

分类号: TP242.6

密 级: 公开

学校代码: 10363

学 号: 2210320117



安徽工程大学

Anhui Polytechnic University

硕士学位论文

题 目 面向节能的柔性车间动态调度规则
自动化生成方法研究

论 文 作 者 费宝林

校 内 导 师 徐彬梓

校 外 导 师

专业学位类别 控制科学与工程

研究方向(领域) 控制理论与控制工程

论文提交日期: 2024 年 5 月 31 日

分类号: TM74

单位代码: 10363

密 级: 公开

学 号: 2210320117

题 目 面向节能的柔性车间动态调度规则
自动化生成方法研究

题 目 Research on Automatic Generation of
Energy-efficient Dispatching Rules for
Dynamic Flexible Job Shop Scheduling

论文作者: 费宝林

指导教师: 徐彬梓

专 业: 控制科学与工程

研究方向: 控制理论与控制工程

论文答辩日期: 2024/5/15

安徽工程大学学位论文原创性声明

本人郑重声明：我恪守学术道德，崇尚严谨学风。所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已明确注明和引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品及成果的内容。论文为本人亲自撰写，我对所写的内容负责，并完全意识到本声明的法律后果由本人承担。

学位论文作者签名： 费宝林

日期： 2024 年 5 月 31 日

安徽工程大学学位论文版权使用授权书

学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅或借阅。本人授权安徽工程大学可以将本学位论文的全部内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密 ☐，在 _____ 年解密后适用本版权书。

本学位论文属于

不保密 ☒。

学位论文作者签名：

费宝林

指导教师签名：

徐林祥

日期： 2024 年 5 月 31 日

日期： 2024 年 5 月 31 日

面向节能的柔性车间动态调度规则自动化生成方法研究

摘 要

柔性制造系统是一种先进的制造工艺和管理方法，旨在提高制造业生产线的灵活性、适应性以及效率。目前，柔性制造系统已经在众多行业得到广泛应用，包括电子产业、航空航天、汽车制造、医疗设备制造等领域。与此同时，柔性车间动态调度(Dynamic Flexible Job Shop Scheduling, DFJSS)问题突显了在车间生产过程中实现柔性调度和动态调度的复杂性，成为一个现实而具有挑战性的组合优化问题。

启发式方法由于能在一定时间内获得合理的调度方案而被大量研究证实为解决车间调度问题的有效方法，但它在处理动态环境时仍存在计算量大、难以实时调度和调度稳定性等问题。超启发式遗传规划算法(Genetic Programming Hyper Heuristic, GPHH)由于可以根据当前车间的状态，通过自动化生成调度规则进行实时决策，相较于启发式算法在处理动态事件上具有明显优势。然而，目前大多数基于启发式规则的研究都侧重于完工时间和延迟时间等时间相关的优化目标，却忽略了对能耗的关注，这无法满足当今社会对绿色制造的迫切需求。

为此，本文通过对柔性制造系统的能量流进行分析，确定了系统中的主要能耗来源及其能耗方式。在此基础上，设计了适用于 GPHH 算法的终端符来表征这些的能耗特征信息。之后，研究了基于双重特征权重集合的 GPHH 算法，并将其应用于面向节能的调度规则自动启发式设计。

本文的主要研究内容如下：

(1) 基于动态柔性车间生产调度过程中能量流动的分析，设计了面向节能的终端符和人工调度规则。本文重点考虑了设备加工能耗和待机能耗，并将它们的能耗之和作为整个车间的优化目标。通过在不同生产情境下与已有时间相关终端符进行对比实验，验证了所提出的能耗终端符在节能降耗方面的有效性。

(2) 考虑到多目标问题中特征数量较大所导致的 GPHH 算法搜索空间大、计算时间长、收敛速度较慢等问题，本文在 GPHH 算法中融入双重特征权重集合的思想，提出了一种面向多目标动态柔性车间调度问题的特征权重度量方法，旨在优化 GPHH 算法的性能。所提特征权重方法根据特征的多目标贡献度为其分配适当权重，并在后续迭代过程中作为特征被选中的概率，在保证搜索空间的同时为 GPHH 算法提供方向性指引。实验结果表明，相较于已有的相关研究，所提方法在处理多目标问题时能在相近时间内显著提高所得节能调度规则的优化性能和可解释性。

关键词：动态柔性作业车间，超启发式遗传规划，调度规则，节能调度，双重特征权重集合

RESEARCH ON AUTOMATIC GENERATION OF ENERGY-EFFICIENT DISPATCHING RULES FOR DYNAMIC FLEXIBLE JOB SHOP SCHEDULING

ABSTRACT

Flexible Manufacturing Systems represent an advanced manufacturing process and management approach aimed at enhancing flexibility, adaptability, and efficiency in production lines. Presently, it has found widespread application across various industries, including electronics, aerospace, automotive manufacturing, medical equipment production, among others. Concurrently, the emergence of Dynamic Flexible Job Shop Scheduling (DFJSS) underscores the complexity of achieving flexible and dynamic scheduling in job shop production processes, presenting a practical and challenging combined optimization problem.

Heuristic methods have been extensively researched and proven effective for addressing job shop scheduling problems due to their capability to generate reasonable scheduling solutions within a specified timeframe. However, challenges persist when dealing with dynamic environments, including computational complexity, real-time scheduling, and scheduling stability. Genetic Programming Hyper Heuristic (GPHH) algorithms offer distinct advantages over heuristic methods in handling

dynamic events by automatically generating dispatching rules based on the current workshop state. Nevertheless, most heuristic-based studies predominantly focus on time-related optimization objectives such as makespan and tardiness while neglecting energy consumption, which fails to meet the urgent societal demand for green manufacturing.

Therefore, this paper analyzes the energy flow of flexible manufacturing systems and determines the main sources of energy consumption in the system and their energy consumption patterns. On this basis, terminals suitable for GPHH algorithms are designed to represent these energy consumption feature information. Afterwards, the GPHH algorithm based on dual feature weight sets is studied and applied to the automatic heuristic design of energy-efficient dispatching rules.

The main research contributions of this paper are as follows:

(1) Analysis of energy flow in dynamic flexible workshop production scheduling processes and design of energy-efficient terminals and manual dispatching rules. The paper particularly considers equipment processing energy consumption and standby power consumption, taking their sum as the optimization objective for the entire workshop. Comparative experiments with existing time-related terminals under different production scenarios validate the effectiveness of the proposed energy consumption terminals in energy savings.

(2) Considering the challenges posed by the large number of features

in multi-objective problems, leading to issues such as a large search space, lengthy computation time, and slow convergence in GPHH algorithms, this paper integrates the idea of dual-feature weight sets into the GPHH algorithm. It proposes a feature weight measurement method for multi-objective dynamic flexible workshop scheduling problems, aiming to optimize the performance of the GPHH algorithm. The proposed feature weight method assigns appropriate weights to features based on their multi-objective contribution and subsequently utilizes these weights as the probability of feature selection in subsequent iterative processes, providing directional guidance for the GPHH algorithm while ensuring the exploration of the search space. Experimental results demonstrate that compared to existing relevant studies, the proposed method significantly improves the optimization performance and interpretability of obtained energy-efficient dispatching rules for multi-objective problems within a comparable timeframe.

KEY WORDS: dynamic flexible job shop scheduling, genetic programming hyper heuristic, dispatching rules, energy-efficient scheduling, dual-feature weight sets.

目录

摘 要.....	I
ABSTRACT.....	III
第 1 章 绪论.....	1
1.1 研究背景及意义.....	1
1.2 国内外研究现状.....	2
1.2.1 车间调度问题类型.....	2
1.2.2 面向能耗的车间调度问题.....	3
1.2.3 智能优化方法.....	4
1.4 本文主要研究内容.....	7
第 2 章 基于 GPHH 求解多目标动态柔性车间调度问题.....	9
2.1 引言.....	9
2.2 动态柔性车间调度问题描述.....	9
2.3 基于调度规则的动态柔性车间调度决策.....	10
2.3.1 调度规则的表达形式.....	10
2.3.2 人工调度规则.....	11
2.3.3 基于调度规则的 DFJSS 调度决策过程	12
2.4 超启发式遗传规划算法.....	13
2.4.1 GPHH 算法的基本流程.....	14
2.4.2 GPHH 的初始化方法.....	15
2.4.3 GPHH 的遗传操作算子.....	16
2.5 本章小结.....	17
第 3 章 面向节能的终端符设计.....	18
3.1 引言.....	18
3.2 柔性车间制造系统的能流分析.....	18
3.3 能耗相关终端符设计.....	21
3.4 能耗相关人工规则设计.....	21
3.5 仿真实验设计.....	22

3.5.1 GPHH 算法的终端集合和函数集合.....	22
3.5.2 实验对比算法.....	23
3.5.3 仿真模型构建.....	24
3.5.4 GPHH 参数设置.....	26
3.5.5 算法性能指标.....	26
3.7 实验结果分析.....	28
3.7.1 算法总体性能分析.....	28
3.7.2 不同终端符集合对不同目标的优化效果分析.....	32
3.7.3 算法运行时间.....	33
3.7.4 终端符出现频率.....	34
3.8 本章小结.....	37
第 4 章 基于双重特征权重集合的节能调度规则自动化生成.....	38
4.1 引言.....	38
4.2 基于双重特征权重集合的 GPHH 算法.....	38
4.2.1 算法总体框架.....	38
4.2.2 基于适应度值的特征权重度量方法.....	40
4.2.3 基于多样性的特征权重度量方法.....	41
4.2.4 基于特征权重的种群调整策略.....	42
4.3 仿真实验设计.....	43
4.3.1 实验对比算法.....	43
4.3.2 仿真模型构建.....	44
4.3.3 GPHH 参数设置.....	45
4.3.4 算法性能指标.....	46
4.4 实验结果分析.....	47
4.4.1 算法总体性能分析.....	47
4.4.2 算法训练时间和测试时间对比.....	48
4.4.3 特征权重机制的有效性分析.....	49
4.4.4 种群调整策略的有效性分析.....	51
4.4.5 特征种类数量分析.....	53

4.4.6 特征权重与特征频率的联合分析.....	54
4.5 本章小结.....	57
第 5 章 总结与展望.....	59
5.1 全文总结.....	59
5.2 工作展望.....	60
参考文献.....	61
攻读硕士期间取得的学术成果.....	68
致谢.....	69

第 1 章 绪论

1.1 研究背景及意义

制造业是国民经济的支柱，同时也是全球能源消耗和碳排放的主要行业之一，占据全球能源消耗的 33%和全球碳排放的 36%^[1]。根据《2023 年世界能源统计报告》数据，2022 年，全球能源消耗和碳排放量分别增长 1%和 0.8%，而中国能源消耗和碳排放量分别增长 3.1%和 1.6%；全球煤炭消耗增加 0.6%，需求的主要推动力来自中国和印度；全球煤炭产量增长 7.9%，达到 88 亿吨，创下历史最高水平^[2]。因此，在确保经济快速发展的同时，缓解能源和环境之间的紧张关系，实现人与自然的和谐共生，已经成为全球共同面临的挑战。在这一背景下，节能减排、绿色制造已经成为未来发展的主要趋势。

截至目前，已有超过 130 个国家和地区提出实施净零排放或碳中和目标的计划。2019 年 6 月，英国修改《气候变化法》，将净零排放的目标从原来的 80%提升至 100%，是全球首个立法承诺在 2050 年实现净排放的主要经济体^[3]。2020 年 9 月，习近平总书记在联合国大会上发表演讲表示，中国将努力争取在 2030 年前实现碳达峰，并计划在 2060 年前实现碳中和^[4]。2020 年 12 月，日本政府发布了《2050 年碳中和绿色增长战略》，此战略涵盖了 14 个产业，包括能源、运输、制造业等，提出了各个产业在碳减排方面的具体目标和措施^[5]。2021 年 6 月，欧盟在《欧洲气候法》中，要求各成员国承诺到 2030 年将温室气体排放量减少 55%，并计划在 2050 年前实现碳中和^[6]。

中国作为全球制造业的重要参与者，其制造业在国家经济中占据重要地位。根据中国国家统计局的数据，2023 年第二产业（制造业、采矿业等）占国内生产总值的 39.3%^[7]。《中国能源统计年鉴 2022》指出 2021 年中国制造业能耗总量在全国能耗总量的 55.7%^[8]。为解决中国制造业存在的能耗高、能源利用率低以及环保问题，政府在十四五现代能源规划中明确了任务，即继续优化能源结构、推动能源绿色变革，以实现在制造业升级过程中降低能耗和碳排放的目标。

随着社会的进步和科技的发展，人类正迈入以高科技为主导的知识经济时代，制造业也正经历着信息化、绿色化、智能化的时代变革。新型制造系统通过引入

现代智能技术，致力于改善当前的生产模式，以更好地应对市场竞争和客户需求的个性化和多样化。生产调度的智能化是确保整个生产系统高效运转的关键问题，它涉及如何在有限的时间内，合理分配可用的生产资源给不同的生产任务，从而确保产品交付和资源成本的最优利用。

基于以上背景，本文以柔性车间调度问题为出发点，研究超启发式遗传规划算法与规则调度相融合的作业车间调度方法，利用自动生成的调度规则解决面向节能的多目标动态优化调度问题。通过对柔性车间加工过程的能量流动分析和特征权重方法的研究，从而为超启发式遗传规划算法自动生成节能调度规则提供理论支持，在调度规则自动生成的过程中，通过迭代学习来提高调度规则的性能。通过深入研究面向节能的柔性车间动态调度规则自动化生成方法，我们有望为制造业的智能化和绿色升级提供有力支持。这不仅有助于提高中国制造业的国际竞争力，也为全球制造业的可持续发展贡献智慧和解决方案。

1.2 国内外研究现状

1.2.1 车间调度问题类型

作业车间调度(Job Shop Scheduling, JSS)是指针对一项可分解的加工任务，在满足约束条件的前提下，对生产资源进行合理分配，以完成加工任务，并获得一些性能指标的最优化（例如，最小化总加工时间、流程时间和延迟时间）^[9]。在典型的作业车间调度问题中，每个工件需要按照特定的顺序在不同的设备上完成一系列的加工操作，且每个设备同时只能处理一个工件。因此需要对加工顺序进行合理安排，以最大程度地提高生产效率。

在柔性车间调度问题(Flexible Job Shop Scheduling, FJSS)中，设备资源的使用比 JSS 更灵活，它定义一个工件的工序可以在多台候选设备上加工，这更符合实际情况，同时也增加了问题的复杂性^[10, 11]。因此，FJSS 问题的解决涉及到路径决策和排序决策的设计，即需要先将工件分配给合适的加工设备，然后在设备的等待队列中将待加工工件合理排序并依次加工。

此外，实际车间环境通常是动态的，在动态柔性车间调度(Dynamic Flexible Job Shop Scheduling, DFJSS)过程中经常会发生不可预测的随机事件（例如新工件

到达、设备故障、工件返工等)^[12]。因此, DFJSS 需要实时调整生产计划和调度方案, 以适应突发事件并最大程度地减少生产中断和资源浪费。这也使得 DFJSS 问题相较于 JSS 和 FJSS 问题更加复杂和具有挑战性, 需要结合智能算法和实时数据分析等技术来实现高效的调度和优化。

1.2.2 面向能耗的车间调度问题

过去, 企业为了降低能源消耗, 主要集中于开发更节能的生产设备或改进生产工艺。而在车间调度问题的研究中, 更主要关注的是诸如最大完工时间、交货期、生产成本成本等时间和经济方面的指标。事实上, 车间调度方案对制造系统的能耗有着重要影响。在倡导制造业可持续发展的大背景下, 制造公司正致力于寻求更节能的调度策略^[13]。

近年来涌现了许多有关车间调度的节能研究, 主要包括优化加工设备选择和优化工序加工顺序两个方面。前者说明不同的加工设备, 对应着不同的加工功率和时间, 从而导致加工产生的能耗也不同, 对设备进行优选可以实现节能。后者说明通过调整各工序在设备上的加工顺序, 能够改变相邻工序之间的空闲时间, 从而影响设备空闲状态的能耗。

He 等人^[14]提出了一种能源感知优化方法, 结合加工设备选择和工序加工顺序, 致力于解决柔性车间的节能问题。卫少鹏等人^[15]构建了一个具有调整时间的绿色柔性车间调度模型, 综合考虑了设备的调整和加工环节所产生的时间和能耗, 并提出了一种新型头脑风暴优化算法, 实现对完工时间和总能耗两个目标的优化。戴敏等人^[16]针对作业车间调度问题, 构建了一种自动引导小车运输和加工资源集成调度的多目标优化调度模型, 并提出了一种结合模拟退火搜索策略的分布估计算法来求解这个模型。Lei 等人^[17]构建了一个最小化碳排放和完工时间的作业车间多目标优化模型, 并提出了一种改进动态邻域搜索的优化方法。

可以看出, 为了实现作业车间的绿色调度, 首先需要深入了解作业车间的能量流动过程。最初, 对作业车间的能流分析主要集中于加工设备的能耗过程以及在设备进行加工时涉及的能量流动机理。易茜等人^[18]从广义能量效率的角度出发, 建立了数控加工系统的优化模型。Xie 等人^[19]用广义随机 Petri 网来预测加工过程中的能耗和生产时间, 并通过实验数据分析了加工参数对平均加工时间、

平均功率、平均耗能的影响。此外，很多专家学者也对作业车间系统层面的能流分析进行了大量的研究。Brundage 等人^[20]利用传感器构建了一个面向能耗的性能指标体系，并基于此分析制造系统的动态特性和能耗瓶颈。Thiede 等人^[21]提出了一种基于任务加工过程链的机械加工车间的能耗建模方法，并用来预测机械加工车间不同层次的能耗。

1.2.3 智能优化方法

(1) 精确算法与元启发式算法

JSS 和 FJSS 问题作为典型的 NP-hard 问题，由于在制造领域的广泛应用且具有复杂性，已经引起了专家学者的广泛关注和深入研究。在问题规模较小时，研究人员倾向于采用精确求解方法，如数学规划法^[22]和分支界定法^[23]等。这些精确求解方法在小规模问题上表现出色，能够为问题提供确切的最优解，为高效生产提供了可靠的支持。然而，随着问题规模的增加，这些精确求解方法面临指数级的计算复杂度，导致难以在合理的时间内找到最优解。

对于规模较大的 JSS 和 FJSS 问题，元启发式方法成为了备受关注的选择，如遗传算法^[24]、禁忌搜索算法^[25]、粒子群优化算法^[26]、模拟退火算法^[27]等。这些启发式方法能在相对较短的时间内寻找到问题的近似最优解，为实际生产中的决策提供了可行的解决方案。Wu 等人^[28]考虑了设备开关机数量和转速水平两种节能措施，提出了一种基于非支配排序的遗传算法来优化最大完工时间和总能耗。Liu 等人^[29]同时考虑能源效率和完工时间的多目标 FJSS 问题，提出了一种基于非支配排序和混合整数数学模型的改进遗传算法，并将所提出方法与一些特定的禁忌策略相结合，以增强算法的局部搜索能力。

对于 DFJSS 问题，传统优化方法通常以重调度机制来解决车间动态调度问题。尽管重调度方法能够灵活应对动态性，但在实际应用中，计算每次重调度所需的计算量和资源较大，尤其是在问题规模较大或生产环境复杂的情况下。这不仅增加了计算的复杂性，还可能导致系统性能和稳定性下降^[30]。Tao 等人^[31]提出了一种改进的混合多阶段量子粒子群优化算法，并为一个有新工件插入、设备故障的动态柔性作业车间定期进行重调度。他们发现，调度效率和稳定性可能会受到不同的重调度周期影响。Wen 等人^[32]研究了设备故障下动态集成工艺规划与

调度问题的动态调度方法，设计了遗传算法与邻域搜索算法相结合的混合算法，并提出了一种自适应分层迁移策略来提高种群的多样性和收敛性，提高了重调度的稳定性。

（2）基于调度规则的优化方法

调度规则因其可扩展性、可重用性以及动态事件的快速反应等优点，已被证明是一种有效的启发式策略^[33,34]。简单地说，调度规则的本质是由包含车间加工信息的特征和数学运算所组成的优先级函数。在每个路径和排序决策点，调度规则计算每个调度对象（例如候选加工设备和待加工工序）的优先级值，并选择最优先的调度对象进行下一步的处理和动作。近几十年来，为了应对各种作业车间场景和优化目标，学者和工人们提出了许多不同类型的特征（例如工序加工时间、设备待机时间等）和调度规则（例如先进先出规则等）^[35]。

考虑到 DFJSS 问题涉及柔性车间中多种生产要素的复杂交互耦合，且不同调度决策的决策环境和信息需求也不同，单纯依靠工人经验来理清各特征之间的潜在关联并设计出合适的调度规则是一项非常困难的任务，甚至可能是不可行的。为解决这一挑战，Burke 等人^[36]首次提出了超启发式遗传规划算法(Genetic Programming Hyper Heuristic, GPHH)的框架。通过采用 GPHH，他们能够自动生成调度规则以解决车间调度问题。他们的研究指出，GPHH 生成的解不是启发式解本身，而是一个启发式程序或启发式规则。该方法的创新性在于通过遗传规划结合超启发式方法，从而克服了直接设计调度规则的复杂性和困难性，这为后续更深入研究车间调度问题以及其他组合优化问题提供了新的方向。

GPHH 可以通过预先设定的参数（例如树的最大深度、函数集和、特征集合等），从而在进化迭代过程中通过不断调整调度规则中特征和函数运算符的组合方式来自动设计有效的调度规则，而无需依赖大量的特定领域知识^[37]。与大部分求解车间调度的元启发式算法相比，GPHH 具有灵活的编码表示方式、强大的搜索能力以及更易应用于实际环境等优点，其生成的调度规则天然适用于求解大规模动态柔性车间调度问题。在求解车间调度问题时，GPHH 通常用一组包含工件、设备和车间状态的特征（对应叶节点）和一组数学运算函数（对应非叶节点）以树的形式产生规则个体^[38]。

