

# 带有动态到达工件的分布式柔性作业 车间调度问题研究

张洪亮, 童超, 丁倩兰

(安徽工业大学管理科学与工程学院, 安徽马鞍山 243032)

**摘要:** 分布式柔性作业车间调度是生产调度的一个重要分支, 工件的动态到达作为实际生产中的1种常见扰动情况, 进一步增加了作业车间调度问题的复杂性和不确定性。针对带有工件动态到达的分布式柔性作业车间调度问题(DA-DFJSP), 提出1种分批调度策略, 将原本的动态调度问题转化成一系列连续调度区间上的静态调度问题, 构建以最大完工时间为优化目标的混合整数规划模型; 在此基础上, 结合问题特征采用批次、工厂、工序、机器的4层染色体编码及快速贪婪搜索插入的解码方式改进遗传算法, 同时引入多种交叉、变异算子来增强染色体的多样性; 最后, 基于FJSP标准算例构建DA-DFJSP测试算例进行仿真对比实验, 验证所提策略和改进算法的求解优势。结果表明: 相较于传统的重调度策略和改进前的遗传算法, 采用分批调度策略和改进的遗传算法(IGA)所求调度方案具有更短的完工周期、更均匀的工厂加工负荷及更高的设备工作效率, IGA与分批调度策略之间有高度的契合性, 能够有效提升生产效率。

**关键词:** 分布式柔性作业车间调度; 工件动态到达; 分批调度; 染色体编码; 遗传算法; 混合整数规划模型; 最大完工时间

中图分类号: TP 18 文献标志码: A doi: 10.12415/j.issn.1671-7872.24008



## Research on Distributed Flexible Job-shop Scheduling Problem with Dynamic Arrival of Jobs

ZHANG Hongliang, TONG Chao, DING Qianlan

(School of Management Science and Engineering, Anhui University of Technology, Maanshan 243032, China)

**Abstract:** Distributed flexible job shop scheduling is an important branch of production scheduling. As a common disturbance in the actual production, the dynamic arrival of jobs further increases the complexity and uncertainty of the job shop scheduling problem. Aiming at the distributed flexible job-shop scheduling problem with dynamic arrival of jobs (DA-DFJSP), a batching scheduling strategy was proposed, which transformed the original dynamic scheduling problem into a series of static scheduling problems over continuous scheduling intervals, and a mixed integer programming model was constructed with the maximum completion time as the optimization objective. On this basis, combined with the characteristics of the problem, the genetic algorithm was improved by the four-layer chromosome coding of batch, factory, process and machine, as well as a decoding method of fast greedy search insertion. At the same time, a variety of crossover and mutation operators were introduced to enhance the diversity of chromosomes. Finally, based on the FJSP standard example, a DA-DFJSP test case was constructed for simulation comparison experiments to verify the solution advantages of the proposed strategy and improved algorithm. The

收稿日期: 2024-01-08

基金项目: 安徽省哲学社会科学规划项目(AHSKY2022D117)

作者简介: 张洪亮(1979—), 男, 河北沧州人, 博士, 副教授, 主要研究方向为生产调度与优化。

引文格式: 张洪亮, 童超, 丁倩兰. 带有动态到达工件的分布式柔性作业车间调度问题研究[J]. 安徽工业大学学报(自然科学版), 2024, 41(5):573-582.

results show that compared to the traditional rescheduling strategy and the pre-improved genetic algorithm, the scheduling scheme proposed by the batched scheduling strategy and the improved genetic algorithm (IGA) has shorter completion period, more uniform plant processing load, and higher equipment work efficiency. There is a high degree of compatibility between IGA and the batch scheduling strategy, which can effectively improve production efficiency.

**Keywords:** distributed flexible job shop scheduling; jobs dynamic arrival; batch scheduling; chromosome coding; genetic algorithm; mixedinteger programming model; the maximum completion time

随着市场消费结构的升级以及生产系统的个性化、定制化、时效性需求的增强,多工厂、跨区域的分布式柔性制造模式正逐步上升为制造企业主流的生产模式。此外,实际生产过程中存在的各种因素影响,如设备故障、物料供应不足等均会增加生产的不确定性和不可预测性,给制造业的稳定生产带来巨大的考验和挑战,如何有效管理这些不确定性因素已成为当前制造业领域研究的1个重要课题。分布式柔性作业车间调度问题(distributed flexible job-shop scheduling problem, DFJSP)是指在生产过程中,通过合理地分配机器、工厂等资源,以较优的方式解决一系列工序加工的问题。这种问题在制造业中普遍存在,直接关系到生产效率、成本控制、交货期保证等关键方面。因此,研究DFJSP对于制造型企业优化生产资源配置、提高产能、降低生产风险以及增强自身可持续发展能力具有重要实践意义。

DFJSP涉及多个工件的工序在多工厂环境下生产资源的调度和管理,当面临如工件动态到达、紧急插单等突发性扰动事件时,可通过优化资源整合与分配实现生产过程的高效运行,确保任务的顺利完成。因此,DFJSP成为近年的研究热点,被学者们广泛关注。目前的研究主要分为两类:单目标DFJSP和多目标DFJSP。对于单目标DFJSP,为最大程度地减少最大完工时间,Giovanni等<sup>[1]</sup>设计1种新型遗传算法用于提高解的质量;Chang等<sup>[2]</sup>提出1种带有混合交叉策略的遗传算法,可提高算法勘探全局的能力;吴锐等<sup>[3]</sup>将1种嵌入三维向量编码的新型NSGA-II应用到DFJSP;孟磊磊等<sup>[4]</sup>提出1种混合蛙跳算法,采用多层编码以及基于机器选择规则的解码策略,同时结合变邻域搜索进一步增强算法的局部搜索能力。对于多目标DFJSP,通常着眼于同时优化如完工时间、能耗等在内的多个目标,在考虑不确定的加工时间、资源限制以及生产负载均衡等多种影响因素的基础上,探索有效的调度策

