明显优于其他算法.

### 4 工程实例应用研究

#### 4.1 实例数据

本节基于汽车差速器制造车间的实际生产数据, 对所提出的调度模型与方法进行工程应用验证. 汽车差速器核心部件包括输入轴、差速器壳体、行星齿轮组件、输出半轴与轴承等. 工程实例部分数据如表 10 所示, 其中零件均为轴类零件, 加工过程具有以下特点: 部分工序需要进行二次装夹; 部分工序可使用辅助模块提高加工效率, 如工件 1 的第 3 道工序在机器 1 上加工需 10 s, 配合辅助模块  $A_1$  则仅需 8 s; 此外, 也存在必须依赖特定辅助模块才能完成的工序, 如工件 1 的第 4 道工序必须要在机器 3 上使用辅助模块  $A_2$  进行加工.

表10 部分工程实例参数

| $i$ | $j$ | $t/s$     | $tw_1$ | $tw_2$ | 可操作机器                       |
|-----|-----|-----------|--------|--------|-----------------------------|
| 1   | 1   | 10 8      | 4      | 1      | 5 8                         |
|     | 2   | 4 8 7 4 2 | 1      | 1      | 1 4 4( $A_4$ ) 5 6( $A_2$ ) |
|     | 3   | 10 8 8    | 1      | 2      | 1 1( $A_1$ ) 8              |
|     | 4   | 5         | —      | —      | 3( $A_2$ )                  |
| 2   | 1   | 5         | 4      | 1      | 2                           |
|     | 2   | 7 6       | 2      | 3      | 6 6( $A_3$ )                |
|     | 3   | 5 10      | 2      | 2      | 1 2                         |
| 3   | 4   | 6 5       | 1      | 1      | 4 2( $A_1$ )                |
|     | 5   | 5 4       | 1      | 4      | 3 3( $A_3$ )                |
|     | 6   | 10 8      | 1      | 3      | 8 8( $A_4$ )                |
|     | 7   | 7         | —      | —      | 5                           |

#### 4.2 算法对比分析

按照 3.2 节所得到的参数  $Pop = 500$ ,  $Cr = 0.9$ ,  $Mr = 0.1$ ,  $tSize = 2$ , 将本文所提出方法与 PSO、SA、ACO 和 IG 进行对比实验, 每次实验均独立运行 10

次, 终止标准为  $m \times n \times 10$  s, 得到的 IGD 和 HV 值如表 11 所示. 发现第 6 次实验计算得到的 HV 值相较于其他数据明显偏高, 其中 ACO 的 HV 值为 0, 表明此次实验结果均为异常值, 可以剔除. 对表 11 中的数据绘制得到箱形图如图 14 所示, Pareto 前沿对比结果如图 15 所示.

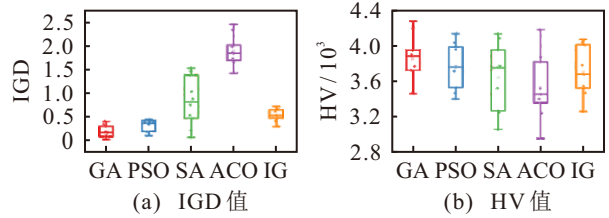


图14 工程案例算法对比实验

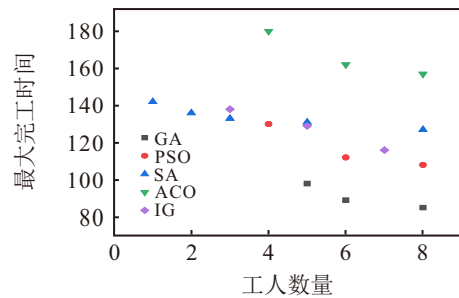


图15 工程案例 Pareto 前沿对比结果

观察图 14 可以发现, 本文提出的算法无论是在 IGD 还是 HV 中均明显优于其他算法, 且结果分布最集中, 稳定性和可靠性最好. 观察图 15 中的 Pareto 前沿对比结果可以发现, 所提出算法生成的前沿均能支配其他算法, 明显优于其他算法.