近年来，运用超启发式遗传规划自动化设计和生成合适的调度规则成为了有

效解决动态车间调度问题的公认方法。已有很多研究证明使用 GPHH 能够自动演化出比人工设计更为优秀的调度规则。Park 等人^[39]提出了用集成式 GPHH 算法处理动态车间调度问题，并对四种常见的组合方案进行研究和分析，结果表明 GPHH 线性组合方法性能更优。Zhang 等人^[37]提出了一种新的重组机制，引导父代个体交叉进化，从而得到性能和适应性更好的规则，实验从所选子树的深度比、所选子树的相关性以及进化过程中的概率差异等方面验证了其有效性。Zhou 等人^[40]提出基于双染色体协同进化策略的遗传规划方法来求解 DFJSS 问题，实验通过代理仿真模型辅助终端符的选择和调度规则的进化，验证了其方法的有效性和鲁棒性。

(3) 基于 GPHH 的特征筛选方法

文献[41]中已说明，树的最大深度、函数集、特征集是决定调度规则组合形式多样性和 GPHH 搜索空间的三个主要因素。若“调度规则树”的最大深度为 d ，使用“Full 方法”（即完全生长法）生成树，则 GPHH 搜索空间的大小为 $|\mathcal{F}|^{2^d-1} \cdot |\mathcal{T}|^{2^d}$ ，其中 $\mathcal{F}$ 和 $\mathcal{T}$ 分别表示函数集和特征集。然而，实际的解树数量远小于上述理论值，这一事实说明了通过 GPHH 寻找有效的解决方案仍然是一个艰巨的任务。

学者们通过研究发现，减少特征数量是缩小 GPHH 搜索空间、增强 GPHH 搜索效率的一种有效途径^[42, 43]。GPHH 的特征集中可以包含大量从不同侧面和形式描述车间调度过程相关信息的特征，例如系统相关、设备相关和工件相关的特征等，但这些特征并非都对调度决策起到积极的作用。例如，工件的交货截止日期被认为是优化平均流程时间的无用特征。通过将特征筛选技术与 GPHH 相结合，能够删除与所求问题无关和冗余的特征，使得算法能够更有针对性地探索包含有用信息的区域，提高算法效率。

然而，目前关于 JSS 及其变体的特征筛选方法的研究还相对较少，且主要集中在单目标问题的研究上。最初的研究侧重于通过计算最优调度规则中特征的出现频率来度量特征的重要性^[44]。Mei 等人^[45]首次提出了基于特征对个体适应度值贡献度的度量方法，用于衡量特征的重要性。相比于特征频率，这种度量方法可以避免冗余特征的影响，具有更好的准确性。然而，这些方法主要在动态 JSS 问题中进行了研究，仅考虑了排序规则。Zhang 等人^[46]首次将特征筛选方法用于

DFJSS 问题中,它涉及两个特征集合,分别用于路径决策和排序决策。在他们的后续工作^[47]中,还进一步开发了一种连续的两阶段 GPHH 框架,以在特征选择过程中充分利用已生成的优秀个体。

尽管基于 GPHH 的特征筛选方法已成功应用于单目标 DFJSS 问题中,降低了特征集合的维度并增强了生成规则的可解释性,但并未实现对算法优化性能的提升。至于 DFJSS 多目标问题,目前尚未发现有效的特征筛选方法,这也是本文的一个主要研究方向和动机。

1.4 本文主要研究内容

本文围绕面向节能的动态柔性车间调度问题进行了深入研究。通过对柔性车间动态调度过程中的能量流动进行分析,设计了与能耗相关的终端符和人工规则。随后,基于 GPHH 自动化生成面向能耗的节能调度规则,以实现平均延迟时间和总能耗两个目标的最优化。在此基础上,为解决 GPHH 算法搜索空间大、计算效率低等问题,本文借鉴了已有单目标问题中的特征筛选方法思路,提出了适用于多目标问题的特征权重方法,并将其引入到 GPHH 算法中,实现基于双重特征权重集合的节能调度规则自动化设计。

第 1 章为绪论,介绍了本文的研究背景,并对国内外相关研究现状进行了详细阐述。在此基础上,清晰阐明了本文的研究思路 and 具体研究内容。

第 2 章详细介绍相关理论知识,着重阐述了面向节能的动态柔性车间多目标调度问题的数学模型和调度规则的表达形式,以及如何通过 GPHH 算法自动生成调度规则来求解动态柔性车间调度问题。

第 3 章主要研究了面向节能的终端符设计。通过对柔性车间制造加工过程的能流分析,设计了能耗相关的终端符和人工规则,为 GPHH 算法提供了车间加工过程中的各项能耗特征参数的重要信息。仿真实验证明了所提终端符能够有效实现对能耗目标的优化。

第 4 章研究了基于双重特征权重集合的多目标动态柔性车间节能调度规则自动生成方法。通过在 GPHH 算法中引入了双重特征权重集合的概念,进一步优化了算法性能。仿真实验结果验证了特征权重方法的有效性,能演化出更优秀

和可解释的调度规则。

第 5 章对本文的研究工作进行了全面总结,分析了当前工作的局限性,并指出了未来研究的方向。

论文结构图如图 1-1 所示。

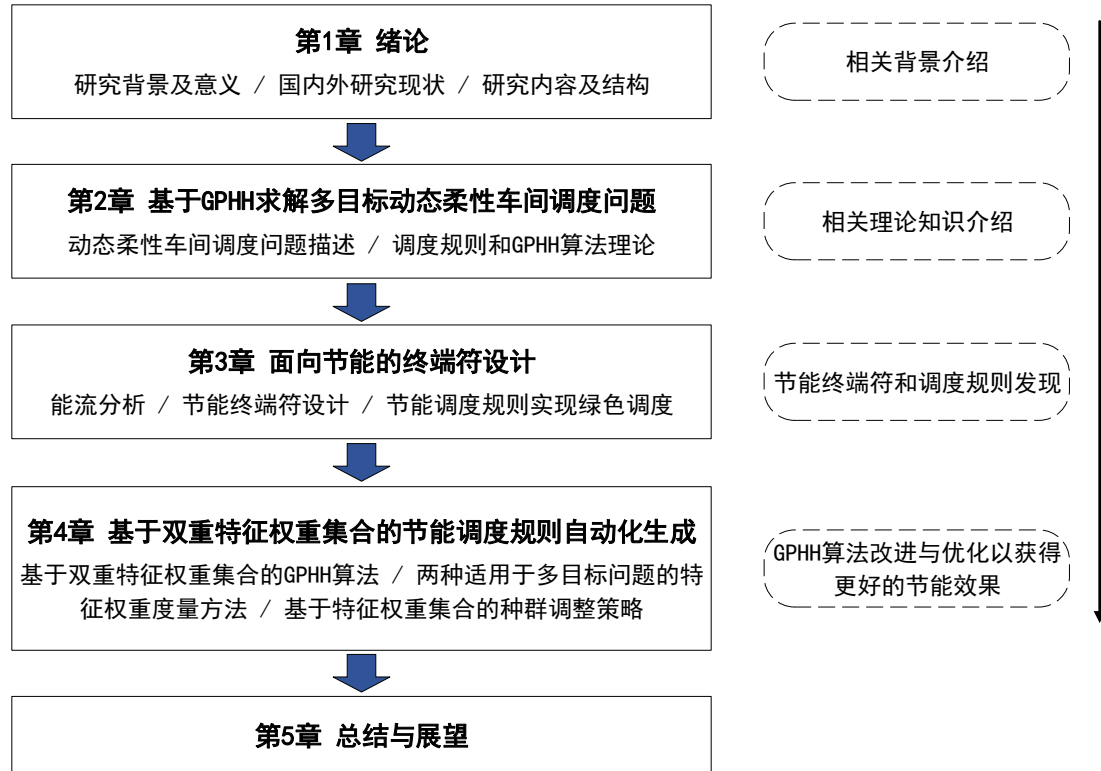


图 1-1 论文结构图

第 2 章 基于 GPHH 求解多目标动态柔性车间调度问题

2.1 引言

柔性车间调度问题属于典型的组合优化问题,其动态特性使得调度策略需要灵活适应不断变化的生产需求。学术界对此问题的关注主要体现在如何利用调度规则来提高生产效率和适应性上。本章详细介绍了动态柔性车间节能调度问题的数学模型,并强调了利用超启发式遗传规划算法生成调度规则的有效性,从而解决这一复杂挑战。这种方法的应用不仅能够有效指导车间生产,提高生产效率,还能够增强车间应对各种生产需求变化的能力,从而实现更加灵活和高效的生产调度管理。

2.2 动态柔性车间调度问题描述

本文所研究的面向节能的 DFJSS 问题的数学描述如下^[48, 49]:有 m 台设备 $\mathcal{M}=\{M_1, \dots, M_m\}$ 被用来处理工件的不同工序,每台设备 M_k 都有给定的待机功率 MP_k 。有 n 个工件 $\mathcal{J}=\{J_1, \dots, J_n\}$ 需要被加工,这些工件在柔性车间加工过程中随着时间的推移陆续到达,并在到达的同时分配相关的加工信息,包括到达时间 at_i 、交货期 dd_i 和指定的工序序列 $(P_{i1}, \dots, P_{in})$ (其中 n_i 是工件 J_i 的工序数),此为动态性条件。每道工序 P_{ij} 都有其可选择的加工设备集合 $\mathcal{M}_{ij}$, 工序可从中任选一台设备 $M_k \in \mathcal{M}_{ij}$ 加工,对应的加工时间为 PT_{ijk} , 加工能耗为 EC_{ijk} , 此为柔性条件。结合实际生产条件, DFJSS 问题还需要满足以下约束条件:

(1) 工件在到达柔性车间之前,其加工信息处于未知状态,因此在当前加工过程中不做考虑。

(2) 工件在到达柔性车间之后才能开始加工,其所有工序必须按给定的顺序进行加工,且每个工序只能在其可选设备集合中选择一台进行加工。

(3) 设备在同一时刻最多只能加工一个工件,加工过程不能被打断。设备在不加工的时刻处于空闲待机状态,以待机功率消耗能量。

本文同时优化 DFJSS 问题中的两个生产目标：平均延迟时间(Mean Tardiness, MT)和车间总能耗(Total Energy Consumption, TEC)，数学表达式如式(2-1)和(2-2)所示，其中式(2-2)来源于本文 3.2 章节的能耗分析与计算过程。平均延迟时间用于确保工件满足交货期要求，以保证订单整体上按时交付。车间总能耗反映了车间加工系统的能耗情况，主要包括设备加工能耗和设备待机能耗两类。

$$MT = \frac{1}{n} \sum_{i=1}^n \max \{c_i - dd_i, 0\} \quad (2-1)$$

$$TEC = \sum_{k=1}^m \left(c_{\max} - \sum_{i=1}^n \sum_{j=1}^{n_i} PT_{ijk} \right) MP_k + \sum_{i=1}^n \sum_{j=1}^{n_i} EC_{ijk} \quad (2-2)$$

其中 c_i 是工件 J_i 的完工时间， $c_{\max}$ 是最后一个工件的完工时间，即总加工时间。

2.3 基于调度规则的动态柔性车间调度决策

2.3.1 调度规则的表达形式

调度规则因其可扩展性、可重用性以及对于未知生产事件的快速反应等优点，被广泛应用于 DFJSS 问题的求解^[34]。简单地说，调度规则的本质是由包含车间加工信息的特征和数学运算符号所组成的优先级函数。在每个路径和排序决策点，调度规则计算每个调度对象（例如候选加工设备和待加工工序）的优先级值，并选择最优先的调度对象进行下一步的处理和动作。本文求解柔性车间调度问题，一个调度规则个体包含两个子调度规则：路径调度规则和排序调度规则，分别用于处理路径决策和排序决策。

调度规则通常以图 2-1 所示的树形结构表示^[38]。这样的树形结构有助于计算机进行编码和解码，也有助于 GPHH 进行复制、交叉、变异等遗传操作来生成新的规则树。图 2-1 展示了一个规则树及其相应的解码示例，解码时通常按照深度优先的顺序从左到右解释树。树底部的叶节点为常量或表示车间信息的终端特征符号，例如 PT 表示工件当前工序的加工时间、WIQ 表示设备的待加工队列中所有工序的总加工时间等。非叶节点由函数运算符连接，如 $\{+, -, \times, /\}$ 等。

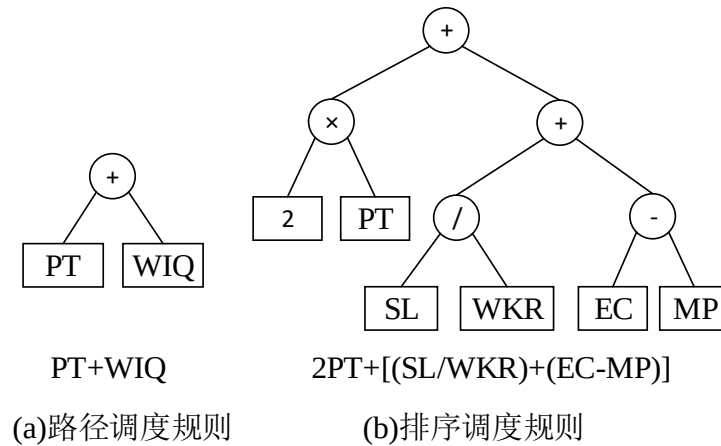


图 2-1 调度规则的表达式树示例

2.3.2 人工调度规则

人工设计调度规则是一个耗时且复杂的过程，通常需要综合考虑各种相关要素来创建这些规则，并通过实验评估它们的效果。Panwalkar 等人^[35]对调度规则进行了归纳总结，根据调度规则的结构主要可分为三类：单一规则、复合规则和加权复合规则。在本文中，使用了一些在以往研究中表现较好的规则作为本研究的基准调度规则，如表 2-1 所示。

单一规则通常只涉及一个特征参数。例如，表 2-1 中工件工序加工时间最短规则，简称为最短加工时间规则(Shortest Processing Time, SPT)，就是一个典型的单一调度规则。当一台设备处于空闲状态时，该规则会指导设备从其待加工队列中选择所需加工时间最短的工件进行优先处理。

复合规则，是由几个单一调度规则组合而成的调度规则，例如 CR+PT、LWKR+PT 等。相关研究表明，与单一规则相比，复合规则在优化调度方面更为有效^[50]。如果设计良好，组合规则可以解决具有多个优化目标的现实问题。然而，如何找到一个良好的复合规则仍是一个巨大的挑战。

加权复合规则，是复合规则的扩展，针对调度问题中的不同优化性能指标的重要性，对其相关性较高的单一规则给予更高的权重，以实现复合规则的更高效利用，例如 2PT+WIQ+NPT 等。在实践中，通过对各种规则进行调整和优化，可以进一步提高调度的效率和性能，以更好地满足实际需求。

表 2-1 基准调度规则

序号	人工规则	函数表达式	含义
1	SPT	PT	工件工序加工时间最短
2	LPT	-PT	工件工序加工时间最长
3	SL	SL	工件宽裕时间
4	FIFO	ORT	工件先进先出
5	LIFO	-ORT	工件后进先出
6	LWKR	WKR	工件剩余工序总加工时间最少
7	MWKR	-WKR	工件剩余工序总加工时间最多
8	NPT	NPT	工件下一个工序的平均加工时间
9	CR	$(DD-t)/WKR$	(到货期-当前时间)/工件剩余工作量
10	CR+PT	$(DD-t)/WKR+PT$	(到货期-当前时间)/工件剩余工作量 +当前工序加工时间
11	LWKR+PT	WKR+PT	剩余加工时间+当前工序加工时间
12	SL/WKR+PT	SL/WKR+PT	工件宽裕时间/剩余工作量+当前工序加工时间
13	2PT+WIQ+NPT	2PT+WIQ+NPT	2 倍当前工序加工时间+队列加工时间+ 下一工序加工时间
14	PT/WKR	PT/WKR	工序加工时间/剩余工作量
15	DD	DD	工件交货期最短
16	min NOR	NOR	工件剩余工序数最少
17	max NOR	-NOR	工件剩余工序数最多
18	PT+LIFO	PT-ORT	工序加工时间-工序就绪时间
19	SL/WKR	SL/WKR	宽裕时间/剩余工作量
20	$\max(t+PT, DD)$	$\max(t+PT, DD)$	比较当前时间+工序加工时间和工件交 货期
21	$\max(SL, WKR)$	$\max(SL, WKR)$	比较工件宽裕时间和工件剩余工作量

2.3.3 基于调度规则的 DFJSS 调度决策过程

图 2-2 给出了如何利用调度规则实时快速地解决 DFJSS 问题的调度决策过程示意图。在每个调度决策点，通过调度规则来计算每个调度决策对象的优先级值，并选择拥有最小值的调度决策对象作为决策输出。从图中可以看出，新工件

在到达之后，其加工信息变为已知并进入待加工工件区域。路径决策会根据路径规则的指导，为工件的当前加工工序从对应的可选设备集合中指定一台设备，并在该设备处于加工状态时将工件放入设备的待加工队列中。排序决策则会在设备刚刚完成一道工序时，根据排序规则从设备待加工队列中指定下一个要加工的工件或工序。例如，图中 1 号工件的第 2 道工序 P_{12} 经过路径决策和排序决策在设备 M_3 上加工完成后，其下一道工序 P_{13} 还将被指派到设备 M_2 的待加工队列中等待下一步加工，如此循环往复，直至所有工序全部加工完成，输出完工工件。路径决策和排序决策协同运作，从而产生一个满足约束条件的调度方案，并使生产目标得到优化。

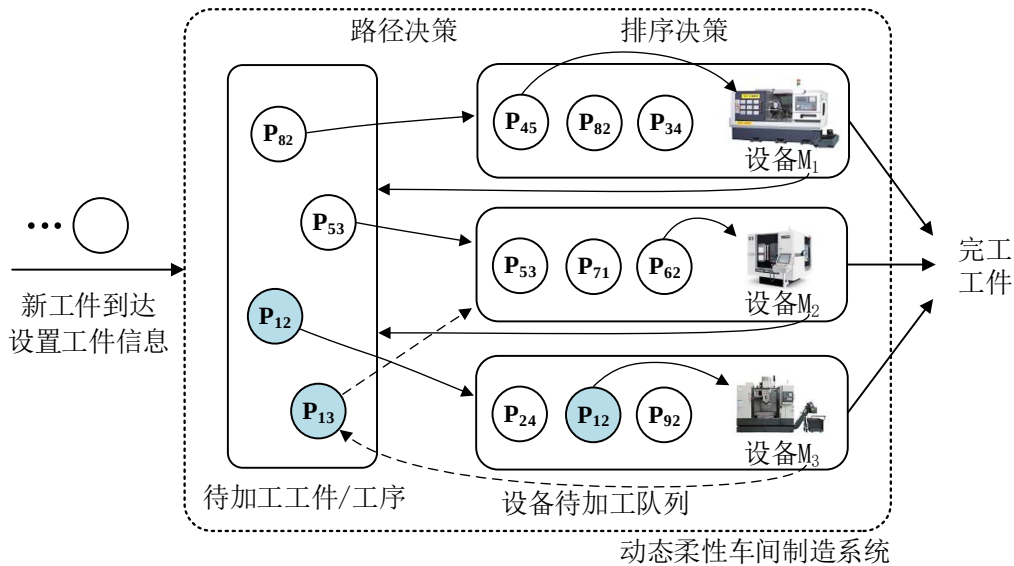


图 2-2 DFJSS 调度决策过程

2.4 超启发式遗传规划算法

GPHH 算法广泛应用于各类优化问题之中，特别是以车间调度为代表的组合式优化问题^[51]。作为一种超启发式方法，GPHH 并不直接寻找问题的具体解，而是通过迭代生成一组启发式规则，来指导解的生成。在车间调度问题中，GPHH 生成的则是调度规则（用于进行调度决策以生成解决问题实例的具体调度方案），而不是调度方案本身。

2.4.1 GPHH 算法的基本流程

GPHH 算法的基本流程图如图 2-3 所示，其基本思想与遗传算法相似，主要有种群适应度评估和种群演变两个部分。首先，算法随机生成初代种群，评估模块将种群中的每个个体（即调度规则）应用于调度问题中，并以此形成对应的调度方案，从而计算每个个体的适应度值。在种群演变阶段保留其中的精英个体，再经过遗传操作产生新的子代种群^[47]。在种群的进化过程中，每一次进化都会使种群整体更优，直到算法收敛或找到理论最优解。

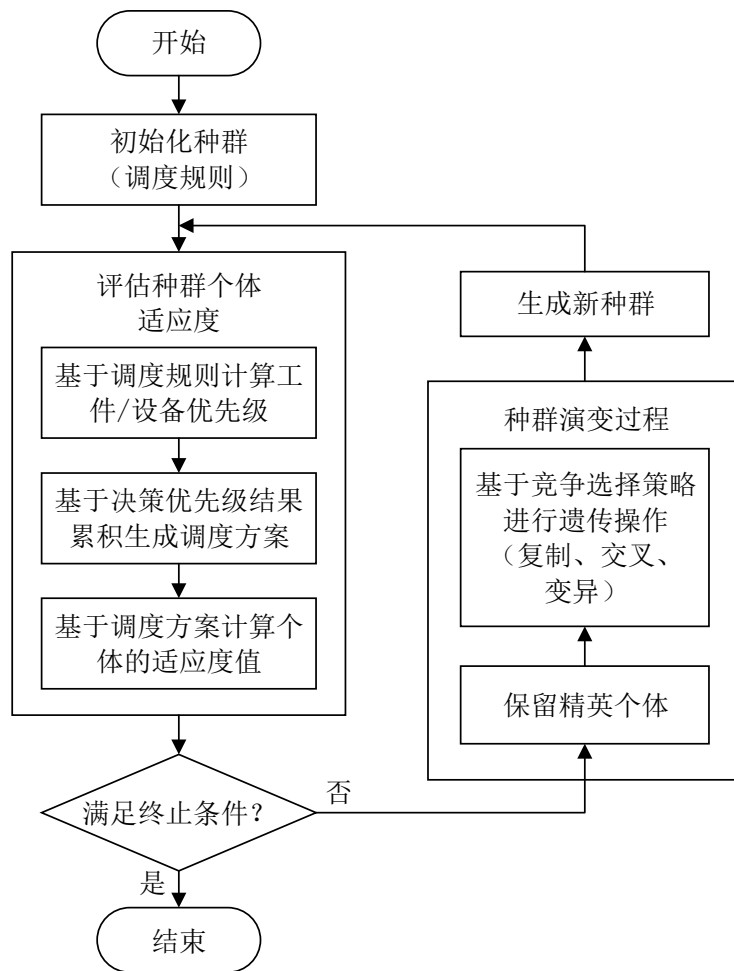


图 2-3 GPHH 求解调度问题基本流程

由于本文求解的是多目标优化问题，故采用 Pareto 非支配解的思想比较不同个体的优劣^[52]。GPHH 求解多目标 DFJSS 问题的伪代码如算法 2.1 所示，其输出为最优非支配解集。