略和算法。吴秀丽等<sup>[5]</sup>建立考虑总成本和提前/延期惩罚的数学模型,并设计1种改进差分进化算法进行求解;Li等<sup>[6]</sup>以最小化最大完成时间、负载和提前或延期惩罚为目标,提出1种基于帕累托分布的禁忌搜索算法;Lu等<sup>[7]</sup>采用更精确的三维染色体和1种新型编码方式来加快算法求解速度;王凌等<sup>[8]</sup>以最小化最大完工时间和加工过程总能耗为优化目标,提出1种协同群智能算法;唐红涛等<sup>[9]</sup>考虑加工时间的不确定性,在引入三角模糊数的基础上,提出1种改进的灰狼优化算法以最小化最大模糊完工时间;Chang等<sup>[2]</sup>提出1种混合遗传算法,其中使用轮盘赌编码方法来解决工序分配问题;Meng等<sup>[10]</sup>基于不同建模思想建立4种混合整数规划模型和1种高效约束规划策略来求解多目标的DFJSP;Luo等<sup>[11]</sup>设计1种集成多亲本交叉、变异算子和多组邻域结构搜索操作的改进模因算法。DFJSP是1个复杂的研究领域,涵盖各种目标的集成优化。现有研究普遍基于静态调度环境的假设,即工件的信息和处理都是预先已知且不会随时间变化的,但由于工件的到达往往具有不确定性,这一假设在实际生产环境中并不完全成立。

动态车间调度问题的研究致力于解决生产过程中出现的动态事件,工件的动态到达(dynamic arriving, DA)作为实际生产普遍存在的动态事件,也引起学者们的关注。Wang等<sup>[12]</sup>建立11个混合整数规划模型,对传统的4种匹配策略进行改进,在此基础上提出1种改进的粒子群优化算法;Duan等<sup>[13]</sup>提出1种考虑系统综合可重用性的动态事件响应策略;Liu等<sup>[14]</sup>采用双深度Q网络强化学习算法,用以捕捉生产信息与调度目标之间的关系,基于此设计1种实时重调度策略;Zhang等<sup>[15]</sup>提出1种自适应的资源分配策略,设计可自动调度并启发式学习的重要性感知遗传算法;Zambrano等<sup>[16]</sup>提出1种融合遗传算法与工件到达时间信息的动态任务重调度策略;Li等<sup>[17]</sup>考虑设备加工时空转和待机状态

的能耗, 提出 1 种工件动态到达时更新工件与设备状态的优化方法, 并设计 1 种快速非支配排序遗传算法求解。综上可发现: 近年学者们对动态生产调度问题的研究多集中于单个车间类型的生产调度环境; 对于分布式柔性作业车间制造环境下工件 DA 问题的研究, 不论是在处理策略还是优化算法层面仍较少涉及。

重调度是动态车间调度问题常用的响应策略之一, 但是重调度策略往往会中断相邻工序的加工, 导致部分工件的整体加工时间延长, 对生产的连续性和产品质量带来负面影响。例如需要特殊生产环境制造业中的温控设备、精密仪器和测量仪器, 加工过程中较长的停滞时间会导致缺陷产品数量的增加。因此, 以带有动态到达工件的分布式柔性作业车间调度问题 (distributed flexible job-shop scheduling problem with dynamically arriving workpieces, DA-DFJSP) 为研究对象, 建立以最小化完工时间为目标的混合整数规划模型, 并提出 1 种分批调度策略; 在此基础上, 根据 DA-DFJSP 特点设计 1 种改进的遗传算法 (improved genetic algorithm, IGA) 对其进行求解, 获得较优的调度方案, 以期在响应动态到达工件的同时合理规划工件的加工顺序以及工厂、机器的分配, 实现企业最大化生产效率。

## 1 DA-DFJSP 问题模型的建立

对于带有动态到达工件的生产调度问题, 传统的重调度策略通常采用预设固定的调度周期或对动态到达的工件实施即时响应, 以启动相应的重调度操作。然而, 固定的重调度周期难以适应生产系统的动态变化, 尤其是在生产环境变化迅速的场景中。另一方面, 事件驱动型的重调度要求对每一个动态到达的工件均做出反应, 易导致重调度的频率过高, 不仅增加计算资源的消耗, 还削弱调度计划的稳定性。因此, 需结合 DA-DFJSP 设计与之契合的调度策略。

### 1.1 分批调度策略

设计的分批调度策略示意如图 1。由图 1 可看出: 在分批调度模型中 1 个批调度计划将从时刻  $t_n$  开始, 产生 1 个周期为  $\Delta T_n$  的调度区间, 每个调度区间需从当前等候区选择数量为  $q$  的工件进行调度, 未被选中的工件继续保留在等候区, 且与当前调度区间内动态到达的工件合并为下一个批调度计划候选工件集, 重复此操作直到等候区不存在待加工工件。分批调度策略需根据新工件随机到达的时刻, 对其进行调度区间的划分, 将原本的动态调度问题转化成一系列连续调度区间上的静态调度问题, 从而方便求解。

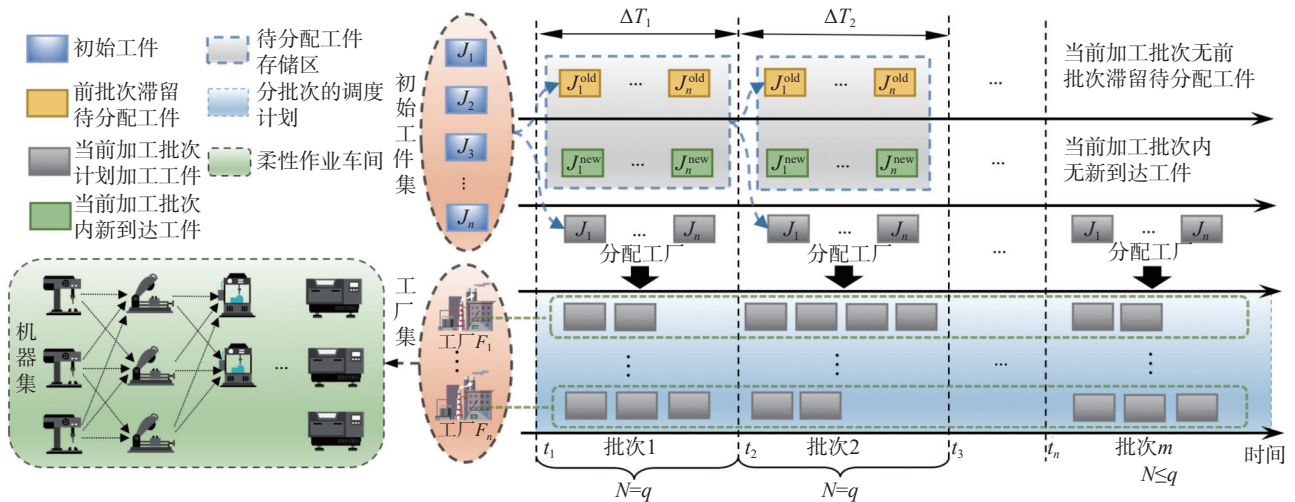


图1 分批调度策略示意图

Fig. 1 Schematic diagram of batch scheduling strategy

图 1 中字母符号与文中其他处用到的变量字母符号及含义如表 1。

### 1.2 问题描述

DA-DFJSP 问题可描述为: 存在  $N_i$  个拥有不同工序数的工件需被分配到  $N_n$  个柔性作业车间类型工厂加工, 其中部分工件  $J^{new}$  会在加工过程中随机到达。

