

# 中国科学技术大学

# 硕士学位论文



## 多部件系统的质量-维护-调度联合优化 研究

作者姓名： 陈明

学科专业： 控制科学与工程

导师姓名： 康宇 教授 赵云波 教授

完成时间： 二〇二四年五月二十四日



University of Science and Technology of China  
A dissertation for master's degree



# **Research on Joint Optimization of Quality-Maintenance-Scheduling in Multi-component Systems**

Author: Ming Chen

Speciality: Control Science and Engineering

Supervisors: Prof. Yu Kang, Prof. Yun-Bo Zhao

Finished time: May 24, 2024



## 中国科学技术大学学位论文原创性声明

本人声明所呈交的学位论文，是本人在导师指导下进行研究工作所取得的成果。除已特别加以标注和致谢的地方外，论文中不包含任何他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的贡献均已在论文中作了明确的说明。

作者签名： 陈明

签字日期： 2024.5.25

## 中国科学技术大学学位论文授权使用声明

作为申请学位的条件之一，学位论文著作权拥有者授权中国科学技术大学拥有学位论文的部分使用权，即：学校有权按有关规定向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅，可以将学位论文编入《中国学位论文全文数据库》等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、汇编学位论文。本人提交的电子文档的内容和纸质论文的内容相一致。

控阅的学位论文在解密后也遵守此规定。

公开  控阅 (\_\_\_\_ 年)

作者签名： 陈明

导师签名： 秦宇 志诚

签字日期： 2024.5.25

签字日期： 2024.5.25



## 摘 要

随着生产系统的复杂度不断提升，多部件系统逐渐成为现代生产制造过程的核心，其生产管理研究引起了学术界和产业界的广泛关注。在此背景下，产品质量、维护决策和生产调度是生产过程中的三个主要因素，且彼此间存在紧密的相互作用。系统中任一部件故障会导致生产停机，影响生产计划的执行；在生产调度过程中需要考虑维护时机，以便在计划的生产停机时间进行维护，减少维护产生的停机成本；通过维护可以减少系统状态退化造成的质量损失，但频繁的维护会增加维护成本。由于三者之间相互影响，同时优化质量、维护成本和交货期这三个关键目标面临挑战。

针对以上问题，论文以多部件系统为研究对象，将产品质量退化过程纳入到维护决策中，考虑维护与调度之间的相关性，在强化学习和遗传算法框架下，实现质量、维护与调度的联合优化，主要研究内容包括：

(1) 针对固定生产计划下多部件系统的产品质量和维护决策联合优化问题，将维护决策建模为马尔可夫决策过程，引入当前生产批次剩余时间的概念，在奖励函数中加入质量损失以及与生产计划相关的停机成本，使用深度强化学习对决策问题求解，实现降低维护成本的同时减少质量损失，并能够根据生产计划变化动态调整维护策略。

(2) 针对多部件系统的产品质量、维护决策和生产调度联合优化问题，设计了一种两阶段联合优化方法。第一阶段以维护决策为主，在维护中考虑生产计划以及产品质量，利用强化学习实现能够根据生产计划变化动态调整的维护策略。第二阶段利用第一阶段得到的维护策略，计算生产过程中的维护时间、维护成本和质量损失，使用遗传算法求解联合优化模型，实现减少延误成本的同时减少维护成本和质量损失。

(3) 针对生产计划扰动下并行多机系统的动态调度问题，设计了基于周期和事件混合驱动策略，减少静态调度随时间累积的误差以及扰动事件对生产带来的影响。并针对不同扰动事件分别设计了相应的调度修复策略，实现了处理生产计划扰动的同时保持生产过程的稳定性。

**关键词：**产品质量，维护决策，生产调度，联合优化，多部件，强化学习，遗传算法，动态调度

## ABSTRACT

With the rising complexity of production systems, multi-component systems have progressively taken center stage in modern manufacturing processes, prompting a surge of interest in the research of their production management. In this context, product quality, maintenance decisions, and production scheduling serve as critical, interdependent factors within the production process. Any component failure can trigger production downtime, disrupting the execution of established plans. Conducting maintenance during scheduled production pauses effectively minimizes downtime costs. Although maintenance can mitigate quality losses resulting from system degradation, excessive maintenance may conversely escalate overall costs. Given the intricate interplay among these factors, simultaneously optimizing quality, maintenance costs, and delivery times proves to be a challenging endeavor.

In addressing the aforementioned challenges, this thesis focuses on multi-component systems, integrating the phenomenon of product quality degradation into maintenance decision-making while considering the interrelation between maintenance and scheduling activities. Under the framework of reinforcement learning and genetic algorithms, joint optimization of quality, maintenance, and scheduling is implemented. The main content includes:

(1) For the joint optimization of product quality and maintenance decisions within a multi-component system operating under a fixed production plan, maintenance decisions are modeled using Markov decision processes. The concept of remaining time within the current production batch is incorporated, and the reward function incorporates quality losses and downtime costs related to the production plan. Deep reinforcement learning is employed to solve this decision problem, effectively reducing maintenance costs while minimizing quality losses. Moreover, it facilitates the dynamic adjustment of maintenance strategies in response to fluctuations in the production plan.

(2) A two-stage joint optimization methodology was developed for the concurrent optimization of product quality, maintenance, and production scheduling within multi-component systems. In the first stage, the emphasis is placed on maintenance, taking into account production plans and product quality. Reinforcement learning is utilized to devise maintenance strategies that can adapt dynamically to variations in production plans. In the second stage, the maintenance strategy derived from the first stage serves as the basis for calculating maintenance time, maintenance costs, and quality losses

throughout the production cycle. Genetic algorithm is employed to solve the joint optimization model with the objective of concurrently minimizing delay costs, maintenance costs, and quality losses.

(3) To address the dynamic scheduling challenges faced by parallel multi-machine systems under the influence of production plan disturbances, a scheduling recovery method grounded in hybrid-driven scheduling is proposed. This strategy is designed to mitigate the cumulative errors inherent in static scheduling over time and to counteract the disruptive effects of various disturbance events on production. Additionally, corresponding scheduling repair strategies are tailored for different types of disturbance events, enabling the handling of production plan disturbances while maintaining the stability of the production process.

**Key Words:** Product quality; maintenance optimization; production scheduling; joint optimization; multi-component; reinforcement learning; genetic algorithm; dynamic scheduling



## 目 录

第 1 章 绪论	1
1.1 研究背景和意义	1
1.2 研究现状	3
1.2.1 多部件系统的维护决策研究现状	3
1.2.2 产品质量、维护决策与生产调度联合优化研究现状	4
1.2.3 动态调度研究现状	6
1.3 研究内容和结构安排	7
1.3.1 研究内容	7
1.3.2 结构安排	8
第 2 章 相关基础知识	10
2.1 多部件系统视情维护	10
2.2 强化学习	11
2.2.1 强化学习简介	12
2.2.2 马尔可夫决策过程	12
2.2.3 强化学习原理	14
2.2.4 Q-learning 算法	16
2.3 遗传算法	17
第 3 章 固定生产计划下多部件系统的产品质量和维护决策联合优化方法	20
3.1 引言	20
3.2 考虑生产计划和产品质量的维护决策建模	21
3.2.1 退化过程建模	22
3.2.2 质量损失建模	23
3.2.3 维护决策过程	24
3.3 基于深度强化学习的维护决策方法	25
3.3.1 多部件系统维护决策	25
3.3.2 深度强化学习算法	27
3.3.3 Restricted-DQN 算法	28
3.4 实验结果及分析	29
3.4.1 实验设置	29
3.4.2 三部件系统	30

3.4.3 敏感性分析	33
3.5 本章小结	34
第4章 多部件系统的产品质量、维护决策和生产调度联合优化方法	35
4.1 引言	35
4.2 问题描述与假设	37
4.3 产品质量、维护决策与生产调度联合优化方法	37
4.3.1 联合优化模型建模	37
4.3.2 两阶段联合优化方法	40
4.4 模型求解	41
4.4.1 染色体编码设计	42
4.4.2 遗传算法过程	42
4.5 实验结果及分析	44
4.5.1 实验设置	44
4.5.2 实验结果	45
4.6 本章小结	47
第5章 生产计划扰动下并行多机系统的动态调度方法	48
5.1 引言	48
5.2 问题建模	49
5.3 并行多机系统维护与生产静态调度	51
5.4 并行多机系统维护与生产动态调度	52
5.4.1 重调度驱动机制设计	53
5.4.2 重调度方法设计	54
5.4.3 动态调度流程	56
5.5 实验结果及分析	57
5.6 本章小结	61
第6章 总结与展望	62
6.1 工作总结	62
6.2 研究展望	62
参考文献	64
致谢	71
在读期间发表的学术论文与取得的研究成果	72

## 插图清单

图 1.1	论文结构安排	9
图 2.1	P-F 曲线	11
图 2.2	强化学习框架	12
图 2.3	Q-learning 学习过程	16
图 2.4	遗传算法流程	18
图 3.1	Gamma 过程示意图	22
图 3.2	维护决策流程图	25
图 3.3	当前生产批次剩余时间	26
图 3.4	三个部件的退化过程	30
图 3.5	场景 1 中不同维护策略结果	31
图 3.6	场景 2 中不同维护策略结果	32
图 3.7	场景 3 中不同维护策略结果	32
图 4.1	不同功能部门间的冲突	36
图 4.2	基于阈值的机会性维护示意图	38
图 4.3	两阶段联合优化方法流程图	40
图 4.4	基于强化学习的维护和生产调度联合优化模型染色体编码	42
图 4.5	基于阈值的机会性维护和生产调度联合优化模型染色体编码	42
图 4.6	交叉操作示意图	43
图 4.7	五个部件的退化过程示意图	44
图 5.1	并行多机调度染色体编码示意图	51
图 5.2	扰动因素分类	52
图 5.3	动态调度流程图	56
图 5.4	静态调度甘特图	57
图 5.5	动态调度甘特图	58
图 5.6	交货期变更扰动调度修复策略甘特图	59
图 5.7	交货期变更扰动完全重调度策略甘特图	59
图 5.8	订单取消扰动调度修复策略甘特图	60
图 5.9	订单取消扰动完全重调度策略甘特图	60
图 5.10	订单增加扰动调度修复策略甘特图	60
图 5.11	订单增加扰动完全重调度策略甘特图	61



## 表格清单

表 3.1	三个部件退化过程的参数	30
表 3.2	不同策略下的成本	31
表 3.3	成本的均值和方差	33
表 3.4	五部件系统不同维护策略的成本均值和方差	33
表 3.5	八部件系统不同维护策略的成本均值和方差	33
表 3.6	十部件系统不同维护策略的成本均值和方差	34
表 3.7	长时间跨度下成本	34
表 4.1	五部件退化过程参数	44
表 4.2	五部件系统维护成本设置	45
表 4.3	作业相关信息	45
表 4.4	不同优化方法结果	46
表 4.5	大规模作业优化方法结果	46
表 5.1	初始调度作业相关信息	57
表 5.2	无扰动下静态调度与动态调度结果	58
表 5.3	交货期变更、订单撤销和订单增加下静态调度结果	58
表 5.4	交货期变更扰动下调度修复策略与完全重调度策略结果	59
表 5.5	订单取消扰动下调度修复策略与完全重调度策略结果	59
表 5.6	订单增加扰动下调度修复策略与完全重调度策略结果	60



## 符号说明

$m$	部件数量
$d_t$	部件 $i$ 的退化状态
$H_i$	部件 $i$ 的失效阈值
$D(t)$	系统的退化状态
$\alpha_i$	部件 $i$ 的形状参数
$\beta_i$	部件 $i$ 的尺度参数
$K(t)$	产品质量特征偏差
$c_q$	产品质量损失
$c_m$	维护成本
$c_f$	操作成本, 包括零部件更换成本和人力成本
$c_d$	维护所造成的停机成本
$\tau_t$	当前批次的剩余时间
$x_{ij}$	单机调度决策变量
$n$	待加工作业数量
$p_i$	作业 $i$ 加工时长
$td_i$	作业 $i$ 截止时间
$w_i$	作业 $i$ 惩罚权重
$T_m$	设置时间
$C_{[i]}$	调度顺序为 $i$ 的完成时间
$H_i^p$	部件 $i$ 的预防性维护阈值
$H_i^o$	部件 $i$ 的机会性维护阈值
$l$	并行机器数量
$y_{ijk}$	多机调度决策变量
$c_{[ik]}$	第 $k$ 台机器上的调度顺序为 $i$ 的作业的完成时



# 第1章 绪 论

## 1.1 研究背景和意义

制造业作为国民经济发展的关键支柱行业，其发展水平深刻影响国家综合国力的提升，是驱动国家走向繁荣富强的核心力量之一。2015年中国推出《中国制造2025》国家战略，体现着我国对制造业发展的高度重视与坚定决心。近十年来，得益于这一政策的引导和支持，我国制造业经历了快速的发展，实现了规模扩张和技术突破。然而，在全球一体化和产业变革浪潮下，制造业的竞争也日益激烈，尤其智能工厂、精益生产及服务型制造等新型模式对制造企业提出了更高要求。如何通过技术创新和管理优化，实现系统的可靠运行、高效维护及灵活调度，以最小的成本损耗和高效的生产效率满足市场需求，成为了我国制造业的关键课题。

伴随工业技术的日臻成熟，各种生产系统的复杂性不断提升，而结构化和模块化的设计理念普遍运用于复杂系统中，使其能够从功能层面和结构层面上被合理地拆解为多个关键部件<sup>[1]</sup>。例如工业机器人由机械臂、伺服电机、控制器等多个关键部件组成；装配线生产系统中，每个部件代表负责不同生产过程的生产设备。相比于单部件系统，多部件系统中的部件数量和部件种类更多，且部件之间存在复杂的相关性，任一部件发生故障都会影响整个生产系统的正常运行。因此，有效管理多部件系统对于实现智能制造目标至关重要。

不论是单部件系统还是多部件系统，产品质量、维护决策和生产调度都是生产过程的三个主要因素：

1. 在当今激烈的市场竞争环境下，消费者对产品的期望值不断攀升，追求个性化、定制化和高品质体验成为主流趋势。这意味着企业在追求市场份额和盈利能力的同时，必须将提高产品质量作为其核心战略之一。任何程度的质量问题，无论是微小的性能不足还是重大的安全隐患，都有可能导​​致消费者的信任度下降、满意度滑坡，进而影响企业的口碑和市场地位。对于制造企业提升财务和运营绩效，质量管理至关重要<sup>[2]</sup>。借助于智能制造技术和先进的质量管理体系，通过对生产系统的全过程监控和精准调控，有望实现产品质量的持续改进与优化。
2. 随着设备工作运行，其性能不可避免地会发生退化直至故障。设备的突然故障会带来高昂的维护成本和停机成本，直接影响企业的经济效益。停机期间，生产线停滞不仅意味着直接产出的损失，还可能引发一系列后续问题，如交货延迟等间接经济损失。此外，随着制造系统的集成化程度不断提高，各个设备和关键部件之间的相互依赖性也随之增强，一旦某一环节

出现故障,就可能迅速蔓延至整个系统,导致大面积停机甚至瘫痪,由此产生的经济损失往往是巨大的。研究显示,维护成本占到总生产成本的15%到40%<sup>[3]</sup>。有效的维护决策可以显著降低因设备故障、停机而导致的经济损失,延长资产使用寿命,从而为企业节省大量运维成本。

3. 生产调度是企业针对生产计划进行调控与执行的过程,它在大多数制造业中起着重要作用<sup>[4]</sup>。由于人力资源、生产设备等的有限性,在同一时间企业无法满足多样化的生产需求。因此需要对企业的生产过程进行合理有效的调度规划,依据资源状况和市场需求,组织和协调生产作业顺序、资源负荷,以确保生产流程的顺畅,保证准时交货。此外,生产调度还能灵活应对各类不确定性因素,如市场需求波动、设备故障等,迅速调整生产节奏以保持生产系统的稳定运行。

在生产过程中,产品质量、维护决策与生产调度这三大因素是相互关联的,但现有的生产理论模型大都没有对三者同时考虑。产品质量不仅仅取决于初始的设计与制造工艺,还与设备的维护状态密切相关<sup>[5]</sup>。良好的维护决策能够确保设备保持高效稳定的运行状态,保证生产计划的正常执行。设备维护的时机又受到生产调度安排的影响,合适的生产计划能够为设备维护提供充足的窗口期,避免因维护活动对生产连续性造成过大干扰。可见,在生产流程中,质量、维护及调度形成了一个相互依存、彼此制约的关系网。如何在实际生产运作中妥善协调三者之间的动态平衡,从而最大化整体生产效率和经济效益,是当前制造业的重要课题。

本课题的研究意义主要有以下几个方面:

- (1) 为了减少单个部件的故障导致多部件系统整体停机带来的损失,建立多部件系统的退化模型,在制定维护决策时考虑了不同部件之间的相关性,采用深度强化学习方法给出多部件系统的机会性维护策略,减少了维护成本。

- (2) 在维护决策过程中考虑了生产计划以及部件之间的相关性,并将设备退化导致的产品质量损失加入优化目标,实现在降低维护成本的同时减少质量损失以及对生产计划的影响。

- (3) 将产品质量、维护决策与生产调度联合优化,在生产调度中考虑了维护策略对生产过程的影响,解决了设备故障导致停机无法执行生产计划的问题。在优化过程中考虑产品质量、维护决策与生产调度之间的约束关系,兼顾维护成本、质量损失和交货期,实现整体效益的最优。

## 1.2 研究现状

在多部件系统的产品质量、维护决策和生产调度的联合优化研究中,需要考虑以下问题:多部件系统的不同部件之间存在复杂的相关性,任一部件故障会导致整个系统失效,这给维护决策带来挑战;产品质量、维护决策与生产调度这三个因素相互关联,优化过程需要同时考虑三个因素以提升生产系统整体收益;生产过程中的动态扰动事件会使得原本的维护和生产计划不符合实际生产过程,需要制定能够处理扰动情况的动态调度方案。针对以上问题下文分别给出相应的研究现状。

### 1.2.1 多部件系统的维护决策研究现状

针对多部件系统, Tian 等<sup>[6]</sup>提出了一个基于比例风险模型的多部件系统维护优化模型。若某个部件的故障率超过一级阈值,则该部件将被预防性更换;而当其他部件的故障率超过二级阈值时,这些部件会一起更换。Zhu 等<sup>[7]</sup>采用周期性的维护策略,在维护期间,对那些退化程度超过相应限制的部件进行集体维护以节省设置成本。Poppe 等<sup>[8]</sup>针对部件的退化水平设置了两个阈值,一个阈值用于决定在何时与其它部件一同维护该部件;另一个阈值则用于判断何时为了防止失效而对该部件进行维护,类似地, Bakir 等<sup>[9]</sup>建立了风力发电厂的双阈值机会维护模型。Biao 等<sup>[10]</sup>在预防性维护被触发时,引入基于成本的改进因子来识别预防性维护组的重要性排名,然后确定预防性维护的一组机器。Dong 等<sup>[11]</sup>研究了多组件系统的最优机会维护方案,建立了考虑故障依赖性的多部件系统的故障率模型和可靠性评估模型,确定每个单独部件的预防性维护计划,根据机会性维护时的可靠性阈值,建立多部件系统的二维保修成本模型。

这些维护策略通常通过设置健康指标的阈值来实现机会性维护,通常在建立数学模型后通过网格搜索来确定各分量的最佳维护阈值。然而,许多工业设备,例如发动机和机床,通常由大量部件组成,并且这些部件通常会随着时间的推移而老化<sup>[12]</sup>。因此,找到最优的维护阈值变得困难,系统状态和部件维护决策的空间将呈几何级数增加。幸运的是,近年来,深度强化学习方法为我们利用人工智能解决问题提供了新的思路。深度强化学习结合了深度学习的特征提取能力和强化学习的智能决策优势,是对顺序决策问题进行建模的强大工具。在与环境模型交互生成的数据的激励下,智能体能够利用神经网络感知退化状态,输出相应维护行为的评价值,并选择最优的维护行为。通过不断优化,智能体无需任何专业知识即可模拟人类活动。该方法超越了传统维护方法需要计算和设置最优维护阈值的限制,使得在复杂系统中实施基于状态的维护成为可能。