算法 2.1: GPHH 求解多目标 DFJSS 问题的总体框架

输入: 设备使用水平 u , 交货期宽裕度系数 α , 初始随机种子 $seed$, 优化目标 $O(object1, object2)$, 种群大小 $popsiz$, 迭代次数 $genmax$

输出: 最终的 Pareto 前沿 (即最优非支配解集)

```

1   $generation \leftarrow 0$ 
2  随机初始化种群  $\mathcal{P} \leftarrow \{r_1, r_2, \dots, r_{popsiz}\}$ 
3  while 未满足终止条件 do
4      //种群评估
5      根据随机种子  $seed$  产生当前一代的仿真实例  $S_{gen}$ 
6      for  $i = 1 : popsiz$  do
7          将个体  $r_i$  应用到  $S_{gen}$  中, 产生对应的调度方案
8          根据适应度函数计算调度方案对应的适应度值
9      end for
10     根据个体适应度值, 计算个体的非支配等级和拥挤距离[52]
11     //种群演变
12     保留一定数量的精英个体
13     repeat
14         随机选择一定数量的个体, 根据非支配等级和拥挤距离竞争出优秀个体作为
            父代, 通过遗传操作 (复制、交叉、变异) 产生子代个体
15     until 子代种群大小到最大值  $popsiz$ 
16      $seed \leftarrow seed + 10000$ 
17      $generation \leftarrow generation + 1$ 
18 end while
19 return 最终的 Pareto 前沿

```

2.4.2 GPHH 的初始化方法

在 GPHH 第一代种群随机初始化过程中, 调度规则的生成主要有以下三种方法^[53]:

(1) Full 方法, 即完全生长法, 是指在达到最大深度前, 节点都从函数符集中随机选择, 在最大深度时, 节点都从终端符集中随机选择。此方法生成的树具有相同的形状, 所有叶节点都在同一深度, 也就是在最大深度。

(2) Grow 方法, 即成长生长法, 是指从函数符集中随机选择函数作为根节点, 在最大深度时, 节点只能在终端符集中选择; 未达到最大深度时, 节点可在终端符集和函数符集中随机选择, 若分支选择了终端符, 则停止生长。与 Full 方法相比, Grow 方法生成的树有着更为多样化的大小和形状。

(3) Ramped half-and-half 方法, 是指指定一个深度范围, 令种群中一半的

个体使用 Full 方法生成，一半的个体使用 Grow 方法生成。此方法摆脱了单一深度的限制，将 Full 和 Grow 方法混合使用，所形成的树更具多样性，且依旧简单容易实现。

2.4.3 GPHH 的遗传操作算子

在 GPHH 算法迭代过程中，新种群的生成主要通过基于竞争选择策略的遗传操作完成。在进行选择操作时，因种群规模较大，通常采用锦标赛选择法。该方法从父代种群中随机选取一定数量的个体，比较它们的适应度值，适应度值最低者被选为父代个体，用于进行下一步的遗传操作。由于每次选择都是随机选取一定数量的个体进行比较，即使是适应度值相对较低的个体也有机会被选中作为父代个体。结合遗传操作的随机性，保证了种群的多样性。遗传操作算子对父代调度规则个体进行遗传操作，主要包括复制、交叉和变异三种^[54]。

复制操作是指个体直接被复制到下一代，并保留全部的特征信息。

两个父代个体的交叉操作如图 2-4 所示，在每一个父代个体中，随机选择一个交叉节点，互相替换以交叉节点为根节点的子树，生成两个新的子代个体。

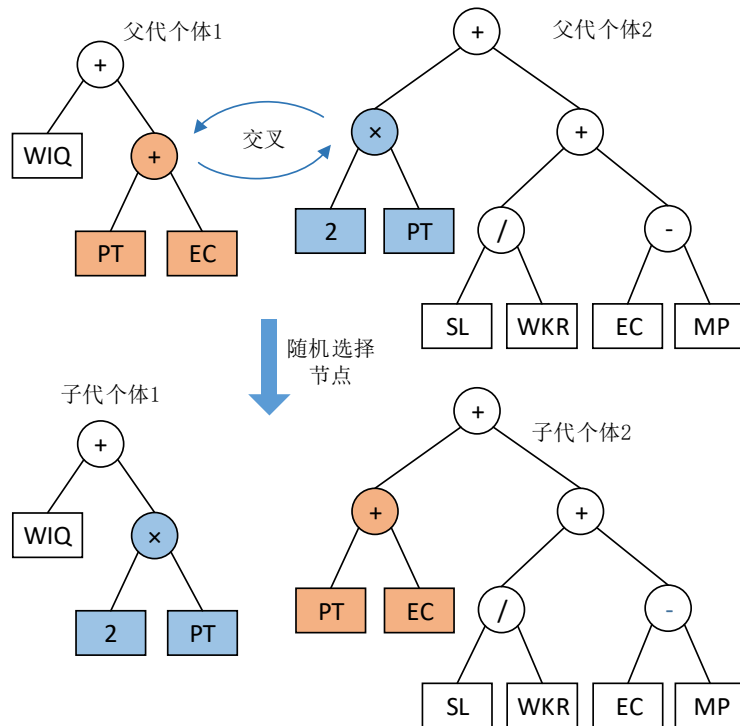


图 2-4 交叉操作

变异操作如图 2-5 所示，在父代个体中随机选择一个子节点作为突变节点，随机生成一个子树替换父代个体中以突变节点为根节点的子树，生成一个新的子代个体。

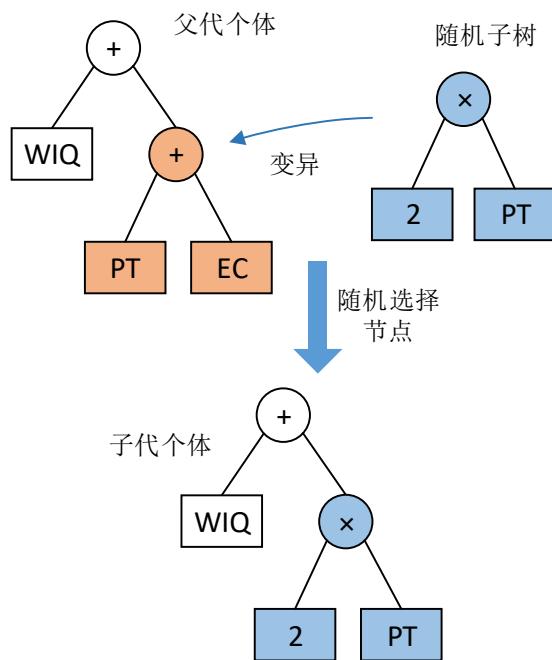


图 2-5 变异操作

2.5 本章小结

本章阐述了多目标动态柔性车间调度问题及其在面向节能方面的数学模型。通过对调度规则和超启发式遗传规划算法的研究，本文旨在寻找高质量的非支配 Pareto 解集。在后续章节中，将进一步讨论节能调度规则的设计和 GPHH 算法的实施，以解决多目标动态柔性车间调度问题。

第3章 面向节能的终端符设计

3.1 引言

在 GPHH 算法中，终端符为调度规则的构建和决策提供了必要的实时车间信息，是调度规则的核心所在。尽管 DFJSS 的终端符集已得到较充分的研究，但有关能耗方面的研究却相对较少。为了实现面向节能的调度规则自动化生成这一目标，本章的重点是通过对柔性车间加工过程中的能量流动和能耗分析，针对节能降耗的具体需求，挖掘车间中与节能降耗密切相关的车间信息，并结合终端符设计理念，设计面向节能的终端符，最终为 GPHH 自动化生成节能调度规则提供有效的能耗信息。此外，本章节还基于已有的人工规则的构建思路，给出了面向能耗的人工规则，可作为后续相关研究的测试基准。

3.2 柔性车间制造系统的能流分析

柔性制造系统是我国工业的重要组成部分，正随着科学技术的不断发展朝着智能制造和知识自动化的方向迈进。以飞机、船舶、汽车、武器装备、制造装备等包含多个零部件生产和组装的柔性制造过程为例，复杂柔性制造系统涉及以物料流、信息流和能量流为主的多类型子系统动态变化和交互耦合，其运作流程如图 3-1 所示。

本文着重对柔性制造系统中的能量流动进行深入分析，以揭示系统中的主要能耗来源及其能耗方式。如图 3-1 所示，柔性车间制造系统主要以物料和能源作为输入，在信息资源的辅助下，通过一系列设备加工技术（如机床车、铣削等）和辅助加工系统的辅助加工（如照明、运输等），完成对产品零部件的制造和组装，最终产出机械产品和零部件作为输出。在这个过程中，系统消耗了大量以电能为主的能源，并且还会产生各种废料和碳排放等加工副产物。

从图 3-1 可以看出，柔性制造系统的能耗主体主要可分为三类，具体能耗构成如表 3-1 所示。

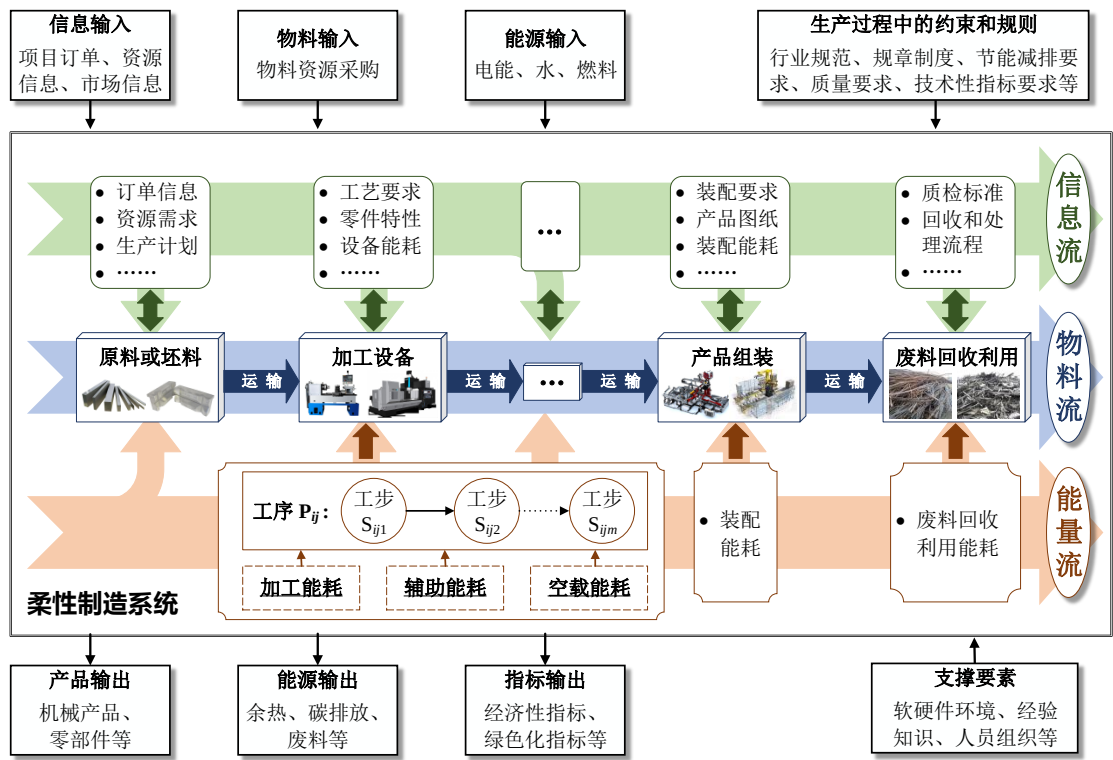


图 3-1 柔性制造系统运作流程示意图

表 3-1 制造车间能耗组成

能耗分类	耗能主体
设备能耗	工件加工能耗、设备待机能耗、辅助能耗（换刀、吹屑等）
公共能耗	照明能耗、LED 看板能耗、风扇能耗、电脑能耗等
运输能耗	运输小车能耗、起重机能耗、传送带能耗等

（1）设备能耗主要用于设备的日常运行和实际生产加工，包含待机能耗、工件加工能耗、换刀能耗和吹屑能耗等。其中，待机和加工是设备主要的耗能方式，而换刀和吹屑则属于辅助加工系统的耗能，其耗能较少且其能耗优化涉及到工件具体的工艺层次，从调度角度难以优化。因此，本文主要研究设备的加工能耗和待机能耗。

设备加工能耗，由式(3-1)计算得到，为设备完成工件/工序加工所需的必要能耗，从节能降耗的角度考虑，在路径决策阶段，工件应尽量选择加工能耗较低的设备进行加工。

$$EC_{process} = \sum_{i=1}^n \sum_{j=1}^{n_i} EC_{ijk} \quad (3-1)$$

设备待机能耗，由式(3-2)计算得到，为设备加工完一个加工工序后以待机状态等待下一个加工工序时所产生的能耗，属于无用能耗，应尽量减少。设备待机能耗产生原因主要有两种：(a)工件工序之间的装卸、运输和更换；(b)调度的不合理导致设备被迫等待，例如设备待加工区域为空时，设备无法获取下一个加工工件，或者当下一个设备待加工区域为满时，加工完成的工件不能及时运走。

$$EC_{stdby} = \sum_{k=1}^m MP_k \times T_k \quad (3-2)$$

其中， T_k 表示设备 M_k 的空闲等待时间。

(2) 公共能耗，由式(3-3)计算得到，为车间加工过程中辅助设备所消耗的能量，如照明、LED看板、空调、电脑等。这些辅助设备大具有固定的额定功率，耗能水平可视为与时间挂钩，优化空间较小。因此，在柔性车间制造系统中，公共能耗常常不在节能降耗的优化范围内。

$$EC_{aux} = P_{aux} \times c_{max} \quad (3-3)$$

其中， P_{aux} 表示单位时间内柔性车间的公共能耗。

(3) 运输能耗，由式(3-4)计算得到，主要为车间中运输设备运输工件所消耗的能量，包含小车运输能耗、传送带能耗等。虽然运输能耗在车间生产过程中是一个重要的能耗来源，但它通常在总能耗中所占比例较小，而且相对固定。此外，由于工件的大小、重量和运输路径的差异，准确评估单个工件的运输能耗是非常困难的。因此，本文更多的将注意力集中在更具影响力和可控性的能耗目标上，以实现更有效的生产调度优化。

$$EC_{trans} = \sum_{i=1}^n \sum_{j=1}^{n_i} EC_{trans_{ij}} \times \beta_{ij} \quad (3-4)$$

其中，如果工序 P_{ij} 需要被运输到对应加工设备上加工，则 β_{ij} 等于1，否则为0。 $EC_{trans_{ij}}$ 为此次运输能耗。

综合以上分析可以发现，柔性制造车间中的能耗种类繁多、动态关联强、耗能机理复杂。然而，在考虑各种能耗的占比和优化潜力时，从车间调度的角度来看，本文着重优化设备能耗中的加工能耗和待机能耗，并将它们的能耗之和作为整个车间的优化目标，如式(2-2)所示。

3.3 能耗相关终端符设计

针对本文主要优化的能耗类型，从工件、设备和相对性指标三方面设计了能耗相关的终端符，如表 3-2 所示。

表 3-2 能耗相关终端符集

分类	终端符	含义
面向工件的能耗	EC	工序加工能耗
	NEC	下一个工序的加工能耗
	ECR	工件剩余工序总加工能耗
面向设备的能耗	MP	设备待机功率
	EIQ	设备的待加工队列中的所有工序的加工总能耗
	EINQ	下一台设备待加工队列的总能耗
相对性能耗指标	REC	相对加工能耗 = $EC_{ijk} / \min\{EC_{ijk} k=1, \dots, \mathcal{M}_j \}$
	RMP	相对待机能耗 = $MP_k / \min\{MP_k k=1, \dots, m\}$

面向工件的能耗终端符以工件/工序为主体，着眼于调度决策角度，特别考虑了工件为了完成加工所涉及到的能耗相关量。结合时间相关终端符的设计思想，从当前和未来、局部和整体四个方面考虑进行全面考量，设计了 EC、NEC 和 ECR 三个终端符。

面向设备的能耗终端符关注于与设备相关的能耗信息，包括设备本身的待机功率和设备对应加工队列的加工能耗等情况，设计了 MP、EIQ 和 EINQ 三个终端符。

相对性能耗指标与时间相关复合性终端符一样，旨在为 GPHH 自动化生成的节能调度规则提供更加丰富和有效的能耗相关信息。因此，设计了 REC 和 RMP 两个终端符。

3.4 能耗相关人工规则设计

能耗规则的设计旨在优化能耗目标，通过考虑工艺和设备方面的能耗特征，提高调度决策的节能性能。基于 3.3 节中所提出的能耗终端符，表 3-3 给出了本文设计的面向能耗目标的人工规则，包含 5 个单一人工规则和 8 个复合人工规

则。这些能耗规则与以往时间相关人工规则相同，主要用于处理调度过程中的排序决策。对于路径决策，统一使用设备指派规则 **WIQ+PT**，该规则会将工序指派到待加工队列中所有工序加工总时间和当前工序加工时间之和最小的设备上，主要作用是平衡设备负荷。

表 3-3 能耗调度规则

序号	能耗人工规则	函数表达式	含义
1	min EC	EC	工件工序加工能耗最小
2	max EC	-EC	工件工序加工能耗最大
3	min ECR	ECR	工件剩余工序加工总能耗最少
4	max ECR	-ECR	工件剩余工序加工总能耗最多
5	NEC	NEC	工件下一个工序的平均加工能耗
6	CR+EC	$(DD-t)/WKR+EC$	(交货期-当前时间)/工件剩余工作量+当前工序加工能耗
7	min ECR+EC	ECR+EC	剩余加工能耗+当前工序加工能耗
8	SL/WKR+EC	SL/WKR+EC	宽裕时间/剩余工作量+当前工序加工能耗
9	2EC+WIQ+NPT	2EC+EIQ+NPT	2 倍当前工序加工能耗+队列加工能耗+下一工序加工能耗
10	EC/ECR	EC/ECR	工序加工能耗/剩余加工能耗
11	PT+EC	PT+EC	工序加工时间+工序加工能耗
12	NPT+NEC	NPT+NEC	工件剩余工序加工时间+剩余加工能耗
13	max (SL, ECR)	max (SL, ECR)	比较工件宽裕时间和工件剩余总加工时间

3.5 仿真实验设计

3.5.1 GPHH 算法的终端集合和函数集合

表 3-4 给出了以往研究中常用到的时间相关终端符和其他一些终端符^[55]，结合表 3-2 中提出的能耗相关的终端符，构建出了一个相对完整的车间特征属性集合，用于支撑本章的实验验证工作。

算法函数符集为 $\{+, -, \times, /, \max, \min\}$ ^[47, 56]。前四个是传统的算术运算符，其中除法运算具有除数保护功能，如果分母为零，结果将返回 1。另外， $\max$ 和 $\min$ 是接受两个参数并返回它们中的较大值或较小值的函数。

表 3-4 GPHH 算法所使用的终端符集

分类	终端符	含义
时间相关 终端符	PT	工序加工时间
	SL	工件宽裕时间
	OWT	工件等待时间
	NPT	下一个工序的加工时间
	WKR	工件剩余工序总加工时间
	TIS	工件在制造系统中的时长
	WIQ	设备的待加工队列中的所有工序的加工总时间
	MWT	设备等待时间
	DD	工件交货期
	MRT	设备就绪时间
时间相对性 终端符	ORT	工件工序就绪时间
	WINQ	下一台设备待加工队列的工作量
	RPT	相对加工时间 = $PT_{ijk} / \min\{PT_{ijk} k=1, \dots, \mathcal{M}_j \}$
	NOR	工件剩余工序数
	NOS	工件工序加工候选设备数
	NIQ	设备的待加工队列中的工序数
	NINQ	下一台设备待加工队列的工序数
	NOPS	工件工序数
	CT	当前时间
	AT	工件到达时间

3.5.2 实验对比算法

为了验证所提能耗相关终端符的有效性，本文在统一的 GPHH 优化框架下将能耗相关终端符和已有的时间相关终端符相比较，并根据不同终端符集合设计了 7 种不同的比较算法。在实验过程中，所有算法都在模拟车间复杂环境的不同情境下进行了训练和测试。7 种算法的设置如下：

(1) 基准调度规则，因人工规则结构简单，常作为相关研究的测试基准。而在节能减排方面，本文将表 3-3 中设计的能耗相关人工调度规则也考虑其中，即表 3-3 和表 2-1 的调度规则一起组成了基准调度规则的测试集。

(2) GPHH-TIME, 该算法仅采用时间相关的终端符作为 GPHH 的输入, 并以此产生调度规则进行调度决策。

(3) GPHH-ENG, 该算法仅采用能耗相关的终端符作为 GPHH 的输入, 并以此产生调度规则进行调度决策。

(4) GPHH-TIME+R, 该算法仅采用时间相关的终端符和时间相对性终端符 (RPT) 作为 GPHH 的输入, 并以此产生调度规则进行调度决策。

(5) GPHH-ENG+R, 该算法仅采用能耗相关的终端符和能耗相对性终端符 (RMC, RMP) 作为 GPHH 的输入, 并以此产生调度规则进行调度决策。

(6) GPHH-TIME+ENG, 该算法仅采用时间和能耗相关的终端符作为 GPHH 的输入, 并以此产生调度规则进行调度决策。

(7) GPHH-TIME+ENG+R, 该算法采用时间和能耗相关的的终端符以及相对性终端符 (RPT, RMC, RMP) 作为 GPHH 的输入, 并以此产生调度规则进行调度决策。

3.5.3 仿真模型构建

在本文实验中, 所提出的基于 GPHH 算法实现优化车间总能耗和平均延迟时间两个优化目标的柔性车间动态调度规则自动化生成的方法, 包含训练和测试两个阶段, 每个阶段都由计算机随机生成对应生产情境的问题实例, 用于模拟实际的动态生产情况。问题仿真实例的具体参数设置如表 3-5 所示, 该参数模型被广泛应用于目前已有的研究工作中^[57]。