表 1 符号变量定义

Tab. 1 Definition of symbolic variables

符号	含义
$i, i'$	工件索引
$j, j'$	工序索引
$k$	机器索引
$m, m', m''$	批次索引

续表

符号	含义
$n, n'$	工厂索引
$N_i$	工件总数
$N_j^i$	工件 <i>i</i> 的工序总数
$N$	每个批次实际加工工件数量
$q$	每批次固定加工工件数量
$N_m$	批次总数
$N_n$	工厂总数
$N_k^n$	工厂 <i>n</i> 机器总数
$J_i$	工件 <i>i</i>
$B_m$	批次 <i>m</i>
$F_n$	工厂 <i>n</i>
$M_k$	机器 <i>k</i>
$J^{\text{new}}, J^{\text{old}}$	工件属性, $J^{\text{new}}$ 为新到达的工件, $J^{\text{old}}$ 为新工件到达前已经存在的工件
$O_{ij}$	工件 <i>i</i> 的第 <i>j</i> 道工序
$t_i^C$	工件 <i>i</i> 的完工时间
$t_{\max}^C$	最大完工时间
$t_i^d$	工件 <i>i</i> 到达的时刻
$t_{mn}^S$	工厂 <i>n</i> 的第 <i>m</i> 批次开工时间
$t_{mn}^D$	工厂 <i>n</i> 的第 <i>m</i> 批次完工时间
$t_{mn}^P$	工厂 <i>n</i> 的第 <i>m</i> 批次加工时长
$t_{ijnk}^S$	工序 <i>O<sub>ij</sub></i> 在工厂 <i>n</i> 的第 <i>k</i> 台机器上的开工时间
$t_{ijnk}^D$	工序 <i>O<sub>ij</sub></i> 在工厂 <i>n</i> 的第 <i>k</i> 台机器上的完工时间
$t_{ijnk}^P$	工序 <i>O<sub>ij</sub></i> 在工厂 <i>n</i> 的第 <i>k</i> 台机器上的加工时长
$L$	一个极大的正数
$X_{ij}^{nk}$	如果工序 <i>O<sub>ij</sub></i> 在机器 <i>M<sub>nk</sub></i> 上加工则为1, 否则为0
$Y_{ij'j''}^{nk}$	如果机器 <i>M<sub>nk</sub></i> 加工完 <i>O<sub>ij'</sub></i> 后加工 <i>O<sub>ij''</sub></i> 则为1, 否则为0
$Z_i^{mn}$	如果工件 <i>i</i> 在工厂 <i>n</i> 的第 <i>m</i> 批次工作时间内到达则为1, 否则为0
$\alpha_i^{mn}$	如果工件 <i>i</i> 被分配在工厂 <i>n</i> 中第 <i>m</i> 批次加工则为1, 否则为0
$\beta_{mm'}^n$	如果工厂 <i>n</i> 的第 <i>m</i> 批次完工后加工第 <i>m'</i> 批次则为1, 否则为0

文中使用可处理动态随机到达工件的分批调度策略, 在各工件基本加工信息已知的情况下, 对各工件的加工批次、工厂以及作业机器进行合理分配, 调度目标为最小化最大完工时间。基于该问题提出以下假设:

- 1) 所有初始工件和机器0时刻可用, 新到达工件根据到达时间 $t_i^d$ 判定可用时间;
- 2) 每台机器同一时刻最多只能加工1个工件;
- 3) 工件不得跨越工厂加工;
- 4) 每个工件同一时刻只能在1台机器上加工;
- 5) 每个工件必须严格按照工序顺序完成加工;
- 6) 机器一旦开始加工不允许中断;
- 7) 同一时刻只允许执行1个批调度计划;
- 8) 分配至各批次的工件必须在该批次调度区

间内完成所有工序的加工;

9) 同一机器上不同工件间的转换时间忽略不计。

### 1.3 问题模型

基于以上描述与假设, 构建 DA-DFJSP 问题的数学模型:

$$\min t_i^C = \min \left( \max_{i=1}^{N_i} \{t_i^C\} \right) \quad (1)$$

s.t.

$$t_i^C = \max \left( t_{ijnk}^D \cdot X_{ij}^{nk} \right) \quad (2)$$

$$\sum_{m=1}^{N_m} \sum_{n=1}^{N_n} \alpha_i^{mn} = 1 \quad (3)$$

$$\sum_{n=1}^{N_n} \sum_{k=1}^{N_k^n} X_{ij}^{nk} = 1 \quad (4)$$

$$t_{mn}^D = t_{mn}^S + t_{mn}^P \quad (5)$$

$$t_{ijnk}^D = t_{ijnk}^S + t_{ijnk}^P \quad (6)$$

$$|t_{m'n}^S - t_{mn}^S| = t_{mn}^P \cdot \beta_{mm'}^n + L(1 - \beta_{mm'}^n) \quad (7)$$

$$\{t_{m'n}^S\} \cap \{t_{mn}^D\} = \emptyset \quad (8)$$

$$|t_{ij'j''}^S - t_{ijnk}^S| \geq t_{ijnk}^P \cdot Y_{ij'j''}^{nk} + L(1 - Y_{ij'j''}^{nk}) \quad (9)$$

$$\{t_{ij'j''}^S\} \cap \{t_{ijnk}^D\} = \emptyset \quad (10)$$

$$\begin{cases} t_{mn}^D - t_i^d \geq 0 \\ Z_i^{mn} = 1 \\ \min(t_{mn}^D - t_i^d) \end{cases} \quad (11)$$

$$Z_i^{mn} = \alpha_i^{m'n} \sum_{m'=m+1}^{N_m} \alpha_i^{m''n} + (1 - \alpha_i^{m'n}) \sum_{m''=m+1}^{N_m} \alpha_i^{m''n'} \quad (12)$$