在基于强化学习的维护策略中, Rocchetta 等<sup>[13]</sup>利用强化学习为具有退化元素并配备预测和健康管理能力的电网寻找最佳运营和维护策略,强化学习的动

作是功率输出的水平以及每台发电机的维护操作。Huang 等<sup>[14]</sup>提出了一种基于深度强化学习的方法,以最小化总体维护成本优化串行生产线的维护策略。强化学习动作指示何时以及哪台机器应该维护。Wei 等<sup>[15]</sup>提出了一种带有卷积神经网络的深度强化学习框架,以获得桥梁的最佳维护策略,通过最小化维护成本来使其保持在可接受的状态。利用卷积神经网络进行状态特征学习,利用全连接网络进行 Q 函数逼近。强化学习行动分为四个离散级别:无修复、小修复、大修复和更换。每个组件都被分配了这些维护级别之一。Yousefi 等<sup>[16]</sup>利用 Q-learning 根据组件当前的寿命和退化状态找到最佳维护操作。他们通过串联、并联和串并联组合三种不同的系统配置来评估所提出的方法,每个系统中包含三个组件。Hu 等<sup>[17]</sup>首先将飞机维护问题转换为强化学习框架,然后开发基于极限学习机的 Q 学习的强化学习驱动的维护策略。维护决策是根据任务概况、飞机剩余使用寿命和可用备件数量做出的。Andriotis 等<sup>[18]</sup>采用部分可观测马尔可夫决策过程在不完善的观测和约束下对随机、非平稳、10 分量恶化系统进行检查和维护规划。Yang 等<sup>[19]</sup>引入强化学习算法,通过考虑处理成本、维护成本和完成奖励来最小化长期预期平均奖励。Andriotisa 等<sup>[18]</sup>提出了一种基于 DRL 的随机最优控制框架,用于在不完整信息和约束下运行的恶化系统的检查和维护计划。部件退化被离散成有限的状态,并且状态之间的转移矩阵是固定的。Zhang 等<sup>[12]</sup>将随机依赖和经济依赖引入到多部件系统中,提出了以部件退化值作为输入的深度强化学习的基于状态的维护模型。Hao 等<sup>[20]</sup>研究了大状态和行动空间下多部件系统的维护策略和备件库存联合优化问题,使用马尔可夫决策过程对其进行建模,并提出一种基于随机策略和行动者批评家框架的改进深度强化学习算法。

总结来说,目前多部件系统的维护方法主要为基于阈值的机会性维护,但随着系统内部件数量的增长,寻找最优的维护阈值会变得困难。尽管基于强化学习的维护技术已显示出在处理多部件系统复杂性方面的潜力,但目前的研究往往只考虑了部件之间的相关性,并没有充分探究系统退化状态对产品质量的潜在影响,以及生产计划对维护决策的影响。

### 1.2.2 产品质量、维护决策与生产调度联合优化研究现状

在制造业体系中,生产设备的性能衰减和可靠性下滑至需维护状态时,其产出的产品品质往往呈现显著滑坡,从而加剧不合格产品的生成率<sup>[21]</sup>。鉴于此,众多研究者已着手将产品质量退化融入维护策略的决策框架中,致力于构建兼顾产品质量与维护决策的整合优化模型。

通过统计过程控制与预防性维护的结合,有学者从控制图、验收抽样等不同方面研究了维护优化与质量改进的结合<sup>[22-24]</sup>。然而,这些研究并未解决机器或系统退化对质量偏差影响的根本机制。Jin 等<sup>[25]</sup>提出了状态空间模型来建立生

产过程中产品特征偏差的形成机制。在此基础上, Jin 等<sup>[26]</sup>提出了 QR-Co-Effect 方法来分析产品质量偏差与部件的可靠性之间的关系。Chen 等<sup>[27]</sup>基于响应面模型<sup>[28]</sup>和田口损失函数<sup>[29]</sup>, 进一步研究了生产系统的产品质量退化模型。

设备的维护决策过程通常以提高设备的可靠性为目标制定维护计划, 在计划的维护时间会对设备进行停机维护, 但这会影响正常的生产计划的正常执行。相反, 生产调度通常以提高生产效率为目标制定生产计划, 因此会充分利用设备来满足生产需求, 但设备长期工作会导致可靠性降低。由于维护决策与生产调度之间的相互关系, 只有通过集成维护决策与生产调度这两个因素的模型, 才能得到全局最优的方案。目前, 维护决策与生产调度联合优化的模型可以分为两大类: 相关模型与集成模型<sup>[30]</sup>。

相关模型在考虑某一个因素时将另一个因素视为约束条件。因此相关模型具体有考虑给定维护策略的生产调度模型与考虑给定生产计划的维护决策模型。考虑给定维护策略的生产调度模型例如朱传军等<sup>[31]</sup>以开放式车间为研究对象, 分别讨论了基于故障率的预防性维护与周期性维护的生产调度问题, 并将禁忌搜索算法与遗传算法相结合用于模型求解。Li 等<sup>[32]</sup>针对考虑维护活动的柔性作业车间调度问题, 构建了以设备最大完工时间、关键设备负荷、总负荷为优化目标的多目标数学模型。宋文佳等<sup>[33]</sup>以确定的预防性维护可靠度阈值为基础, 以优化最大完工时间、生产成本以及平均总维护成本为目标, 建立了柔性作业车间生产调度问题的数学模型。Sheikhalishahi 等<sup>[34]</sup>针对考虑人为错误和预防性维护的开放车间调度问题, 开发了三种元启发式方法来寻找近最优解。Lu 等<sup>[35]</sup>研究了在所有作业完成前必须对一台机器进行一次维护的不相关并行机调度问题。Cheng 等<sup>[36]</sup>同时探讨了单台机器上退化作业与维护活动的调度问题, 在这种情况下, 维护取决于起始时间。

考虑生产的维护决策模型例如周元辉等<sup>[37]</sup>以采用固定周期预防维护的三设备串行生产系统为研究对象, 在组合优化的基础上, 建立一种基于设备可靠性的动态机会维护策略。Hao 等<sup>[38]</sup>专注于动态调整分配给机器的工作负载以保持系统可靠性。为了降低运营成本, Sun 等<sup>[39]</sup>提出了双部件系统的作业重新分配策略。Xia 等<sup>[40]</sup>在具有退化机器的多单元批量生产系统中提出了交互式双层维护策略。在机器级调度中, 使用多属性模型根据单个机器的退化情况来获取维护间隔。在系统级调度中, 考虑机器退化和批量生产的特点, 开发了一种新颖的生产驱动机会维护策略。Van 等<sup>[41]</sup>引入了一种基于状态的维护模型, 该模型考虑了生产系统的维护成本和生产能力。Liu<sup>[42]</sup>等针对动态需求下多部件批量生产系统的维护问题, 为减少停机次数和维护成本, 提出了一种三阶段的维护决策框架。

集成模型将维护决策和生产调度视为同等重要, 在建模时同时考虑二者的优化目标及约束条件。陆志强等<sup>[43]</sup>构建了生产批次可分的批量流调度与预防性

维护的联合优化模型。Xu 等<sup>[44]</sup>研究了单机生产计划和预防性维护调度的集成优化。预防性维护可以按照固定或灵活的维护方案进行。固定维护方案中连续两次维护活动之间的时间间隔是恒定的，而灵活方案中可以提前进行维护。完工时间和总流程时间都被视为调度期间的性能指标。Wang 等<sup>[21]</sup>提出基于每台机器预测故障概率的预测维护策略和基于目标服务水平生产控制策略，以满足每个周期的动态随机需求。开发了生产、质量和维护的集成模型，以最大限度地降低总成本。Ghaleb 等<sup>[45]</sup>考虑了机器的性能会随着时间的推移而恶化，并且机器会遭受基于随机恶化的故障。他们假设机器退化状态之间的转换遵循指数分布，认为加工时间和能源消耗受到机器老化和故障的影响。Salih 等<sup>[46]</sup>开发了一种模型，可以同时集成和优化单台机器的生产、维护和过程控制决策。预防性维护和纠正性维护这两种类型都被考虑在内。该方法首先制定最佳的预防性维护计划。然后开发一个集成模型，确定决策变量并优化生产调度、库存持有和维护以及过程控制产生的每单位时间的总成本。Liu 等<sup>[47]</sup>考虑了系统的退化状态和虚拟年龄。他们建立了一个总金额最小的综合模型，以最小化期望成本为目标。Cui 等<sup>[48]</sup>考虑了故障不确定性的影响，提出了一种以最小化质量鲁棒性为优化目标，融合生产调度和维护研究的梯度下降算法。Paprocka 等<sup>[49]</sup>提出了一种创新的生产调度和维护方法，重点关注车间调度问题，并建立了鲁棒性最大化的模型。Sharifi 等<sup>[50]</sup>研究了并行机生产环境中生产计划和维护计划的联合优化，机器的可靠性被建模为多状态系统，其中引入两个恶化阈值来启动维护并防止基于恶化的故障。提出了一种混合整数规划模型，通过最小化包括预期延迟、维护和能源消耗成本的加权和目标函数来联合优化生产计划和维护计划。一些研究<sup>[51-54]</sup>建立了流水车间和柔性作业车间的生产调度和维护的集成模型。

总结来说，目前大部分产品质量、维护决策和生产调度联合优化研究主要考虑维护对生产调度的影响，很少对三者同时研究，且很少考虑生产调度对维护决策的影响，如生产批次转换时的设置时间产生的维护机会。且过去的联合优化文献通常假设机器加工过程不能中断，也即不允许在生产作业批次工作时进行维护，这种假设容易造成设备意外故障或过度维护造成浪费。

### 1.2.3 动态调度研究现状

生产调度问题的研究策略大致分为两大类：静态调度与动态调度<sup>[55]</sup>。静态调度方法建立在所有作业信息和车间状态事先完全确定的基础上，一旦制定出详尽的调度计划，便严格按照预定计划执行生产流程。动态调度方法将制造系统视为一个持续演进的动态系统，要求系统具备即时应对各种动态事件的能力，如交货期变更、订单撤销、订单增加、生产设备的突发故障及其维修等情况。相较于静态调度，动态调度方法更能贴合实际情况，然而也因此增加了问题的复杂

性。

当前处理动态调度问题的方法主要有三种：完全反应式调度、鲁棒性调度以及预-反应式调度<sup>[56]</sup>。完全反应式调度，也称为在线调度，采用优先级调度规则进行局部且实时的决策，而非依赖预生成的确定性计划<sup>[57]</sup>。然而，这种方法在做调度决策时并未考虑这些决策对后续批次的影响。鲁棒性调度关注于在动态环境中生成能够可预见地满足性能要求的调度计划，但确定其可预见性的度量较为困难。目前普遍使用的方法是预-反应式调度，又称重调度，该方法根据动态事件调整原有计划。例如，Nasiri 等<sup>[58]</sup>提出了一种到达不确定性的周期性预-反应式调度系统。该系统由两个主要部分组成：重新调度优化模型和短间隔修复策略。Tighazoui 等<sup>[59]</sup>考虑了一个相同的并行机器随着时间的推移而重新调度作业。基于预-反应式调度策略，开发了混合整数线性规划模型以及处理在线部分的迭代方法。Li 等<sup>[60]</sup>研究了动态柔性作业车间调度问题，并设计了一种基于蒙特卡罗树搜索算法的重新调度方法，其目标是 minimized 完工时间。Wang 等<sup>[61]</sup>考虑如何重新调度随机到达的新作业以追求作业车间调度的性能和稳定性的问题。建立混合整数规划模型，以最小化三个目标，包括加工过程中新作业的不连续率、初始计划的完工时间偏差和机器上的顺序偏差。Melissa 等<sup>[62]</sup>针对动态柔性作业车间调度问题提出了一种启发式模型，目标是在任务发生任何变化后实现接近最佳的调度，以最大限度地减少最大完成时间。在所提出的模型中，首先根据估计的处理时间进行调度，然后在考虑机器设置确定处理时间后执行重新调度。Gao 等<sup>[63]</sup>针对考虑机器修复的柔性作业车间重调度问题，以尽量减少不稳定性和完工时间为目标，通过在原始 Jaya 算法中引入两个新颖的局部搜索算子，提出了一种改进的 Jaya 算法。

总结来说，目前大多数动态调度研究对于动态事件的选取以设备故障和订单变化为主，其中设备故障被认为是最常见的扰动事件并被假设作为随机扰动发生。而通过监测设备的状态和退化过程，以视情维护与生产调度集成优化的静态调度为基础的动态调度方法相对欠缺。

## 1.3 研究内容和结构安排

### 1.3.1 研究内容

论文以多部件系统为研究对象，考虑了设备退化对产品质量的影响，以及维护决策与生产调度之间的约束关系，提出了产品质量、维护决策和生产调度联合优化模型，并根据模型特点，设计相应的优化算法进行求解。具体的研究内容主要包含：

- (1) 固定生产计划下多部件系统的产品质量和维护决策联合优化方法

针对固定生产计划下多部件系统的产品质量和维护决策联合优化问题，首先对部件的退化过程进行建模，并分析了多部件系统退化状态与产品质量之间的关系，以及生产计划对维护决策的影响。然后提出了一种基于强化学习的维护决策方法，重新构建了决策模型，引入当前生产批次剩余时间的概念。此外，还在奖励函数中加入了与健康状态相关联的质量损失以及与生产计划相关的停机成本。进一步地，结合深度强化学习方法与基于阈值的机会性维护，提出了Restricted-DQN方法，以适应实际生产环境。最后通过实验将所提方法与传统的基于阈值的机会性维护方法进行对比。

### (2) 多部件系统的产品质量、维护决策和生产调度联合优化方法

针对多部件系统的产品质量、维护决策和生产调度联合优化问题，设计了一种两阶段联合优化方法。第一阶段以维护决策为主，在维护中考虑生产计划以及产品质量，利用强化学习实现能够根据生产计划变化动态调整的维护策略。第二阶段利用第一阶段得到的维护策略，计算生产过程中的维护时间、维护成本和质量损失，在调度模型中计算质量损失、维护成本和延误成本。根据优化模型设计相应的染色体编码，利用遗传算法对优化问题求解。最后通过实验将联合优化模型与单独优化方法的效果进行对比。

### (3) 生产计划扰动下并行多机系统的动态调度方法

针对生产计划扰动下并行多机系统的动态调度问题，首先对并行多机系统的动态调度问题建模，并在模型中考虑了交货期变更、订单撤销和订单增加这些扰动事件。将产品质量、维护决策和生产调度的联合优化方案作为静态调度方法，以最小化质量损失、维护成本和延误成本调度目标，建立动态调度模型。然后分析了不同的调度驱动机制，设计了基于周期和事件混合驱动策略以解决静态调度随时间累积的误差以及扰动事件对生产带来的影响。进一步分析了两种不同的重调度方法，并针对三种不同的扰动事件分别设计了相应的调度修复策略。最后通过实验将所提出的动态调度方法与静态调度方法进行比较。

## 1.3.2 结构安排

全文共六章，论文结构如图1.1，具体描述如下：

第1章绪论。介绍了论文的研究背景和意义，以及多部件系统产品质量、维护决策与生产调度联合优化的相关研究现状，最后给出全文的研究内容与结构安排。

第2章相关基础知识。描述了全文研究所涉及到的相关理论知识，以及后续研究内容所涉及到的基础算法。

第3章固定生产计划下多部件系统的产品质量和维护决策联合优化方法。对部件的退化过程进行建模，并分析了多部件系统退化状态与产品质量之间的关

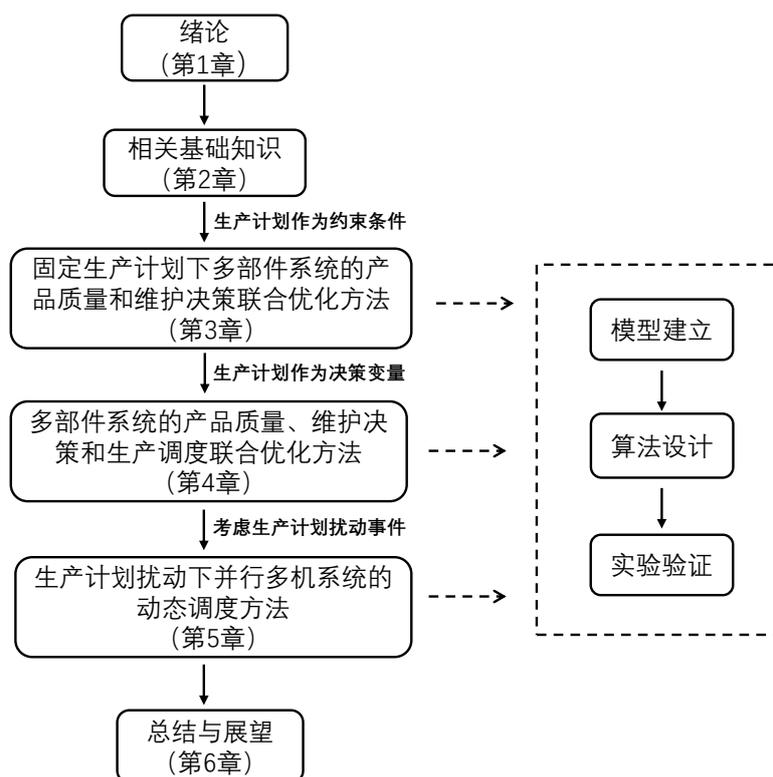


图 1.1 论文结构安排

系，以及生产计划对维护决策的影响。将决策过程建模为马尔可夫决策过程，并利用强化学习方法求解，最后进行相关实验验证。

第 4 章多部件系统的产品质量、维护决策和生产调度联合优化方法。对维护决策与生产调度的联合优化问题进行建模，并设计了两阶段联合优化方法和相应求解算法，最后进行相关实验验证。

第 5 章生产计划扰动下并行多机系统的动态调度方法。分析了实际生产环境中可能的扰动因素，对维护与生产动态调度问题建模，设计了基于周期和事件混合驱动策略，并针对三种主要的扰动事件分别设计了相应的调度修复策略，最后进行相关实验验证。

第 6 章总结与展望。对论文所完成的工作进行总结，并提出了论文的不足及进一步的研究展望。

## 第2章 相关基础知识

### 2.1 多部件系统视情维护

在现代制造生产中，设备是生产过程中不可或缺的重要组成部分。然而，设备的频繁故障和停机会导致生产中断和成本增加，降低生产效率，对企业的竞争力造成严重影响。目前主流的维护策略主要包括三大类型：一是事后维护 (Corrective Maintenance, CM)，仅在设备部件发生故障后才进行维护修理，日常运行阶段不做主动干预；二是定期维护 (Time-Based Maintenance, TBM)，按照预定的时间间隔对部件实施维护，通常依据平均故障间隔来设定维护周期；三是视情维护 (Condition-Based Maintenance, CBM)，主要是利用先进的状态监测系统实时监测和收集设备部件的运行状态数据，预测其未来的性能衰退趋势，进而采取精准适宜的维护措施。

CBM 和 TBM 本质上都是预防性维护策略，但两者有所不同。TBM 主要依据设备的历史平均故障时间来设定维护周期；而 CBM 则侧重于实时监测设备的工作状态，能够迅速掌握设备存在的问题并据此采取针对性的维护措施。由于只在必要时进行维护，CBM 可以避免不必要的过度维护，从而节约维护成本和资源消耗。同时，通过实时监测设备状态，可以有效避免设备的意外故障，减少非计划性停机。