所用仿真模型中, 设有 10 台设备, 1000 个工件, 500 个预热工件。每个工件的工序数和工序候选加工设备数都服从均匀分布 $U(1,10)$, 由于同一个工序可以在多台设备上加工, 故可以认为不同设备加工同一个工序所花费的时间和能耗是相关的, 并不会太大差距。因此, 对于工序 P_{ij} , 仿真模型先按照均匀分布 $U(1, 99)$ 随机产生一个平均加工时间 $\overline{PT}_{ij}$ 和平均加工能耗 $\overline{EC}_{ij}$, 再根据这个平均值对每一个候选设备 M_k 赋予一个服从正太分布 $PT_{ijk} \sim N(\overline{PT}_{ij}, \overline{PT}_{ij} / 10)$ 和 $EC_{ijk} \sim N(\overline{EC}_{ij}, \overline{EC}_{ij} / 10)$ 的加工时间 PT_{ijk} 和加工能耗 EC_{ijk} 。设备待机功率的设定值差异较大, 以保证运用于不同的调度方案中时会有较大的差异。

表 3-5 DFJSS 问题仿真模型参数

参数	设定
加工设备数	10
工件个数	1000
前期工件个数	500
工件工序数及其分配	$U(1, 10)$
工件到达过程	泊松分布
设备使用水平	0.85, 0.95
交货期宽裕度系数	2, 4, 6
工序平均加工时间及其分布	$U(1, 99)$
工序平均能耗及其分布	$U(1, 99)$
设备待机功率	{10, 12.5, 4.5, 3.6, 7.0, 1.5, 8.5, 2.2, 22.9, 6.4}

根据已有的研究^[58]，设备使用水平 u 和交货期宽裕度系数 α 是定义负载条件的主要因素，代表了不同作业车间的负载情况。因此，本文设置了 2 个 u 值 {0.85, 0.95} 和 3 个 α 值 {2, 4, 6}，共构成了 $2 \times 3 = 6$ 种不同的生产情境。在训练阶段，将 GPHH 算法在每种情境下基于不同的随机种子进行 30 次独立运行。为了测试生成的调度规则的有效性，将每次运行的结果进行了 100 次独立测试，然后取这 100 次测试结果的平均值作为所得调度规则的最终适应度值，以验证其实际泛用性和普适性。

新工件的到达过程可以看作连续不断发生的独立随机事件，满足泊松分布，相对应的工件到达时间间隔满足指数分布，工件的平均到达时间间隔 λ 由下式计算得到：

$$\lambda = \frac{u_j \times u_p}{u \times m} \quad (3-5)$$

其中， u_j 表示工件平均工序数， u_p 表示工序平均加工时间， u 表示车间使用水平，反应了车间里设备加工的不同紧张程度， m 表示设备数。

工件的交货期 dd_i 由下式计算得到：

$$dd_i = at_i + \alpha \times \sum_{j=1}^{n_j} \min PT_{ijk} \quad (3-6)$$

其中， α 是交货期宽裕度系数，交货期宽裕度系数越小，说明工件交货期越紧张。

3.5.4 GPHH 参数设置

表 3-6 给出了 GPHH 算法的具体参数设置。实验中，规则树的最大深度为 8，初代种群采用 Ramped half-and-half 方法随机生成，深度范围设置为 2~6，为后续的随机交叉和变异操作留有余量，保持种群的多样性。采用容量为 7 的锦标赛选择法选出个体作为父代，使用遗传算子（复制、交叉和变异）对父代进行遗传操作，生成新的子代。此外，在迭代过程中融合精英保留策略，防止种群中最优个体在遗传过程中丢失，提高算法的收敛速度^[42]。

表 3-6 GPHH 算法参数设计

参数	设定
初始化方法	Ramped half-and-half
种群大小	600
规则树最大深度	8
交叉率	80%
变异率	15%
复制率	5%
父代选择方法	容量为 7 的锦标赛选择法
精英保留个体	10
迭代次数	51

3.5.5 算法性能指标

在衡量算法所得到的 Pareto 前沿的质量时，本文采用了两个常用的多目标优化性能评价指标：反转世代距离(Inverted Generational Distance, IGD)和超体积度量(Hyper-Volume, HV)^[59]。

(1) IGD 指标

IGD 通过计算真实 Pareto 前沿上的点（个体）到算法获得的 Pareto 前沿之间的最小距离来评价算法的综合性能，主要反应了算法所得 Pareto 前沿与真实 Pareto 前沿的逼近程度。IGD 的定义式如下：

$$IGD = \frac{1}{n_{APF}} \left(\sum_{i=1}^{n_{APF}} d_i \right) \quad (3-7)$$

其中， n_{APF} 是真实的 Pareto 前沿上个体的个数， d_i 是真实的 Pareto 前沿上个体到算法所得 Pareto 前沿的最小欧几里得距离。

IGD 指标数值越小，表明算法所得 Pareto 前沿越逼近真实 Pareto 前沿并且分布更均匀，解集的最优性和多样性更好。图 3-2 给出了 IGD 指标示意图，由于本文求解的是双目标问题，图像为二维图。

对算法 IGD 指标的分析主要包括以下两步：首先分别计算出 30 组测试集数据的 IGD 指标，然后计算其平均值和标准差。计算时需要进行归一化处理，IGD 还需要有一个真实的 Pareto 前沿，实验中，将所有算法的解集混合在一起，得到一个尽可能最优的非支配解集，作为近似的真实 Pareto 前沿。

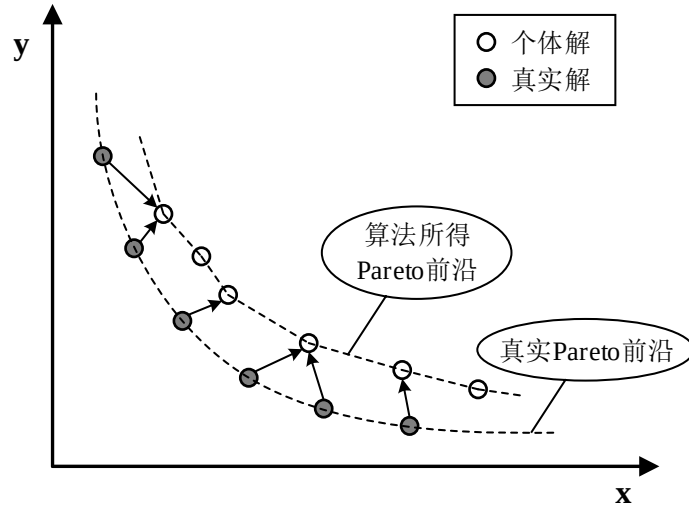


图 3-2 IGD 指标

(2) HV 指标

HV 用于测量最优化问题所得到的非支配解所覆盖的目标空间的体积，主要反应了算法所得 Pareto 前沿上个体分布的均匀程度和多目标下的优化效果。HV 的定义式如下：

$$HV = \bigcup_{i=1}^{n_{PF}} v_i \quad (3-8)$$

其中， n_{PF} 是算法所得 Pareto 前沿上个体的个数， v_i 是 Pareto 前沿上的个体与参考点之间形成的超立方体体积。

HV 指标数值越大，说明对应 Pareto 解集性能越好，图 3-3 给出了 HV 指标

二维示意图。对算法 HV 指标的分析主要包括以下两步：首先分别计算出 30 组测试集数据的 HV 指标，然后计算其平均值和标准差。计算时需要进行归一化处理，HV 还需要有一个参考点，设置为 $1 + 1/(n_{APF} - 1)^{[57]}$ ，其中 n_{APF} 是 IGD 中所得近似真实前沿中解的个数。

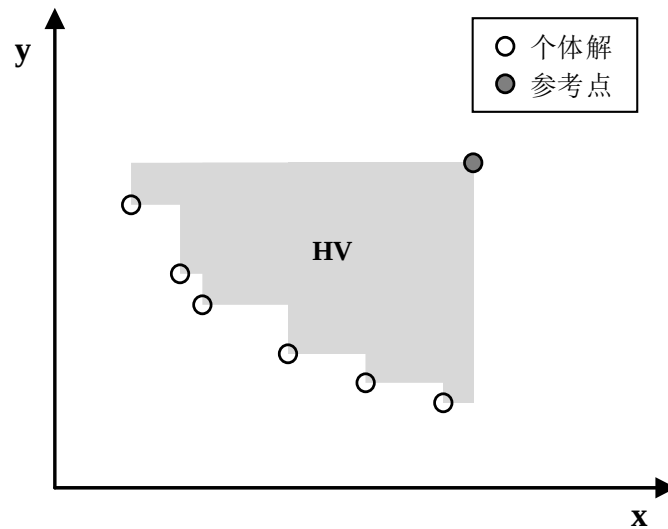


图 3-3 HV 指标

3.7 实验结果分析

为了分析上述 7 种算法在六种不同车间情境/配置下的性能，基于 Java 语言开发了相应的仿真调度系统，在 Eclipse IDE 开发环境下进行了仿真运行，并对结果进行了分析。

3.7.1 算法总体性能分析

实验中，将 GPHH 算法应用到 6 种生产情境上，共独立运行 6×30 次，自动化生成适用于对应生产情境的调度规则，并将每次训练得到的最终 Pareto 解集在测试集合（包含 100 次独立运行）上验证其性能。

表 3-7 和表 3-8 分别给出了 7 种算法在 6 种生产情境下得到 HV 和 IGD 指标，以此对算法进行整体性能分析。为了进一步验证所提能耗和相对性终端符在柔性车间节能调度方面的有效性，基于置信水平 5% 的 Wilcoxon 秩和检验对 7 种

算法进行显著性分析^[60]。本文主要做了两组秩和检验实验，分别为实验一（TIME+ENG 和 TIME+ENG+R 分别与其他 5 种算法的实验结果进行秩和检验）和实验二（TIME+ENG 与 TIME+ENG+R 的实验结果进行秩和检验）。如果 TIME+ENG 和 TIME+ENG+R 显著好于其他 5 种算法，则将结果在表 3-7 和表 3-8 中加粗表示；如果 TIME+ENG+R 显著好于 TIME+ENG，则将结果在表 3-7 和表 3-8 中用“●”标出。

表 3-7 7 种算法在不同生产情境下的 HV 指标平均值(标准差)

情境	人工规则	TIME	ENG	TIME+R	ENG+R	TIME+ENG	TIME+ENG+R
0.85-2	0.04(0)	0.72(0.05)	0.90(0.01)	0.73(0.02)	0.91(0.01)	0.95(0.03)	0.95(0.03)
0.85-4	0.11(0)	0.78(0.03)	0.95(0.01)	0.79(0.01)	0.96(0.01)	1.00(0.02)	1.01(0.02) ●
0.85-6	0.31(0)	0.87(0.01)	1.00(0.01)	0.86(0.01)	1.00(0.01)	1.04(0.02)	1.05(0.02) ●
0.95-2	0.22(0)	0.86(0.04)	0.79(0.01)	0.87(0.04)	0.79(0.01)	0.91(0.03)	0.92(0.03) ●
0.95-4	0.41(0)	0.96(0.02)	0.95(0.01)	0.97(0.01)	0.96(0.01)	1.00(0.02)	1.02(0.02) ●
0.95-6	0.57(0)	0.99(0.04)	0.99(0.01)	1.00(0.02)	0.99(0.01)	1.02(0.04)	1.04(0.03) ●

表 3-8 7 种算法在不同生产情境下的 IGD 指标平均值(标准差)

情境	人工规则	TIME	ENG	TIME+R	ENG+R	TIME+ENG	TIME+ENG+R
0.85-2	0.71(0)	0.16(0.05)	0.07(0.01)	0.14(0.02)	0.07(0.01)	0.05(0.02)	0.04(0.02)
0.85-4	0.73(0)	0.17(0.03)	0.07(0.02)	0.17(0.02)	0.07(0.03)	0.06(0.03)	0.06(0.03)
0.85-6	0.65(0)	0.14(0.01)	0.05(0.01)	0.14(0.01)	0.06(0.01)	0.04(0.01)	0.04(0.01)
0.95-2	0.58(0)	0.09(0.02)	0.15(0.01)	0.09(0.02)	0.15(0.01)	0.06(0.01)	0.05(0.02) ●
0.95-4	0.55(0)	0.07(0.01)	0.07(0.01)	0.07(0.01)	0.07(0.01)	0.05(0.01)	0.04(0.01) ●
0.95-6	0.37(0)	0.11(0.04)	0.11(0.02)	0.10(0.03)	0.10(0.02)	0.11(0.04)	0.10(0.04)

整体上看，无论使用哪种终端符集合，GPHH 算法得到的结果都远比人工规则好。这是因为 GPHH 考虑了更多的车间特征信息，并且在算法迭代过程中剔除了适应性不好的调度规则，最终生成的解集也是经过融合 NSGA-II 算法的非支配等级和拥挤距离选择而来，这些都使得 GPHH 算法生成的调度规则拥有更加丰富灵活的结构，从而可以从容面对车间的动态环境，证明了 GPHH 算法在进化调度规则方面的有效性。

对比 TIME 与 ENG 和 TIME+R 与 ENG+R 两组数据，包含能耗终端符的算

法优化效果更好些。虽然 ENG 中没有时间相关终端符，但终止条件已经把大部分在时间上调度效果差的规则剔除了，因此 ENG 也是有考虑时间问题的，对时间目标的优化结果并不会太差。而 TIME 中既没有能耗相关终端符，也没有设置如能耗阈值等的终止条件，无法对能耗目标进行优化，所以从双目标的角度看 ENG 和 ENG+R 的效果更好。然而，ENG 和 ENG+R 并不是在所有情境下都优于 TIME，当设备使用水平达到 0.95 时，特别是在<0.95-2>的生产情境下，工件交货期较为紧张，算法更侧重于对时间的优化，并且，较高的设备使用率水平也使得待机能耗的优化空间减少，无形中降低了能耗的优化难度。

对于 HV 指标而言，TIME+ENG 和 TIME+ENG+R 在所有情境中的表现都优于其他算法，TIME+ENG+R 在 5 种情境下表现优于 TIME+ENG。对于 IGD 指标而言，TIME+ENG 和 TIME+ENG+R 都是在除<0.96-6>以外所有情境中的表现优于其他算法，TIME+ENG+R 在 2 种情境下表现优于 TIME+ENG。这是因为在优化柔性车间的平均延迟时间、总能耗这两个优化目标时，这两个算法拥有更健全的终端符集，对于动态柔性车间的加工信息参考的更多。

此外，在 TIME+ENG+R 中的 3 个时间和能耗相关的相对性终端符也起到了重要作用。这也表明相对性终端符可以提高算法性能，它提供的加工信息是其他终端符所没有的，而且对优化确实有效果，但相对性终端符占比小，且与已有的终端符有一定的相关性，所以对算法的性能提升有限。从 TIME 与 TIME+R 和 ENG 与 ENG+R 这两组实验数据中也能得出相似结论。

在<0.95-6>生产情境下，由于设备使用水平较高，导致设备的待机能耗大幅降低，并且设备的加工能耗本身差距也较小，所以算法对能耗目标的优化有限。又因为宽裕度系数取值为 6，工件交货期较为宽裕，算法所得调度方案几乎都能在交货期内完成加工目标，并且真实 Pareto 前沿也是由 6 种算法的 Pareto 解混合后近似取得，所以从 IGD 指标来看 6 种 GPHH 算法的优化效果并无明显区别。

图 3-4 给出了 7 种算法在 6 种生产情境下得到的 Pareto 前沿，图形结果基本符合上文中 HV 和 IGD 的指标分析。可以看出，基于 GPHH 算法生成的调度规则要明显优于人工规则。当设备使用水平不变，交货期宽裕度系数增加时，算法所得调度规则的平均延迟时间明显降低，而能耗无明显变化。这是由于交货期宽裕度系数与工件的交货期紧张程度有关，与能耗无关。当交货期宽裕度系数不变，

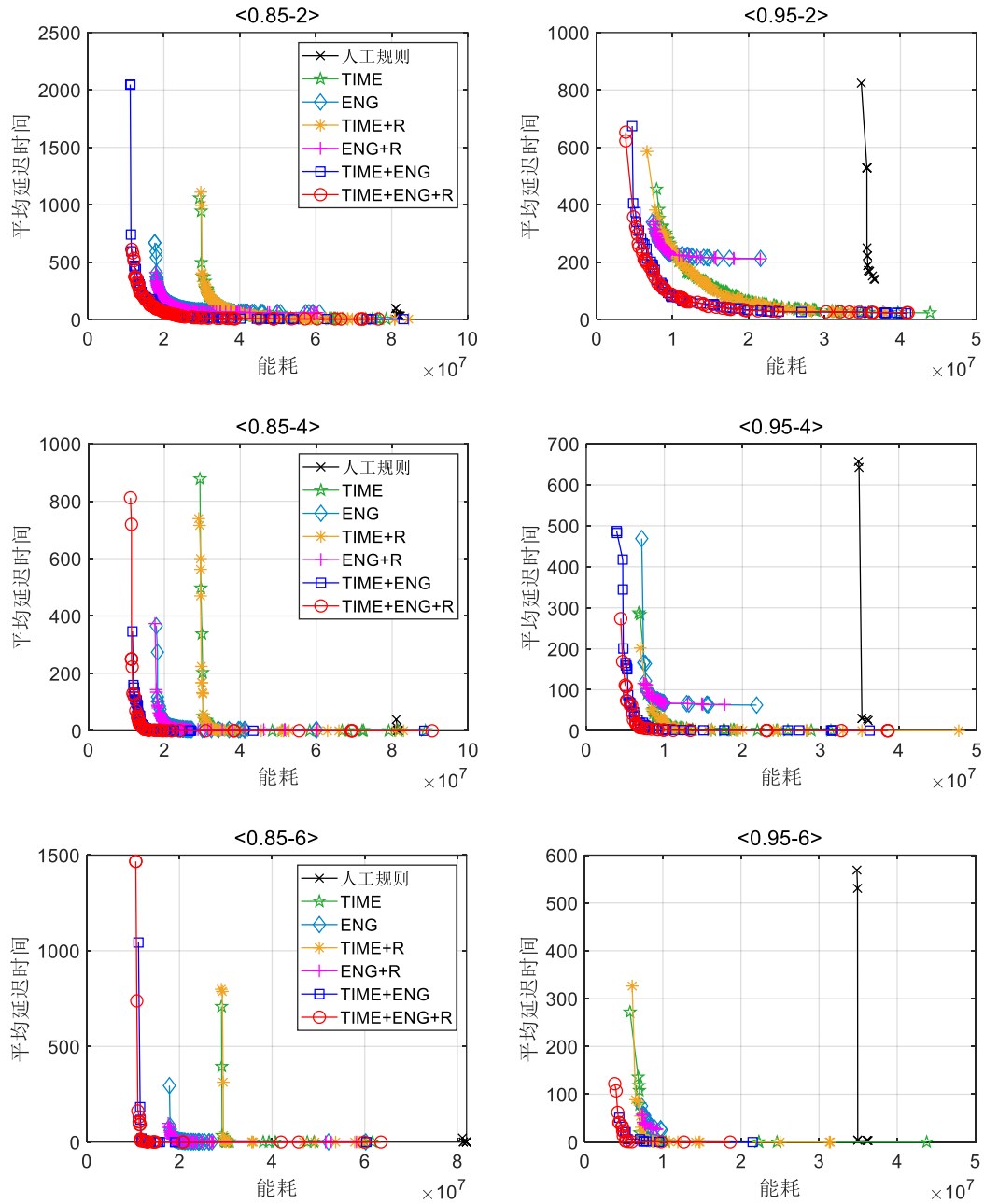


图 3-4 7 种算法在不同生产情境下的非支配解集

设备使用水平增加时，算法所得调度规则的能耗就会随之降低，这是由于设备使用水平越高，设备就越忙碌，其待机能耗就降低了。

此外，同样可以得出结论，在设备使用水平为 0.85，即设备不忙碌的时候，TIME 和 TIME+R 在优化能耗时效果不好。而 ENG 和 ENG+R 在设备使用水平为 0.95 时对平均延迟时间的表现也较差。对于相同的终端符集合，有和没有相对性的终端符，其所得的 Pareto 前沿都比较接近。

3.7.2 不同终端符集合对不同目标的优化效果分析

为了分析不同终端符集合在优化平均延迟时间和总能耗时的倾向性，图 3-5 以误差折线图的形式给出了 6 种 GPHH 算法在不同情境下分别对平均延迟时间和总能耗两个单目标的优化结果。统计点为算法所得 Pareto 前沿的中间点，在计算时表现为训 30 次独立运行实验得到的 30 个非支配解集的中位数的平均值，误差为中位数的标准差。