#### 4.3 调度结果分析

为验证二次装夹对生产调度的实际影响, 本节实验分别在两种情况下进行: 1) 考虑二次装夹约束; 2) 不考虑二次装夹约束. 两种情况均独立运行 20 次, 对不考虑二次装夹得到的结果, 根据该方案执行时各工件真实的加工需求, 在相应工序间强制加入必要的二次装夹时间, 重新计算此时的实际最大完

表11 工程案例算法对比实验

| 实验次数 | GA(IGD)     | PSO(IGD)    | SA(IGD) | ACO(IGD) | IG(IGD) | GA(HV)         | PSO(HV)  | SA(HV)         | ACO(HV) | IG(HV)         |
|------|-------------|-------------|---------|----------|---------|----------------|----------|----------------|---------|----------------|
| 1    | <b>0.40</b> | 0.43        | 1.52    | 1.83     | 0.65    | <b>3898.54</b> | 3714.92  | 3522.50        | 3520.64 | 3681.11        |
| 2    | <b>0.19</b> | 0.19        | 1.03    | 2.00     | 0.52    | <b>3461.95</b> | 3401.03  | 3267.63        | 3453.52 | 3257.74        |
| 3    | <b>0.09</b> | 0.11        | 1.37    | 2.44     | 0.30    | <b>4277.19</b> | 4135.36  | 3248.98        | 3239.50 | 3468.73        |
| 4    | <b>0.17</b> | 0.40        | 0.47    | 1.85     | 0.66    | 3770.98        | 3530.96  | 3772.67        | 3402.35 | <b>4010.55</b> |
| 5    | 0.34        | <b>0.19</b> | 0.21    | td1.41   | 0.53    | 3726.00        | 4034.21  | <b>4135.19</b> | 3818.54 | 3547.33        |
| 6    | <b>0.02</b> | 0.43        | 0.53    | 1.97     | 0.71    | 65535.00       | 65535.00 | 65535.00       | 0.00    | 65535.00       |
| 7    | <b>0.30</b> | 0.38        | 0.06    | 1.72     | 0.41    | 3951.24        | 3956.83  | 3953.40        | 3358.01 | <b>4039.29</b> |
| 8    | <b>0.02</b> | 0.34        | 0.75    | 1.68     | 0.61    | <b>3882.02</b> | 3760.08  | 3755.09        | 3872.89 | 3783.09        |
| 9    | <b>0.11</b> | 0.45        | 0.87    | 2.32     | 0.52    | <b>4194.94</b> | 3987.30  | 4084.89        | 4180.86 | 3678.76        |
| 10   | <b>0.17</b> | 0.35        | 1.46    | 1.65     | 0.47    | 3461.03        | 3465.96  | 3057.68        | 2957.83 | <b>3523.15</b> |

工时间,最终得到的实验结果如表12和图16所示.

表12 工程案例对比实验

| 实验次数 | 考虑二次装夹 |      | 不考虑二次装夹 |      |
|------|--------|------|---------|------|
|      | 完工时间   | 工人数量 | 完工时间    | 工人数量 |
| 1    | 103    | 4    | 130     | 2    |
| 2    | 98     | 3    | 126     | 3    |
| 3    | 99     | 2    | 139     | 2    |
| 4    | 98     | 3    | 147     | 2    |
| 5    | 102    | 5    | 177     | 1    |
| 6    | 92     | 5    | 155     | 2    |
| 7    | 94     | 3    | 203     | 1    |
| 8    | 97     | 3    | 135     | 2    |
| 9    | 104    | 2    | 141     | 2    |
| 10   | 97     | 3    | 152     | 1    |
| 11   | 98     | 3    | 168     | 1    |
| 12   | 96     | 3    | 131     | 2    |
| 13   | 96     | 3    | 181     | 1    |
| 14   | 100    | 3    | 147     | 2    |
| 15   | 98     | 6    | 121     | 3    |
| 16   | 99     | 3    | 140     | 2    |
| 17   | 93     | 3    | 137     | 2    |
| 18   | 100    | 3    | 120     | 2    |
| 19   | 90     | 3    | 200     | 1    |
| 20   | 89     | 5    | 112     | 3    |