CBM 依赖于这样的假设：大多数故障不会立即发生，并且可以在恶化过程的早期阶段被检测到。主要挑战是确定应该执行维护的确切时刻并确定最合适的操作。P-F 曲线可用于表示设备随时间的变化状况<sup>[64]</sup>，如图 2.1所示，其中 P 点表示为潜在故障点，F 点表示为功能故障点。维护时机在 P 点之后，可以有效利用设备或者部件的寿命；维护的时机在 F 点之前，可以避免设备或者部件出现意外故障导致生产系统停机。视情维护的基本思想就在 P 点与 F 点之间查找一个合适的时间点，以平衡维护成本和故障导致的停机成本。

随着生产系统的复杂性不断增加，系统往往由多个关键部件组成，单一部件的故障可能会导致整个系统的停机和生产中断，对企业的影响显著。系统内部各部件之间存在复杂的关系，倘若简单将各个部件的维护需求机械叠加，很难达成预期的理想维护效果。事实上，部件之间的相关性主要分为经济相关性、结构相关性和随机相关性。经济相关性是指在系统中对某个部件进行维护时，对其他退化严重的部件同时维护可以降低系统的总体维护成本；结构相关性适用于在结构上形成连接集的组件，此时维护一个组件需要拆卸其他组件；随机相关性是指系统内的某一个部件退化过程会对系统中的其他部件的退化过程产生影响。

针对多部件系统，目前主要有三种维护策略：批量更换策略、成组维护策略

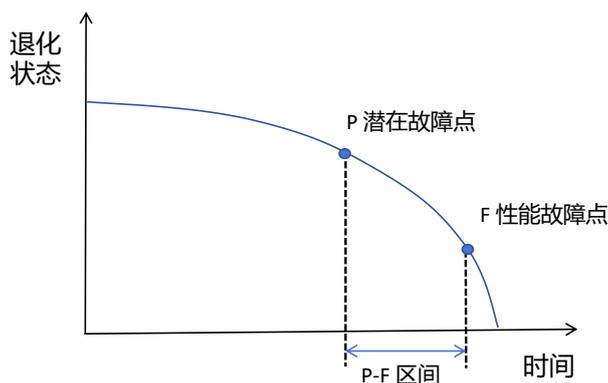


图 2.1 P-F 曲线

和机会性维护策略。

(1) 批量更换策略是指所有部件仅允许在预设的固定替换时刻进行集体更换<sup>[65]</sup>。一旦有部件在某一固定替换时间点之前发生故障，将在该时间点立即对其进行替换，或直至下一指定批量更换时机到来再统一进行替换。

(2) 成组维护策略是指将系统内的部件按照特定属性或特性进行归类分组，当系统内任一部件出现故障，或当某一组内的故障部件数量累积至设定阈值时，将对该组的所有部件一并进行维护操作。

(3) 机会性维护策略是指在对某部件维护时，充分考量系统内各部件间的经济效益关联性，借由已发生的停机时机，以较低的成本趁势对其他已显著衰退的设备进行预防性维护。

相较于批量更换和成组维护这两种策略，机会性维护策略更侧重于关注系统内各部件的实际损耗程度，并能够在出现适宜的维护时机时，针对已达维护标准的部件同步进行维护安排，可以更有效地降低维护成本和停机次数，减少成本。

## 2.2 强化学习

在面对维护决策这一复杂挑战时，强化学习（Reinforcement Learning, RL）展现出了独特的适用性和潜力。传统维护策略往往基于预设规则或历史数据统计分析，难以灵活应对多变的系统状态和环境条件，而强化学习则通过动态的学习过程，使维护决策系统能在不断与环境互动中自我优化，寻求长期利益最大化。强化学习的核心理念与维护决策的实质需求不谋而合。它将维护决策问题构建成智能体与环境的交互过程，其中智能体代表维护决策系统，环境则涵盖了系统的运行状态、维护成本、设备性能衰减等多种因素。智能体通过执行不同的维护动作（如定期检查、部件更换等），从环境中接收反馈（奖励或惩罚），这一过程直观地模拟了现实世界中维护活动。

### 2.2.1 强化学习简介

强化学习是关于解决序贯决策问题，其目标是最大化一些奖励。当考虑现实世界的问题时，例如玩游戏或优化维护策略，人们可以轻松地在脑海中构建问题。在这种游戏过程中，需要采取一系列行动才能取得胜利。同样的想法可以应用于维护策略的优化，其中需要提高资产可用性，同时降低成本。人类通过利用可用信息并选择可以推断为最有益的行动来解决此类问题。强化学习在解决此类问题方面具有巨大潜力，因为它模仿了人类的决策过程。

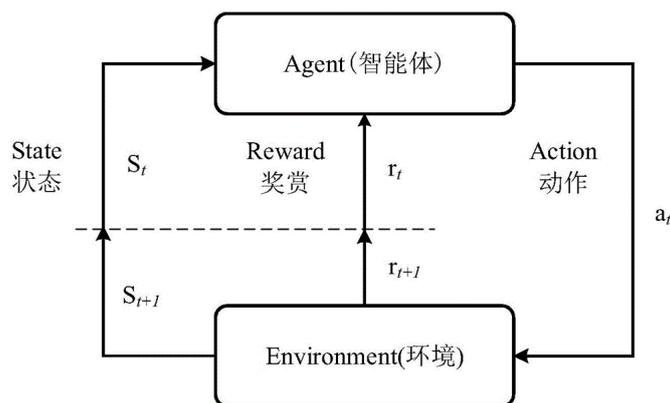


图 2.2 强化学习框架

如图 2.2所示，智能体和环境是强化学习的两个核心要素。环境是智能体看到并与之交互的可观察世界的表示。当环境在每个交互步骤发生变化时，智能体会观察状态的变化，并选择在环境中采取的动作，这反过来会导致环境发生变化。智能体从环境中接收奖励信号，该信号强化了在当前环境状态的情况下动作的好坏。智能体的目标是学习最大化其累积未来奖励的决策策略。

强化学习的核心思想可以理解为外界环境通过对个体奖励或惩罚来激励个体，使其对外界进行探索，不断进行各种尝试，逐渐输出对不同环境刺激下的预期值，产生能获得最大回报或者最小代价的行动。在智能体与具体环境的交互过程中，如果智能体选择的动作能往达成目标的趋势发展，则该动作会提高一定的优先级，即后面再产生这个行为的趋势更加强烈，反之则趋势减弱。

### 2.2.2 马尔可夫决策过程

马尔可夫决策过程 (Markov Decision Process, MDP) 为求解强化学习问题提供了一个数学框架。马尔可夫过程是指状态转移过程满足马尔可夫性质的随机过程，其中马尔可夫性质指随机过程中的下一时刻状态的条件概率分布，仅取决于当前时刻的状态，并不依赖过去所有的状态。马尔可夫决策过程在每个时刻增加动作变量，此时下一时刻的状态取决于当前时刻的状态以及动作。一个马尔可夫决策过程由一个四元组  $(S, A, P, R)$  定义，其中每个元素的含义如下：

- $S$  为系统所有可能的状态的集合；
- $A$  为系统所有可能的动作的集合；
- $P$  为状态转移概率， $P(s_{t+1}|s_t, a_t)$  表示在  $t$  时刻状态  $s_t$  下采取动作  $a_t$  的情况下在  $t$  时刻系统状态转移到状态  $s_{t+1}$  的概率，其符合马尔可夫性质，即：

$$P(s_{t+1}|s_t, a_t, s_{t-1}, a_{t-1}, \dots, s_0, a_0) = P(s_{t+1}|s_t, a_t) \quad (2.1)$$

- $R$  为奖励函数， $r(s_t, a_t)$  表示在  $t$  时刻，机器在状态  $s_t$  下执行动作  $a_t$  所获得的奖励。

上述的 MDP 模型是假设环境完全可观察的，即系统总能观察到自身的状态。如果关于环境的一些信息对状态隐藏，则环境被部分观察，此时可以将问题建模成部分可观察马尔可夫过程 (Partially Observable Markov Decision Process, POMDP)。

根据环境的不同，动作可以是离散的，也可以是连续的。系统具有一组有效的动作，称为动作空间。在离散动作空间中，只能执行有限数量的动作，例如在国际象棋中，只能对规则允许的特定棋子采取行动。在连续动作空间中，动作可以是实值向量，例如在控制车辆的转向角、油门位置和制动器的环境中，动作可以是连续向量。

在强化学习中，并不知道转移概率函数  $P$  和奖励函数  $R$ ，只能采取行动并观察结果状态和奖励。为了通过简单地观察状态和奖励来学习采取最佳行动，需要学习策略。当给定当前状态  $s_t$ ，用来决定采取什么动作  $a_t$  的决策规则集称为策略，可以用  $\pi$  表示：

$$a_t \sim \pi(a_t|s_t) \quad (2.2)$$

在深度强化学习中，策略是通过函数来计算的，这些函数基于所使用的神经网络的权重和偏置进行参数化。神经网络的参数可以被更新以修改策略的行为。为了表示参数化的策略函数，通常使用符号  $\theta$  作为下标：

$$a_t \sim \pi_\theta(a_t|s_t) \quad (2.3)$$

来自环境的一系列状态和动作称为轨迹  $\tau = (s_0, a_0, s_1, a_1, \dots)$ 。如果一条轨迹包含了从第一步到最后一步的所有状态和动作，它也被称为一个回合 (episode)。在强化学习中，当讨论算法性能指标时经常会提到回合，而在谈论算法训练步骤或者基于收集到的数据进行训练的过程时，则更多地会提及轨迹。需要注意的是，在训练过程中使用的数据可能并不一定包含从开始到结束的完整序列。

奖励  $r_t = r(s_t, a_t)$  也可以称为回报，智能体的目标是最大化轨迹上的累积未

来奖励  $R(\tau)$ 。有限范围内的未折扣回报是在固定步数上获得的奖励总和：

$$R(\tau) = \sum_{t=0}^T r_t \quad (2.4)$$

但无限运行的持续任务无法使用的回报公式(2.4)。因为  $T = \infty$ ，要最大化的回报很容易是无限大。这里引入折扣的概念，无限时间折扣回报是指未来获得的所有奖励之和，这些奖励会按照一个因子  $\gamma \in (0, 1)$  进行折扣处理，这个因子的作用是强调近期获得的奖励相对于未来获得的奖励的重要性。

$$R(\tau) = \sum_{t=0}^T \gamma^t r_t \quad (2.5)$$

通常情况下，会选择  $\gamma$  小于 1 的值，以便优先考虑有助于强化学习算法收敛的短期奖励。 $\gamma$  的值越小，对未来的奖励给予的权重就越轻，从而使得奖励更加偏向短期。而当  $\gamma$  接近 1 时，则意味着对未来奖励给予了更多的权重。

### 2.2.3 强化学习原理

强化学习本质上是利用累计回报来学习，最终学会最优策略来最大化整体的回报。但是由于状态转移都具有一定的随机性，所以算法优化的目标函数不应该是一条轨迹的累计折扣回报，而应该是所有可能轨迹的期望回报  $J(\pi)$ ，即：

$$J(\pi) = \mathbb{E}_{\tau \sim \pi} [R(\tau)] \quad (2.6)$$

一条轨迹的概率分布为：

$$P(\tau|\pi) = \rho_0(s_0) \prod_{t=0}^{T-1} P(s_{t+1}|s_t, a_t) \pi(a_t|s_t) \quad (2.7)$$

其中  $\rho_0(s_0)$  代表初始状态分布，因此期望回报可以表示为：

$$J(\pi) = \sum_{\tau} P(\tau|\pi) R(\tau) \quad (2.8)$$

最优策略  $\pi^*$  为：

$$\pi^* = \arg \max_{\pi} J(\pi) \quad (2.9)$$

一些强化学习算法利用值函数来计算状态或状态-动作对的值。从这个意义上说，价值是指智能体从特定状态或状态-动作对开始，并在后续根据特定策略采取行动的预期回报。

状态值函数  $V^\pi(s)$  代表如果智能体以状态  $s$  开始并始终从策略  $\pi$  中选择动作得到的预期回报：

$$V^\pi(s) = \mathbb{E}_{\tau \sim \pi} [R(\tau) | s_0 = s] \quad (2.10)$$

动作-状态值函数  $Q^\pi(s, a)$  代表如果智能体以状态  $s$ 、动作  $a$  开始, 并始终从策略  $\pi$  中选择动作得到的预期回报：

$$Q^\pi(s, a) = \mathbb{E}_{\tau \sim \pi} [R(\tau) | s_0 = s, a_0 = a] \quad (2.11)$$

最优状态值函数  $Q^*(s)$  代表如果智能体以状态  $s$ 、动作  $a$  开始, 并始终从最优策略中选择动作得到的预期回报：

$$V^*(s) = \max_{\pi} \mathbb{E}_{\tau \sim \pi} [R(\tau) | s_0 = s] \quad (2.12)$$

最优动作-状态值函数  $Q^*(s, a)$  代表如果智能体以状态  $s$ 、动作  $a$  开始, 并始终从最优策略中选择动作得到的预期回报：

$$Q^*(s, a) = \max_{\pi} \mathbb{E}_{\tau \sim \pi} [R(\tau) | s_0 = s, a_0 = a] \quad (2.13)$$

贝尔曼方程用于用已知状态的价值来表达未知状态的价值。通过使用贝尔曼方程, 当前状态的价值可以根据未来状态的价值递归定义, 迭代求解最优策略从而变得可能。

状态值函数和动作-状态值函数的贝尔曼方程为：

$$V^\pi(s) = \mathbb{E}_{a \sim \pi, s_{t+1} \sim P} [r(s, a) + \gamma V^\pi(s_{t+1})] \quad (2.14)$$

$$Q^\pi(s, a) = \mathbb{E}_{s_{t+1} \sim P} [r(s, a) + \gamma \mathbb{E}_{a_{t+1} \sim \pi} [Q^\pi(s_{t+1}, a_{t+1})]] \quad (2.15)$$

最优状态值函数和最优动作-状态值函数的贝尔曼方程为：

$$V^*(s) = \mathbb{E}_{a \sim \pi, s_{t+1} \sim P} [r(s, a) + \gamma V^*(s_{t+1})] \quad (2.16)$$

$$Q^*(s, a) = \mathbb{E}_{s_{t+1} \sim P} [r(s, a) + \gamma \mathbb{E}_{a_{t+1} \sim \pi} [Q^*(s_{t+1}, a_{t+1})]] \quad (2.17)$$

对于上述最优贝尔曼方程, 强化学习有以下三种基本解法：

(1) 动态规划法：

动态规划法基于预先给定的模型参数, 包括详尽的状态转移概率函数以及明确的奖励函数机制, 运用贝尔曼方程逐步递归地求解出最优价值函数。然而, 这种方法的实施依赖于对模型全面精确的认知, 且一旦状态空间规模急剧扩张时, 其计算复杂度将呈现指数级增长, 大大降低了算法的实际运行效率。

## (2) 蒙特卡洛法:

蒙特卡洛法摆脱了对状态转移函数和奖励函数精确预知的需求，转而采取与环境互动的方式，通过不断试错和探索，积累丰富的实践经验数据，从而推断出马尔可夫决策过程中的最优策略。尽管该方法在理论上具有较强的通用性，但其实践过程中存在的明显短板在于，机器只能在完成一条完整的采样轨迹后方可进行学习更新，这无疑降低了学习效率。此外，由于数据主要来源于随机抽样，故而方差较大，可能影响到算法性能的稳定性和最终收敛效果。

## (3) 时序差分法:

时序差分法成功地融合了动态规划严谨的理论框架与蒙特卡洛法的实践经验，它能够在每经历一个或多个时间步长后，灵活运用贝尔曼方程进行即时的价值评估更新。不同于传统的累积平均估值方式，时序差分法采用增量计算的方式来逐步逼近真实值函数。其中，Q-learning 算法是时序差分法的典型代表。

## 2.2.4 Q-learning 算法

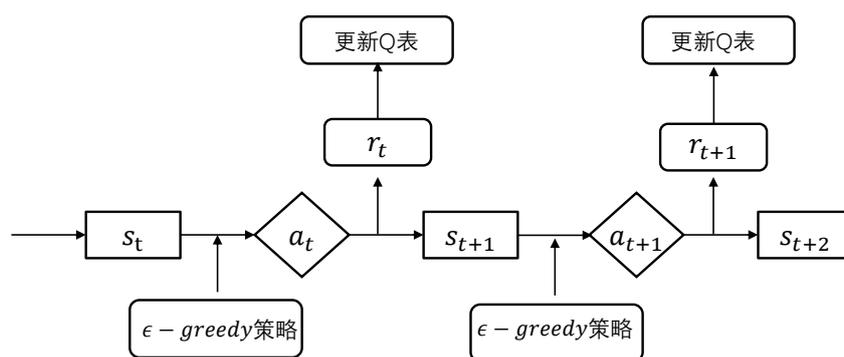


图 2.3 Q-learning 学习过程

作为时序差分学习方法的代表算法，Q-learning 算法是一种建立在价值函数基础之上的强化学习算法。它的核心目标在于指导智能体在环境中通过不断地试错探索，学习各种不同行为动作所带来的结果。智能体借助一种名为 Q-table 的二维表格结构，用来存储和更新每一个状态-动作对  $(s, a)$  所对应的预期未来收益，即动作-状态值函数  $Q(s, a)$ 。在实际运作过程中，智能体会遍历所有可能遇到的状态，并针对每个状态尝试执行各种不同的动作。每当智能体在新的状态下完成一次动作尝试，并收到相应的奖励信号及下一状态信息后，它就会依据一定的更新规则，实时调整 Q-table 中的对应项值，以体现最新学到的经验知识。最终，经过无数次的自我迭代优化，智能体能在不同的外界环境下做出使得累计奖励最大化的动作选择。Q-learning 学习过程见图 2.3，具体的学习流程如下：

## • 步骤 (1): 初始化

初始化一个 Q-table (也称为 Q 函数表或动作-状态值函数表)，该表是一个

二维数组，行表示所有可能的状态，列表示每个状态下可执行的所有动作。将 Q-table 中的所有初始值设置为 0 或其他预设值。

- 步骤 (2): 循环迭代学习阶段

对于每一个训练回合 (episode):

1. 初始化状态: 从环境的某个起始状态开始。
2. 选择动作: 根据当前状态, 根据某种策略选择下一个动作。常用的策略有  $\epsilon$ -greedy 策略, 即以概率  $\epsilon$  随机选择一个动作, 以概率  $1 - \epsilon$  选择当前 Q-table 中对应状态下的最优动作 (即具有最大 Q 值的动作)。
3. 执行动作并观察结果: 在环境中执行选定动作, 并接收新状态以及相应的即时奖励。
4. 更新 Q 值: 使用贝尔曼期望方程更新 Q-table 中的 Q 值:

$$Q_{new}(s_t, a_t) = Q(s_t, a_t) + \alpha * (r_t + \gamma * \max(Q(s_{t+1}, a)) - Q(s_t, a_t))$$

其中:

- $Q_{new}(s_t, a_t)$  是当前状态  $s_t$  下执行动作  $a_t$  的更新 Q 值。
- $\alpha$  是学习率, 控制了新经验对旧 Q 值的影响程度。
- $r_t$  是在执行动作  $a_t$  后立即获得的即时奖励。
- $\gamma$  是折扣因子, 决定了对未来奖励的重视程度。
- $\max(Q(s_{t+1}, a))$  是下一状态  $s_{t+1}$  下所有可能动作的最大 Q 值估计。

5. 状态转移: 将当前状态更新为新状态  $s_{t+1}$ , 继续进行下一轮迭代直至达到终止状态或者达到预先设定的最大步数限制。

- 步骤 (3): 重复步骤 (2)