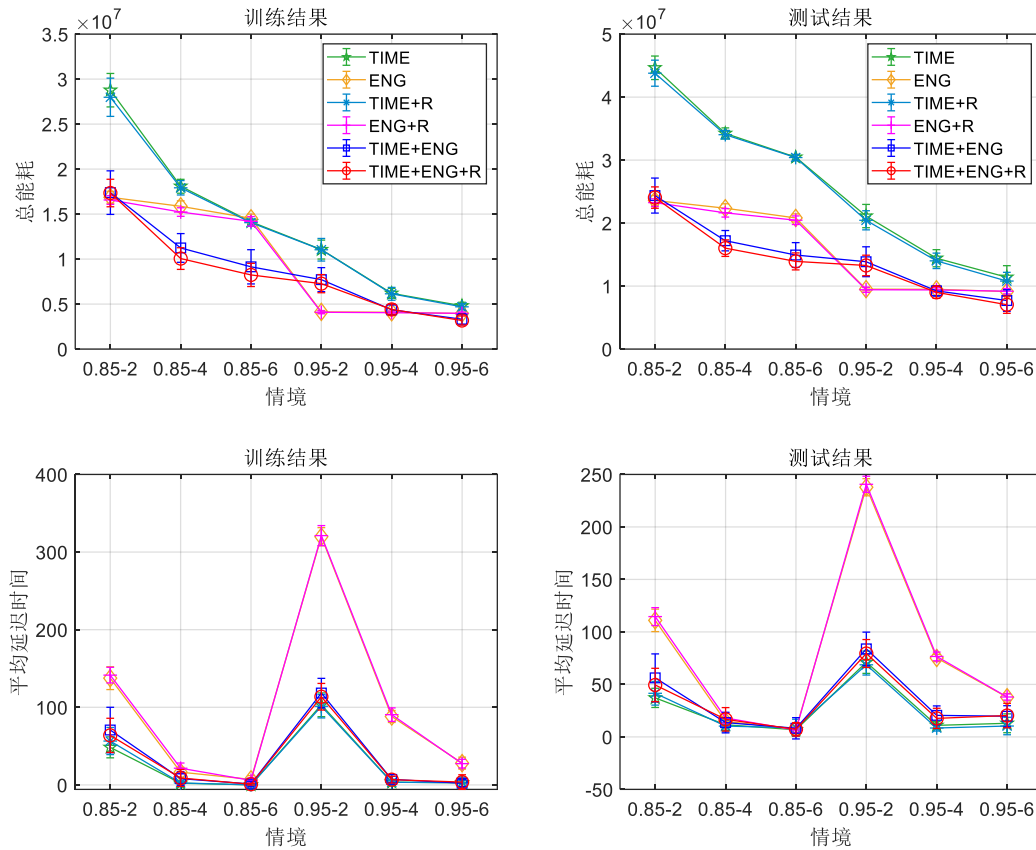


图 3-5 不同算法在不同情境下对单目标的优化结果

总体上看，无论是总能耗还是平均延迟时间，训练结果和测试结果的趋势都基本保持一致。考虑到训练结果只由最后一代的 1 个仿真实例得到的，而测试结果是 50 个仿真实例的平均值，说明仿真实例的多少对规则的优劣评估没有太大影响，也说明了算法所得规则具有较强的泛化性和普适性。

从交货期宽裕度系数 2-4-6 的变化来看，结果总体呈现下降趋势，说明工件的交货期越宽裕，由式(2-1)计算的平均延迟时间就越低，同时，算法就有更多的

空间用于优化能耗目标。

从图 3-5 中还可以看出,算法的不同终端符集合在优化调度的过程中侧重点是不同的,如 TIME 和 TIME+R 在设备使用水平为 0.85 时,因不具备能耗相关终端符,优化能耗的效果明显不好,当设备使用水平为 0.95 时,设备更加繁忙,导致待机能耗大幅降低,哪怕不针对能耗做调度优化,车间总能耗也会大幅降低,这也侧面说明了能耗优化主要优化的是设备的待机能耗。相反,TIME+ENG 和 TIME+ENG+R 拥有较为全面的终端符集合,在所有情境中对两个目标都有较好的优化结果。

对于平均延迟时间而言,ENG 和 ENG+R 也确实比 TIME 表现更差,而从宽裕度系数 2-4-6 看,ENG 和 ENG+R 与 TIME 的差距越来越小,因为宽裕系数变大,对平均延迟时间的要求就变低了,ENG 和 ENG+R 在宽裕度系数为 6 时也能同 TIME 一样完成交货期任务。

3.7.3 算法运行时间

表 3-9 给出了 6 种 GPHH 算法在不同情境下求解柔性车间动态调度问题所用的平均时间。可以看出,由于本文只针对 GPHH 所用终端符集合进行了设计,并没有更改 GPHH 算法本身的迭代流程,故各算法之间的主要差异也只是顺序结构上的不同,求得调度问题解的时间相差并不大。但也可以看出,随着终端符集合中终端符的增多,时间是在增长的。这是因为终端符集合变多,虽然提高了种群的多样性,但也扩大了算法的解空间,增大了算法的求解难度以及差个体的出现概率,所以时间也随之增加。而 ENG+R 中,仅采用能耗相关的和相对性的

表 3-9 6 种算法在不同情境下所得调度问题解的平均时间 (单位 s)

算法	0.85-2	0.85-4	0.85-6	0.95-2	0.95-4	0.95-6
TIME	6249.7	6455.4	6151.0	6646.3	6963.5	6995.2
ENG	7068.8	7119.5	7135.2	6549.8	6558.2	7087.2
TIME+R	6743.8	7037.8	6924.5	7120.8	7343.2	7376.7
ENG+R	7842.7	8416.9	8520.3	7941.2	8152.5	8356.0
TIME+ENG	6678.5	7007.4	7326.7	6573.0	6709.6	6730.2
TIME+ENG+R	7712.6	7795.3	8074.0	7487.2	7367.9	7640.1

终端符作为 GPHH 的输入，无法对工件的交货时间做专门的优化，导致较多的工件长时间滞留在车间中，增加了算法的计算量，在仿真时花费的时间最多。

3.7.4 终端符出现频率

为了理清不同终端符对于不同优化目标的作用，需要计算处于非支配解集不同区域的调度规则中各终端符出现的频率。本节以 GPHH-TIME+ENG+R 在 $\langle 0.85-2 \rangle$ 、 $\langle 0.85-6 \rangle$ 、 $\langle 0.95-2 \rangle$ 和 $\langle 0.95-6 \rangle$ 上的结果为例，计算其独立运行 30 次所得非支配解集中各终止符出现的频率。对于每一次运行的结果，本节首先对非支配解集按 MT 升序排序。之后，选择第一个，最后一个和中间三个调度规则，分别代表只考虑 MT 的解，只考虑 TEC 的解和同时考虑两个不同目标的解。对于以上三种调度规则，分别计算路径规则和排序规则中各终端符出现的频率，其结果如图 3-6 和图 3-7 所示。

从图 3-6 可知，GPHH-TIME+ENR+R 在 4 种情境下各终止符出现频率的趋势相似，PT 和 RPT、MP 和 RMP、WIQ 和 NIQ 都是路径规则中较常出现的几个终止符，可见他们对于优化 MT 和 TEC 来说是至关重要的。这与实际情况相符，为了尽可能降低 MT 并保证所有工件都能尽早完工，工件应该被安排到加工时间最短的那个设备上。为了降低 TEC，路径规则更倾向于将工件安排到拥有较大待机功率的设备，从而减少其待机时间。在路径决策过程中，也不能过于追求局部的最优解，兼顾整个加工过程，平衡设备负荷对于优化完成加工目标同样重要。MWT 也经常出现在路径规则中，这表明，在路径决策中，MWT 对于保证工件按时完成极其重要，同时对于优化 TEC，MWT 也是需要被考虑的关键因素。主要是为了减少设备的等待时间和待机能耗，工件经常被安排到处于待机状态的设备上加工。

从排序规则的角度看，PT、SL、WKR、和 NOR 是在所有解中频繁出现的终止符。可以看出，这 4 个终止符都与加工时间直接相关，对于 MT 的优化起到至关重要的作用。图 3-7 中偏向于 REC 的实验结果也表明这 4 个终止符对于优化 TEC 同样重要。通过在排序决策中考虑这 4 种信息，排序规则可以保证工件尽可能快的被加工，从而提高了设备的使用时间，降低了它们的待机时间，进而降低 TEC。在交货期宽裕度系数为 2 时，排序规则更倾向于选择 PT 和 WKR，而在交

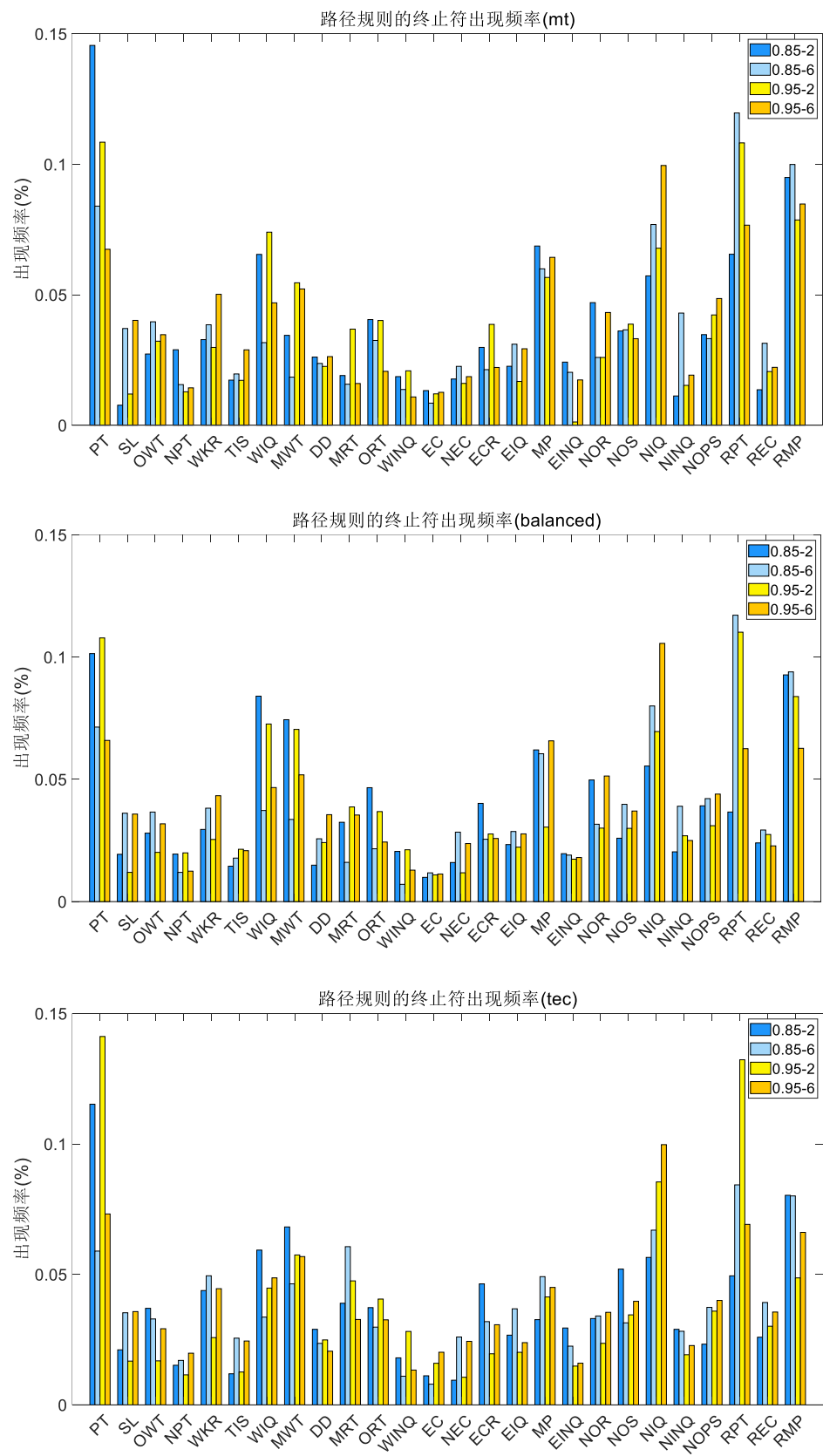


图 3-6 GPHH-TIME+ENG+R 中考虑不同目标倾向的解的路径规则平均终止符出现频率

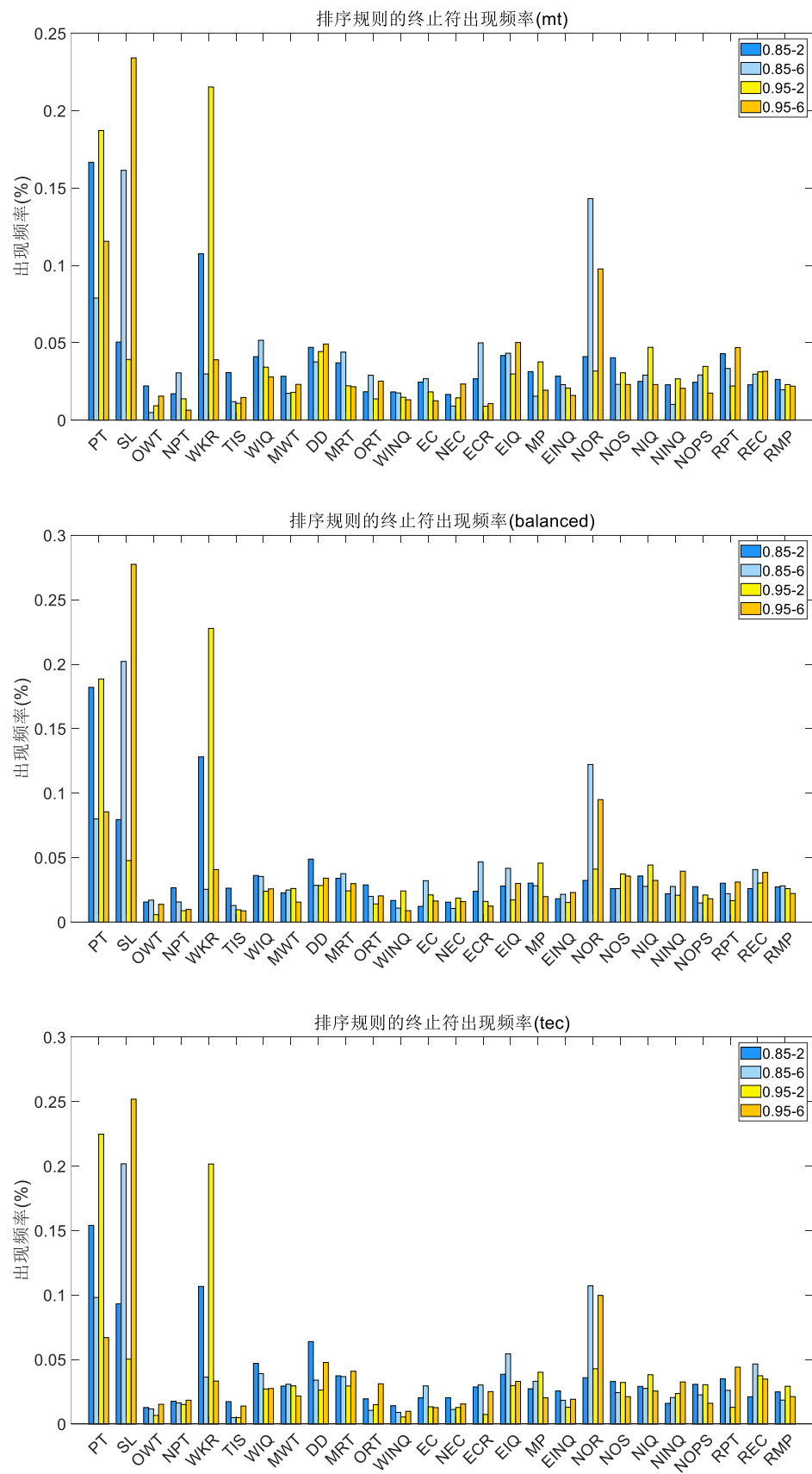


图 3-7 GPHH-TIME+ENG+R 中考虑不同目标倾向的解的排序规则平均终止符出现频率

货期宽裕度系数为 6 时，排序规则更倾向于选择 SL 和 NOR。

除 MP 在路径规则中所占比重较高以外，其它能耗相关终止符在路径和排序规则中的表现并不突出。虽然如 EC 等的能耗规则直接影响着 TEC，但都对优化 TEC 产生不了大的影响。这是因为工序在不同候选设备上的能耗差异项多只有 $\overline{EC}_{ij}/10$ 。换言之，对于相同的待机能耗，在最好的情况下只能提升 9% 的 TEC。然而，最大的待机功率是最小的 15 倍，也就是说在相同待机时间下，让待机功率小的设备待机可以有效降低 TEC，这也是 MP 要远比 EC 重要的原因。而这一操作主要通过路径决策来实现，在排序规则中 MP 的比重也并不高。

3.8 本章小结

本章节基于超启发式遗传规划算法求解柔性作业车间动态调度问题，针对现有对能耗方面研究的缺失，在车间能流分析的基础上提出了面向节能的终端符和人工规则，并通过实验分析了能耗相关终端符在节能调度优化中的效果和机理。

实验结果表明了本文所设计的能耗终端符能有效优化能耗目标。具体而言，包含能耗相关终端符集合的 GPHH 算法在优化总能耗时，明显比仅有时间相关终端符的算法效果好。此外，所提出的相对性终端符也在一定程度上提高了算法的最终优化性能。

第 4 章 基于双重特征权重集合的节能调度规则自动化生成

4.1 引言

已有研究证明,特征的重要性并不完全相同,而且在特征集中包含不相关的特征会降低优化性能^[42]。然而,现有 GPHH 特征筛选方法更注重使用更少的特征以获得更紧凑的规则,而不是提高有效性。在文献[47]中,通过在 GPHH 算法中应用特征筛选方法,仅使用选定的特征来演化更多可解释的调度规则。然而,由于特征筛选的结果过度依赖于人为设定的阈值^[42, 45],在删除无关和冗余特征的同时,很容易删除对问题有用的特征,导致 GPHH 在生成规则时未能获取足够多的有效信息,而最终影响算法性能的提升。此外,在多目标问题中,不同目标对特征信息的需求不同,对特征贡献度和重要性的度量也更加困难,这进一步增加了特征筛选的难度。

因此,本章节尝试使用特征权重方法来对不同特征的重要性进行更准确的度量,并根据特征在多目标问题求解和优化中的贡献程度为其分配一个合适的权重值。这样做,一方面有效避免了人为设定参数对特征集合的影响,另一方面也可以通过特征权重为算法搜索提供指引,提高算法的搜索效率,自动化生成更高质量和可解释的调度规则。

4.2 基于双重特征权重集合的 GPHH 算法

4.2.1 算法总体框架

考虑到路径决策和排序决策的信息需求存在差异,本文引入了双重特征集合的概念。通过结合特征权重机制和种群调整策略,以更有效地利用迭代过程中积累的优秀个体信息,从而生成更为优秀的调度规则。

图 4-1 给出了所提方法的详细流程图。与传统 GPHH 相比,图 4-1 中增添了特征权重度量模块和特征权重使用模块(图中颜色加深区域并标红)。需要注意的是,初始的特征集合是无权重的,也可以理解为此时特征具有相同的权重。随

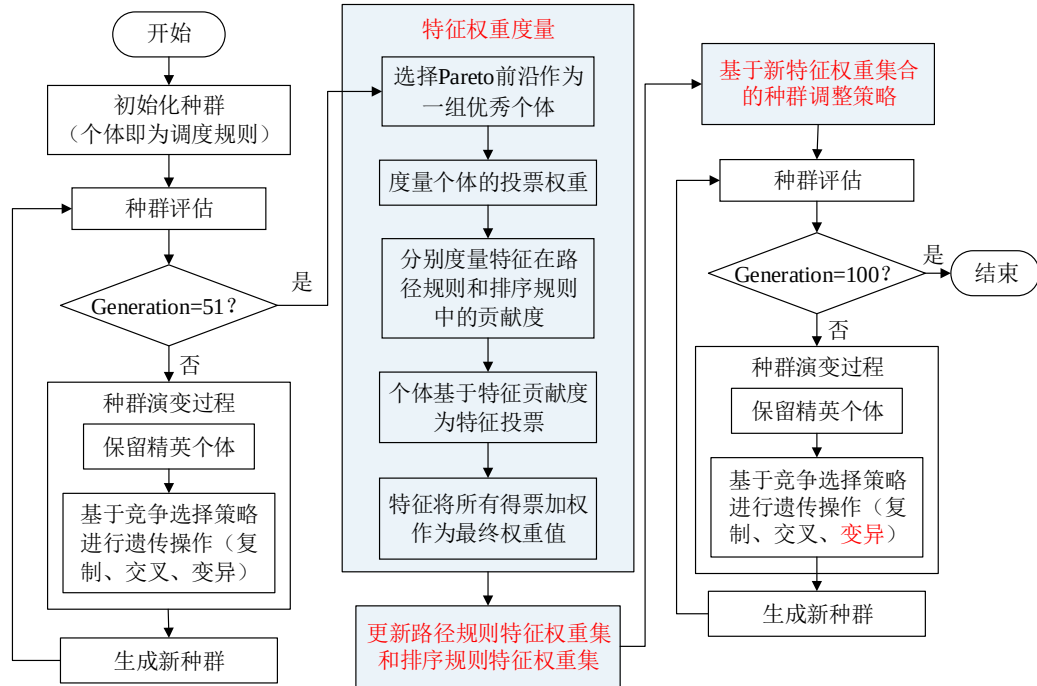


图 4-1 融入特征权重机制和种群调整策略的 GPHH 流程图

着 GPHH 的初始化和迭代（即前 50 代）的进行，能够得到一组优秀个体，满足高精度的特征权重评估条件。

在特征权重度量模块中，由于涉及多目标问题，直接选取 Pareto 前沿作为一组优秀且多样化的个体。通过度量特征在个体和种群中的贡献度（重要程度），为每个特征分配一个权重值，用于指导种群后续的进化方向。考虑到特征权重评估操作的耗时性质，为了兼顾效率，在整个算法迭代过程中只执行一次特征权重评估操作。本章节创新点之一在于将路径规则和排序规则独立处理，分别生成路径规则特征权重集合和排序规则特征权重集合。

所得新特征权重集合在算法后续的 50 代中主要用于种群调整策略和个体的变异操作。其中，种群调整策略的作用是利用新的特征权重集合调整种群中的个体，例如，在这一过程中删除权重为 0 的无关特征和使用新的特征权重集合随机生成新个体等。

值得注意的是，在第 51 代特征权重评估完成后，新的特征权重集合会替换初始无权重特征集合，这使得在后续的个体初始化和变异算子生成新的子树时，具有高权重的特征将有大概率被选中。因此，准确度量特征的权重对指导 GPHH 生成更优秀的调度规则至关重要。为此，本文提出了两种适用于多目标的特征权

重度量方法：一是基于适应度值的特征权重度量方法，二是基于多样性的特征权重度量方法。

4.2.2 基于适应度值的特征权重度量方法

Mei 等^[42]提出了一种用于 GPHH 的特征筛选机制来解决单目标车间调度问题，并指出特征的重要性既取决于个体适应度值，也与特征本身的贡献度有关。本文在此基础上，研究多目标中个体的非支配关系，并提出适用于多目标问题环境下的基于适应度值的特征权重度量方法。