式(1)为优化目标函数, 指最小化最大完工时间; 式(2)表示工件的完工时间为该工件最后一道工序完工的时间; 式(3)保证每个工件只能在1个批次中的1个工厂加工; 式(4)保证工件的每道工序只能选择在1个工厂的1台机器上加工; 式(5)保证批次的完工时间等于该批次的开工时间加上加工时长; 式(6)保证工序的完工时间等于该道工序的开始时间加上加工时长; 式(7)用来约束同一工厂内不同批次的先后关系; 式(8)保证1个工厂不得同时进行多个批次的加工; 式(9)用来约束在同一台机器上加工工序的先后关系; 式(10)保证1台机器不得同时进行多道工序的加工; 式(11)用来判定新到达工件所在的批次; 式(12)保证新到达工件的加工批次必须排在当前正在加工批次之后。

## 2 问题模型求解算法的改进

遗传算法(genetic algorithm, GA)用来模拟生物进化过程中的自然选择和遗传变异过程, 是1种

具有较强全局搜索能力的启发式算法, 可处理大规模复杂的问题。DA-DFJSP 涉及多个工厂、多台机器及众多的工件工序, 是 1 种较为复杂的 NP-hard 问题, 故文中选用 GA 求解 DA-DFJSP。为增强 GA 对动态环境的适应性和对多目标优化的处理能力, 在处理 DA-DFJSP 时能够发挥显著作用, 有效提升生产系统的决策质量, 对 GA 进行改进和拓展。

2.1 基于批次、工厂、机器、工序的 4 层编码

针对 DA-DFJSP 的特征, 设计 1 组 4 层染色体编码方式, 采用工序排序 (operation sequence, OS)、机器分配 (machine assignment, MA)、工厂分配 (factory assignment, FA)、批次分配 (batch assignment, BA) 4 个部分进行编码, 2 工厂、2 批次以及工序分别为 3, 4, 2 的 3 个工件编码方案如图 2。其中不同颜色代表不同工件: OS 序列中 1 个数字代表 1 个工件, 数字第几次出现表示对应工件的第几道工序, 如第三个数字 2 是第二次出现, 则表示工序  $O_{22}$ ; MA 序列为按工序排列的机器码, 对应数字代表该工序分配的机器序号, 如第 4 个数字 1 表示工序  $O_{21}$  在选定工厂内的第一台可用机器上加工; FA 序列为按工件排列的工厂码, 对应数字代表该工件分配工厂的序号; BA 序列为按工件排列的批次码, 对应数字代表该工件分配批次的序号。

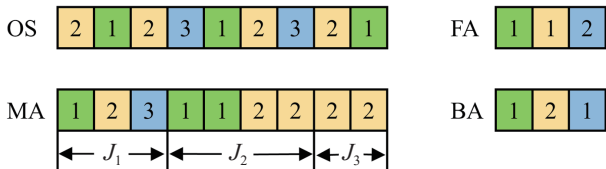


图 2 4 层编码示意图

Fig. 2 Schematic diagram of four-layer encoding

2.2 考虑批次的贪婪搜索插入解码

DA-DFJSP 中工件需按批次顺序加工, 但 OS 序列在经过进化操作后工件对应的批次会出现错乱的情况, 故在正式解码前需对 OS 序列进行批次修正。同样, MA 序列在进化过程中也可能出现所选机器不存在的情况, 故针对 MA 序列设计 1 种读码规则, 用于提高算法的容错率。假设工序  $O_{ij}$  在工厂中的可用机器数为  $p$ , 对应的 MA 序列位置为  $q$ , 若  $p \geq q$ , 则正常读取机器码  $q$ ; 若  $p < q$ , 则重复  $q - kp (k = 1, 2, 3, \dots, n)$  操作, 直到满足  $q - kp \leq p$ , 读取机器码  $q - kp$ , 即工序  $O_{ij}$  在工厂内的第  $q - kp$  台可用机器上进行加工。解码操作的具体步骤如下:

- 1) 根据 FA 序列确定所有工件的工厂分配情况;
- 2) 根据 BA 序列对 OS 序列进行批次修正, 读

取 MA 码并结合加工信息确定工序  $O_{ij}$  在加工机器  $M_k$  的加工时间  $t_{ijnk}^H$ ;

3) 引入贪婪思想, 全局搜索工序  $O_{ij}$  的加工机器  $M_k$  在工件  $i$  所在批次  $B_m$  内的所有可用时间段, 得到该机器的所有可用时间段;

4) 选择工序  $O_{ij}$  可插入的最早可用时间段, 如果无可插入的时间段就转步骤 5);

5) 确定机器  $M_k$  当前可开工时间  $T_k$ 、工序  $O_{ij}$  的紧前工序  $O_{ij}$  的结束时间  $t_{ij'nk}^G$  以及工件  $i$  的到达时间  $t_i^d$ , 再根据式 (13) 得出工序  $O_{ij}$  的最早可加工时刻  $t_{ij}^S$ ;

$$\begin{cases} T_{ij}^S = \max(T_k, t_i^d), & j = 1 \\ T_{ij}^S = \max(T_k, t_{i(j-1)nk}^G), & j > 1 \end{cases} \quad (13)$$

6) 依次进行以上操作获得最终解码结果并计算目标值。

2.3 种群初始化

种群初始化是算法的关键步骤, 直接影响算法的性能和收敛速度。由随机生成初始种群的方法得到的解质量易不稳定, 故设计 1 种考虑新到达工件的种群初始化方法。针对初始存在的工件, 根据单次加工的批量确定批次数, 结合初始工件数量随机生成批次码, 工序、机器及工厂序列则采用随机生成的方式; 以动态到达的新工件为对象在原先编码的基础上进行更新, 对当前染色体进行解码, 得到各工厂的批次加工时间区间, 并将工件到达时刻  $t_i^d$  同属于区间内的工件归为一组; 将同组的工件根据相关信息依次随机生成批次、工序、机器序列段并插入原先序列的对应位置, 具体操作示意如图 3。