在多个 episode 中不断迭代这个过程, 随着学习的推进, Q-table 会逐渐收敛到代表最优策略的状态-动作值。

- 步骤 (4): 策略提取

当 Q-table 收敛时, 可以从中提取最优策略, 即对于任意状态  $s$ , 选择具有最高 Q 值的动作  $a$  作为最优动作。

## 2.3 遗传算法

遗传算法 (Genetic Algorithm, GA) 作为一种基于自然选择和遗传学原理的全局优化技术, 为解决生产调度问题提供了强大而灵活的工具。生产调度面临资源分配、任务排序、时间约束等多重挑战, 其求解空间巨大且往往带有复杂的约束条件。遗传算法正是通过模拟生物进化过程中的遗传、突变和自然选择机制, 有效探索这些高维度、非线性问题的解决方案。

遗传算法在计算机环境中模拟了自然界中的生物遗传系统，模仿生物遗传学中的一系列进化操作，保留适应性强的个体，淘汰适应性弱的个体。在求解优化问题时，将问题抽象成编码，对应生物遗传学上的染色体，不同位置的染色体编码控制优化问题不同的优化变量。再通过选择、交叉、变异操作，对染色体进行进化，使得适应度高的基因遗传到下一代，适应度低的染色体被淘汰。

相较于传统的数学解析手段，遗传算法的独到优势在于其能够直接依据适应度函数值来评估候选解的优劣，在追求全局最优解的过程中，并不苛求对目标函数有详尽严谨的数学解析。遗传算法凭借其灵动的编码策略和群体智能搜索机制，能够在多个方向上探索潜在的最优解集。另外，遗传算法的编码方式可以根据不同问题的具体需求进行个性化设计，这一特点使它具备了解决多种复杂问题类型的普适性和灵活性。遗传算法流程图见图 2.4，具体流程如下：

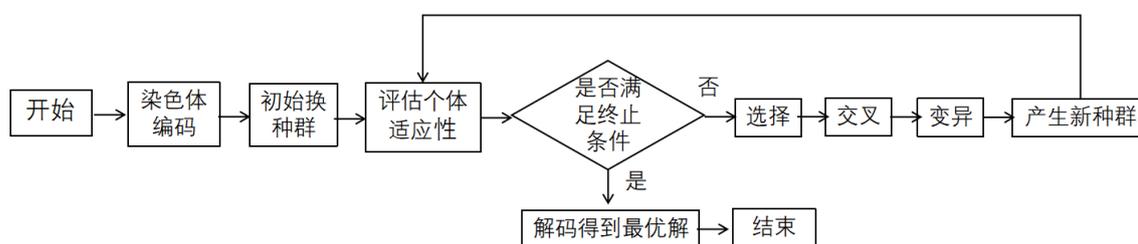


图 2.4 遗传算法流程

- 步骤 (1)：染色体编码。遗传算法无法直接操作问题的实际参数，因而采用对问题所有可能解的编码转换策略。不同的编码方法将在整个算法运行中显著影响计算效率及最终优化结果的质量。针对性地为各类问题设计编码方案有助于简化问题复杂度，提高求解效率。
- 步骤 (2)：初始化种群。在确立编码规则后，每一个编码实例代表一条独特的染色体，若干条染色体组合成为一个初始种群，形成第一代个体集合。
- 步骤 (3)：计算个体适应度值。在遗传算法框架下，适应度是指某个体携带特定基因组时传递至后代的能力指标，其数值大小直接决定了个体在生存竞争中繁衍下一代的概率。
- 步骤 (4)：检查是否已满足预设的终止条件，若当前已找到满意解，则输出最优解并终止算法；反之，继续执行后续步骤。
- 步骤 (5)：选择。基于适应度评估结果，挑选出种群中表现优越的个体作为父代参与下一代的生成。常见的选择策略如轮盘赌选择法，该方法通过对种群内个体适应度进行标准化处理，随后按适应度对应的概率选取父代个体，适应度越高的个体越有可能被选中。
- 步骤 (6)：交叉。运用各种交叉算子对所选父代的染色体结构进行重组或

交换，由此产生新的子代染色体，丰富解空间。

- 步骤 (7): 变异。在种群内的个别个体基因位点上实施变异操作，导致染色体编码发生改变。返回步骤 (3)。

## 第3章 固定生产计划下多部件系统的产品质量和维护决策联合优化方法

本章提出了固定生产计划下多部件系统的产品质量和维护决策联合优化方法。考虑了系统退化对产品质量的影响、生产计划对维护决策的约束以及多部件系统部件之间的相关性，将决策过程建模为马尔可夫决策过程，并设计了一种基于强化学习的维护决策方法。具体内容包括质量损失与维护成本建模、考虑生产计划的维护决策方法建模、强化学习算法设计以及相关仿真实验。

### 3.1 引言

多部件生产系统在现代工业场景中很常见，尤其是在装配线生产系统中，每个部件代表负责不同过程的生产设备。以笔记本电脑主板生产线为例，印刷机负责焊膏印刷，焊膏检测设备负责印刷的质量检测，贴片机负责元件安装。在这样的系统中，设备的意外故障往往严重影响生产效率和质量。主要原因是定位和维护故障可能会导致长时间的计划外停机，且处于故障边缘的设备很容易产生缺陷产品。对于生产系统来说，设备故障引起的维护成本通常占生产成本的很大一部分<sup>[66]</sup>，因此需要有效的维护方法。

在实际生产过程中，仅根据设备健康状态制定维护策略并不一定能达到最佳效率。主要原因是生产系统的效率与质量控制和生产计划密切相关，这两个因素也受到维护策略的影响。详情如下：

1. 质量控制是指将产品质量控制在符合制造商要求的范围内。影响产品质量的主要因素之一是设备的健康状态。当设备健康状况不佳时，产品质量恶化的风险大大增加，导致缺陷产品数量增加。处理缺陷产品的方法主要有两种：返工和丢弃。然而，这两种方法都会导致质量损失，即维修成本和生产浪费。为了减少损失，一个有效的手段是通过维护使设备保持良好的健康状态。然而，频繁的维护会增加维护成本，因此维护决策需要平衡维护成本和质量损失。
2. 生产计划决定了未来的生产任务和日程安排。当设备按照生产计划正常运行时，任何停止设备工作的维护行动都将导致停机成本<sup>[67]</sup>。相反，在生产部门计划的停机期间（如清洁、换班、批量或换工具）进行维护时，不计算停机成本<sup>[40,68]</sup>。为了降低成本，可以在生产计划中的计划停机期内尽可能地安排维护行动。

另外，多个部件之间的相关性给维护策略的制定带来了挑战。实际上，部件

之间的相关性主要可以分为三种类型：随机相关性、结构相关性和经济相关性。随机相关性是指某个部件的状态会影响其他部件的退化过程。结构相关性适用于在结构上构成连通集合的部件，维护一个部件时需要拆卸其他部件。经济相关性是指在系统中对某个部件进行维护时，对其他退化严重的部件同时维护可以降低系统的总体维护成本。由于经济相关性直接影响维护决策并通过维护成本体现，研究人员对此问题进行了广泛研究，本文主要关注的是经济相关性。

考虑到经济相关性，维护决策必须从全局视角出发，而不是仅仅优化单个部件。当前针对多部件系统维护优化的 CBM 方法主要分为两大类：基于阈值的机会性维护和基于深度强化学习（Deep Reinforcement Learning, DRL）的方法。基于阈值的机会性维护不仅为每个部件设置失效阈值以判断是否需要维护，还额外设置一个阈值来决定是否联合维护其他部件。然而，基于阈值的机会性维护的效果完全依赖于这两个阈值，且如何确定这些阈值也十分棘手。此外，这种方法在设置阈值时无法考虑质量损失，也无法根据环境和系统状态动态调整阈值。第二种方法则是基于 DRL 的方法。与前一种方法相比，DRL 方法直接基于监控到的系统状态信息做出决策，无需设定维护阈值。具体来说，它通过最小化总成本将部件的退化信息直接映射到相应的维护决策空间。将强化学习融入维护决策过程使维护优化过程更加灵活，且能够根据系统状态动态调整维护策略。

鉴于以上动机，本章研究了固定生产计划下多部件系统的产品质量和维护决策联合优化问题。考虑生产计划、质量和维护之间的耦合关系以及多个部件间的相互依赖关系，提出了一种新的基于深度强化学习的维护优化方法。与现有的基于深度强化学习的方法相比，重新构建了决策模型。具体来说，引入了当前生产批次剩余时间的概念，并分析了其状态转移过程（包含了生产计划的信息），分别对状态空间和状态转移函数进行重构。此外，还在奖励函数中加入了与健康状态相关联的质量损失以及与生产计划相关的停机成本。借助新的决策模型和 Deep Q Network (DQN) 算法，相应的最优决策问题得以求解。进一步地，结合深度强化学习方法与基于阈值的机会性维护，提出了 Restricted-DQN 算法，使其适应实际生产环境。

本章结构安排如下：第3.2节介绍了考虑生产计划和产品质量的维护决策建模；第3.3节介绍了基于深度强化学习的维护方法；第3.4节通过数值结果验证了所提方法的有效性；第3.5节总结了本章的研究内容。

## 3.2 考虑生产计划和产品质量的维护决策建模

考虑一个多部件生产系统，系统由  $m$  个部件构成。在生产过程中每个部件都经历累积的退化过程，部件的退化水平的表示可以使用物理健康指标，如累

积磨损、裂纹扩展、腐蚀等，或根据不同测量值构建的综合健康指标<sup>[69]</sup>。每个部件  $i(i = 1, 2 \dots m)$  的退化水平使用标量随机变量  $d_i(t)$  来描述。当  $d_i(t) > H_i$  时，部件  $i$  被认为失效也即故障，其中  $H_i$  是部件  $i$  的失效阈值。整个系统由  $m$  个部件串联组成，也即其中任意一个部件故障都会导致系统失效，因此系统的退化水平可以使用向量随机变量  $D(t) = (d_1(t), d_2(t) \dots d_m(t))$  来描述。假定经过所有部件加工后系统产出最后的产品，系统按批次生产，不同批次生产不同的产品类型，通过生产计划可以知道未来一段时间每个批次所需的加工时间。

### 3.2.1 退化过程建模

随着时间推进，大多数设备均会表现出不同程度的性能退化<sup>[70]</sup>。监测并量化设备退化程度有助于深入评估和预测可靠性，比单纯依赖故障时间数据更具洞察力。因此，基于退化的可靠性分析方法在学术和业界受到高度重视。此类方法的核心是对设备关键性能参数动态变化过程的研究，这些参数涉及设备物理属性以及反映退化进展的可量化指标。利用适宜的退化模型来描述设备可靠性随时间的衰减规律，从而全面展现设备运行中的功能特性演变情况。

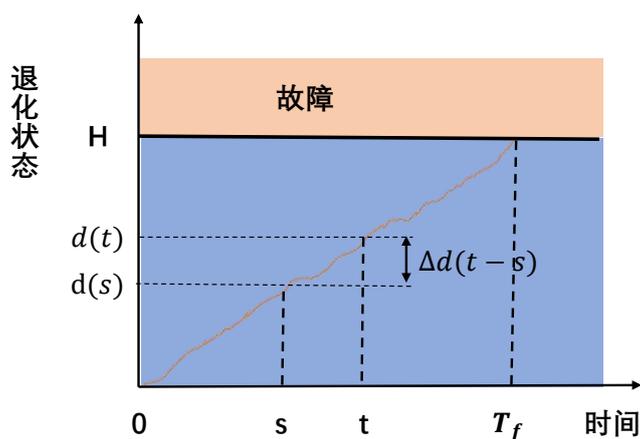


图 3.1 Gamma 过程示意图

在过去的文献中，提出了几种模型来模拟部件的退化过程，其中 Gamma 过程是最受欢迎的几种之一<sup>[71-73]</sup>。实际上，Gamma 过程是具有独立且非负增量的单调递增随机过程，适合用于描述由磨损、蠕变、疲劳、腐蚀等累积引起的退化过程。在制造过程中，相关部件的退化，如刀具的磨损或老化，一般是单调渐进的随机退化过程，退化量随时间单调累计，且增量是独立、非负的，非常符合 Gamma 退化过程模型的特点。因此，使用 Gamma 过程来模拟系统部件的退化过程，也即在没有任何维护干预的情况下，每个部件的退化水平的增量遵循 Gamma 过程。图 3.1 给出了 Gamma 退化过程的示意图。数学上，给定部件  $i$  在

时间  $s$  的退化水平  $d_i(s)$  下, 部件  $i$  在时间  $t$  的退化水平  $d_i(t)$  可以表示为:

$$d_i(t) = d_i(s) + \Delta d_i(t-s) \quad (3.1)$$

其中  $\Delta d_i(t-s)$  遵循形状参数为  $\alpha_i$ 、尺度参数为  $\beta_i$  的 Gamma 分布, 其概率密度函数定义为:

$$f(d; \alpha_i(t-s), \beta_i) = \frac{\beta_i^{(\alpha_i(t-s))} d^{(\alpha_i(t-s)-1)} \exp(-\beta_i d)}{\Gamma(\alpha_i(t-s))} \quad (3.2)$$

其中  $\Gamma(\alpha) = \int_0^\infty z^{\alpha-1} \exp(-z) dz (\alpha > 0)$  是关于  $\alpha$  的 gamma 函数。

$\Delta d_i(t-s)$  的期望和方差为:

$$\mathbb{E}[\Delta d_i(t-s)] = \alpha_i(t-s)/\beta_i, i = 1, 2, \dots, m \quad (3.3)$$

$$\text{Var}[\Delta d_i(t-s)] = \alpha_i(t-s)/\beta_i^2, i = 1, 2, \dots, m \quad (3.4)$$

其中 Gamma 过程的形状参数  $\alpha_i$  和尺度参数  $\beta_i$  可以通过极大似然估计。多部件系统的退化水平由各个部件的退化水平构成  $\mathbf{D}(t) = (d_1(t), d_2(t), \dots, d_m(t))$ 。事实上, 在多部件系统设计中, 工程师会特意设计部件或子系统减少或消除部件间的依赖性, 确保即使某个部件失效, 也不至于引起其他部件的连锁反应, 从而提高整体系统的可靠性。且退化过程是否独立不影响后续维护决策建模, 因此假设各个部件的退化过程相互独立。

### 3.2.2 质量损失建模

产品的质量可以用关键产品特征如直径、高度、强度等表征。且关键产品特征通常存在一个标准值, 代表合格标准。经由一系列加工、装配等工艺环节后, 产品的关键产品特征可能与预设标准值不符, 二者之间的偏差定义为产品质量特征偏差, 并用符号  $K(t)$  表示。产品质量特征偏差可用于直观体现产品质量的好坏。例如在加工流程中, 零部件几何尺寸的实际测定值与规定的标准尺寸之差, 即可反映出该零部件质量水平的高低。

产品质量特征偏差受到多种因素的影响, 包括质量相关部件的退化变量以及噪音变量, 且这些变量与产品质量特征偏差之间的关系可能是非线性的。参考 Chen 等<sup>[27]</sup>以多部件退化过程中的退化状态  $\mathbf{D}(t) = (d_1(t), d_2(t), \dots, d_m(t))$  以及噪音变量  $\mathbf{V} = (v_1, v_2, \dots, v_m)$  为设计变量, 建立如式 (3.5) 所示的多元二次模型, 该模型建立了多部件系统退化状态与产品质量特征偏差的量化关系。

$$k(t) = c + \mathbf{aD}(t)^T + \mathbf{bV}^T + \mathbf{D}(t)\mathbf{UV}^T \quad (3.5)$$

其中  $c$  是模型常数； $\mathbf{a}$  和  $\mathbf{b}$  分别表示  $\mathbf{D}(t)$  和  $\mathbf{V}$  对  $k(t)$  线性影响效应的向量； $\mathbf{U}$  表示  $\mathbf{D}(t)$  和  $\mathbf{V}$  之间相互影响关系的矩阵。假设噪音变量  $\mathbf{V}$  服从正态分布，即  $\mathbf{V} \sim N(0, \Sigma)$ 。

针对由设备退化导致的质量特征偏差而产生的损失成本，田口函数 (Taguchi Function) 是目前最主要的计算方法，田口函数的基本形式如下所示：

$$L(k) = qk^2 \quad (3.6)$$

其中， $L(k)$  是成本损失， $q$  是成本系数， $k$  是产品质量特征偏差。

参考田口函数的表达式，产品质量特征偏差造成的质量损失成本函数可以表示为：

$$c_q(t) = \rho(c + \mathbf{aD}(t)^T + \mathbf{bV}^T + \mathbf{D}(t)\mathbf{UV}^T)^2 \quad (3.7)$$

其中， $\rho$  表示单位时间内产出产品的数量。

### 3.2.3 维护决策过程

为了避免系统故障并减少因系统退化导致的质量损失，需要制定合适的维护策略。在每个时刻决定是否进行维护是不可能的，只能在离散的时间间隔上进行维护活动。即每隔时间间隔  $T_0$  设置一个检查点，在此检查点对系统进行检查以确定是否应该进行维护。根据维护效果，维护行动可以分为不完美维护和完美维护。假设部件在维护前的退化状态为  $d_{be}$ ，维护后的退化状态为  $d_{af}$ 。对于不完美维护的情况，有  $0 < d_{af} < d_{be}$ ；而对于完美维护的情况，则有  $d_{af} = 0$ 。无论是不完美维护还是完美维护，都不会影响维护决策框架，因此假设所有维护行动均为完美维护。

执行一次维护行动产生的维护成本  $c_m$  可分为两部分： $c_f$  和  $c_d$ 。其中  $c_f$  是操作成本，包括零部件更换成本和人力成本。 $c_d$  是由于维护所造成的停机成本，它取决于维护所需的时间。令  $\mathbf{a}_t = (a_{t,1}, \dots, a_{t,n})$  代表在时间  $t$  的维护行动，其中  $a_{t,i} \in \{0, 1\}$ ， $a_{t,i} = 0$  表示第  $i$  个部件未被维护，而  $a_{t,i} = 1$  表示第  $i$  个部件被维护。则有如下公式：

$$c_f(\mathbf{a}_t) = \sum_{i=1}^m c_i a_{t,i} \quad (3.8)$$

其中  $c_i$  是第  $i$  个部件的操作成本。

维护所需的时间与被维护的部件数量有关，但同时维护多个部件所需的时间要小于单独维护每个部件所需的时间之和。维护时的停机时间也受到生产计划的影响。一方面，执行维护活动需要消耗时间，所以维护活动会导致停工停机成本；另一方面，生产系统按批次生产，在批次转换时需要调整以满足下一生产

批次的生产条件。而在批次转换期间，由维护引起的停机成本可以忽略不计。则有如下公式：

$$c_d(a_t) = \begin{cases} f(a_{t,1}, \dots, a_{t,m}) & t \notin T_c \\ 0 & t \in T_c \end{cases} \quad (3.9)$$

其中  $f(a_{t,1}, \dots, a_{t,n})$  是维护各部件引起的停机成本函数， $T_c$  是所有批次转换间隔区间的并集。

综上所述，在每个检查点，需要基于系统的退化状态、质量损失、维护成本以及生产计划，做出最优的维护决策。

### 3.3 基于深度强化学习的维护决策方法

强化学习是三大基础机器学习方法之一，其目的是在一个由一个或多个智能体与环境互动组成的系统中，选择最优的动作策略以最大化奖励或最小化损失。智能体基于观察到的环境状态来选取行动，每一步行动都会产生一个奖励和下一个状态。因此，强化学习非常适合研究马尔可夫决策过程<sup>[74]</sup>。对于 MDP 而言，强化学习的目标通常是学习一个最优策略  $\pi : S \rightarrow A$ ，该策略旨在最大化长期奖励，学习过程可以通过智能体与环境之间的互动完成。