由图16中的折线图可以发现,采用考虑二次装夹约束的调度方案所获得的最大完工时间明显低于不考虑二次装夹约束的方案,且结果分布更集中,稳定性更好.观察图16的Pareto前沿对比结果可以发现,不考虑二次装夹的方案在工人数量指标上表现更优,但是其他Pareto前沿被考虑二次装夹约束的

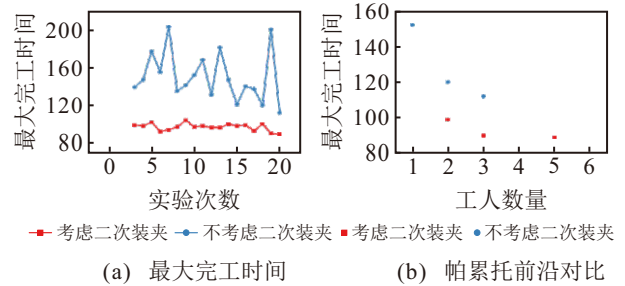


图16 工程案例对比实验

方案支配.这是因为算法中基于关键工人的邻域结构设计可以对工人配置进行优化,减少了工人数量,但在完全不考虑二次装夹时间的情况下,实际执行必然产生效率损失.综上所述,二次装夹作为车间调度中普遍存在的现象,考虑二次装夹的影响可以将最大完工时间减少约56%,但对工人资源的需求也提升了约67%,在效率与资源之间需要根据实际产能和交付压力做出平衡决策.最终得到考虑二次装夹的汽车差速器生产调度方案如图17所示,其中斜线条纹表示工人操作进行二次装夹.例如机器7加工的第1道工序为 $O_{5-1}$ ,中间的斜线条纹表示工人 $W_3$ 操作进行二次装夹.

### 5 结论

本文针对带有二次装夹约束的可重构柔性作业车间调度问题,构建了最小化最大完工时间和工人数量的混合整数线性规划数学模型,并提出了一种改进的多目标遗传算法.然后通过测试案例与工程案例实验表明,与传统方法及近期文献中的先进算法相比,本文算法具有更好的性能,可以将最大完工