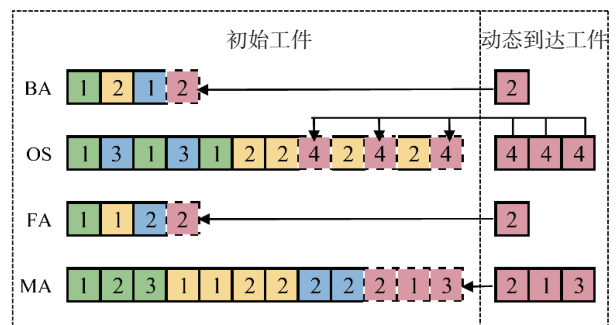


图 3 初始化操作示意图

Fig. 3 Schematic diagram of initialization operation

2.4 交叉算子

交叉操作有助于个体间优良特性的遗传和组合, 可增强算法的全局搜索能力。针对 FA, BA, OS, MA 4 个序列分别采用以下交叉方式。

- 1) FA 序列采用多点交叉操作, 过程如图 4。随机选择 1 对亲本染色体  $P_1, P_2$ , 再随机生成 1 组与 FA 序列等长且仅包含 0 和 1 的数组  $A$ , 并将亲本染

染色体中对应数组  $A$  所有位置为 1 的基因交换得到 2 个新的  $C_1, C_2$  子代。

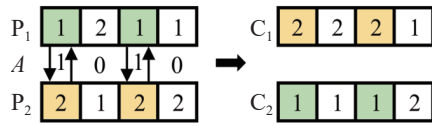


图4 FA序列的多点交叉操作示意图

Fig. 4 Schematic diagram of multi-point crossover operation for FA sequence

2) BA序列采用映射交叉操作。该操作在多点交叉操作的基础上增加 1 项冲突检测步骤,总的操作过程如图 5。随机选择 1 对亲本染色体  $P_1, P_2$ , 再随机生成 1 组与 BA 序列等长且仅包含 0 和 1 的数组  $A$ , 并将亲本染色体中对应数组  $A$  所有位置为 1 的基因交换, 此时根据所有交换的 2 组基因建立映射关系。以图 5 中 1-2 的映射关系为例, 可看到子代 1 比亲本多出 1 个基因 2, 子代 2 比亲本多出 1 个基因 1, 此时应通过映射关系随机选择交换位置以外的 1 个多余基因转变为映射关系对应的基因, 反复执行该操作直到子代没有冲突为止, 最后得到  $C_1, C_2$  的 2 个最终子代。

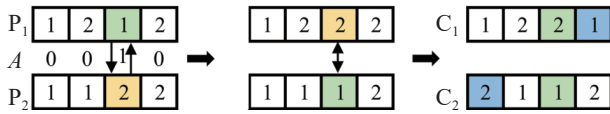


图5 BA序列的映射交叉操作示意图

Fig. 5 Schematic diagram of mapping crossover operation for BA sequence

3) OS序列采用顺序交叉操作, 操作过程如图 6。随机选择 1 对亲本染色体  $P_1, P_2$ , 并将总工件集合随机划分为 2 个无交集集合  $S_1, S_2$ ; 将  $P_1$  中包含集合  $S_1$  元素的基因以及  $P_2$  中包含集合  $S_2$  元素的基因以相同位置复制到子代  $C_1$ ; 将  $P_2$  中包含集合  $S_2$  元素的基因以及  $P_1$  中包含集合  $S_1$  元素的基因复制并顺序插入到子代  $C_2$ , 得到  $C_1, C_2$  的 2 个最终子代。

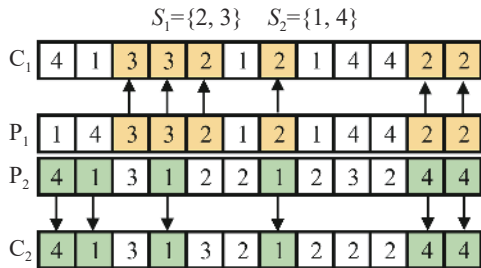


图6 OS序列的顺序交叉操作示意图

Fig. 6 Schematic diagram of sequential crossover operation for OS sequence

4) MA序列采用定位交叉操作, 操作过程如图 7。随机选择 1 对亲本染色体  $P_1, P_2$ , 再随机生成 1 组与

MA 序列等长且仅包含 0, 1 的数组  $A$ ; 选出  $P_1$  中与数组  $A$  中位置 1 对应的基因和  $P_2$  中与数组  $A$  中位置 0 对应的基因, 并复制到子代  $C_1$  相同的位置; 在  $P_2$  中选出与数组  $A$  中位置 1 对应的基因和  $P_1$  中选出与数组  $A$  中位置 0 对应的基因, 并复制到子代  $C_2$  相同的位置, 最后得到  $C_1, C_2$  的 2 个新子代。

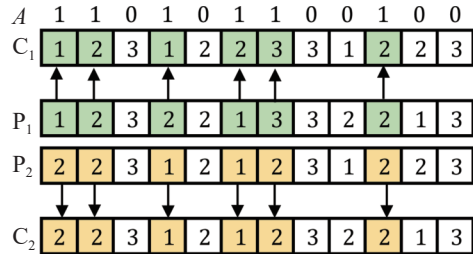


图7 MA序列的定位交叉操作示意图

Fig. 7 Schematic diagram of localization crossover operation for MA sequence

### 2.5 变异算子

变异操作引入随机性, 有助于跳出当前解的局部区域, 从而提高算法的全局搜索能力, 发现更多可能优的解。对于 FA, BA, OS, MA 4 个序列分别采用如下变异方式。

1) 对于 FA, BA 序列采用倒换变异方式, 具体操作如图 8。随机选择一段基因序列, 并将该序列中的基因顺序打乱, 未选中的基因序列不做任何变动。

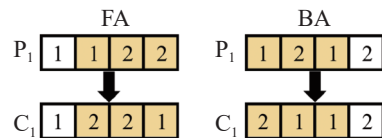


图8 FA和BA序列的倒换变异操作示意图

Fig. 8 Schematic diagram of inversion mutation operation for FA and BA sequences

2) 对于 OS 序列采用首尾交叉变异方式, 具体操作如图 9。在序列上随机选择 2 个互不相邻的位置  $L_1, L_2$  ( $L_1 < L_2$ ), 并将  $L_1$  之前 (包括  $L_1$ ) 的基因段与  $L_2$  之后 (包括  $L_2$ ) 的基因段交叉互换。