#### 3.3.1 多部件系统维护决策

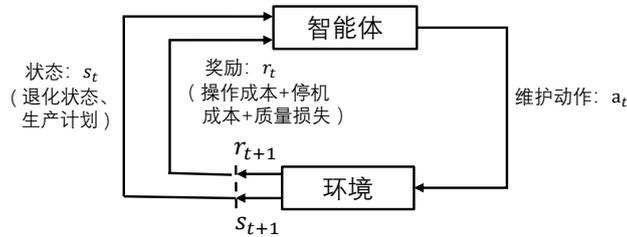


图 3.2 维护决策流程图

维护决策过程包括确定每个检查点需要维护哪些组件，执行维护操作，系统返回两个检查点之间的生产成本。这个过程符合强化学习的框架，如图 3.2 所示。此外，系统的退化过程 Gamma 过程满足马尔可夫性质。因此，强化学习适用于解决维护优化问题，下文介绍强化学习的相关元素：

状态空间  $S$  是指所有可能状态的集合。显然，系统状态  $s_t$  包含了每个部件的退化状态，这是智能体作出决策的主要依据。考虑到生产计划会影响维护决策，将当前批次的剩余时间  $\tau_t$ （如图 3.3 所示）加入到系统状态中。由于生产持续时间大致与产品数量成正比，故可将  $\tau_t$  设置为当前批次剩余的产品数量。因此，系统状态定义为：

$$s_t = (d_1, d_2 \dots d_m, \tau_t) \quad (3.10)$$

其中  $d_i$  是第  $i$  个部件的退化状态。

确定了系统状态空间后，需要确定状态转移过程。每个部件的退化状态遵循 Gamma 过程，而  $\tau_t$  会根据生产计划递减，即当前批次的剩余产品数量。

$$\tau_t = \begin{cases} \tau_{t-1} - 1 & \tau_{t-1} > 0 \\ 0 & \tau_{t-1} = 0 \quad \& \quad o_t = 0 \\ B_t & \tau_{t-1} = 0 \quad \& \quad o_t = 1 \end{cases} \quad (3.11)$$

其中  $B_t$  表示时间  $t$  时的批次持续时间； $o_t$  表示工作状态，即在批次转换点  $o_t = 0$ ，生产过程中  $o_t = 1$ 。 $B_t$  由生产计划决定。从(3.11)可以看出， $\tau_t$  由  $\tau_{t-1}$  决定，因此对于状态  $s_t$  来说，马尔可夫性质仍然成立。

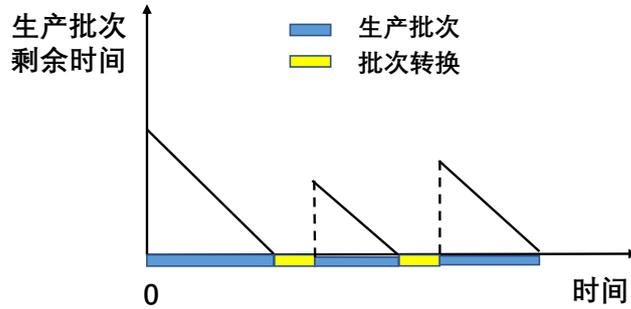


图 3.3 当前生产批次剩余时间

动作空间  $A(S)$  是指所有可能动作的集合。在时间  $t$ ，动作  $a_t \in A(S)$ ，每个动作都会影响下一个状态  $s_{t+1}$ 。在这种情况下，动作是选择要维护的部件。每个部件可以选择维护或不维护，因此可用的动作数量为  $2^m$ ，其中  $m$  为部件的数量。

奖励函数  $r(s_t, a_t)$  是在状态  $s_t$  下采取动作  $a_t$  时系统的奖励或损失。此处，奖励函数取负值，表示损失，包括质量损失  $c_q$ 、操作成本  $c_f$  和停机成本  $c_d$ 。因此，总的奖励函数为：

$$r(s_t, a_t) = c_q(s_t, a_t) + c_f(s_t, a_t) + c_d(s_t, a_t) \quad (3.12)$$

其中  $c_q$  可根据(3.7)得到， $c_f$  可根据(3.8)得到， $c_d$  可根据(3.9)得到。实际上，它们都是关于  $s_t$  和  $a_t$  的函数（或者部分相关）。

折扣因子  $\gamma \in (0, 1)$  在强化学习中广泛使用，用于确保奖励无限和的收敛性。因此，长期奖励函数为：

$$R_t = \sum_{l=0}^{\infty} \gamma^l r(s_{t+l}, a_{t+l}) \quad (3.13)$$

### 3.3.2 深度强化学习算法

深度强化学习是强化学习与深度学习相结合的一种方法，它提供了强大的近似能力和学习能力，极大地提高了计算速度，因此适用于高维 CBM 优化问题。DRL 在诸如机器人控制<sup>[75]</sup>、对话系统训练<sup>[76]</sup>和维护决策<sup>[77]</sup>等多个领域取得了广泛应用和成功。

首先，定义动作价值函数  $Q_\pi(s_t, a_t)$  和最优动作价值函数  $Q^*(s_t, a_t)$ 。其中， $Q_\pi(s_t, a_t)$  定义为：

$$Q_\pi(s_t, a_t) = \mathbb{E}[R_t | s_t, a_t, \pi] \quad (3.14)$$

这里  $R_t$  代表长期回报函数。上述等式表明，如果从初始状态  $s_t$  出发执行动作  $a_t$ ，然后始终按照策略  $\pi$  选择后续动作，则可以获得期望长期奖励。基于动作价值函数，定义最优动作价值函数如下：

$$Q^*(s_t, a_t) = \max_{\pi} Q_\pi(s_t, a_t) \quad (3.15)$$

$Q^*(s_t, a_t)$  代表给定初始状态下的可能最优长期奖励，是求解的目标。一旦估计出最优动作价值函数，就可以得到以下最优策略：

$$\pi(s_t) = \arg \max_{a_t} Q^*(s_t, a_t) \quad (3.16)$$

强化学习中的主要挑战之一是如何计算(3.15)中的最优动作价值函数。传统方法如标准 Q-learning 创建了一个名为 Q 表的表格，然后在表格中初始化、更新并存储最优动作价值函数。然而，这种方法仅限于离散的动作和状态空间，对于高维问题缺乏可扩展性。DQN 利用神经网络来近似最优动作价值函数，相比于基于表格的 Q-learning 更加灵活。在各种 DQN 算法中，标准 DQN 在噪声环境下容易过高估计动作价值，而 Double DQN (DDQN) 算法可以缓解这个问题。因此采用 DDQN 方法来估计(3.15)。

具体来说，令  $Q(s_t, a_t; \theta)$  表示参数化为  $\theta$  的动作价值函数。在标准 DQN 训练过程中，使用单一神经网络来近似动作价值函数。相比之下，DDQN 包含两个神经网络：在线网络和目标网络，分别参数化为  $\theta$  和  $\theta^-$ 。在线网络设计用于根据当前状态选择动作，而目标网络主要用于评估策略。目标值  $y_t$  定义为：

$$y_t = r(s_t, a_t) + \gamma Q(s_{t+1}, \arg \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta_t^-); \theta_t^-) \quad (3.17)$$

$y_t$  是对动作价值函数的一个更好的估计，因此，在每次训练步骤中，基于平方损

失  $(y_t - Q(s_t, a_t; \theta))^2$  更新参数化为  $\theta$  的在线网络:

$$\theta_{t+1} = \theta_t + lr(y_t - Q(s_t, a_t; \theta))\partial_{\theta_t} Q(s_t, a_t; \theta) \quad (3.18)$$

其中  $lr$  代表学习率。

---

### 算法 3.1 DDQN 算法

---

- 1 初始化  $\theta$  并复制  $\theta$  给  $\theta^-$
  - 2 对每个回合重复:
  - 3 初始化状态  $s$
  - 4 对回合中的每一步重复:
  - 5 使用从  $Q(s, a; \theta)$  中得到的策略选择动作  $a$  (例如,  $\epsilon$ -greedy 策略)
  - 6 执行动作  $a$ , 观察到回报  $r$ , 下一状态  $s'$
  - 7 计算更新目标  $y = r + \gamma Q(s', \arg \max_a Q(s', a; \theta); \theta^-)$
  - 8 根据损失函数  $(y - Q(s, a; \theta))^2$  进行梯度下降
  - 9 每  $N$  步替换参数:  $\theta^- \leftarrow \theta$
- 

训练过程如算法3.1所示。训练完成后, 在线网络可用于维护决策。在每个检查点, 获取每个部件的退化状态及当前生产批次的剩余时间, 将这些值输入在线网络, 得到所有维护动作的动作价值。对应最大动作价值的动作即为当前状态下最优的维护动作。

### 3.3.3 Restricted-DQN 算法

在实际生产环境中, 任何操作都必须首先考虑安全因素。首要目标是通过避免被认为存在风险的操作, 确保生产过程的顺利进行。然而, 第 3.3.2 节中实现的算法依赖深度学习方法, 引入了不确定性的元素。虽然通过训练可以获得良好的结果, 但不能保证算法的所有动作在任何时候都绝对安全。例如, 算法可能会在设备仍处于健康状态时做出维护决策, 或者未能对即将失效的设备进行必要的维护。因此, 算法3.1可能在真实生产环境中做出错误的决定而产生严重的后果。

为了适应实际生产环境, 有必要对算法3.1得到的结果进行一些修改。类似于基于双阈值的方法, 设定两个阈值  $H_{max}$  和  $H_{min}$  来确保决策的安全性。具体而言, 输入退化状态和当前批次剩余时间到 DQN 算法中, 经过训练后可以输出相应的维护动作。获得 DQN 算法的结果后, 可以增加一些限制条件。如果某个部件退化程度较低, 而 DQN 方法建议维护, 则取消该维护动作。相反, 如果某个部件退化程度较高, 而 DQN 方法建议不维护, 则增加维护动作。为方便起见, 添加限制条件的方法称为受限-DQN, 具体流程可见算法3.2。

**算法 3.2 Restricted-DQN 算法**


---

```

1 使用算法3.1训练 DDQN 算法
2 在检查点观察得到状态  $s$ 
3 输入状态  $s$  给 DDQN 算法得到动作  $a$ 
4 根据动作  $a$  决定维护部件集合
5 对每个部件  $i, i = 1, 2, \dots, n$ :
6 判断部件  $i$  是否在维护部件集合中
7 if 部件  $i$  在维护部件集合中 then
8   | if 部件  $i$  的退化状态满足  $x_i < H_{min}$  then
9   | | 将部件  $i$  移除维护部件集合
10  | end
11 else
12  | if 部件  $i$  的退化状态满足  $x_i > H_{max}$  then
13  | | 将部件  $i$  添加进维护部件集合
14  | end
15 end
16 返回维护集合

```

---

### 3.4 实验结果及分析

为了证明所提方法的有效性,本节以多部件系统为对象展开实验,其中每个部件都经历随机退化过程。每个生产批次的生产时长服从正态分布  $N(70, 5)$ 。假设检查点间隔为一个单位时间(例如小时、天等),在每个检查点决定是否进行维护。将提出的 DRL 方法与基于阈值的机会性维护进行比较。3.4.1节展示实验设置,3.4.2节展示三部件系统的实验结果,3.4.3节展示敏感性分析。

#### 3.4.1 实验设置

表 3.1列出了各部件退化过程的参数,图 3.4显示了系统部件的退化过程,虚线表示它们各自的故障阈值。如图 3.4所示,三个部件设置了相同的故障阈值,但它们的退化速率不同。部件 3 的退化速度比部件 1 和 2 更快,因此相比其他部件更有可能发生故障,而部件 1 最不容易发生故障。

根据第 3.2.2节的内容,质量损失由常数  $c$ 、向量  $\mathbf{a}$  和  $\mathbf{b}$ 、矩阵  $\mathbf{U}$  和噪音变量  $\mathbf{V}$  决定。而质量损失受退化状态向量影响最大,因此设置  $c = 0.0$ ,  $\mathbf{a} = [0.0012, 0.0016, 0.0019]$ ,  $\mathbf{b} = [0.0022, 0.0014, 0.0017]$ ,  $\mathbf{U} = 0.0001\mathbf{I}$ ,  $\mathbf{V} \sim N(0, \mathbf{I})$ 。

与维护相关的成本包括两部分:操作成本  $c_f$  和停机成本  $c_d$ 。设置每个部件的操作成本  $c_i$  为 2.5,停机成本  $c_d$  与停机时间有关。停机时间包括维护时间和

生产系统设置时间。维护时间与被维护的部件数量成正比，维护的部件越多，所需时间越长。生产系统设置时间代表每次维护动作中断和恢复生产所需的时间。设置每次维护动作所需的生产系统设置时间为 2.0，每个部件的维护时间成本为 1.5。因此， $f(a_{t,1}, \dots, a_{t,m})$  为：

$$f(a_{t,1}, \dots, a_{t,m}) = \begin{cases} 3.5 & \sum_i a_{t,i} = 1 \\ 5.0 & \sum_i a_{t,i} = 2 \\ 6.5 & \sum_i a_{t,i} = 3 \end{cases} \quad (3.19)$$

表 3.1 三个部件退化过程的参数

参数	描述	部件 1	部件 2	部件 3
$\alpha$	Gamma 过程形状参数系数	0.4	0.2	0.2
$\beta$	Gamma 过程尺度参数	1.2	1.4	1.0

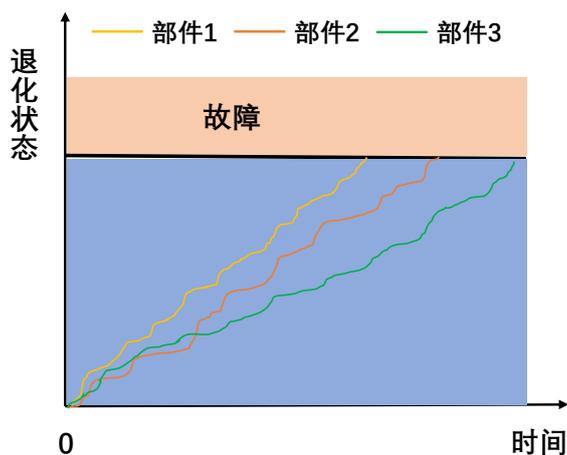


图 3.4 三个部件的退化过程

DQN 算法使用了一个具有两个隐藏层的神经网络。第一个隐藏层有 28 个神经元，第二个隐藏层有 36 个神经元。学习率设置为 0.001，贪心选择概率  $\epsilon$  为 0.945，折扣因子  $\gamma$  为 0.993。基于阈值的机会性维护策略的两个阈值设置为 35、28。

### 3.4.2 三部件系统

为了直观地演示提出的考虑生产计划和质量的维护决策方法的有效性，模拟了三个场景以对比不同的维护策略，每个场景包含六个批次。最优维护策略通过穷举搜索获得。在模拟维护决策时，三个部件在 400 个时间步内的退化过程预

先随机确定。在固定所有部件的退化过程后，可以确保不同维护策略都在同一退化情景下进行试验，并且能够通过穷举搜索得到实际最优维护策略。

表 3.2 不同策略下的成本

	最优维护	机会性维护	DQN	Restricted-DQN
场景 1	35.0	62.5	36.6	-
场景 2	42.0	72.3	44.2	-
场景 3	44.1	-	63.1	44.1

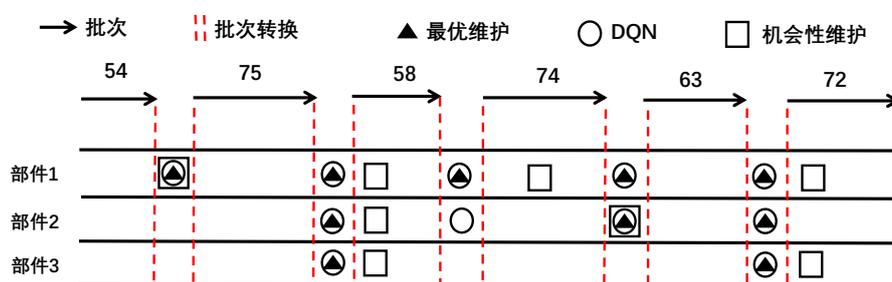


图 3.5 场景 1 中不同维护策略结果

对于场景 1，批次规模为 {54,75,58,74,63,72}，总生产时间为 396。实验结果见表 3.2 和图 3.5。从图表中可以看出，DQN 方法基本与最优维护策略相同，DQN 方法能够在不向算法提供机会维护概念的情况下学会同时维护多个部件的策略。与基于阈值的机会性维护方法相比，主要区别在于第二批次和第三批次之间是否维护的选择。在第二批次结束时，三个部件的退化状态分别为 27、31 和 30，系统已经开始生产较多不合格产品。此时，同时维护三个部件可避免质量损失，并且在批次转移点进行维护能避免停机成本。基于阈值的机会性维护并未充分利用这一时机。

在场景 2 中，批次规模为 {72,63,87,64,58,79}，总生产时间为 423。实验结果在表 3.2 和图 3.6 中展示。总体上，DQN 方法仍然接近最优维护策略。从图 3.6 可以看出，基于阈值的机会性维护不仅存在未能充分利用批次转换期间计划停机时间的机会的缺点，而且过于保守。比如，最优维护策略是在第一批次结束时维护部件 1，当时部件 1 的退化状态为 35。尽管提前维护可以减少质量损失，但在生产过程中进行维护会导致更严重的停机成本。相较于机会性维护，DQN 方法能够平衡质量损失和停机成本做出更好的选择。此外，从图 3.6 可以看出，DQN 方法和最优维护策略分别进行了 5 次维护，而机会性维护进行了 8 次，这也体现了所提方法的优势。

从场景1和场景2可以看出，DQN方法优于机会性维护方法，但与最优维护策略之间仍存在差距。主要原因有两个：一是DQN方法陷入局部最大值。从图3.5和图3.6可以看出，主要差异出现在批次转换点。在某些批次转换点，系统无需维护，但DQN方法选择了维护。这是因为DQN方法陷入了在每次批次转换时尽可能多做维护的局部最优策略。在场景1中，被维护的部件的退化状态为19.0，在场景2中，被维护的部件的退化状态为18.0。它们都处于健康状态，但DQN方法错误地选择了维护。另一个原因是DQN算法本身的局限性。DQN算法利用神经网络拟合系统状态与维护行动之间的关系，但深度学习本质上存在不确定性，不能保证对任何系统状态都能得出准确结果。

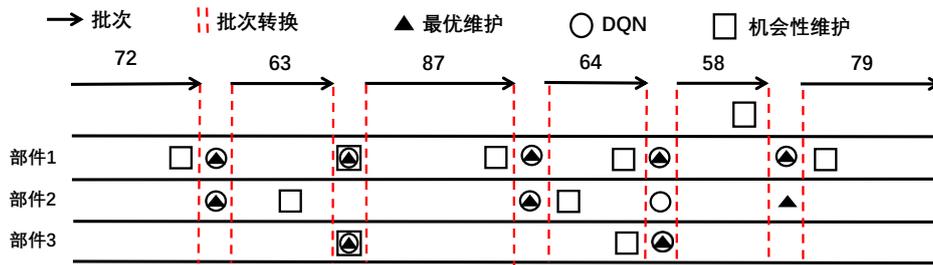


图 3.6 场景 2 中不同维护策略结果

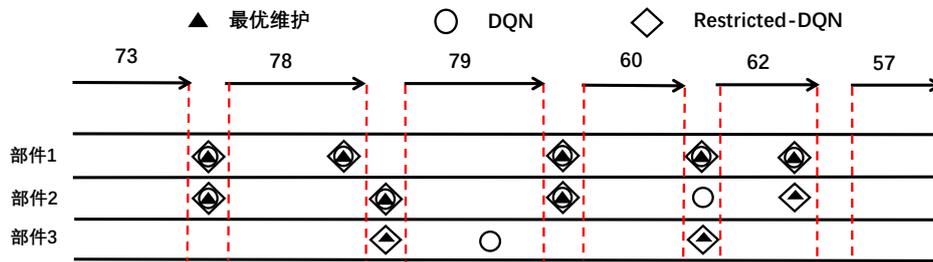


图 3.7 场景 3 中不同维护策略结果

为了弥补上述差距，需要对DQN方法添加一些限制。如果部件退化程度低而提出的方案选择维护，则取消维护操作；反之，如果部件退化程度高而方案选择不维护，则增加维护操作。在场景3中进行了DQN方法与Restricted-DQN的对比实验，结果见表3.2和图3.7。可以看到，DQN方法在整个时间段内完全忽略了部件3，导致在第三批中期产生了巨大的事后维护成本。并且在第四批次和第五批次的批次转换点时被困在局部最大值。相比之下，Restricted-DQN缓解了上述问题并在场景3中实现了最优维护策略。