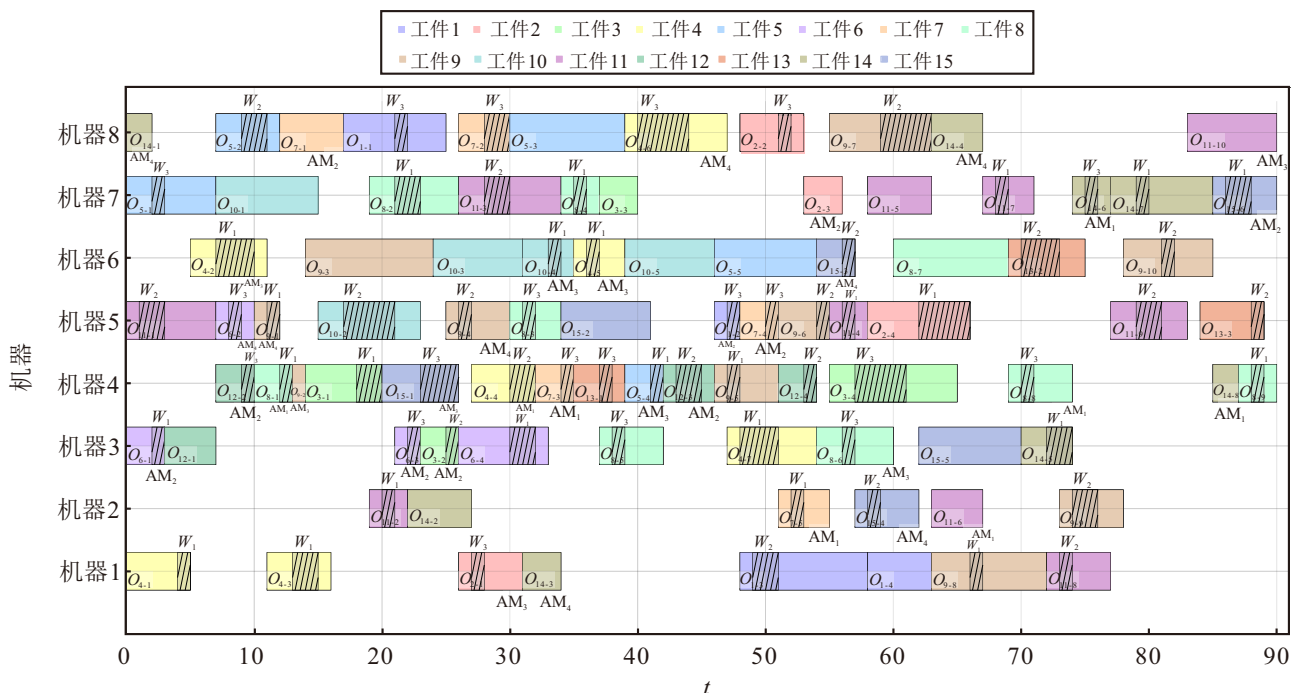


图17 工程案例甘特图

时间减少约 56%。在工厂实际生产过程中,二次装夹的操作可能会导致装夹的定位精度产生误差,影响最终产品的质量。因此,未来可进一步探索二次装夹对产品质量的影响及工人技能熟练度在调度中的协同优化策略。