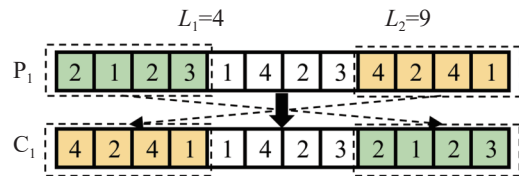


图9 OS序列的首尾交叉变异操作示意图

Fig. 9 Schematic diagram of head-tail crossover mutation operation for OS sequence

3) 对于 MA 序列的采用随机转换变异方式, 具体操作如图 10。随机选择 MA 序列上的 2 个基因,

分别替换成该基因位置对应工序可用机器集内的其他任意机器。

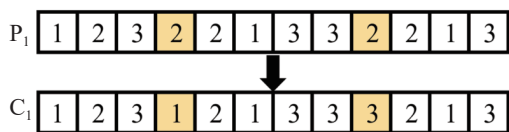


图10 MA 序列的随机转换变异操作示意图

Fig.10 Schematic diagram of random transformation mutation operation for MA sequence

### 3 实验与结果分析

为验证所提分批调度策略及改进 IGA 算法在处理 DA-DFJSP 的优势, 构建 DA-DFJSP 测试算例进行仿真实验。策略及算法基于 Python.3.10 实现, 并在配置 2.50 GHz/16 GB RAM 的 Intel(R) Core (TM) i5-12400F 的 PC 电脑上运行。

#### 3.1 算例构建及评价指标

鉴于目前没有 DA-DFJSP 测试算例, 基于 FJSP 标准算例 (MK01~MK10)<sup>[18]</sup>, 对其进行拓展修改作为测试算例。具体方式如下: DFJSP 中各个工序可在多个工厂的多台机器上进行加工且操作时间相互独立, 此外各工件的可选机器加工同一工序所需时间不同。因此设工序  $O_{ij}$  在  $F_n$  的第  $k$  台机器上的加工时长  $Q_{ij}^{nk}$  服从均匀分布  $Q_{ij}^{nk} \sim U(0, P_{\max}^a)$ ,  $Q_{ij}^{nk}=0$  表示工

序  $O_{ij}$  不能在机器  $M_k^n$  上加工,  $P_{\max}^a$  表示标准算例 a 所涉工件中耗时最长工序的加工时长; 工件  $i$  的到达时间  $t_i^d$  服从均匀分布  $t_i^d \sim U(0, \rho D_{\max}^a)$ <sup>[19]</sup>,  $t_i^d=0$  表示工件  $i$  为初始工件,  $D_{\max}^a$  为当  $\sum_{i=1}^{N_i} t_i^d=0$  且  $\rho=0.1, 0.5, 1.0$  时首次实验得到的  $t_{\max}^c$ 。基于生成的算例, 采用调度计划稳定性 (stability, STA)<sup>[19-20]</sup> 评估分批调度策略及 IGA 算法求解 DA-DFJSP 的有效性。

#### 3.2 策略对比实验结果分析

考虑加工设备数和工厂数恒定, 动态到达工件数在总工件数中所占不同比例时, 分别采用文中设计的分批调度策略 (Stra1)、传统的周期型重调度策略 (Stra2)、事件驱动型重调度策略 (Stra3) 与 IGA 算法对构建的 DA-DFJSP 算例进行测试。其中考虑总工件数  $N_i$  分别为 10, 20, 40, 60 时工厂数量为 2, 动态到达工件数与总工件数之比  $r$  为 0.1, 0.2, 0.3, 0.4, 0.5 的场景,  $N_k$  为所有工厂包含的机器总数量。IGA 算法的参数设置为: 单批加工工件数与初始待加工工件数之比为 0.8、种群规模为 200、最大迭代次数为 400。以 20 组算例为测试对象, 每组优化求解 20 次, 最终得到优化目标最大完工时间及 STA 指标数据, 结果如表 2, 表中每组算例求得指标的最优结果用粗体突出表示。

表 2 策略对比实验结果

Tab. 2 Comparison experimental results of strategies

算例	$N_i/N_k/r$	最大完工时间/min			STA		
		Stra1	Stra2	Stra3	Stra1	Stra2	Stra3
DA-01	10/4/0.1	<b>86</b>	99	93	<b>0.178 2</b>	0.368 5	0.442 4
DA-02	10/4/0.2	<b>85</b>	98	95	<b>0.175 7</b>	0.380 3	0.452 3
DA-03	10/4/0.3	<b>86</b>	97	94	<b>0.153 2</b>	0.379 3	0.438 0
DA-04	10/4/0.4	<b>89</b>	100	97	<b>0.123 6</b>	0.397 5	0.458 6
DA-05	10/4/0.5	<b>89</b>	103	99	<b>0.139 8</b>	0.379 2	0.431 9
DA-06	20/6/0.1	<b>80</b>	97	87	<b>0.324 3</b>	0.503 4	0.563 2
DA-07	20/6/0.2	<b>81</b>	95	89	<b>0.291 2</b>	0.441 9	0.473 1
DA-08	20/6/0.3	<b>83</b>	93	90	<b>0.253 5</b>	0.437 1	0.504 2
DA-09	20/6/0.4	<b>84</b>	95	93	<b>0.206 3</b>	0.467 6	0.521 3
DA-10	20/6/0.5	<b>83</b>	97	89	<b>0.182 4</b>	0.441 7	0.558 5
DA-11	40/12/0.1	<b>118</b>	149	132	<b>0.373 2</b>	0.582 5	0.624 0
DA-12	40/12/0.2	<b>120</b>	150	136	<b>0.393 7</b>	0.561 7	0.638 5
DA-13	40/12/0.3	<b>117</b>	153	142	<b>0.362 5</b>	0.547 3	0.592 6
DA-14	40/12/0.4	<b>123</b>	154	145	<b>0.338 1</b>	0.572 6	0.617 5
DA-15	40/12/0.5	<b>123</b>	151	140	<b>0.302 9</b>	0.564 1	0.625 1
DA-16	60/12/0.1	<b>176</b>	228	203	<b>0.524 1</b>	0.616 3	0.687 4
DA-17	60/12/0.2	<b>176</b>	232	212	<b>0.490 2</b>	0.637 2	0.691 6
DA-18	60/12/0.3	<b>179</b>	234	214	<b>0.471 3</b>	0.594 7	0.653 9
DA-19	60/12/0.4	<b>179</b>	249	204	<b>0.437 1</b>	0.562 0	0.664 0
DA-20	60/12/0.5	<b>177</b>	241	227	<b>0.381 9</b>	0.537 1	0.647 2