表3.3展示了100次实验结果的均值和标准差。可以看出，不进行任何维护时成本最高。所提DQN方法的平均成本低于机会性维护方法，但方差较高，即DQN方法的稳定性不足，而Restricted-DQN不仅具有最低的平均成本，与DQN方法相比的还显著降低了方差。

表 3.3 成本的均值和方差

	Restricted-DQN	DQN	OM	CM
均值	-52.41	-53.27	-66.28	-128.16
方差	78.75	91.32	56.88	140.86

### 3.4.3 敏感性分析

为了展示所提出的基于 DRL 的 CBM 策略的灵活性和可扩展性, 考虑了具有五个、八个和十个部件的系统。对于包含五个部件的系统, 每次检查时面临 32 种不同的维护操作选择 ( $2^5 = 32$ )。为了应对这一较大的动作空间, 在 DDQN 模型的两层网络中将神经元数量扩展至 (60, 100), 以实现更好的近似能力; 对于八个部件的系统, 维护动作的选择增加到了 256 种 ( $2^8 = 256$ ), 此时神经元数量进一步扩展到 (100, 200); 而对于包含十个部件的系统, 维护动作多达 1024 种 ( $2^{10} = 1024$ ), 相应的神经元配置调整为 (300, 300)。Gamma 过程的参数  $\alpha_i$  遵循均匀分布  $U(0.3, 0.1)$ ,  $\beta_i$  遵循  $U(1.3, 0.1)$ , 以此确保各部件的退化过程各不相同。

表 3.4 五部件系统不同维护策略的成本均值和方差

	Restricted-DQN	DQN	机会性维护	事后维护
均值	<b>-64.81</b>	-66.43	-80.88	-367.37
方差	61.02	139.50	<b>57.78</b>	480.60

表 3.5 八部件系统不同维护策略的成本均值和方差

	Restricted-DQN	DQN	机会性维护	事后维护
均值	<b>-93.76</b>	-101.42	-126.68	-504.0
方差	286.83	489.65	<b>142.89</b>	1121.5

在上述参数设置下, 我们计算了所提基于 DRL 的维护策略下的成本, 并为了比较计算了 Restricted-DQN 策略、DQN 策略、基于阈值的机会维护策略以及事后维修策略的成本。成本的均值和标准差结果列于表3.4、表3.5和表3.6中。可以看出, 无论是对于五部件系统、八部件系统还是十部件系统, 所提出的 Restricted-DQN 策略都实现了最低的平均成本, 并达到了较低的成本方差。与基于阈值的机会维护相比, Restricted-DQN 方法能为五部件系统降低成本 16.07, 为八部件

系统降低成本 32.92，为十部件系统降低成本 40.73。随着部件数量的增加，相比于基于阈值的机会维护策略，所提方法展现出更为显著的成本降低效果，这表明了所提方法的有效性和良好的扩展性。

**表 3.6 十部件系统不同维护策略的成本均值和方差**

	Restricted-DQN	DQN	机会性维护	事后维护
均值	<b>-329.29</b>	-333.34	-370.02	-1122.71
方差	631.85	878.55	<b>531.11</b>	2525.35

为了验证模型是否适用于长期成本，将时间跨度从 400 增加到 4000。表 3.7 展示了 100 次实验结果的均值和方差。结果显示所提方法适合长时间跨度场景。

**表 3.7 长时间跨度下成本**

	Restricted-DQN	DQN	OM	CM
五部件	<b>-647.00</b>	-667.08	-808.34	-3687.95
八部件	<b>-922.65</b>	-980.05	-1972.6	-5037.5
十部件	<b>-3692.79</b>	-3722.45	-4096.96	-12337.39

### 3.5 本章小结

本章针对固定生产计划下多部件系统的产品质量和维护决策联合优化问题，对部件的退化过程进行建模，并分析了多部件系统退化状态与产品质量之间的关系，以及生产计划对维护决策的影响。提出了一种基于强化学习的维护决策方法，引入当前生产批次剩余时间的概念，分别对状态空间和状态转移函数进行重构。此外，还在奖励函数中加入了与健康状态相关联的质量损失以及与生产计划相关的停机成本。最终实验表明所提方法优于传统的基于阈值的机会性维护方法，在维护过程中能够考虑到产品质量和生产计划，降低了生产成本。

## 第4章 多部件系统的产品质量、维护决策和生产调度联合优化方法

第3章提出了固定生产计划下多部件系统的产品质量和维护决策联合优化方法,在维护过程中能够考虑到产品质量和生产计划,降低了生产成本。但其仅将生产调度的结果作为约束条件,并不是联合优化维护决策与生产调度。本章针对多部件系统的产品质量、维护决策和生产调度联合优化问题,设计了一种两阶段联合优化方法。具体内容包括联合优化问题建模、两阶段联合优化方法设计、求解算法设计以及相关仿真实验。

### 4.1 引言

产品质量、维护决策和生产调度是生产过程中的三个主要因素。为了减少维护成本、质量损失和因设备故障导致的生产停机,增加生产效率,企业需要协同调整其质量、维护和调度管理策略。较于传统规划方法将质量、维护和调度视为孤立问题,集成控制策略已展现出更好的性能表现<sup>[78]</sup>。这在过去几十年间促使科学家们不断探索最优的集成控制模型。然而,尽管如此,目前文献中同时涵盖质量、维护和调度三者联合优化的研究仍然相对较少<sup>[79]</sup>。

当设备健康状况不佳时,产品质量恶化的风险大大增加,导致缺陷产品数量增加<sup>[80]</sup>。可以通过维护使设备保持良好的健康状态从而减少质量损失,然而频繁的维护会增加维护成本。为平衡维护成本和质量损失,需要将产品质量退化纳入到维护决策中。

维护决策过程和生产调度过程相互影响。设备维护部门根据监测设备运行状态以及历史运行数据,安排维护计划让设备发生故障之前维护,维护的时间将传达给生产部门。安排的维护可以最大限度地提高机器可用性,但可能会影响生产计划的顺利执行。同样,生产调度部门可能倾向于充分利用设备的能力来满足生产需求。在这种情况下,由于持续生产导致故障较多,机器的可用性将会降低。因此,单独优化维护决策与生产调度会导致冲突情况,如图4.1。

针对维护决策与生产调度的联合优化模型,可以分为两类:相关模型和集成模型<sup>[30]</sup>。相关模型在优化一个因素时将另一个因素作为约束条件。例如将维护部门安排的维护计划作为约束条件,生产调度部门需要考虑维护对生产的影响,合理安排生产计划以达到目标函数最优,如最小化总延迟时间。相关模型在优化过程中将一个因素作为约束条件,并没有将其作为决策变量,因此并不能达到全局最优。

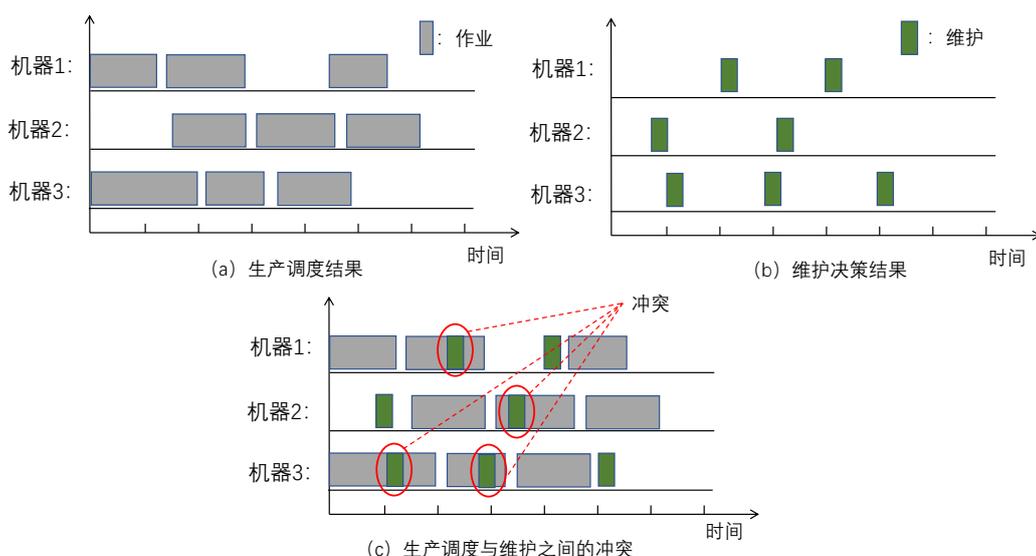


图 4.1 不同功能部门间的冲突

集成模型同时优化两个因素，优化模型的决策变量由两个因素组成，目标函数也包含两个因素的优化目标，如维护成本、总延迟时间等。在集成模型中，维护决策不仅考虑何时进行维护以减少故障停机时间，还会考虑维护时机与生产节奏的契合，以避免维护活动对正常生产造成过大冲击。生产调度则会在全面了解维护计划的基础上，灵活调整生产计划，尽可能减小维护活动带来的不利影响，同时优化生产流程，实现整体效益的最大化。相较于相关模型，集成模型考虑维护决策与生产调度相互之间的影响，从而能够寻找到维护与生产之间的最佳平衡点，以实现整个生产系统的全局最优化。

因此本章提出一种适用于多部件系统的两阶段集成优化方案，第一阶段以维护决策为主，在维护中考虑生产计划以及产品质量，利用强化学习实现能够根据生产计划变化动态调整的维护策略。第二阶段利用第一阶段得到的维护策略，计算生产过程中的维护时间、维护成本和质量损失，以最小化质量损失、维护成本和延误成本为优化目标建立调度优化模型。根据优化模型设计相应的染色体编码，利用遗传算法对优化问题求解。

本章结构安排如下：第4.2节介绍了维护决策和生产调度的联合优化问题描述；第4.3节介绍了产品质量、维护决策和生产调度联合优化方法；第4.4节给出了相应的求解算法；第4.5节给出了相关的实验结果及分析；第4.6节总结了本章的研究内容。

## 4.2 问题描述与假设

本章考虑了多部件生产系统，系统由  $m$  个部件串联组成，调度作业的加工需所有部件共同参与。所有部件在加工过程中会磨损退化，且任一部件退化状态超过故障阈值会导致系统停机。有  $n$  个作业待加工，每个作业  $i$  都有对应的加工时长  $p_i$ 、截止时间  $td_i$  和惩罚权重  $w_i$ 。惩罚权重代表该作业的紧急程度，权重越大越紧急。

相关假设如下：

1. 系统由  $m$  个相同且独立的关键部件串联组成，任何一个部件的维护都会影响系统的正常运行。
2. 单机系统一次只能处理一项作业，并且每个作业只能由一台机器处理。
3. 完成当前处理的作业之前不允许插入新的作业。
4. 在两个作业之间需要一段时间对系统进行调整，使其满足下一个作业的工作条件，称为设置时间  $T_m$ ，在设置时间段对系统维护不会产生停机成本。
5. 构成系统的部件在工作过程中会经历连续的退化过程，且不同部件退化过程相互独立。
6. 当系统的某个部件退化状态超过故障阈值时会发生故障，且任一部件故障都会导致整个系统不能工作。
7. 在系统工作过程中发生故障会进行事后维护，也可以在故障发生前进行预防性维护。
8. 事后维护和预防性维护均可以将故障的部件恢复到完好如新的状态。

## 4.3 产品质量、维护决策与生产调度联合优化方法

### 4.3.1 联合优化模型建模

产品质量、维护决策与生产调度联合优化的目标是在确保产品质量、满足交货期要求的前提下，最小化维护成本、最大化生产效率。为实现这样的联合优化，需要构建优化模型将维护决策和生产调度看作一个整体，通过合理调配维护资源、灵活调整生产计划，以应对设备状态的变化和市场需求的波动。该优化模型的优化目标是 minimized 加权延误成本、维护成本、与质量损失。下文以基于阈值的机会性维护为例建立联合优化模型。

在基于阈值的机会性维护中，首先会设定一系列反映设备健康状态的阈值指标，如设备的磨损度、故障率、性能衰退程度等。当监测数据表明设备状态接近或达到某个阈值时，就认为出现了进行维护的“机会”，此时实施维护可以防止设备性能急剧下降或突然失效，从而避免高昂的非计划停机时间和维护成本。

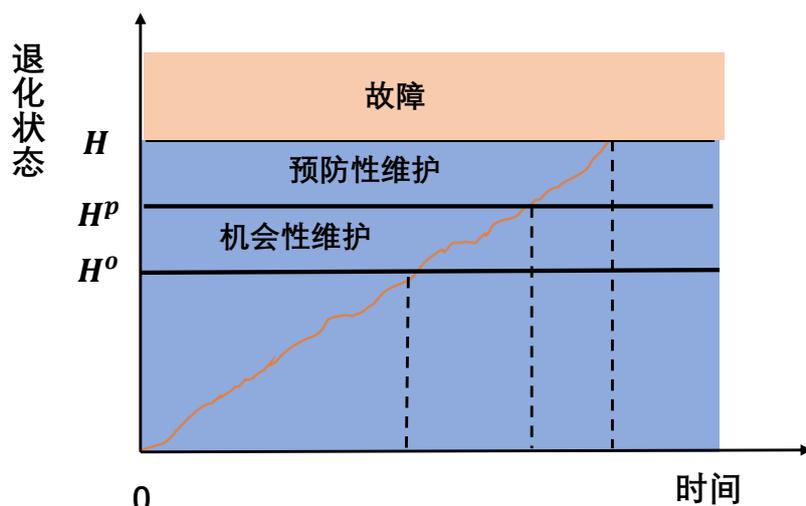


图 4.2 基于阈值的机会性维护示意图

具体而言,对于  $m$  部件系统,每个部件都定义了一个三元组  $(H_i^o, H_i^p, H_i)$ , 分别对应了机会性维护阈值、预防性维护阈值和事后维护阈值,有  $H_i^o < H_i^p < H_i$ , 示意图如图 4.2。基于这三种阈值的维护方法如下:

1. 若任意部件  $i$  的退化状态  $d_i$  满足  $d_i > H_i$ , 此时则部件  $i$  失效立即对该部件进行事后维护。
2. 若任意部件  $i$  的退化状态  $d_i$  满足  $H_i^p < d_i < H_i$  时, 对部件  $i$  进行预防性维护。
3. 在任意部件进行预防性维护或事后维护时, 若部件  $i$  的退化状态  $d_i$  满足  $H_i^o < d_i < H_i^p$ , 此时对部件  $i$  进行机会性维护。
4. 若  $m$  个部件的退化状态值均低于预防性维护阈值  $H_i^p$ , 则不安排维护活动。

机会性维护的基本假设在于同时维护多个部件比单独维护每个部件的成本更少。也即在某个部件快失效必须要进行维护时, 因为维护造成整个系统停机, 此时对其他部件而言是个“维护机会”, 也即不需要再额外造成系统停机。在维护决策与生产调度联合优化中, 考虑到连续两个作业之间有一段设置时间, 此时系统停止工作也即计划性的停机。类似于部件维护产生的“维护机会”, 此时也是所有部件维护的机会。也即当处于设置时间段时, 若部件  $i$  的退化状态  $d_i$  满足  $d_i > H_i^o$ , 此时对部件  $i$  进行机会性维护。

生产调度的决策变量是 0-1 变量  $x_{ij}, i, j = 1, 2, \dots, n$ ,  $x_{ij} = 1$  代表调度顺序为  $i$  的作业序号为  $j$ 。决策变量的不同组合将导致不同的调度安排。生产调度的目标是最小化加权延误成本, 而作业的完成时间受到维护时间的影响。加权延误成本计算如下:

调度顺序为  $i$  的作业的加工时间为:

$$p_{[i]} = \sum_{j=1}^n p_j x_{ij} \quad (4.1)$$

调度顺序为  $i$  的作业的完成时间为:

$$c_{[i]} = \sum_{k=1}^i (p_{[k]} + tm_{[k]}) + (i-1)T_m \quad (4.2)$$

事实上, 调度顺序为  $i$  的作业的完成时间由三部分组成: 第一部分是前  $i$  个作业所需的生产时间  $\sum_{k=1}^i p_{[k]}$ ; 第二部分是在工作过程中的维护时间之和  $\sum_{k=1}^i tm_{[k]}$ , 其中  $tm_{[k]}$  为在调度顺序为  $k$  的作业加工时间段总的维护时间; 第三部分是相邻作业之间的设置时间之和  $(i-1)T_m$ 。

调度顺序为  $i$  的作业的惩罚权重为:

$$w_{[i]} = \sum_{j=1}^n w_j x_{ij} \quad (4.3)$$

调度顺序为  $i$  的作业的截止时间为:

$$td_{[i]} = \sum_{j=1}^n td_j x_{ij} \quad (4.4)$$

因此调度顺序为  $i$  的作业的加权延误成本为:

$$c_{delay[i]} = w_{[i]} \max(0, c_{[i]} - td_{[i]}) \quad (4.5)$$

联合优化的目标是 minimized 质量损失、维护成本和加权延误成本, 多部件系统的联合优化模型建立如下:

$$f = \min_t (\sum (c_q(t) + c_m(t)) + \sum_{i=1}^n c_{delay[i]}) \quad (4.6)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad j = 1, 2, \dots, n \quad (4.7)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad i = 1, 2, \dots, n \quad (4.8)$$

$$x_{ij} = 0, 1 \quad i, j = 1, 2, \dots, n \quad (4.9)$$

$$0 < H_i^o < H_i^p < H^i \quad i = 1, 2, \dots, m \quad (4.10)$$

其中公式(4.6)描述了联合优化的目标函数,  $c_q(t)$  由式(3.7)给出,  $c_m(t)$  由式(3.8)和式(3.9)给出,  $c_{delay[t]}$  由式(4.5)给出; 公式(4.7)和公式(4.8)描述了目标函数中的单机调度约束条件, 也就是说同一时刻仅能安排一个作业进行加工, 且在任何特定时间, 某一作业只能安排在唯一的一个位置; 而公式(4.9)和公式(4.10)则给出了决策变量的取值范围。

### 4.3.2 两阶段联合优化方法

基于强化学习的维护决策原理将设备退化状态等信息建模为系统状态, 维护动作构建为系统动作, 维护相关成本构建为系统奖励。然后计算所有状态动作价值函数, 通过比较选择具有最大状态动作价值函数的动作即为此状态下的最优维护动作。通过第3章分析可知在多组件系统的维护决策中, 基于强化学习的维护策略优于基于阈值的机会性维护策略, 因此下文给出产品质量、生产调度与基于强化学习的维护的联合优化方法。