### 参考文献 (References)

- [1] Morgan J, Halton M, Qiao Y S, et al. Industry 4.0 smart reconfigurable manufacturing machines[J]. *Journal of Manufacturing Systems*, 2021, 59: 481-506.
- [2] Khan A S, Homri L, Dantan J Y, et al. An analysis of the theoretical and implementation aspects of process planning in a reconfigurable manufacturing system[J]. *The International Journal of Advanced Manufacturing Technology*, 2022, 119(9): 5615-5646.
- [3] Zhou W M, Yu F, Zhang J F, et al. An adaptive clamp system for deformation control of aerospace thin-walled parts[J]. *Journal of Manufacturing Processes*, 2023, 107: 115-125.
- [4] Napoleone A, Andersen A L, Brunoe T D, et al. Towards human-centric reconfigurable manufacturing systems: Literature review of reconfigurability enablers for reduced reconfiguration effort and classification frameworks[J]. *Journal of Manufacturing Systems*, 2023, 67: 23-34.
- [5] Gadalla M, Xue D Y. Recent advances in research on reconfigurable machine tools: A literature review[J]. *International Journal of Production Research*, 2017, 55(5): 1440-1454.
- [6] Ameer M, Dahane M. NSGA-III-based multi-objective approach for reconfigurable manufacturing system design considering single-spindle and multi-spindle modular reconfigurable machines[J]. *The International Journal of Advanced Manufacturing Technology*, 2023, 128(5): 2499-2524.
- [7] Fan J X, Zhang C J, Liu Q H, et al. An improved genetic algorithm for flexible job shop scheduling problem considering reconfigurable machine tools with limited auxiliary modules[J]. *Journal of Manufacturing Systems*, 2022, 62: 650-667.
- [8] Hu Y F, Zhang L P, Zhang Z K, et al. Flexible assembly job shop scheduling problem considering reconfigurable machine: A cooperative co-evolutionary matheuristic algorithm[J]. *Applied Soft Computing*, 2024, 166: 112148.
- [9] Pang Z, Yang B, Chen R H, et al. A multi-phase scheduling method for reconfigurable flexible job-shops with multi-machine cooperation based on a scout and mutation-based aquila optimizer[J]. *CIRP Journal of Manufacturing Science and Technology*, 2023, 46: 116-134.
- [10] Mahmoodjanloo M, Tavakkoli-Moghaddam R, Baboli A, et al. Flexible job shop scheduling problem with reconfigurable machine tools: An improved differential evolution algorithm[J]. *Applied Soft Computing*, 2020, 94: 106416.
- [11] Wang J, Xi X C, Ma J Y, et al. Study on material removal mechanism and control strategy of multi-axis fast ED-milling[J]. *Journal of Manufacturing Processes*, 2023, 101: 354-370.
- [12] Arrabiyeh P A, Setti D, Basten S, et al. Micro grinding 16MnCr<sub>5</sub> hardened steel using micro pencil grinding tools with diameters ~50 $\mu$ m[J]. *CIRP Journal of Manufacturing Science and Technology*, 2019, 27: 1-10.
- [13] Wang J, Ma J Y, Lu Y F, et al. Efficient machining of diffuser-shaped film cooling holes by multi-axis fast ED-milling[J]. *CIRP Annals*, 2023, 72(1): 173-176.
- [14] Luo Q, Deng Q W, Gong G L, et al. A distributed flexible job shop scheduling problem considering worker arrangement using an improved memetic algorithm[J]. *Expert Systems with Applications*, 2022, 207: 117984.
- [15] 王凌, 李瑞, 陈靖方. 面向人机协同能效车间调度的群智能优化算法[J]. *中国科学: 技术科学*, 2024, 54(9): 1676-1692.  
(Wang L, Li R, Chen J F. A swarm intelligence optimization algorithm for human-robot collaborative energy-efficient shop scheduling[J]. *Scientia Sinica(Technologica)*, 2024, 54: 1676-1692.)
- [16] 胡金昌, 吴颖颖, 王艳艳, 等. 考虑学习效应的单人作业车间调度算法[J]. *控制与决策*, 2022, 37(1): 37-46.  
(Hu J C, Wu Y Y, Wang Y Y, et al. Worker job shop scheduling algorithm considering learning effect[J]. *Control and Decision*, 2022, 37(1): 37-46.)
- [17] 张佳朋, 庄存波, 刘检华, 等. 基于超启发式算法的可重构装配车间调度[J]. *计算机集成制造系统*, 2025, 31(2): 399-410.  
(Zhang J P, Zhuang C B, Liu J H, et al. Reconfigurable assembly shop scheduling based on hyper heuristics algorithms[J]. *Computer Integrated Manufacturing Systems*, 2025, 31(2): 399-410.)
- [18] Li Y X, Li X Y, Gao L, et al. Multi-agent deep reinforcement learning for dynamic reconfigurable shop scheduling considering batch processing and worker cooperation[J]. *Robotics and Computer-Integrated Manufacturing*, 2025, 91: 102834.
- [19] 李益兵, 黄炜星, 吴锐. 基于改进人工蜂群算法的多目标绿色柔性作业车间调度研究[J]. *中国机械工程*, 2020, 31(11): 1344-1345.  
(Li Y B, Huang W X, Wu R. Research on multi-objective green flexible job-shop scheduling based on improved ABC algorithm[J]. *China Mechanical Engineering*, 2020, 31(11): 1344-1345.)
- [20] 刘超, 韩玉艳, 杨晓宇, 等. 多目标协同进化算法求解具有可变加工速度和自动导引车的柔性作业车间调度问题[J]. *聊城大学学报: 自然科学版*, 2025, 38(3): 317-332.  
(Liu C, Han Y Y, Yang X Y, et al. A multi-objective cooperative coevolutionary algorithm for solving the flexible job shop scheduling problem with variable processing speeds and automated guided vehicles[J].