算法 4.1 给出了基于适应度值的特征权重度量方法的伪代码描述，其本质是一个加权多数投票过程。首先，将当前一代种群的 Pareto 前沿 $\mathcal{R}$ （视为一组优秀个体）和初始无权重特征集合作为输入。然后，逐一评估每个特征对 $\mathcal{R}$ 中个体的适应度贡献，如果某个特征对该个体的适应度值有贡献，那么个体就会投票给它。最终，每个特征将得到的所有投票权重加权作为自己的权重值，用于衡量特征的重要程度。

算法 4.1：基于适应度值的特征权重度量方法

输入：当前一代种群的 Pareto 前沿 $\mathcal{R}$ ，初始无权重特征集合 $\mathcal{F}_w$

输出：新的特征权重集合 $\mathcal{F}_w^*$

```

1  for each  $f \in \mathcal{F}_w$  do
2     $vote(f) \leftarrow 0$ 
3    for each  $r \in \mathcal{R}$  do
4      由式(4-1)计算个体  $r$  投票权重  $w_{fit}(r)$ 
5      由式(4-2)计算特征贡献度  $Con(f, r)$ 
6      if  $Con(f, r) > 0$  then
7         $vote(f) \leftarrow vote(f) + w_{fit}(r)$ 
8      end if
9    end for
10 end for//此时每个特征都赋予了一个权重值
11 将 $\mathcal{F}_w^*$ 中所有权重归一化成百分制数
12 return  $\mathcal{F}_w^*$ 

```

式(4-1)定义了个体的投票权重，其中共有 m 个优化目标， $fit_i(r)$ 表示规则对第 i 个目标的适应度值。本文求解最小化目标值，个体的适应度值越小，其投票权重越大。

$$w_{fit}(r) = \sum_{i=1}^m \frac{\max\{fit_i(r) | r \in \mathcal{R}\} - fit_i(r)}{\max\{fit_i(r) | r \in \mathcal{R}\} - \min\{fit_i(r) | r \in \mathcal{R}\}} \quad (4-1)$$

式(4-2)定义了特征对个体的贡献度，将特征置 1 等价于删除特征，例如， $(PT+WIQ|PT=1)=1+WIQ$ 。通过分子部分的差值判断特征在个体中是否有贡献，如果最终的 $Con(f, r) > 0$ ，表示特征对个体有正向贡献，个体会为此特征投票。

$$Con(f, r) = \sum_{i=1}^m \frac{fit_i(r | f=1) - fit_i(r)}{\max\{fit_i(r) | r \in \mathcal{R}\} - \min\{fit_i(r) | r \in \mathcal{R}\}} \quad (4-2)$$

4.2.3 基于多样性的特征权重度量方法

考虑到求解多目标问题的目标是获取分布更广泛、更多样化的 Pareto 前沿，除了使用适应度值度量方法评估特征的权重以外，本文还提出一种基于多样性的度量方法，用于衡量特征的重要性。此方法更加注重特征和个体对整个 Pareto 前沿的贡献度，算法 4.2 给出了该方法的伪代码描述。

算法 4.2: 基于多样性的特征权重方法

输入: 当前一代种群的 Pareto 前沿 $\mathcal{R}$, 初始无权重特征集合 $\mathcal{F}_w$

输出: 新的特征权重集合 $\mathcal{F}_w^*$

```

1  for each  $f \in \mathcal{F}_w$  do
2     $vote(f) \leftarrow 0$ 
3    for each  $r \in \mathcal{R}$  do
4      由式(4-3)计算个体  $r$  的投票权重  $w_{div}(r)$ 
5      //计算特征贡献度  $Con(f, r)$ 
6      将个体  $r$  中的特征  $f$  置 1, 记为  $r|f=1$ 
7      计算  $fit(r|f=1)$ 
8      if 存在  $r \in \mathcal{R}$  支配了  $r|f=1$  then
9         $Con(f, r) = 1$ ;
10     else
11        $Con(f, r) = 0$ ;
12     end if
13     if  $Con(f, r) > 0$  then
14        $vote(f) \leftarrow vote(f) + w_{spa}(r)$ 
15     end if
16   end for
17 end //此时每个特征都赋予了一个权重值
18 将  $\mathcal{F}_w^*$  中所有权重归一化成百分制数
19 return  $\mathcal{F}_w^*$ 

```

式 (4-3) 定义了个体的拥挤距离，作为个体的投票权重，其中 $fit_i(r+1) - fit_i(r-1)$ 表示与个体 r 相邻的两个个体的适应度差。这种度量方法的优势在于，它能够在空间上更均匀地分配投票权重给 Pareto 前沿中的所有个体，有助于提升多目标非支配关系问题的优化性能。对于局部扎堆聚集的个体，给予一定惩罚，从而更好地避免局部最优解对特征权重度量结果的影响。

$$w_{div}(r) = \sum_{i=1}^m \frac{|fit_i(r+1) - fit_i(r-1)|}{\max\{fit_i(r) | r \in \mathcal{R}\} - \min\{fit_i(r) | r \in \mathcal{R}\}} \quad (4-3)$$

在计算特征对个体的贡献度 $Con(f, r)$ 时，先计算特征 f 设置为常数 1 时规则的适应度值 $fit(r | f = 1)$ ，如果其被 Pareto 前沿中的任何一个个体支配了，则认为此特征对个体和种群有正向贡献。支配条件为：对于 $\forall i \in \{1, 2, \dots, m\}$ ，都有 $fit_i(r) \leq fit_i(r | f = 1)$ ，且至少存在一个 i ，使得 $fit_i(r) < fit_i(r | f = 1)$ 。

4.2.4 基于特征权重的种群调整策略

评估完成特征权重后，重要性较低的特征会赋予一个低权重值。如果一个特征被赋予了 0 的权重，那么就可以确定这个特征为问题的无关特征。虽然 GPHH 本身具备一定的特征筛选功能，但仅通过交叉、变异等遗传操作很难有效地去除这些无关特征。如果放任这些无关特征一直存在于个体中，将会导致种群受到“污染”，使得更多个体带有冗余、重复和不必要的特征信息，不利于种群的进一步迭代演化和调度规则性能的优化。

评估特征重要性之后，为了避免无关特征对种群的负面影响，需要基于所得特征重要性对当前种群进行调整，以删除无关特征。常见的方法有“置 1 法”、“随机替换法”和“随机初始化法”^[47]。

“置 1 法”基于上述特征重要性的计算思想，直接将无关特征置 1，能最大程度地保留当前优良个体的结构和性能。而“随机替换法”则是将无关特征替换为其他重要性更高的特征，这样的随机替换可能会在某些方面改变优良个体的行为，影响其最终优化性能。对于“初始化方法”，其基本思想类似于算法的种群初始化，直接运用评估后的特征集合随机初始化种群，但这样的方法丢失了当前种群中优良个体的有效信息。

已有用于求解单目标问题的相关研究表明^[47]，“置 1 法”相较于“随机替换

法”而言，能更好地提升算法的性能。此外，本文在评估特征权重时将特征置 1 作为判断依据，因此，使用“置 1 法”删除个体中的无关特征不仅有助于消除冗余信息，保留个体结构，还可能提升个体性能。

鉴于此，本文提出了一种基于特征权重的种群调整策略，同时运用“置 1 法”与“初始化方法”对当前种群进行调整，并在实验中验证了其有效性。

算法 4.3 给出了完整种群调整策略的伪代码描述。算法中只对排名前 20% 的优良个体使用“置 1 法”删除个体中的无关特征，以保留优良个体的有效信息。而对于剩余 80% 的非优良个体，则直接删除并基于新的特征权重集合使用“初始化方法”重新生成，以在删除无关特征的同时提高种群中个体的多样性。

算法 4.3: 基于特征权重的种群调整策略

输入：当前一代的种群 $\mathcal{P}$ ，新的特征权重集合 $\mathcal{F}_w^*$

输出：新种群 $\mathcal{P}^*$

```

1  根据个体非支配等级和拥挤距离对个体进行排序
2  for 种群  $\mathcal{P}$  中排名前 20% 的个体
3      for each  $f \in \mathcal{F}_w^*$  do
4          if  $\text{vote}(f) < 0.0001$  then
5              使用“置 1 法”删除这些无关特征
6          end if
7      end for
8  end for
9  for 种群  $\mathcal{P}$  中剩下的个体
10     基于新的特征权重集合  $\mathcal{F}_w^*$  使用“初始化方法”重新生成个体
11 end for
12 return  $\mathcal{P}^*$ 

```

4.3 仿真实验设计

4.3.1 实验对比算法

为了有效分析所提考虑特征权重的双重特征集合 GPHH 算法的具体性能表现，本文将其与已有的三种相关 GPHH 算法进行了对比实验，具体算法如下：

(1) GPLWT^[61](即 GPHH-LWT)，指使用最短等待时间规则(the Least Waiting Time, LWT)作为路径规则，并通过 GPHH 来生成排序规则。换言之，该算法只自动化设计排序规则，而其路径规则是固定的。在本章中，将 GPLWT 视为问题求

解的基准算法。

(2) GPMT^[56](即 GPHH-Multi-Tree), 与 GPLWT 不同之处在于, 该算法的一个调度规则个体包含两个子规则, 分别表示路径规则和排序规则, 并都使用 GPHH 自动生成。在解决 DFJSS 问题方面, 该算法被认为是目前最为有效的 GPHH 算法之一。

(3) GPFS^[47](即 GPHH-Feature-Select), 是一种融入特征筛选方法的 GPHH 算法。在之前的文献中, GPFS 方法通常用于求解单目标问题, 本文基于多目标加权方法将它应用到多目标问题中并分析其性能。

(4) GPFW_{fit}(即 GPHH-Feature-Weight_{fitness}), 是本文提出的考虑双重特征权重集合的 GPHH 算法, 采用基于适应度值的特征权重度量方法。

(5) GPFW_{div}(即 GPHH-Feature-Weight_{diversity}), 是本文提出的考虑双重特征权重集合的 GPHH 算法, 采用基于多样性的特征权重度量方法。

4.3.2 仿真模型构建

本章所采用的面向节能的 DFJSS 问题模型的具体参数设置如表 4-1 所示, 该参数模型被广泛应用于目前已有的研究工作中^[47, 62]。

表 4-1 DFJSS 问题仿真模型参数

参数	设定
加工设备数 m	10
工件个数 n	2000
前期工件个数	500
工件工序数及其分布	$U(1, 10)$
工序候选设备数及其分布	$U(1, 10)$
工件到达过程	泊松分布
设备使用水平 u	0.85, 0.95
交货期宽裕度系数 α	2, 4, 6
工序平均加工时间及其分布	$U(1, 99)$
工序平均能耗及其分布	$U(1, 99)$
设备待机功率	{10, 12.5, 4.5, 3.6, 7.0, 1.5, 8.5, 2.2, 22.9, 6.4}

与第 3 章中的参数相似,仍使用 2 种不同的设备使用水平和 3 种不同的交货期宽裕度系数,共构成了 $2 \times 3 = 6$ 种不同的生产情境。在训练阶段,将 GPHH 算法在每种情境下基于不同的随机种子进行 30 次独立运行。为了测试生成的调度规则的有效性,将每次运行的结果进行了 100 次独立测试,然后取这 100 次测试结果的平均值作为所得调度规则的最终适应度值,以验证其实际泛用性和普适性。

4.3.3 GPHH 参数设置

表 4-2 给出了 GPHH 算法的具体参数设置^[47,63]。实验设置在第 51 代时评估特征权重,且整个算法迭代过程只进行一次特征权重的评估操作。可以简单的理解为,GPLWT 和 GPMT 运行了 100 代,而 GPFS 和本文所提特征权重算法运行了 50+50 代。

结合本文第三章设计的能耗相关终端符,表 4-3 给出了本文实验所用的 28 个与车间调度问题相关的特征,它们被广泛用于相关研究^[55]。这些特征涵盖了多个方面,包括与工件、设备和车间相关的各种信息。

算法函数符集为 $\{+, -, \times, /, \max, \min\}$ ^[56],前四个是传统的算术运算符,其中除法运算具有除数保护功能,如果分母为零,结果将返回 1。另外, $\max$ 和 $\min$ 是接受两个参数并返回它们中的较大值或较小值的函数。

表 4-2 GPHH 算法参数设计

参数	设定
初始化方法	Ramped half-and-half
种群大小	600
规则树最大深度	8
交叉率	80%
变异率	15%
复制率	5%
父代选择方法	容量为 7 的锦标赛选择法
精英保留个体	10
总迭代次数	101
评估特征权重	第 51 代时
种群调整策略	继承并调整排名前 20%的个体,剩余个体初始化

表 4-3 GPHH 算法所使用的特征集

序号	特征	含义
1	PT	工序加工时间
2	SL	工件宽裕时间
3	OWT	工件等待时间
4	NPT	下一个工序的加工时间
5	WKR	工件剩余工序总加工时间
6	TIS	工件在制造系统中的时长
7	WIQ	设备的待加工队列中的所有工序的加工总时间
8	MWT	设备等待时间
9	DD	工件交货期
10	MRT	设备就绪时间
11	ORT	工件工序就绪时间
12	WINQ	下一台设备待加工队列的工作量
13	AT	工件到达时间
14	NRT	下一台设备的就绪时间
15	EC	工序加工能耗
16	NEC	下一个工序的加工能耗
17	ECR	工件剩余工序总加工能耗
18	EIQ	设备的待加工队列中的所有工序的加工总能耗
19	MP	设备待机功率
20	EINQ	下一台设备待加工队列的总能耗
21	NOR	工件剩余工序数
22	NOS	工件工序加工候选设备数
23	NIQ	设备的待加工队列中的工序数
24	NINQ	下一台设备待加工队列的工序数
25	NOPS	工件工序数
26	RPT	相对加工时间 = $PT_{ijk} / \min\{PT_{ijk} k=1, \dots, \mathcal{M}_j \}$
27	REC	相对加工能耗 = $EC_{ijk} / \min\{EC_{ijk} k=1, \dots, \mathcal{M}_j \}$
28	RMP	相对待机能耗 = $MP_k / \min\{MP_k k=1, \dots, m\}$

4.3.4 算法性能指标

本章仍致力于多目标问题的讨论，并在评价算法所得 Pareto 前沿的质量时，

延续使用 3.5.5 节中介绍的 HV 和 IGD 评价指标。一个优秀的算法应该具有较高的 HV 值和较低的 IGD 值，这表明该算法所得 Pareto 前沿更为接近真实 Pareto 前沿，且其解的分布更加均匀，体现了解集在最优性和多样性方面的更优表现。

4.4 实验结果分析

4.4.1 算法总体性能分析

表 4-4 和表 4-5 显示了不同算法在 6 种不同的情境下，针对 TEC 和 MT 两个目标，经过 30 次独立运行得到的 HV 和 IGD 指标的平均值和标准差（括号中）。此外，本文使用 Wilcoxon 秩和检验（基于 5% 的置信水平）对算法性能进行显著性分析^[60]。将其他算法与 GPLWT 和 GPMT 进行比较，如果 Wilcoxon 秩和检验的 p 值都小于 0.05，在表格中使用粗体标记来表示差异显著。结果显示，GPFS 在半数情境下的 p 值存在显著差异。对于本文所提算法 GPFW_{fit} 和 GPFW_{div} ，在所有情境中 p 值在差异上都没有表现出太大的显著性。

从表中可以看出，所有算法的性能都远好于 GPLWT，说明使用 GPHH 生成的路径规则较使用一个固定的 LWT 规则作为路径规则效果要好。因为 GPHH 能够基于当前车间系统的信息生成更为合理的路径规则进行路径决策，为设备产生更合理的待加工队列，以更好的配合后续的排序决策。

在目前已有的 GPHH 算法中，GPMT 和 GPFS 是求解 DFJSS 问题最好的算法。研究表明^[42]，GPFS 在单目标问题中可以筛选出重要特征，通过使用更少的特征数量来使生成的规则更具可解释性且在性能上与 GPMT 相近。本文将 GPFS 应用于多目标问题中，发现在绝大多数情境下，GPFS 算法的性能都不如 GPMT，而且标准差较大，特别是在车间负载较为紧张的<0.85-2>和<0.95-2>情境下。这表明通过加权和的方式将 GPFS 算法简单平移到多目标问题中并不合适，所选择的特征无法很好的满足多目标优化的需求，从而影响了算法的优化性能，且对其稳定性也造成了影响。

将 GPFW_{fit} 和 GPFW_{div} 对比 GPMT 可以看出，两个算法总体比 GPMT 要好，但本身没有显著性差异。可能的原因是，GPHH 算法本身就具有一定的特征筛选的功能，因为融合了无关和非必要特征的调度规则性能大概率也不是很好，从而

在迭代进化过程中会被 GPHH 自然淘汰。因此，最后生成的调度规则本身就是包含了与优化目标强相关的特征，限制了性能提升的效果。

比较 $GPFW_{fit}$ 和 $GPFW_{div}$ 后发现它们的结果非常接近，没有明显的差异。在理论上，基于多样性的特征权重度量方法可能会略微优于基于适应度值的特征权重度量方法。这一结果说明，在经过 GPHH 的 50 次迭代后，获得的 Pareto 前沿已经足够优秀，并没有陷入局部最优解的情况，可以用于准确评估特征权重。

表 4-4 不同算法在 6 种生产情境下的 HV 指标平均值(标准差)

算法	0.85-2	0.85-4	0.85-6	0.95-2	0.95-4	0.95-6
GPLWT	0.324 (0.001)	0.549 (0.000)	0.602 (0.000)	0.392 (0.001)	0.578 (0.001)	0.668 (0.001)
GPMT	0.977 (0.010)	1.000 (0.007)	1.024 (0.005)	0.970 (0.009)	1.005 (0.005)	1.027 (0.006)
GPFS	0.923 (0.101)	0.996 (0.018)	1.024 (0.004)	0.957 (0.033)	0.994 (0.024)	1.017 (0.019)
$GPFW_{fit}$	0.979 (0.009)	1.001 (0.006)	1.024 (0.005)	0.973 (0.012)	1.007 (0.006)	1.027 (0.005)
$GPFW_{div}$	0.979 (0.009)	1.001 (0.006)	1.024 (0.005)	0.973 (0.012)	1.007 (0.006)	1.027 (0.005)

表 4-5 不同算法在 6 种生产情境下的 IGD 指标平均值(标准差)

算法	0.85-2	0.85-4	0.85-6	0.95-2	0.95-4	0.95-6
GPLWT	0.538 (0.001)	0.420 (0.000)	0.398 (0.000)	0.457 (0.001)	0.372 (0.001)	0.325 (0.001)
GPMT	0.014 (0.005)	0.025 (0.006)	0.047 (0.010)	0.025 (0.006)	0.044 (0.013)	0.032 (0.007)
GPFS	0.044 (0.074)	0.025 (0.012)	0.045 (0.010)	0.034 (0.021)	0.045 (0.020)	0.036 (0.012)
$GPFW_{fit}$	0.014 (0.005)	0.025 (0.005)	0.047 (0.008)	0.023 (0.005)	0.038 (0.016)	0.032 (0.007)
$GPFW_{div}$	0.014 (0.005)	0.024 (0.006)	0.047 (0.008)	0.024 (0.006)	0.041 (0.014)	0.031 (0.005)

4.4.2 算法训练时间和测试时间对比

表 4-6 给出了算法 30 次独立运行的平均训练时间（时间花费较大的部分已用粗体标出）。由于 GPLWT 使用固定的路径规则，计算量较小，因此其训练时间最短。相较 GPMT 而言，GPFS、 $GPFW_{fit}$ 和 $GPFW_{div}$ 在特征筛选和特征权重

评估阶段花费了较长的时间。但从总体训练时间来看，这些算法之间并没有明显差异。这表明特征权重评估过程的引入并没有给算法带来过多的计算量和计算时间，整体耗时属于可接受范围。

表 4-7 给出了算法所得规则经过 100 次独立测试的平均时间（时间花费较大的部分已用粗体标出）。可以观察到，这些调度规则仅需 20 毫秒左右就可以完成单次仿真调度过程中的所有调度决策，而平均到每个决策点，花费的时间会更少，满足实时性调度决策的要求。

表 4-6 不同算法经过的 30 次训练的平均时间（单位: s ）

算法	0.85-2	0.85-4	0.85-6	0.95-2	0.95-4	0.95-6
GPLWT	5124	4988	5032	4587	4602	4495
GPMT	10745	11568	11814	10656	12948	13530
GPFS	11670	12923	13086	9657	9502	10183
GPFW _{fit}	13029	13383	13441	10122	10590	10097
GPFW _{div}	9741	10375	13697	13638	13791	13864

表 4-7 不同算法所得的规则经过 100 次测试的平均时间（单位: ms ）

算法	0.85-2	0.85-4	0.85-6	0.95-2	0.95-4	0.95-6
GPLWT	3.90	3.00	3.00	4.40	4.04	3.43
GPMT	14.36	15.58	16.17	14.34	16.93	16.50
GPFS	11.31	15.08	13.26	12.99	13.74	13.95
GPFW _{fit}	21.00	20.95	23.02	19.45	22.10	25.04
GPFW _{div}	15.03	15.90	14.45	14.88	16.73	17.42

4.3.3 特征权重机制的有效性分析

为了验证特征权重机制的有效性，本节额外设置了一组对比实验。

(1) GPFS(noadjust)，表示 GPFS 算法在特征筛选完成后，不使用（跳过）种群调整策略。此时，筛选后的特征集合在后续迭代过程中只用于算法的变异操作。

(2) GPFS(newrun)，表示将 GPFS 算法的特征筛选结果导出，并将其作为初始特征集合，重新运行 100 代，在此过程中不再进行特征筛选操作。

(3) GPFWfit(noadjust)，表示 GPFWfit 算法在特征权重评估完成后，不使用

(跳过) 种群调整策略。此时, 评估后的特征权重集合在后续迭代过程中只用于算法的变异操作。

(4) $\text{GPFW}_{\text{fit}}(\text{newrun})$, 表示将 GPFW_{fit} 算法的特征权重评估结果导出, 并将其作为初始特征权重集合, 重新运行 100 代, 在此过程中不再进行特征权重评估操作。

(5) $\text{GPFW}_{\text{div}}(\text{noadjust})$, 表示 GPFW_{div} 算法在特征权重评估完成后, 不使用(跳过) 种群调整策略。此时, 评估后的特征权重集合在后续迭代过程中只用于算法的变异操作。