一般的联合优化模型通常是将维护决策的决策变量和生产调度的决策变量放在一起优化, 例如将决定维护动作的阈值和决定调度顺序的0-1变量一起作为联合优化的决策变量。然而基于强化学习的维护策略由状态动作价值函数决定, 并没有对应的决策变量, 这给维护与生产调度的联合优化带来困难。

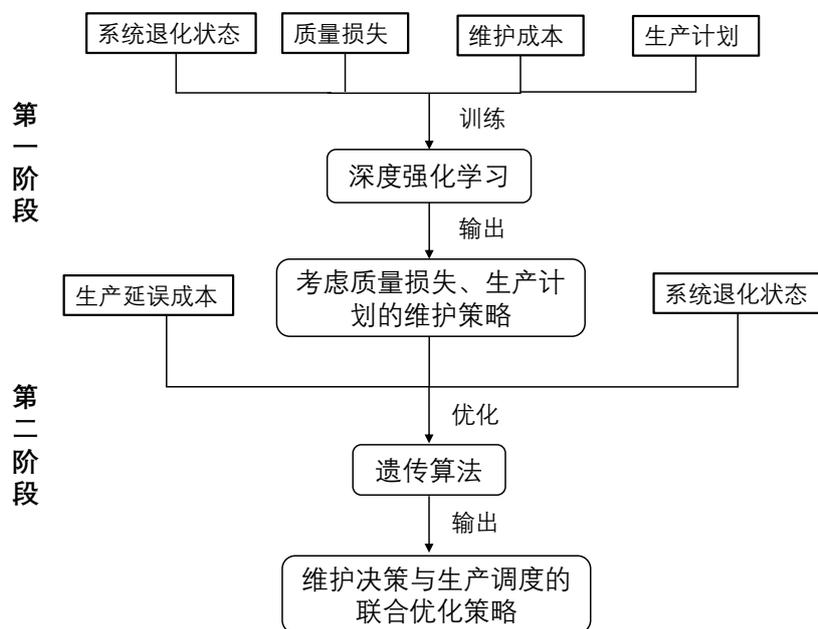


图 4.3 两阶段联合优化方法流程图

为了解决以上难题, 提出了一种两阶段优化方法: 第一阶段基于强化学习实现考虑质量损失、生产计划的维护决策, 将生产信息引入状态空间, 以最小化质量损失、维护成本为目标得到考虑部件相关性和生产计划的多部件系统维护策略。第二阶段利用第一阶段得到的维护策略, 计算生产过程中的维护时间、维护

成本和质量损失，并在优化模型中计算延误成本时考虑维护时间，在优化目标中考虑质量损失、维护成本和延误成本，最后使用遗传算法求解优化问题。流程见图 4.3。

根据第 3 章的内容，考虑产品质量和生产计划的维护决策可以建模为以下马尔可夫决策过程：

1. 状态空间  $S$  是指所有可能状态的集合。 $s_t = (d_1, d_2 \cdots d_m, \tau_t)$ ，其中  $d_i$  是第  $i$  个部件的退化状态， $\tau_t$  是当前作业的剩余加工时间。
2. 动作空间  $A(S)$  是指所有可能动作的集合。在时间  $t$ ，动作  $a_t \in A(S)$ ，每个动作都会影响下一个状态  $s_{t+1}$ 。在这种情况下，动作是选择要维护的部件。每个部件可以选择维护或不维护，因此可用的动作数量为  $2^m$ ，其中  $m$  为部件的数量。
3. 转移概率  $P(s'|a, s)$ ：部件在某一状态  $s$ ，执行动作  $a$  后会以概率  $P(s'|a, s)$  转移至下一状态  $s'$ 。其由式 (3.2) 和 (3.11) 给出。
4. 奖励函数  $r(s_t, a_t)$  是在状态  $s_t$  下采取动作  $a_t$  时系统的奖励或损失。此处，奖励函数取负值表示损失，包括质量损失  $c_q$ 、操作成本  $c_f$  和停机成本  $c_d$ 。因此，总的奖励函数为： $r(s_t, a_t) = c_q(s_t, a_t) + c_f(s_t, a_t) + c_d(s_t, a_t)$ ，其中  $c_q$  可根据(3.7)得到， $c_f$  可根据(3.8)得到， $c_d$  可根据(3.9)得到。

建模为马尔可夫决策过程后使用第 3 章中的算法 3.2 进行训练。

第二阶段中考虑维护的生产调度优化模型建立如下：

$$f = \min \left( \sum_t (c_q(t) + c_m(t)) + \sum_{i=1}^n c_{delay[i]} \right) \quad (4.11)$$

$$\sum_{i=1}^n x_{ij} = 1, j = 1, 2, \dots, n \quad (4.12)$$

$$\sum_{j=1}^n x_{ij} = 1, i = 1, 2, \dots, n \quad (4.13)$$

$$x_{ij} = 0, 1, i, j = 1, 2, \dots, n \quad (4.14)$$

其中公式 (4.11) 描述了联合优化的目标函数；公式 (4.12) 和公式 (4.13) 描述了目标函数中的单机调度约束条件；公式 (4.14) 给出了决策变量的取值范围。

#### 4.4 模型求解

GA 是一种启发式算法，广泛应用于生产调度问题。遗传算法从一个初始种群开始，种群中的每个个体代表一个解。从当前种群中选择一些个体来创建下一

代，并通过一些策略改善当前种群的特征。重复此过程直到算法满足停止条件。遗传算法使用三种算子来获得新一代种群：选择、交叉和变异。选择策略用于从当前种群中选择优秀的个体来创建下一代。通过应用交叉策略，一对个体产生两个后代。突变策略通过对单个个体进行随机改变来创建后代。

#### 4.4.1 染色体编码设计

对于  $x_{ij}$  采用长为  $n$  的自然数编码，且满足  $x_{ij} \in [1, n]$ 。在这套编码体系中，编码的值代表作业编号，编码的位置代表作业在调度序列中的位置。根据联合决策模型的约束条件，同一时刻仅能安排一个作业进行加工，且在任何特定时间，某一作业只能安排在唯一的一个位置，因此编码应该是从 1 到  $n$  的不重复随机排序。基于强化学习的维护和生产调度联合优化模型的染色体设计如图 4.4。



图 4.4 基于强化学习的维护和生产调度联合优化模型染色体编码

基于阈值的机会性维护和生产调度联合优化模型的决策变量除了调度顺序，还包括机会性维护阈值和预防性维护阈值 ( $H_i^o, H_i^p, i = 1, 2, \dots, m$ )。机会性维护阈值  $H_i^o$  和预防性维护阈值  $H_i^p$  的编码分别用 1 位  $[0, H_i^o]$  和  $[0, H_i^p]$  内的实数表示。将其统一编码为长度  $n + 2m$  的染色体。前  $n$  位表示调度顺序编码，与基于强化学习的方法的染色体设计相同。中间  $m$  位表示机会性维护阈值编码，后  $m$  位表示预防性维护阈值编码。编码示意图如图 4.5。



图 4.5 基于阈值的机会性维护和生产调度联合优化模型染色体编码

#### 4.4.2 遗传算法过程

##### (1) 适应值计算

对于某个编码后的染色体，进行解码后可以得到一个对应的联合优化模型的解，计算适应值后可以判断该解是否最优，而适应值即为优化目标式 4.11。由于该目标函数涉及多个部件的随机退化过程，其数值解难以计算，且由于基于强化学习的维护方法中使用了深度学习方法，无法给出维护策略的数学表达式。

因此使用蒙特卡洛估计方法估计适应值，针对某个染色体，解码后可以得到具体的调度顺序和维护策略，给定调度顺序和维护策略后通过模拟多次实验可以得到多次适应值，取其均值作为该染色体的适应值。根据大数定理，当模拟次数够多时，样本均值能够近似总体均值。

(2) 染色体交叉

针对调度序列的编码部分，采用单点交叉的方法。交叉过程需要考虑满足解的约束条件，具体过程如下：产生一个1到n的随机整数作为交叉位置，其中两条染色体中1到交叉点之间的编码保持不变，第一条染色体交叉点到n之间的编码替换为需要替换的编码在第二条染色体对应的序列，第二条染色体交叉点到n之间的编码替换为需要替换的编码在第一条染色体对应的序列。

对于机会性维护阈值和预防性维护阈值的编码部分，使用算术交叉的方法。以  $H_i^o$  为例，针对两个父个体  $H_{i\ old1}^o$  和  $H_{i\ old2}^o$ ，算术交叉后产生两个新个体为  $H_{i\ new1}^o$  和  $H_{i\ new2}^o$ ，交叉过程见式 4.15。对  $H_i^p$  同理。其中  $\mu$  为取值范围在 (0, 1) 之间的常数，交叉操作示意图如图 4.6。

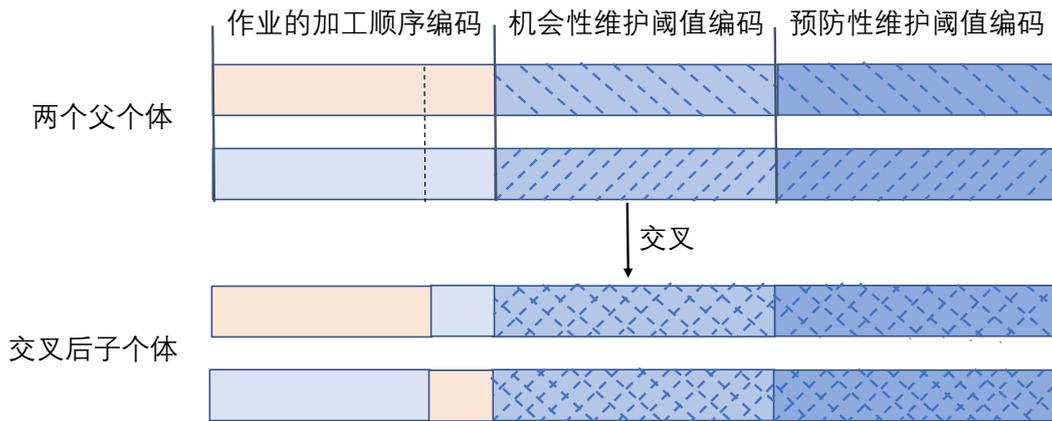


图 4.6 交叉操作示意图

$$\begin{cases} H_{i\ new1}^o = (1 - \mu)H_{i\ old1}^o + \mu H_{i\ old2}^o \\ H_{i\ new2}^o = \mu H_{i\ old1}^o + (1 - \mu)H_{i\ old2}^o \end{cases} \quad (4.15)$$

(3) 染色体变异

针对调度序列的编码部分，变异步骤如下：首先随机选定一个变异位置，随后在这个位置上产生 [1, n] 内的一个整数来替代原有的数值。为了确保满足单机调度的约束条件，对于与变异位置上新数值重复的非变异位置上的数据进行适当的调整；针对机会性维护阈值和预防性维护阈值的编码部分，变异操作为分别随机产生一个 [0,  $H_i^o$ ] 和 [0,  $H_i^i$ ] 的实数替换变异位置上的机会性维护阈值和预防性维护阈值。

## 4.5 实验结果及分析

为了验证所提出的模型的有效性，本节以一个包含 5 个部件的串联系统为例进行说明，其中每个部件都经历了一个随机退化过程，即双参数 Gamma 过程。为了说明所提维护决策与生产调度联合优化方法的有效性，我们将基于强化学习的维护生产调度联合优化方法、基于阈值的机会性维护与生产调度联合优化方法一起，与单独优化维护决策、生产调度的方法比较。为了方便将基于强化学习的维护与生产调度联合优化方法记为联合优化方法一；将单独优化生产调度和基于强化学习的维护的方法记为独立优化方法一；将基于阈值的机会性维护与生产调度联合优化方法记为联合优化方法二；将单独优化生产调度和基于阈值的机会性维护的方法记为独立优化方法二。

### 4.5.1 实验设置

表 4.1 列出了各部件退化过程的参数，退化过程见图 4.7 我们为 5 个部件设置了相同的故障阈值，但它们的退化速率不同。

表 4.1 五部件退化过程参数

参数	描述	部件 1	部件 2	部件 3	部件 4	部件 5
$\alpha$	Gamma 过程形状参数系数	0.2	0.21	0.22	0.3	0.32
$\beta$	Gamma 过程尺度参数	1.0	1.1	0.9	1.35	1.4

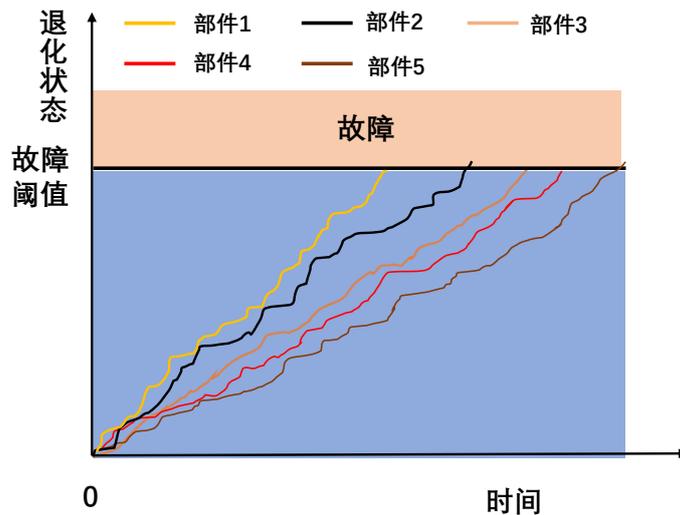


图 4.7 五个部件的退化过程示意图

根据第 3.2.2 节的内容，质量损失由常数  $c$ 、向量  $\mathbf{a}$  和  $\mathbf{b}$ 、矩阵  $\mathbf{U}$  和噪音变量  $\mathbf{V}$  决定。设置  $c = 0.0$ ， $\mathbf{a} = [0.0012, 0.0016, 0.0019, 0.001, 0.0018]$ ， $\mathbf{b} = [0.0022, 0.0014, 0.0017, 0.0021, 0.0008]$ ， $\mathbf{U} = 0.0001\mathbf{I}$ ， $\mathbf{V} \sim N(0, \mathbf{I})$ 。

与维护相关的成本包括两部分：操作成本  $c_f$  和停机成本  $c_d$ 。我们设置每个部件的维护操作成本为 2.5，停机成本  $c_d$  与停机时间有关。停机时间包括维护时间和生产系统设置时间。维护时间与被维护的部件数量成正比，维护的部件越多，所需时间越长。生产系统设置时间代表每次维护动作中断和恢复生产所需的时间。我们设置每次维护动作所需的生产系统设置时间为 2.0，每个部件的维护操作时间成本为 0.5。具体的维护成本见表 4.2。

表 4.2 五部件系统维护成本设置

	部件 1	部件 2	部件 3	部件 4	部件 5
$c_m$	2.5	5.0	7.5	10.0	12.5
$c_d$	3.0	3.5	4.0	4.5	5.0

在 DQN 算法中，我们使用了一个具有两个隐藏层的神经网络。第一个隐藏层有 60 个神经元，第二个隐藏层有 100 个神经元。学习率设置为 0.001，贪心选择概率  $\epsilon$  为 0.945，折扣因子  $\gamma$  为 0.993。遗传算法所使用的参数及其对应值如下：种群大小为 50，交叉概率设为 0.7，变异概率设定为 0.2，最大迭代代数为 100。

#### 4.5.2 实验结果

取表 4.3 中调度作业的加工时间、截止时间及所对应的惩罚权重，利用四种优化策略进行实验计算总成本，实验结果见表 4.4。

表 4.3 作业相关信息

$j$	$p_j$	$td_i$	$w_i$	$j$	$p_j$	$td_i$	$w_i$
1	44	150	1.8	5	48	200	1.1
2	37	150	1.6	6	30	200	0.8
3	46	150	1.4	7	60	200	1.0
4	55	200	0.9	8	50	300	0.5

根据表 4.4 中的结果，联合优化方法一的总成本为 383.39，比独立优化方法一的总成本 404.94 少 21.55；联合优化方法二的总成本为 417.21，比独立优化方法二的总成本 458.88 少 41.67。这说明不管是使用强化学习的维护方法，还是基于阈值的机会性维护方法，都是与生产调度联合优化比单独优化效果更好。且不管是延误成本还是总成本，都是联合优化方法比独立优化方法效果更好。这说明

所提出的联合优化方法能够同时考虑维护决策与生产调度，在维护决策时能够考虑到生产顺序对维护成本的影响，在生产调度能考虑维护对完成时间的影响，从而实现总成本降低。

表 4.4 不同优化方法结果

策略	调度结果	延误成本	总成本
联合优化方法一	3-1-2-6-5-7-4-8	<b>323.88</b>	<b>383.39</b>
独立优化方法一	1-3-2-5-6-4-8-7	341.80	404.94
联合优化方法二	2-3-1-5-6-4-7-8	349.16	417.21
独立优化方法二	1-3-2-5-6-4-8-7	387.5	458.88

比较方法一和方法二，二者的调度结果不同，这验证针对不同的维护策略下的联合优化问题，最优调度结果是不同的。从整体来看，联合优化方法一比联合优化方法二延误成本少 25.28，总成本少 33.82；独立优化方法一比独立优化方法二延误成本少 45.7，总成本少 53.94。这也进一步证明第 3 章中所提出的考虑产品质量与生产计划的维护决策方法的有效性。

为了说明所提出的联合优化方法在不同作业规模下依然具有有效性，设置待加工的作业数量增加为 20。将四种优化方法进行对比，结果如表 4.5 所示

表 4.5 大规模作业优化方法结果

调度作业数量	策略	总成本
20	联合优化方法一	<b>3914.09</b>
20	独立优化方法一	4010.76
20	联合优化方法二	4328.73
20	独立优化方法二	4514.65

在 20 项调度作业情况下，联合优化方法一的总成本为 3914.09，比独立优化方法一的总成本 4010.76 少 96.67；联合优化方法二的总成本为 4328.73，比独立优化方法二的总成本 4514.65 少 185.92。对比表 4.4 中 8 项调度作业的结果，总成本降低幅度更大，这说明所提出联合优化方法在大规模作业调度情况下仍然有效，且规模越大效果越好。

## 4.6 本章小结

本章提出一种适用于多部件系统的两阶段联合优化方案，第一阶段以维护决策为主，在维护中考虑生产计划以及产品质量，利用强化学习实现能够根据生产计划变化动态调整的维护策略。第二阶段利用第一阶段得到的维护策略，计算生产过程中的维护时间、维护成本和质量损失，以最小化质量损失、维护成本和延误成本为优化目标建立调度优化模型。最终通过实验表明，联合优化方法优于单独优化方法，且所提联合优化方法优于基于阈值的机会性维护方法。

## 第5章 生产计划扰动下并行多机系统的动态调度方法

第3、4章主要考虑的是静态调度问题，即假设所有作业信息和车间状态事先完全确定，并没有考虑到实际生产环境中的扰动因素。为解决静态调度方法在存在扰动事件的生产环境中不可行的问题，本章以并行多机系统为研究对象，针对生产计划扰动事件设计了相应的动态调度方法。具体内容包括并行多机动态调度问题建模、设计重调度驱动方式、针对不同扰动事件设计调度修复策略以及相关仿真实验。

### 5.1 引言

随着消费者的多元化需求日益显著，企业接收到的订单结构普遍转向小批量、多批次定制化模式<sup>[81]</sup>，这对企业提供个性化产品的灵活性和服务水平提出了更高的要求。生产车间正逐步向多台并行机器协同加工的方式转变，以提升生产速度和效率。同时，由于定制化生产模式，生产过程会经历不可预测的实时扰动事件，例如新作业到达、截止日期更改等。这些扰动事件可能会导致进度表发生变化，使得之前可行的生产计划失去可行性。若能针对这些扰动事件有效调整调度计划，可较大地提高企业的生产效益<sup>[82]</sup>。为保证生产过程顺利进行，需要构建适应动态环境的动态调度策略。动态调度策略是指在生产过程遭遇突发扰动事件，原有调度计划失效的情况下，依据即时扰动因素调整原计划，确保生产系统维持稳定运行的一种生产管理策略。这一策略可分为三个类别：完全反应式调度、鲁棒性调度和预-反应式调度<sup>[56]</sup>。