- Journal of Liaocheng University: Natural Science Edition, 2025, 38(3): 317-332.)
- [21] 苏建涛, 董绍华, 朱诗敏. 多目标混合流水车间机器故障重调度问题研究[J]. 机械工程学报, 2024, 60(4): 438-448.  
(Su J T, Dong S H, Zhu S M. Research on machine fault rescheduling problem of multi objective hybrid flow shop based on intelligent manufacturing[J]. Journal of Mechanical Engineering, 2024, 60(4): 438-448.)
- [22] 赵仕存, 周泓. 强化学习驱动进化的模因算法求解准时制分布式柔性作业车间调度问题[J]. 控制与决策, DOI: 10.13195/j.kzyjc.2025.0310.  
(Zhao S C, Zhou H. Reinforcement learning-driven evolutionary memetic algorithm for solving the just-in-time distributed flexible job shop scheduling problem [J]. Control and Decision, DOI: 10.13195/j.kzyjc.2025.0310.)
- [23] 张梓琪, 钱斌, 胡蓉, 等. 基于多维EDA算法的低碳分布式装配流水车间调度[J]. 控制与决策, 2022, 37(5): 1367-1377.  
(Zhang Z Q, Qian B, Hu R, et al. ultidimensional estimation of distribution algorithm for low carbonscheduling of distributed assembly permutation flow-shop[J]. Control and Decision, 2022, 37(5): 1367-1377.)
- [24] 付威, 纪青然, 陈录城, 等. 禁忌搜索和NSGA-II算法融合求解多车间作业任务协同调度问题[J]. 计算机工程, DOI: 10.19678/j.issn.1000-3428.0070435.  
(Fu W, Ji Q R, Chen L C, et al. Tabu search and NSGA-II algorithm fusion to solve the problem of multi-workshop job tasks collaboration scheduling[J]. Computer Engineering, DOI: 10.19678/j.issn.1000-3428.0070435.)
- [25] Cheng L X, Tang Q H, Zhang L P, et al. Inventory and total completion time minimization for assembly job-shop scheduling considering material integrity and assembly sequential constraint[J]. Journal of Manufacturing Systems, 2022, 65: 660-672.
- [26] Zhang Z K, Tang Q H, Zhang L P, et al. A Q-learning-based multi-population algorithm for multi-objective distributed heterogeneous assembly no-idle flowshop scheduling with batch delivery[J]. Expert Systems with Applications, 2025, 263: 125690.
- [27] 王凌, 潘子肖. 基于深度强化学习与迭代贪婪的流水车间调度优化[J]. 控制与决策, 2021, 36(11): 2609-2617.  
(Wang L, Pan Z X. Scheduling optimization for flow-shop based on deep reinforcement learning and iterative greedy method[J]. Control and Decision, 2021, 36(11): 2609-2617.)
- [28] Cheng L X, Tang Q H, Zhang L P. Mathematical model and adaptive multi-objective evolutionary algorithm for cellular manufacturing with mixed production mode[J]. Swarm and Evolutionary Computation, 2024, 86: 101545.
- [29] Brandimarte P. Routing and scheduling in a flexible job shop by tabu search[J]. Annals of Operations Research, 1993, 41(3): 157-183.
- [30] Zhao F, Du Y, Zhuang C, et al. An iterative greedy algorithm for solving a multiobjective distributed assembly flexible job shop scheduling problem with fuzzy processing time[J]. IEEE Transactions on Cybernetics, 2025, 55(5): 2302-2315.

## 作者简介

杨志杰 (1988-), 男, 高级实验师, 硕士, 硕士生导师, 主要研究方向为智能制造服务协作、生产调度智能优化, E-mail: yangzhijie@whut.edu.cn;

梁穆熙 (1999-), 男, 硕士生, 主要研究方向为智能优化算法, E-mail: muxialex@whut.edu.cn;

李益兵 (1978-), 男, 教授, 博士, 主要研究方向为生产调度与智能优化, E-mail: ahlyb@whut.edu.cn;

汪开普 (1991-), 男, 副研究员, 博士, 主要研究方向为生产规划与调度、智能优化算法, E-mail: wangkaipu@whut.edu.cn;

梅立文 (1986-), 男, 硕士, 主要研究方向为智能制造与车间调度、设备健康管理, E-mail: meiliwenm@163.com;

郭顺生 (1963-), 男, 教授, 博士, 主要研究方向为CAD/CAM/PDM/ERP基础理论及应用, E-mail: guoshunsheng@whut.edu.cn.