(6) $\text{GPFW}_{\text{div}}(\text{newrun})$, 表示将 GPFW_{div} 算法的特征权重评估结果导出, 并将其作为初始特征权重集合, 重新运行 100 代, 在此过程中不再进行特征权重评估操作。

表 4-8 和表 4-9 显示了不同算法在 6 种不同的情境下, 针对 TEC 和 MT 两个目标, 经过 30 次独立运行得到的 HV 和 IGD 指标的平均值和标准差。将算法与 GPMT 相比, 如果 Wilcoxon 秩和检验结果的 p 值小于 0.05, 在表格中使用粗体标记来表示差异显著。

从两个表中可以看出, $\text{GPFW}_{\text{fit}}(\text{noadjust})$ 和 $\text{GPFW}_{\text{div}}(\text{noadjust})$ 的效果提升不明显, 因为在不使用种群调整策略时, 算法所得特征权重集合只应用于后续 50 代的变异操作。然而, 由于算法设置的变异率仅为 0.15, 其对算法本身的影响较小, 因此不会导致太大的性能差异。

$\text{GPFW}_{\text{fit}}(\text{newrun})$ 和 $\text{GPFW}_{\text{div}}(\text{newrun})$ 相较于 GPMT 而言, 在多数情境下都表现出显著的性能提升。考虑到这三个算法都运行了 100 代, 唯一的不同在于 GPMT 使用的是无权重特征集合, 而 $\text{GPFW}_{\text{fit}}(\text{newrun})$ 和 $\text{GPFW}_{\text{div}}(\text{newrun})$ 使用的是考虑了特征贡献度差异性的特征权重集合。这样的结果说明了所提特征权重度量方法得到的特征权重是合理的, 且确实能有效提高算法的性能。该方法能消除非必要的无关特征, 从而提高算法的搜索效率, 并发现更好的调度规则。

通过观察 GPFS(noadjust)和 GPFS(newrun)的效果, 发现这两个算法的表现很不稳定。GPFS(newrun)在 HV 和 IGD 指标上都要明显差于 GPMT, 此现象较容易解释, 因为特征筛选不可避免的会删除一些有用特征, 导致生成的规则对问题的考虑不全面。

表 4-8 不同算法在 6 种生产情境下的 HV 指标平均值(标准差)

算法	0.85-2	0.85-4	0.85-6	0.95-2	0.95-4	0.95-6
GPMT	0.977 (0.010)	1.000 (0.007)	1.024 (0.005)	0.970 (0.009)	1.005 (0.005)	1.027 (0.006)
GPFS(noadjust)	0.977 (0.010)	1.000 (0.008)	1.020 (0.008)	0.973 (0.016)	1.002 (0.013)	1.020 (0.009)
GPFS(newrun)	0.927 (0.106)	0.967 (0.073)	0.990 (0.067)	0.928 (0.080)	0.975 (0.055)	0.996 (0.046)
GPFW _{fit} (noadjust)	0.977 (0.009)	1.001 (0.006)	1.025 (0.005)	0.974 (0.010)	1.007 (0.004)	1.027 (0.004)
GPFW _{fit} (newrun)	0.982 (0.008)	1.005 (0.004)	1.025 (0.005)	0.975 (0.009)	1.008 (0.005)	1.028 (0.004)
GPFW _{div} (noadjust)	0.977 (0.010)	1.002 (0.004)	1.024 (0.004)	0.970 (0.011)	1.006 (0.005)	1.027 (0.005)
GPFW _{div} (newrun)	0.982 (0.008)	1.004 (0.004)	1.026 (0.004)	0.976 (0.010)	1.007 (0.007)	1.027 (0.004)

表 4-9 不同算法在 6 种生产情境下的 IGD 指标平均值(标准差)

算法	0.85-2	0.85-4	0.85-6	0.95-2	0.95-4	0.95-6
GPMT	0.014 (0.005)	0.025 (0.006)	0.047 (0.010)	0.025 (0.006)	0.044 (0.013)	0.032 (0.007)
GPFS(noadjust)	0.014 (0.004)	0.019 (0.005)	0.039 (0.009)	0.024 (0.007)	0.032 (0.008)	0.019 (0.008)
GPFS(newrun)	0.044 (0.078)	0.044 (0.062)	0.062 (0.055)	0.054 (0.049)	0.053 (0.042)	0.035 (0.037)
GPFW _{fit} (noadjust)	0.014 (0.006)	0.026 (0.005)	0.046 (0.007)	0.024 (0.006)	0.042 (0.014)	0.034 (0.006)
GPFW _{fit} (newrun)	0.011 (0.005)	0.022 (0.004)	0.050 (0.007)	0.023 (0.004)	0.034 (0.013)	0.030 (0.005)
GPFW _{div} (noadjust)	0.014 (0.006)	0.025 (0.005)	0.045 (0.010)	0.024 (0.006)	0.040 (0.014)	0.033 (0.007)
GPFW _{div} (newrun)	0.011 (0.005)	0.022 (0.004)	0.049 (0.007)	0.021 (0.005)	0.034 (0.012)	0.028 (0.005)

4.4.4 种群调整策略的有效性分析

本节对种群调整策略的有效性做进一步分析,以 GPFW_{div} 算法为例,讨论种群调整策略中初始化种群百分比对算法结果的影响。例如,80%random 指的是在种群调整策略中,将种群中排名前 20%的个体中的无关特征置 1,排名后 80%的个体删除并使用初始化方法重新随机生成。表 4-10 和表 4-11 给出了不同算法

在 6 种情境下, 针对 TEC 和 MT 两个目标, 经过 30 次独立运行得到的 HV 和 IGD 指标的平均值和标准差。

表 4-10 GPFW_{div} 算法在种群调整策略中使用不同比例初始化的 HV 指标平均值(标准差)

情境	0%random	20%random	40%random	80%random	100%random
0.85-2	0.978(0.008)	0.977(0.009)	0.978(0.011)	0.979(0.008)	0.976(0.009)
0.85-4	1.001(0.006)	1.001(0.006)	1.001(0.006)	1.001(0.007)	1.000(0.008)
0.85-6	1.025(0.004)	1.024(0.006)	1.025(0.005)	1.025(0.004)	1.022(0.005)
0.95-2	0.974(0.013)	0.972(0.010)	0.970(0.015)	0.972(0.012)	0.966(0.016)
0.95-4	1.006(0.006)	1.006(0.005)	1.005(0.008)	1.006(0.006)	1.004(0.009)
0.95-6	1.026(0.008)	1.025(0.010)	1.026(0.005)	1.027(0.005)	1.019(0.010)

表 4-11 GPFW_{div} 算法在种群调整策略中使用不同比例初始化的 IGD 指标平均值(标准差)

情境	0%random	20%random	40%random	80%random	100%random
0.85-2	0.014(0.004)	0.014(0.005)	0.014(0.004)	0.014(0.005)	0.015(0.004)
0.85-4	0.025(0.006)	0.025(0.005)	0.025(0.005)	0.024(0.006)	0.025(0.007)
0.85-6	0.048(0.008)	0.049(0.009)	0.045(0.010)	0.047(0.008)	0.045(0.007)
0.95-2	0.024(0.008)	0.025(0.006)	0.026(0.007)	0.024(0.006)	0.027(0.007)
0.95-4	0.039(0.015)	0.038(0.014)	0.042(0.016)	0.041(0.014)	0.046(0.013)
0.95-6	0.033(0.008)	0.033(0.010)	0.033(0.007)	0.031(0.005)	0.037(0.008)

可以看出, 在种群调整策略中选择 100%的种群初始化比例时, 算法性能相较于其他情况都有所降低。这是因为在 100%随机初始化的情况下, 新种群并没有继承前一代的优秀个体, 而是随机生成了性能较差的全新种群。前文研究已知应用新的特征权重集合对种群演化更高质量规则有积极作用, 但这里 100%random 并未继承前 50 代种群的演化结果, 所以它实际上相当于仅进行了后 50 次的演化迭代, 在性能上比不上其它算法也是可以理解的。从这也可以看出, 使用特征权重集合指导 GPHH 生成调度规则是有效的。因为仅进行了 50 代的运行, 其性能已经接近其他算法 100 代的算法收敛结果。

然而, 在 0-80%random 时, 算法之间的差异并不明显。可能的原因在于, 在优化过程中真正决定种群质量的主要是排名靠前的一部分个体, 而排名靠后的个体在后续迭代过程中很难被有效地遗传继承。总的来看, 80%random 是一个可以接受且相对合理的折中结果。

4.4.5 特征种类数量分析

特征种类数量指的是调度规则中包含的不同特征的种类数^[46]。本文中讨论由 GPHH 算法自动筛选并保留下来的对性能有积极贡献的有用特征的数量，具体的计算需要查看最终种群中出现的特征的种类数量。特征种类数量越少，意味着由这些特征构造的规则更为简洁，具有更好的可解释性，从而更容易被人理解。

图 4-2 显示了算法经过 30 次独立运行所得调度规则中特征种类数量的平均值及其标准差。其中，GPFS 的特征种类数量相对较少，因为该算法的目标在于使用更少的规则生成高质量规则。其特征种类数量的大小主要取决于特征筛选时的投票阈值设置，本文中特征被选中的条件为投票权重大于总投票权重的一半。因此，权重值低于总投票权重一半的特征都将被删除，这使得结果中特征种类数量明显减少，但也导致了算法优化性能的下降。

相反，特征权重方法并不能直接删除特征，只有权重值为 0 的无关特征才会在执行种群调整策略时被彻底删除。除此以外，特征权重集合在一定程度上影响了 GPHH 中规则的演化方向，重要性较低的特征可能不会出现在最终种群中，这也会导致特征种类数量减少。因此，图 4-2 结果表明考虑了特征权重的 GPHH 算法可以在保持算法性能不变（甚至较好）的情况下有效降低所得调度规则中特

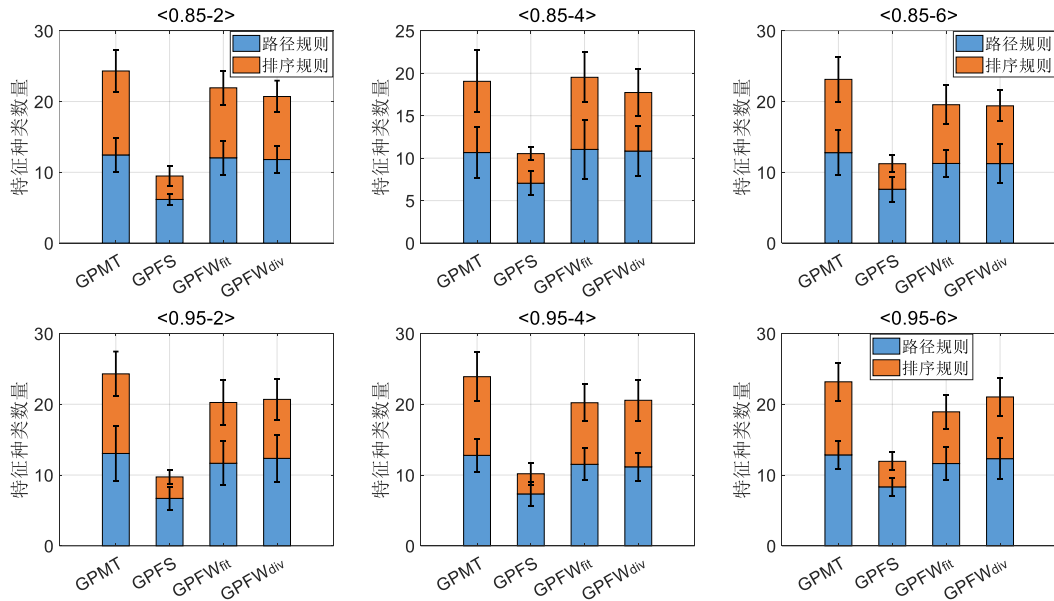


图 4-2 不同算法所得规则中特征种类数量

征种类数量,从而可以演化生成更具有可读性和可解释性的节能调度规则。这也从另一个侧面揭示了将特征权重引入到 GPHH 算法中是合理而有效的,因为即使在复杂的多目标环境中,所提算法赋予的权重也并没有降低算法本身的性能。

从图 4-2 中还可以观察到,排序规则的特征种类数量更容易得到优化,这可能是因为排序决策所需的决策信息相对较少。排序决策通常更加关注与工件相关的加工信息,因为排序决策的范围仅涉及设备待加工队列中工件的加工顺序。相比之下,路径规则需要确定每个工件在设备上的加工路径,除了考虑工件本身的特征以外,还需要考虑设备的可用性、加工队列状态等多方面因素。这使得路径规则的决策信息更加复杂,需要更多的特征信息来进行辅助决策。因此,路径规则的特征种类数量相对较多,对于其性能的提升可能需要更多的优化工作。

4.4.6 特征权重与特征频率的联合分析

为了深入了解算法对于不同特征的重视程度以及这些特征在算法中的相对重要性。本节对 $GPFW_{fit}$ 和 $GPFW_{div}$ 两个算法进行了特征权重与特征频率的联合分析。图 4-3 和图 4-4 分别统计了这两个算法生成的路径规则和排序规则中各特征的权重值以及这些特征的出现频率,统计结果为 30 次独立运行结果的平均值。

(1) $Gen50$ 统计的是 $GPFW_{fit}$ 和 $GPFW_{div}$ 迭代到第 50 代时种群的 Pareto 前沿个体中各特征出现的频率。因为两个算法的前 50 代过程相同,所以输出为相同的结果。

(2) $Gen_{fit}51$ 和 $Gen_{div}51$ 分别表示 $GPFW_{fit}$ 和 $GPFW_{div}$ 在第 51 代时评估得到的各特征权重值。

(3) $Gen_{fit}100$ 和 $Gen_{div}100$ 分别表示 $GPFW_{fit}$ 和 $GPFW_{div}$ 运行 100 代后最终得到的 Pareto 前沿个体中各特征出现的频率。

从图 4-3 和图 4-4 中可以明显看出, $Gen50$ 、 $Gen_{fit}51$ 、 $Gen_{div}51$ 、 $Gen_{fit}100$ 和 $Gen_{div}100$ 的结果基本趋于一致,表明它们之间存在关联性。 $Gen50$ 的特征频率与 $Gen_{fit}51$ 和 $Gen_{div}51$ 的特征权重结果基本吻合,说明特征频率在一定程度上也可以用来度量特征的重要性。同时,这也从侧面说明了 GPHH 本身就具有一定的特征筛选能力,在迭代 50 代后得到的 Pareto 前沿也足够优秀,可以作为一组优秀个体进行特征权重评估。

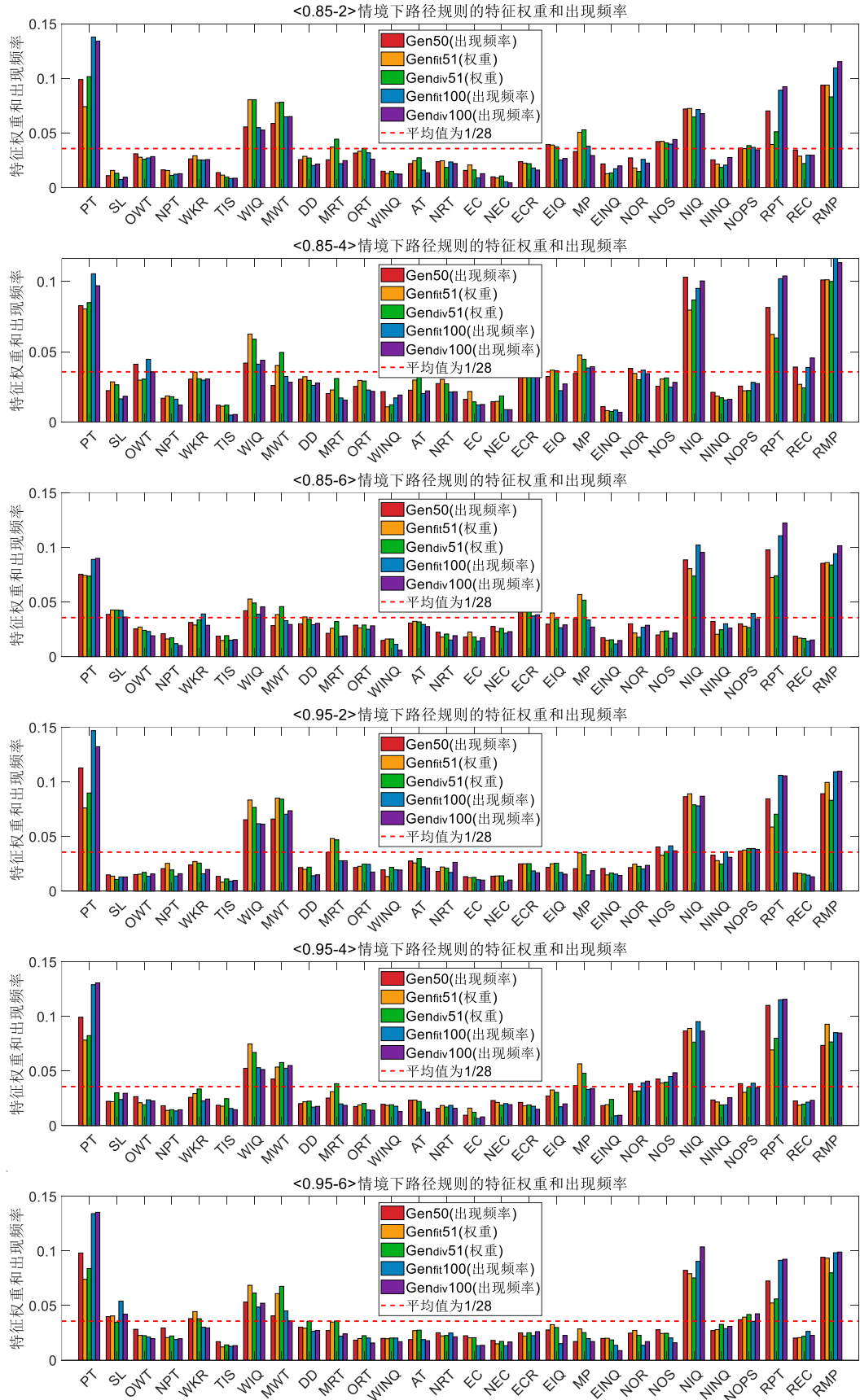


图 4-3 6 种不同情境下路径规则的特征权重和出现频率

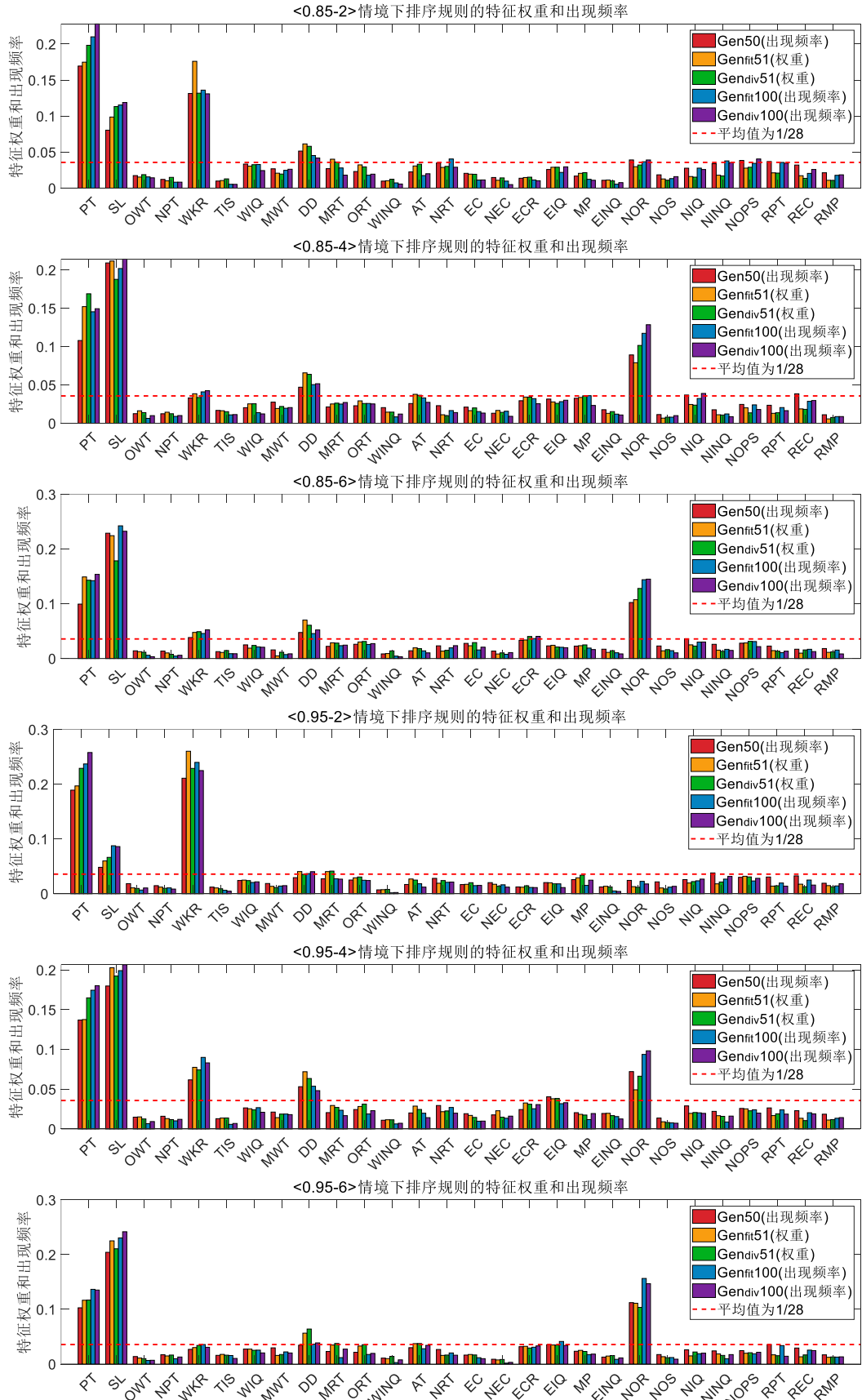


图 4-4 6 种不同情境下排序规则特征的权重和出现频率

Gen_{fit}51 与 Gen_{div}51 的特征权重结果相差不大,说明两种特征权重度量方法具有很强的关联性,它们从不同侧面描述了多目标问题下特征的重要性。这一结论也与表 4-4 和表 4-5 的结论相吻合,即在特征权重度量结果计算相近的情况下,最终的优化效果也趋于相近。