从表2可看出:在求解多组以最大完工时间为目标的DA-DFJSP时,与周期型重调度策略(Stra2)、事件驱动型重调度策略(Stra3)对比,基于IGA的分批调度策略(Stra1)在最大完工时间最小的指标上最优;结合STA计算结果,基于IGA的分批调度策略(Stra1)的STA值远小于事件驱动重调度策略(Stra2)及周期重调度策略(Stra3);此外,随着动态到达工件占比的提高,基于IGA的分批调度策略(Stra1)的求解结果依然保持稳定,另一角度说明本文策略能够解决不同规模的DA-DFJSP。

### 3.3 算法对比实验结果分析

为进一步验证IGA与分批调度策略的适配性,选取改进前的GA及被广泛使用的粒子群优化算法(particle swarm optimization, PSO)作为对照,在统一采用分批调度策略的条件下与IGA进行对比实验。对比算法的基本参数设置与IGA相同。以算例DA-08, DA-13作为测试对象,每个算法独立运行10次,分别记录其中最优化调度方案的关键调度指标,

包括最大完工时间、工厂负载差值之和及设备平均空闲率,其中每个指标的最优结果以粗体突出表示,结果如表3。由表3可见:采用IGA获得的调度方案在所有关键指标上均展现出最优性能。

表3 算法对比实验结果

算例	算法	调度指标		
		最大完工时间/min	工厂负载差值之和/min	设备平均空闲率/%
DA-08	IGA	<b>83</b>	<b>8</b>	<b>15.45</b>
	GA	87	18	19.35
	PSO	96	11	23.13
DA-13	IGA	<b>117</b>	<b>7</b>	<b>26.29</b>
	GA	136	13	36.07
	PSO	138	12	39.74

图11为IGA, GA, PSO对算例DA-08, DA-13求解得到调度方案的甘特图。其中 $F_1$ 和 $F_2$ 表示2个符合工件加工要求的柔性制造车间类型工厂。

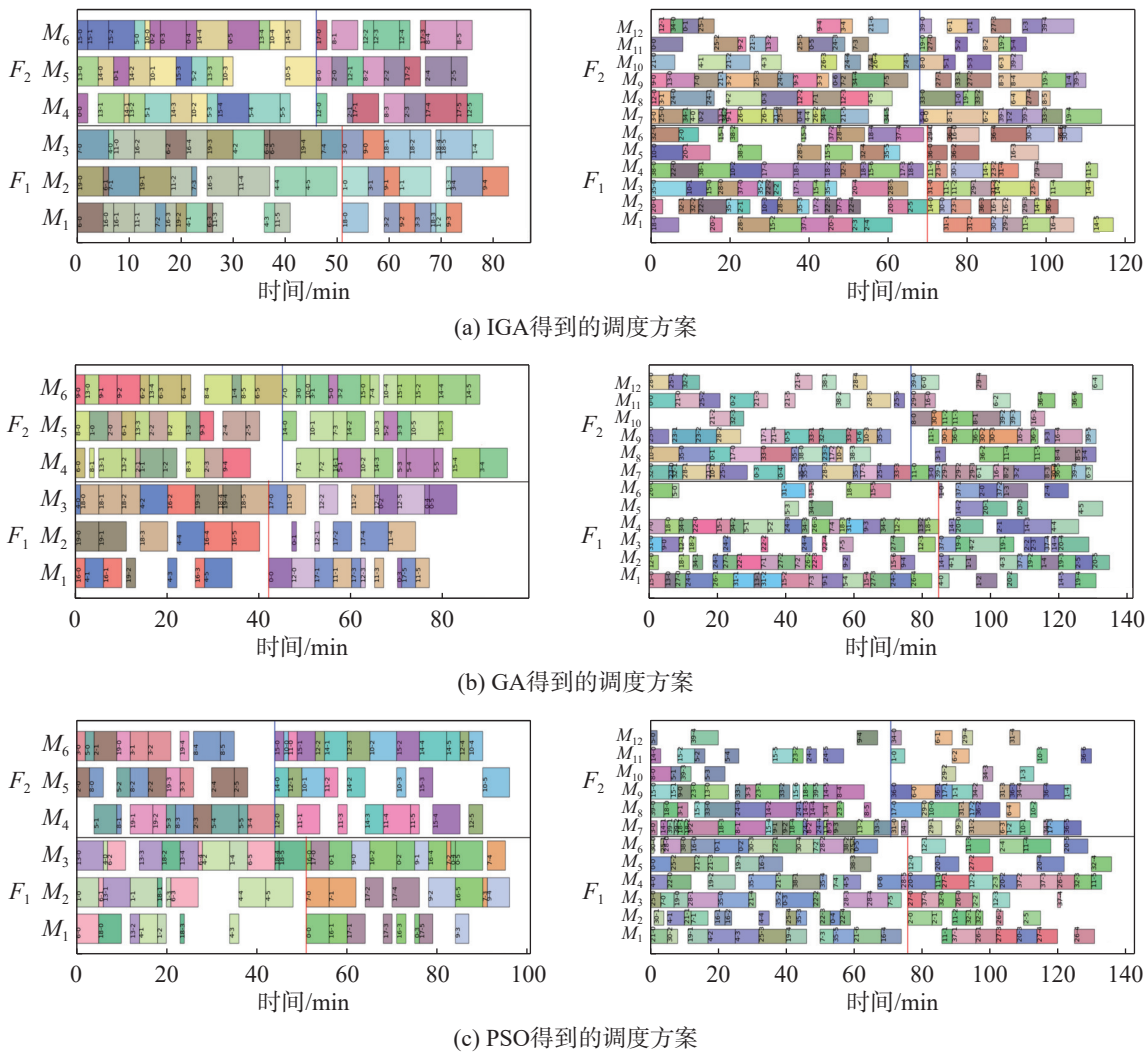


图11 DA-08(左)、DA-13(右)3种算法求得的调度方案

Fig. 11 Scheduling schemes of three algorithms for DA-08(left) and DA-13(right)

由图 11 可看出: 采用 IGA 所得调度方案在完工时间指标上最优, 实现了最短的完工周期; 同时各工厂的加工负荷也更均匀, 设备的总空闲时间也显著小于其他 2 种算法求得的调度方案。这一结果直观地揭示了 IGA 与分批调度策略之间的高度契合性, 能够有效提升生产效率。这主要是因为贪婪搜索插入式解码使工件对空闲时间的搜索范围更有针对性, 能够有效提高算法的求解效率; 考虑新到达工件并区分批次的种群初始化方法能够获取更多的高质量染色体段并加以利用, 有助于种群更高效地完成后续大规模进化操作; 针对批次编码的单组映射交叉操作增加了多段优质染色体间的交互, 防止由于批次限制导致算法陷入局部最优。

## 4 结论

针对带有动态到达工件的分布式柔性作业车间调度问题 (DA-DFJSP), 以最大完工时间为优化目标, 在传统重调度策略的基础上提出 1 种分批调度策略, 构建以最大完工时间为优化目标的混合整数规划模型, 设计 1 种与其适配的改进遗传算法。此外, 设计多组 DA-DFJSP 的测试算例进行对比效果验证。结果表明: 分批调度策略及改进遗传算法 (ICA) 能够有效处理 DA-DFJSP, 与周期型重调度策略、事件驱动型重调度策略对比, 基于 IGA 的分批调度策略在最大完工时间最小的指标上最优; 且随动态到达工件占比的提高, 其求解结果依然保持稳定, 另一角度说明本文策略与改进算法能够解决不同规模的 DA-DFJSP。本文结果可为 DA-DFJSP 领域研究提供新的思路和方法, 也可为制造企业的实际应用提供理论支持和参考。未来的研究需结合实际生产中更为复杂的问题, 如具有不同优先级等级的动态到达工件、包含返工计划的动态调度以及利用强化学习对工件到达时刻进行预测等。

### 参考文献:

- [1] GIOVANNI L D, PEZZELLA F. An improved genetic algorithm for the distributed and flexible job shop scheduling problem[J]. *European Journal of Operational Research*, 2010, 200(2):395–408.
- [2] 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, 28:1973–1986.
- [3] 吴锐, 郭顺生, 李益兵, 等. 改进人工蜂群算法求解分布式柔性作业车间调度问题 [J]. *控制与决策*, 2019, 34(12): 2527–2536.
- [4] 孟磊磊, 张彪, 任亚平, 等. 求解分布式柔性作业车间调度的混合蛙跳算法 [J]. *机械工程学报*, 2021, 57(17): 263–272.
- [5] 吴秀丽, 刘夏晶. 差分进化算法求解分布式柔性作业车间调度问题 [J]. *计算机集成制造系统*, 2019, 25(10): 2539–2558.
- [6] LI J Q, DUAN P Y, CAO J D, et al. A hybrid Pare to-based tabu search for the distributed flexible job shop scheduling problem with E/T criteria[J]. *IEEE Access*, 2018, 6: 58883–58897.
- [7] 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, 29:19–34.
- [8] 王凌, 王晶晶. 考虑运输时间的分布式绿色柔性作业车间调度协同群智能优化 [J]. *中国科学: 技术科学*, 2023, 53(2):243–257.
- [9] 唐红涛, 李悦, 王磊. 模糊分布式柔性作业车间调度问题的求解算法 [J]. *华中科技大学学报 (自然科学版)*, 2022, 50(6):81–88.
- [10] MENG L, ZHANG C, REN Y, et al. Mixed-integer linear programming and constraint programming formulations for solving distributed flexible job shop scheduling problem[J]. *Computers & Industrial Engineering*, 2020, 142:106347.
- [11] LUO Q, DENG Q, GONG G, et al. An efficient memetic algorithm for distributed flexible job shop scheduling problem with transfers[J]. *Expert Systems with Applications*, 2020, 160:113721.
- [12] WANG Z, ZHANG J, YANG S. An improved particle swarm optimization algorithm for dynamic job shop scheduling problems with random job arrivals[J]. *Swarm and Evolutionary Computation*, 2019, 51:100594.
- [13] DUAN J, WANG J. Robust scheduling for flexible machining job shop subject to machine breakdowns and new job arrivals considering system reusability and task recurrence[J]. *Expert Systems with Applications*, 2022, 203:117489.
- [14] LIU R, PIPLANI R, TORO C. Deep reinforcement learning for dynamic scheduling of a flexible job shop[J]. *International Journal of Production Research*, 2022, 60(13):4049–4069.
- [15] ZHANG F, MEI Y, NGUYEN S, et al. Importance-aware

- genetic programming for automated scheduling heuristics learning in dynamic flexible job shop scheduling[C]// International Conference on Parallel Problem Solving from Nature. Cham: Springer International Publishing, 2022:48–62.
- [16] ZAMBRANO R G, BEKRAR A, PRABHU V, et al. Coupling a genetic algorithm with the distributed arrival-time control for the JIT dynamic scheduling of flexible job-shops[J]. *International Journal of Production Research*, 2014, 52(12):3688–3709.
- [17] LI Y, HE Y, WANG Y, et al. An optimization method for energy-conscious production in flexible machining job shops with dynamic job arrivals and machine breakdowns[J]. *Journal of Cleaner Production*, 2020, 254:120009.
- [18] LUNARDI W T, VOOS H. An extended flexible job shop scheduling problem with parallel operations[J]. *ACM SIGAPP Applied Computing Review*, 2018, 18(2):46–56.
- [19] WANG S, ZHANG C, LIU Q, et al. Dynamic scheduling of flexible job shop under different rescheduling cycles[J]. *Computer Integrated Manufacturing System*, 2014, 20(10):2470–2478.
- [20] RANGSARITRATSAMEE R, FERRELL JR W G, KURZ M B. Dynamic rescheduling that simultaneously considers efficiency and stability[J]. *Computers & Industrial Engineering*, 2004, 46(1):1–15.

责任编辑:丁吉海