与 Gen50 相比,在 Gen_{fit}100 和 Gen_{div}100 的特征频率结果中,权重较高的频率会更高,而权重较低的概率会更低,说明了新的特征权重集合在后 50 代中确实有在指引算法的搜索方向,所提基于特征权重的种群调整策略也确实帮助算法消除了无关特征的不良影响。

从图 4-3 和图 4-4 中还可以观察到,在所有情境中,大部分特征的权重和频率的结果都很相似。具体而言,PT、MP 和它们的相对性特征 RPT、RMP,以及 WIQ、MWT、NIQ 等特征在路径规则中都表现出较高的重要性。这与实际情况相符,因为 PT、MP、RPT 和 RMP 对优化平均延迟时间和总能耗至关重要,将它们与 WIQ、MWT、NIQ 等特征配合使用,能够更有效且合理地分配设备资源,实现更快速的工件加工,并确保工件会被安排到 MP 较高的设备上从而降低它们的待机能耗。

从排序规则的角度看,PT、SL 和 WKR 的出现频率和权重值都较高。相关研究已表明,这三个特征对于平均延迟时间的优化起到至关重要的作用^[42]。Xu 等人^[64]的研究也表明,这三个特征对于优化能耗目标同样重要。一个有趣的现象是,WKR(工件剩余工作量)与 NOR(工件剩余工件数)在功能上有一定的相关性,都与工件剩余工作有关,观察结果显示出这两个特征之间存在竞争关系,WKR 高时 NOR 低,WKR 低时 NOR 高,当然也可能同时高。这表明不同特征之间本身存在着相关性,也就更加凸显了本文中引入特征权重机制的必要性。

4.5 本章小结

本章节考虑了 GPHH 在生成调度规则时无关特征对调度决策的负面影响和路径、排序调度决策对不同特征信息的需求,提出一种考虑特征权重的双重特征集合 GPHH 算法,并将其运用到面向节能的多目标 DFJSS 问题中,以期自动演化出高质量和可解释的规则。

具体而言,本章研究了 GPHH 中不同特征在演化高质量路径规则和排序规则时的重要性,并提出了两种特征权重度量方法。在种群迭代过程中,基于 Pareto 前沿个体和所提特征权重度量方法评估特征的重要性,以此赋予特征不同的权重,并使用种群调整策略基于新的特征权重集合遗传更新种群。实验结果表明,所提融入特征权重机制的双重特征集合 GPHH 算法相比于传统 GPHH 算法和特征筛选方法在处理多目标问题时能使演化出的调度规则拥有更好的优化性能和可解释性。

第 5 章 总结与展望

5.1 全文总结

本文考虑制造业中对绿色化制造、碳中和、节能减排和可持续性发展等方面的迫切需求,结合超启发式遗传规划算法在求解动态柔性车间调度问题上的先天优势,研究“面向节能的柔性车间动态调度规则自动化生成”方法。该方法在 GPHH 算法框架下,以节能降耗为目的,通过能耗终端符设计和双终端符集合策略,实现实际生产过程中能耗相关目标的快速、有效优化。本文主要完成以下几方面工作:

(1) 针对 GPHH 已有研究中能耗相关终端符的缺失问题,通过柔性车间制造系统能量流分析,识别车间中的耗能主体及其交互逻辑关系、耗能要素及其耗能规律。并基于终端符设计理念将特征信息转化为 GPHH 可用的节能终端符。基于以往人工规则的设计思路,设计了面向能耗目标的人工规则。

(2) 针对路径与排序决策的决策环境差异问题,设计了一种融入特征权重机制的 GPHH 算法,旨在通过特征权重为算法搜索提供指引,提高算法的搜索效率,自动化生成更有潜力和更易理解的调度规则。该方法有效避免了以往特征筛选方法在筛选特征时难以兼顾多目标问题和过度依赖人为设定参数的局限性。

(3) 针对特征权重(即特征重要性)难以度量的问题,提出了两种适用于多目标车间调度问题的特征权重度量方法:一是基于适应度值的特征权重度量方法,二是基于多样性的特征权重度量方法。这两个方法融合了个体非支配关系和 Pareto 前沿的思想,从不同侧面度量单个调度规则个体在整个种群中的重要性,并以此计算各特征的权重。

(4) 以总能耗和平均延迟时间为优化目标,在面向节能的动态柔性车间调度问题上,与现有基于 GPHH 的相关算法进行比较,验证了所提方法的有效性,并深入分析了特征权重机制在优化调度过程中的具体行为和关联影响。

5.2 工作展望

本文主要围绕面向节能的多目标动态柔性车间调度问题展开了一系列研究，但该领域仍然存在着大量未完成的研究工作，有待进一步的探索。具体来说，以下是未来可开展的工作方向：

（1）本文第 3 章对能耗终端符的设计是运用 GPHH 算法解决绿色化车间调度问题的基础和关键，未来还需要针对车间中的各类耗能要素设计融合不同的节能策略，以期从多个层面优化车间能耗，最终实现车间能耗的多层次、多粒度的协同最优化。

（2）本文第 4 章对融入特征权重机制的 GPHH 算法进行了研究，并针对本文所求问题，设计了一种基于双重特征权重集合的 GPHH 算法。在算法优化方面，还可以考虑研究种群变异概率对算法性能的影响，将算法拓展到高维目标问题上，并进一步改进种群调整策略。这些方向可以为算法的性能提升和更广泛的应用提供有益的探索。

参考文献

- [1] 孟磊磊. 面向高效节能的柔性作业车间调度问题建模与优化[D]. 武汉: 华中科技大学, 2020.
- [2] 马晓敏. 《世界能源统计年鉴 2023》中文编译版发布[N]. 中国矿业报, 2023-11-23(002).
- [3] 李岚春, 陈伟, 岳芳, 等. 英国碳中和战略政策体系研究与启示[J]. 中国科学院院刊, 2023, 38(03): 465-476.
- [4] 刘丛丛, 吴建中. 走向碳中和的英国政府及企业低碳政策发展[J]. 国际石油经济, 2021, 29(4): 83-91.
- [5] 郭玉佳, 杨新平, 刘佳驹, 等. 国内外碳达峰城市减排实践与启示[J]. 环境保护, 2022, 50(18): 67-71.
- [6] 程熙琼, 孙娜. 全球能源低碳发展新趋势—访国际能源转型研究会会长、国际能源论坛原秘书长孙贤胜[J]. 国际石油经济, 2022, 30(01): 14-18.
- [7] 国家统计局. 中华人民共和国 2023 年国民经济和社会发展统计公报[N]. 人民日报, 2024-03-01(010).
- [8] 国家统计局. 中国能源统计年鉴 2022[M]. 北京: 中国统计出版社, 2022.
- [9] Jiang B, Ma Y, Chen L, et al. A Review on Intelligent Scheduling and Optimization for Flexible Job Shop [J]. International Journal of Control Automation and Systems, 2023, 21(10): 3127-3150.
- [10] Boudjemline A, Chaudhry I, Rafique A, et al. Multi-Objective Flexible Job Shop Scheduling Using Genetic Algorithms [J]. Tehnicki Vjesnik-Technical Gazette, 2022, 29(5): 1706-1713.
- [11] Zhang F, Mei Y, Nguyen S, et al. Collaborative Multifidelity-Based Surrogate Models for Genetic Programming in Dynamic Flexible Job Shop Scheduling [J]. IEEE Transactions on Cybernetics, 2021, 52(8): 1-15.
- [12] Chen C, Ji Z, Wang Y. NSGA-II Applied to Dynamic Flexible Job Shop Scheduling Problems with Machine Breakdown [J]. Modern Physics Letters B, 2018, 32(34n36): 1840111.

-
- [13] Kan F A, Nu, Fu Z B, et al. A New Approach to Scheduling in Manufacturing for Power Consumption and Carbon Footprint Reduction[J]. Journal of Manufacturing Systems, 2011, 30(4):234-240.
 - [14] He Y, Li Y, Wu T. An Energy-Responsive Optimization Method for Machine Tool Selection and Operation Sequence in Flexible Machining Job Shops [J]. Journal of Cleaner Production, 2015, 87: 245-254.
 - [15] 卫少鹏, 王婷, 周彤. 考虑调整时间的绿色柔性作业车间调度研究[J]. 机电工程, 2021, 38(02): 158-168.
 - [16] 戴敏, 张玉伟, 曾励. 考虑运输时间的绿色作业车间调度问题研究[J]. 工业工程, 2020, 23(02): 41-48.
 - [17] Lei D, Guo X. An Effective Neighborhood Search for Scheduling in Dual-Resource Constrained Interval Job Shop with Environmental Objective [J]. International Journal of Production Economics, 2015, 159: 296-303.
 - [18] 易茜. 面向广义能量效率的数控加工系统优化模型及方法研究[D]. 重庆: 重庆大学, 2015.
 - [19] Xie N, Duan M, Chinnam R B. An Energy Modeling and Evaluation Approach for Machine Tools Using Generalized Stochastic Petri Nets [J]. Journal of Cleaner Production, 2015, 113: 523-531.
 - [20] Brundage M P, Chang Q, Li Y. Sustainable Manufacturing Performance Indicators for a Serial Production Line [J]. IEEE Transactions on Automation Science & Engineering, 2016, 13(2): 676-687.
 - [21] Thiede S, Seow Y, Andersson J. Environmental Aspects in Manufacturing System Modelling and Simulation-State of the Art and Research Perspectives [J]. Cirp Journal of Manufacturing Science & Technology, 2013, 6(1): 78-87.
 - [22] Lunardi W, Birgin E, Laborie P, at al. Mixed Integer Linear Programming and Constraint Programming Models for the Online Printing Shop Scheduling Problem [J]. Computers & Operations Research, 2020, 123: 105020.
 - [23] Gmys J, Mezmaz M, Melab N, at al. A Computationally Efficient Branch-and-Bound Algorithm for the Permutation Flow-Shop Scheduling Problem [J].

- European Journal of Operational Research, 2020, 284(3): 814-833.
- [24] Lin S, Ying K. Minimising Makespan in Job-Shops with Deterministic Machine Availability Constraints [J]. International Journal of Production Research, 2021, 59(14): 4403-4415.
- [25] Peng B, Lu Z, Cheng T. A Tabu Search/Path Relinking Algorithm to Solve the Job Shop Scheduling Problem [J]. Computers & Operations Research, 2015, 53: 154-164.
- [26] Abdel-Kader R. An Improved PSO Algorithm with Genetic and Neighborhood-Based Diversity Operators for the Job Shop Scheduling Problem [J]. Applied Artificial Intelligence, 2018, 32(5): 433-462.
- [27] Wang B, Xie H, Xia X, et al. A NSGA-II Algorithm Hybridizing Local Simulated-Annealing Operators for a Bi-Criteria Robust Job-Shop Scheduling Problem Under Scenarios [J]. IEEE Transactions on Fuzzy Systems, 2019, 27(5): 1075-1084.
- [28] Wu X, Sun Y. A Green Scheduling Algorithm for Flexible Job Shop with Energy-Saving Measures [J]. Journal of Cleaner Production, 2018, 172: 3249-3264.
- [29] Liu Q, Pan Q, Gao L, et al. Multi-Objective Flexible Job Shop Scheduling Problem Considering Machine Switching Off-On Operation [J]. Procedia Manufacturing, 2019, 39: 1167-1176.
- [30] Wang Y, Leng, J, Liu X, et al. A Method for Dynamic Insertion Order Scheduling in Flexible Job Shops Based on Digital Twins [J]. Applied Sciences-Basel, 2022 ,12(23): 12430.
- [31] Tao N, Ming H, Xu L, et al. A Novel Dynamic Scheduling Strategy for Solving Flexible Job-Shop Problems[J]. Journal of Ambient Intelligence and Humanized Computing, 2016, 7(5): 721-729.
- [32] Wen X, Lian X, Qian Y, et al. Dynamic Scheduling Method for Integrated Process Planning and Scheduling Problem with Machine Fault[J]. Robotics and Computer-Integrated Manufacturing, 2022, 77: 102334.
- [33] Durasevic M, Jakobovic D. A Survey of Dispatching Rules for the Dynamic Unrelated Machines Environment [J]. Expert Systems with Applications, 2018, 113:

555-569

- [34]Zahmani M, Atmani B. Multiple Dispatching Rules Allocation in Real Time Using Data Mining, Genetic Algorithms, and Simulation [J]. Journal of Scheduling, 2020, 24(2): 175-196.
- [35]Panwalkar S, Iskander W. A Survey of Scheduling Rules [J]. Operations Research, 1997, 25(1): 45-61.
- [36]Burke E, Gendreau M, Hyde M, et al. Hyper-heuristics: A Survey of the State of the Art [J]. Journal of the Operational Research Society, 2013, 64(12): 1695-1724.
- [37]Zhang F, Mei Y, Nguyen S, et al. Correlation Coefficient-Based Recombinative Guidance for Genetic Programming Hyper-heuristics in Dynamic Flexible Job Shop Scheduling [J]. IEEE Transactions on Evolutionary Computation, 2021, 25(3): 552-566.
- [38]Fan H, Xiong H, Goh M. Genetic Programming-Based Hyper-Heuristic Approach for Solving Dynamic Job Shop Scheduling Problem with Extended Technical Precedence Constraints [J]. Computers & Operations Research, 2021, 134, 105401.
- [39]Park J, Mei Y, Nguyen S. An Investigation of Ensemble Combination Schemes for Genetic Programming Based Hyper-Heuristic Approaches to Dynamic Job Shop Scheduling [J]. Applied Soft Computing, 2018, 63: 72-86.
- [40]Zhou Y, Yang J, Huang Z. Automatic Design of Scheduling Policies for Dynamic Flexible Job Shop Scheduling Via Surrogate-Assisted Cooperative Co-Evolution Genetic Programming [J]. International Journal of Production Research, 2020, 58(9): 2561-2580.
- [41]Rosca J, Ballard, J. Genetic Programming with Adaptive Representations [M]. Technical Report TR489, University of Rochester, Computer Science Department, Rochester, NY, USA, 1994, pp.1-30.
- [42]Mei Y, Nguyen S, Xue B, et al. An Efficient Feature Selection Algorithm for Evolving Job Shop Scheduling Rules with Genetic Programming [J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2017, 1(5): 339-353.

- [43]Sitahong A, Yuan Y, Li M, et al. Learning Dispatching Rules Via Novel Genetic Programming with Feature Selection in Energy-Aware Dynamic Job-Shop Scheduling [J]. Scientific Reports, 2023, 13(1): 8558.
- [44]Friedlander A, Neshatian K, Zhang M. Meta-Learning and Feature Ranking Using Genetic Programming for Classification: Variable Terminal Weighting [C]. In Proceedings of the 2011 IEEE Congress of Evolutionary Computation (CEC), New Orleans, LA, USA, June 5-8, 2011, pp.941-948.
- [45]Mei Y, Zhang M, Nyugen S. Feature Selection in Evolving Job Shop Dispatching Rules with Genetic Programming [C]. In Genetic and Evolutionary Computation Conference, 2016, Denver Colorado, USA, July 20-24, pp.365-372.
- [46]Zhang F, Mei Y, Zhang M. A Two-stage Genetic Programming Hyper-Heuristic Approach with Feature Selection for Dynamic Flexible Job Shop Scheduling [C]. In Genetic and Evolutionary Computation Conference, Prague, Czech Republic, July 13-17, 2019, pp.347-355.
- [47]Zhang F, Mei Y, Nguyen S, Zhang M. Evolving Scheduling Heuristics Via Genetic Programming with Feature Selection in Dynamic Flexible Job-Shop Scheduling [J]. IEEE Transactions on Cybernetics, 2020, 51(4): 1797-1811.
- [48]Xu B, Mei Y, Wang Y, et al. Genetic Programming with Delayed Routing for Multi-Objective Dynamic Flexible Job Shop Scheduling [J]. Evolutionary Computation, 2021, 29(1): 75-105.
- [49]雷德明, 杨冬婧. 具有总能耗约束的柔性作业车间调度问题研究[J]. 自动化学报, 2018, 44(11): 2083-2091.
- [50]Sels V, Gheysen, N, Vanhoucke M. A Comparison of Priority Rules for the Job Shop Scheduling Problem under Different Flow Time- and Tardiness-Related Objective Functions [J]. International Journal of Production Research, 2012,50(1): 4255-4270.
- [51]Branke J, Nguyen S, Pickardt C, et al. Automated Design of Production Scheduling Heuristics: A Review [J]. IEEE Transactions on Evolutionary Computation, 2016, 20(1): 110-124.

- [52]Deb K, Pratap A, Agarwal S, et al. A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II [J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.
- [53]Schweim D, Wittenberg D, Rothlauf F. On Sampling Error in Genetic Programming [J]. Natural Computing, 2022, 21(2): 173-186.
- [54]张苏雨, 王艳, 纪志成. 基于超启发式遗传规划的动态车间调度方法[J]. 系统仿真学报, 2020, 32(12): 2494-2506.
- [55]Shady S, Kaihara T, Fujii N, et al. A Novel Feature Selection for Evolving Compact Dispatching Rules Using Genetic Programming for Dynamic Job Shop Scheduling [J]. International Journal of Production Research, 2022, 60(13): 4025-4048.
- [56]Zhang F, Mei Y, Zhang M. Genetic Programming with Multi-Tree Representation for Dynamic Flexible Job Shop Scheduling [C]. Proceedings of the 31st Australasian Joint Conference on Artificial Intelligence (AI), Wellington, New Zealand, Dec.11-14, 2018, pp.472-484.
- [57]Ishibuchi H, Imada R, Setoguchi Y, et al. How to Specify a Reference Point in Hypervolume Calculation for Fair Performance Comparison [J]. Evolutionary Computation, 2018, 26(3): 1-29.
- [58]Zeitrag Y, Figueira J, Horta N, et al. Surrogate-Assisted Automatic Evolving of Dispatching Rules for Multi-Objective Dynamic Job Shop Scheduling Using Genetic Programming [J]. Expert Systems with Applications, 2022, 209: 118194.
- [59]Nguyen S, Zhang M, Johnston M, et al. Automatic Design of Scheduling Policies for Dynamic Multi-Objective Job Shop Scheduling Via Cooperative Coevolution Genetic Programming [J]. IEEE Transactions on Evolutionary Computation, 2014, 18(2): 193-208.
- [60]Wilcoxon F. Individual Comparisons by Ranking Methods[J]. Biometrics Bulletin, 1945, 1(6): 80-83.
- [61]Tay J, Ho N. Evolving Dispatching Rules Using Genetic Programming for Solving Multi-Objective Flexible Job-Shop Problems [J]. Computers & Industrial Engineering, 2008, 54(3): 453-473.

- [62]Nguyen S, Mei Y, Xue B, et al. A Hybrid Genetic Programming Algorithm for Automated Design of Dispatching Rules [J]. *Evolutionary Computation*, 2019, 27(3): 467-496.
- [63]Xu B, Tao L, Deng X, et al. An Evolved Dispatching Rule Based Scheduling Approach for Solving DJSS Problem [C]. 2021 40th Chinese Control Conference, Shanghai, China, July 26-28, 2021, pp.6524-6531.
- [64]徐彬梓. 基于知识的离散制造系统能效优化方法研究[D]. 无锡: 江南大学, 2019.

攻读硕士期间取得的学术成果

发表的论文

- [1] **Fei Baolin**, Xu Binzi*, Huang Dengchao, Zhang Yao, Wang Chun, Yang Long. Automatic Generation of Energy-Efficient Dispatching Rules for Dynamic Flexible Job Shop Scheduling [C]. 2023 China Automation Congress (CAC), Chongqin, China, Nov. 17-19, 2023, pp. 533-538 (EI 会议, 已检索, 对应本文第 3 章内容)
- [2] Xu Binzi*, Xu kai, **Fei Baolin**, Huang Dengchao, Tao Liang, Wang Yan. Automatic Design of Energy-Efficient Dispatching Rules for Multi-Objective Dynamic Flexible Job Shop Scheduling Based on Dual Feature Weight Sets [J]. Mathematics, 2024, 12(10), 1463. (中科院三区, 已录用, 对应本文第 4 章内容)

参与的项目

- [1] 芜湖市科技项目应用基础研究项目(参与, 项目编号: 2023jc05): 面向动态柔性制造系统的多源能耗集成式协同优化。
- [2] 高等学校自然科学研究项目重点项目(参与, 项目编号: KJ2020A0345): 基于多粒度协同的离散制造系统能耗智能优化方法研究。
- [3] 检测技术与节能装置安徽省重点实验室开放基金项目(参与, 项目编号: DTESD2020B01): 融合实时工况信息的节能工艺路线知识智能推送方法研究。

致谢

时光匆匆，三年的研究生生活即将画上句号，回首过去的点滴经历，心头涌上一阵感慨。这段时光里，无数美好瞬间交织成丰富的回忆，结交了珍贵的同窗之谊，学到了专业的知识，历经实践的磨练，这将成为我生命中难以忘怀的一段旅程。在此，我要向曾经帮助过我的每一位致以深深的谢意！

首先，衷心感谢我的导师徐彬梓老师，能够在研究生阶段成为徐老师的第一个学生是我人生中的幸运之事。三年来，不论是在学业上还是生活上，徐老师都给予了我无微不至的指导与关怀。从论文选题到完成，每一个细节的处理，徐老师都付出了大量心血。徐老师的勤勉与严谨治学的态度深深影响着我，是我终身受益的人生财富！

其次，感谢室友储曦、李浩宇和黄跃祖一直以来在生活和学业上对我的支持与帮助；感谢师弟师妹们王梦娇、曹新宇、许凯以及实验室的同伴们张陈、严回回、王陈曦、王博、王煜等，在研究生时光里共同度过的欢笑时光。很庆幸能认识大家，愿大家都有一个快乐而充实的生活，愿大家前程似锦！

最后，感谢我的父母和家人，在我追求学业的道路上，你们一直是我坚实的后盾和最坚定的支持者，也是你们无私的关爱成就了如今的我，感谢你们！