完全反应式调度策略依赖于实时环境变化作出即时决策，因其决策基础主要建立在局部实时信息之上，可能会忽视全局调度优化的必要性；鲁棒性调度策略基于当前已知信息或对未来部分可预见信息的估计来构建调度方案，其侧重于对预计扰动的防范，对于实际扰动与预期不符的情形响应和适应能力相对较弱；预-反应式调度是现代制造业中最常见的动态调度机制，它结合了完全反应式和鲁棒性调度的优点。预-反应式调度首先基于现有信息形成初步的静态调度方案，然后在调度周期或者在扰动事件触发时，运用重调度技术对初始方案进行适当修正。在本章研究中，所涉及的扰动事件主要包括交货期调整、订单取消和新增订单，这类事件非高频发生且难以精确预测，使得完全反应式调度和鲁棒性调度效果受限。

鉴于此，本章以并行多机系统为研究对象，提出了在生产计划扰动事件交货期变更、订单撤销和订单增加情况下的预-反应式动态调度方法。将产品质量、

维护决策和生产调度的联合优化方案作为静态调度方法,以最小化质量损失、维护成本和延误成本调度目标,建立动态调度模型。为了解决静态调度随时间累积的误差以及扰动事件对生产带来的影响,设计了基于周期和事件混合驱动策略,并针对交货期变更、订单撤销和订单增加这三种扰动事件分别设计了相应的调度修复策略,实现了处理生产计划扰动的同时保持生产过程的稳定性。

本章结构安排如下:第5.2节介绍了并行多机系统动态调度问题建模;第5.3节介绍了并行多机系统维护与生产静态调度方法;第5.4节给出了并行多机系统维护与生产动态调度方法;第5.5节给出了相关的实验结果及分析;第5.6节总结了本章的研究内容。

## 5.2 问题建模

并行多机多部件系统维护与生产联合动态调度问题具体可以描述为:系统由  $l$  台相同机器组成,每台机器由  $m$  个关键部件串联组成。这  $l$  台相同机器可以并行工作,每个作业可以在任一机器上加工,加工过程需要该机器的所有部件共同参与。所有部件在加工过程中会磨损退化,且任一部件退化状态超过故障阈值会导致该机器停机。有  $n$  个作业待加工,每个作业都有对应的加工时长、截止时间和权重。权重代表该作业的紧急程度,权重越大越紧急。考虑到扰动事件,在  $t$  时刻随机发生交货期变更、订单撤销和订单增加的情况。

第 4 章中的单机系统联合调度问题中的相关假设对并行多机系统的联合调度问题同样成立。考虑到扰动事件的动态调度问题,假设在  $t$  时刻扰动事件发生时,需要完成当前机器的加工的作业才能安排其他的作业,因此在计算作业完成时间时不再从零开始计算。

并行多机系统生产调度的决策变量是 0-1 变量  $y_{ijk}, i, j = 1, 2, \dots, n, k = 1, 2, \dots, l$ 。  $y_{ijk} = 1$  代表第  $k$  台机器上的调度顺序为  $i$  的作业序号为  $j$ 。决策变量的不同组合将导致不同的调度安排。联合优化的目标是最小化质量损失、维护成本和加权延误成本。加权延误成本计算如下:

第  $k$  台机器上的调度顺序为  $i$  的作业的加工时间为:

$$p_{[ik]} = \sum_{j=1}^n p_j y_{ijk} \quad (5.1)$$

第  $k$  台机器上的调度顺序为  $i$  的作业的完成时间为:

$$c_{[ik]} = \sum_{z=1}^i (p_{[zk]} + tm_{[zk]}) + (i-1)T_m + e_k \quad (5.2)$$

事实上,第  $k$  台机器上的调度顺序为  $i$  的作业的完成时间由四部分组成:第一部分是第  $k$  台机器上的前  $i$  个作业所需的生产时间  $\sum_{z=1}^i p_{[zk]}$ ;第二部分是第  $k$  台机器在工作过程中的维护时间之和  $\sum_{z=1}^i tm_{[zk]}$ ,其中  $tm_{[zk]}$  为第  $k$  台机器上的调度顺序为  $z$  的作业加工时间段总的维护时间;第三部分是相邻作业之间的设置时间之和  $(i-1)T_m$ ;第四部分是扰动事件发生时当前机器  $k$  加工完当前作业然后能够加工其他作业所需的时间  $e_k$ 。

第  $k$  台机器上的调度顺序为  $i$  的作业的惩罚权重为:

$$w_{[ik]} = \sum_{j=1}^n w_j y_{ijk} \quad (5.3)$$

第  $k$  台机器上的调度顺序为  $i$  的作业的截止时间为:

$$td_{[ik]} = \sum_{j=1}^n td_j y_{ijk} \quad (5.4)$$

因此第  $k$  台机器上的调度顺序为  $i$  的作业的加权延误成本为:

$$c_{delay[ik]} = w_{[ik]} \max(0, c_{[ik]} - td_{[ik]}) \quad (5.5)$$

多部件系统的联合优化模型建立如下:

$$f = \min(\sum_t c_q(t) + c_m(t) + \sum_{i=1}^n \sum_{k=1}^l c_{delay[ik]}) \quad (5.6)$$

$$\sum_{i=1}^n \sum_{k=1}^l y_{ijk} = 1, j = 1, 2, \dots, n, k = 1, 2, \dots, l \quad (5.7)$$

$$\sum_{j=1}^n \sum_{k=1}^l y_{ijk} = 1, i = 1, 2, \dots, n, k = 1, 2, \dots, l \quad (5.8)$$

$$y_{ijk} = 0, 1, i, j = 1, 2, \dots, n, k = 1, 2, \dots, l \quad (5.9)$$

其中公式 (5.6) 描述了联合优化的目标函数;公式 (5.7) 和公式 (5.8) 描述了目标函数中的单机调度约束条件,也就是说同一时刻仅能安排一个作业进行操作,且在任何特定时间,某一作业只能安排在唯一的一个位置;而公式 (5.9) 则给出了决策变量的取值范围。

### 5.3 并行多机系统维护与生产静态调度

动态调度问题是以静态调度问题为基础发展而来的，为了解决动态调度问题，研究者们开发了一系列方法论和技术手段，旨在将原本复杂的动态问题转换成一系列便于处理的静态子问题或近似静态问题。例如，采用滚动时域方法，通过不断迭代更新调度方案，以便在连续变化的条件下尽可能接近最优解。这样，动态调度问题就能够借鉴静态调度问题的部分优化思路，并在此基础上融入适应动态环境变化的灵活机制，最终得以有效求解。下文给出并行多机系统的维护与生产静态调度方法。

使用第4章中基于强化学习的维护与生产调度联合优化方法。并行多机多部件系统静态调度的思路与方法和第4章提出的单机多部件系统优化方法基本相同，不同点在于二者的决策变量不同。后者的决策变量决定作业的调度顺序，而前者的决策变量不仅要决定作业的调度顺序，还要决定每个作业在哪个机器上工作。因此，针对求解并行多机多部件系统调度问题的遗传算法做以下改动：

#### (1) 染色体编码

使用长为  $2n$  的一维染色体编码，前  $n$  位编码为调度顺序编码，后  $n$  位为机器选择编码。具体而言，将染色体分为两部分，前  $n$  位编码与单机调度的调度顺序编码相同，为不重复的  $[1, n]$  的自然数编码，代表作业调度顺序。后  $n$  位编码为可重复的  $[1, c]$  的自然数编码，其第  $k$  位的编码代表调度顺序编码中第  $k$  位编码代表的作业由哪台机器上工作。也即机器选择编码中有相同编码的作业代表由同一台机器处理，同一台机器处理的作业的顺序由调度顺序编码给出，图 5.1 给出一个 3 台机器 5 个调度作业的染色体编码例子。

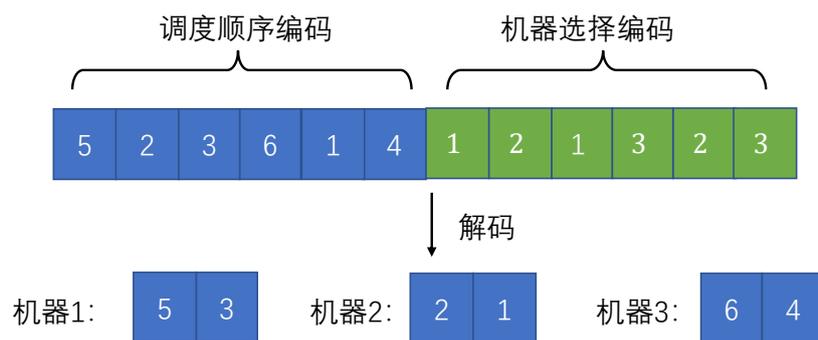


图 5.1 并行多机调度染色体编码示意图

#### (2) 染色体交叉

针对调度顺序编码部分，交叉过程为：产生一个 1 到  $n$  的随机整数作为交叉点，两条染色体中 1 到交叉点之间的编码保持不变，第一条染色体交叉点到  $n$  之间的编码替换为需要替换的编码在第二条染色体对应的序列，第二条染色体交

又点到  $n$  之间的编码替换为需要替换的编码在第一条染色体对应的序列；针对机器选择编码部分，交叉过程为：产生一个 1 到  $n$  的随机整数作为交叉点，两条染色体中 1 到交叉点之间的编码保持不变，剩下部分编码互相交换。

### (3) 染色体变异

针对调度顺序编码部分，变异步骤如下：首先随机选定一个变异位置，随后在这个位置上产生  $[1, n]$  内的一个整数来替代原有的数值。为了确保满足单机调度的约束条件，对于与变异位置上新数值重复的非变异位置上的数据进行适当的调整；针对机器选择编码部分，变异过程为随机产生  $[1, l]$  内的一个整数替换变异位置上的数值。

## 5.4 并行多机系统维护与生产动态调度

随着现代生产加工环境日益复杂多变，传统的静态调度方案逐渐暴露出其局限性，在整个生产流程中任何环节出现扰动，都可能显著影响原有调度计划的执行。因此，面对瞬息万变的生产场景，构建一套动态调度策略显得至关重要。

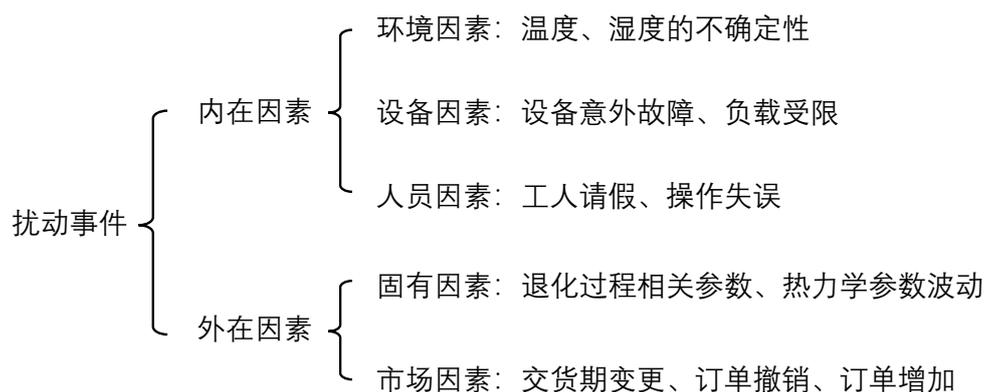


图 5.2 扰动因素分类

在生产过程中，面临的扰动事件可概括性地划分为内在因素与外在因素两大类，如图 5.2 所示。内在因素囊括了环境因素、设备因素以及人员因素三个方面：环境因素主要指加工设备所处作业环境参数的不确定性，诸如空气湿度、温度等变化；设备因素则涵盖了设备意外故障、设备维护等不确定性因素；人员因素则包括操作员的操作错误、临时请假或缺勤等情况产生的不确定性影响。外在因素包括了固有因素与市场因素：固有因素指生产车间内部所固有的设备退化过程相关参数、热力学参数波动等物理化学性质引起的不确定性；市场因素指由市场环境动态演变而导致的生产计划扰动，诸如交货期限的更改、既有订单的撤销以及新订单的增加等商业行为。

在这众多类型的扰动中，设备因素与市场因素对调度效果产生尤为显著的影响。本章采用的联合调度策略已经考虑了维护过程，能够应对设备意外故障等

问题，因此本章着重探讨市场因素带来的生产计划扰动，即交货期变更、订单撤销和订单增加这三种情况。下文给出动态调度中主要关心两个问题的解决思路：何时启动重调度的过程，以及如何制定针对扰动事件的重调度方法。

### 5.4.1 重调度驱动机制设计

针对何时进行重调度的问题，目前有三种策略：基于周期的驱动、基于事件的驱动以及基于周期和事件的混合驱动。基于周期驱动策略采取了滚动时间窗口的概念，按期制定调度方案，形成一种循环性的决策过程。基于周期驱动策略确保了生产活动的连贯性和稳定性，然而，它对扰动事件的响应存在一定的滞后性，无法实时地处理那些可能导致生产中断的重大扰动事件。

基于事件的驱动策略则是围绕实际发生的扰动事件展开反应式调度。在本章研究框架内，交货期变更、订单取消和新增订单均可视为触发重新调度的关键扰动事件。事件驱动机制的核心优势在于其能够即时捕捉并应对生产流程中的突发扰动，消除滞后响应的问题。

基于周期和事件的混合驱动策略融合了两种驱动策略的优点，将基于周期的驱动策略与基于事件的驱动策略有机结合。这种策略允许重调度行动既由调度周期的自然更迭触发，也能因应实时扰动事件的发生而随时启动。

在本章考虑的生产系统中，由于机器部件的退化过程为随机退化过程，因此在联合调度优化问题中，给出的调度结果是期望意义下的最优。也即：

$$f = \min_{d_i(t) \sim \mathbb{E}[d_i(t)], i=1, \dots, m} \mathbb{E} \left[ \sum_t c_q(t) + c_m(t) + \sum_{i=1}^n \sum_{k=1}^l c_{delay[ik]} \right] \quad (5.10)$$

其中  $d_i(t)$  为部件  $i$  的退化状态，质量损失  $c_q(t)$ 、维护成本  $c_m(t)$  和延误成本  $c_{delay[ik]}$  均与设备的退化状态有关。

而随着生产的进行，机器的各个部件退化过程并不会遵循期望的退化轨迹，即存在概率事件：

$$P(d_i(t) > \mathbb{E}[d_i(t)]) \quad (5.11)$$

$$P(d_i(t) < \mathbb{E}[d_i(t)]) \quad (5.12)$$

表明存在概率某个部件退化量远大于期望退化量导致该部件比期望故障时间更早，也存在概率某个部件退化量远小于期望退化量导致该部件比期望故障时间更晚。这种与期望值之间的误差会随着时间增加而累积，导致离调度时刻很远的部分调度结果并不合理。

基于周期的驱动策略可以很好地解决这个问题，通过周期性的调度可以抹除随时间累积的误差。设置调度周期  $T_{sch}$ ，每经过  $T_{sch}$  时刻就更新设备退化状

态并重新优化调度过程：

$$Y(t + T_{sch}) = \arg \min_Y f(Y, d_1(t + T_{sch}), \dots, d_m(t + T_{sch})) \quad (5.13)$$

其中  $Y = (y_{ijk}, i, j = 1, 2, \dots, n, k = 1, 2, \dots, l)$  代表调度问题的决策变量， $t + T_{sch}$  代表周期调度时间点。

另一方面，为了处理交货期变更、订单撤销和订单增加这些扰动事件，需要使用基于事件的驱动策略：

$$Y(t + \Delta t) = \arg \min_Y f(Y, d_1(t + \Delta t), \dots, d_m(t + \Delta t), Event) \quad (5.14)$$

其中  $t + \Delta t$  为扰动事件发生时间点， $Event$  代表具体的扰动事件类型和参数。

最终，通过结合周期驱动与事件驱动两种策略，设计了一种混合驱动的优化调度方法，以实现生产系统的动态适应和高效运行。

#### 5.4.2 重调度方法设计

如何制定针对扰动事件的重调度方法的问题，目前有两种主要的重调度策略：完全重调度和调度修复。

完全重调度是指当扰动发生时，使用静态调度策略重新生成新的安排。完全重调度表述可以表示为：

给定扰动事件  $Event$  发生时的系统状态，完全重调度的目标是在满足新约束条件下找到一个全局最优的调度方案  $Y^*$ ：

$$Y^* = \arg \min_Y f(Y, Event) \quad (5.15)$$

其中， $f(Y, Event)$  是在考虑扰动事件  $Event$  后的调度目标函数。

完全重调度能给出扰动事件发生后的最佳解决方案，但这些解决方案在实践中很少实现。原因在于完全重调度给出的新调度结果往往与原调度方案有很大差异，这可能会导致工厂计划的不稳定和缺乏连续性。此外完全重调度需要大量的计算时间，其反应时间较长。

调度修复是对当前调度策略进行针对性优化与调整的过程。当系统遭遇扰动时，调度修复策略会在初始调度方案基础上，仅针对那些受到扰动事件直接影响或间接影响的作业集合进行重新调度。相比完全重调度，调度修复策略更加注重实际操作中的效率和可行性。针对特定扰动事件  $Event$ ，调度修复问题可以被定义为：

假设原始调度方案为  $Y_0$ ，扰动事件影响的作业集合为  $J_0$ ，那么修复调度的目标是求解局部最优解  $Y'$ ：

$$Y' = \arg \min_{Y \sim J_0} f(Y, J_0, Event) \quad (5.16)$$

其中,  $Y \sim J_0$  表示决策变量只允许改变作业集合  $J_0$  的调度顺序;  $f(Y, J_0, Event)$  表示在初始调度方案  $Y_0$  的基础上, 针对受扰动影响的作业集合  $J_0$  调整得到的局部目标函数。

针对具体扰动事件的情况, 可以进一步细化:

(1) 交货期变更: 若某作业  $j$  的交货期发生变化, 由于考虑的是多机系统, 因此只影响与该作业在同一台机器上工作的作业, 重调度问题可退化为单机调度。调度修复问题可表示为:

$$Y' = \arg \min_{Y \sim M_j} f(Y, M_j, \Delta T_j) \quad (5.17)$$

其中,  $M_j$  表示作业  $j$  所在的机器;  $Y \sim M_j$  表示决策变量只允许机器  $M_j$  上的调度顺序;  $\Delta T_j$  是交货期变化量;  $f(Y, M_j, \Delta T_j)$  表示在机器  $M_j$  上对作业  $j$  及相关作业进行局部调整的目标函数。

(2) 订单撤销: 若某作业  $j$  的订单被撤销, 与交货期变更类似, 订单撤销只影响与该作业在同一台机器上工作的作业。调度修复问题可表示为:

$$Y' = \arg \min_{Y \sim M_j} f(Y, M_j) \quad (5.18)$$

(3) 订单增加: 若新增作业  $a$ , 与交货期变更和订单撤销不同, 新的作业并没有出现在原本的调度方案中, 需要先选择由哪台机器处理该作业。可知多机系统的最优的调度方案是使所有机器的作业工作时长均等, 因此选择在原有调度安排中完工时间最短的机  $M_a$  来处理新增加的作业:

$$M_a = \arg \min_k (\max_i c_{ik}) \quad (5.19)$$

然后, 在选定的机器  $M_a$  上进行调度修复:

$$Y' = \arg \min_{Y \sim M_a} f(Y, M_a) \quad (5.20)$$

以上所有的局部重调度都会运用第 4 章提出的生产与维护联合调度策略进行优化。

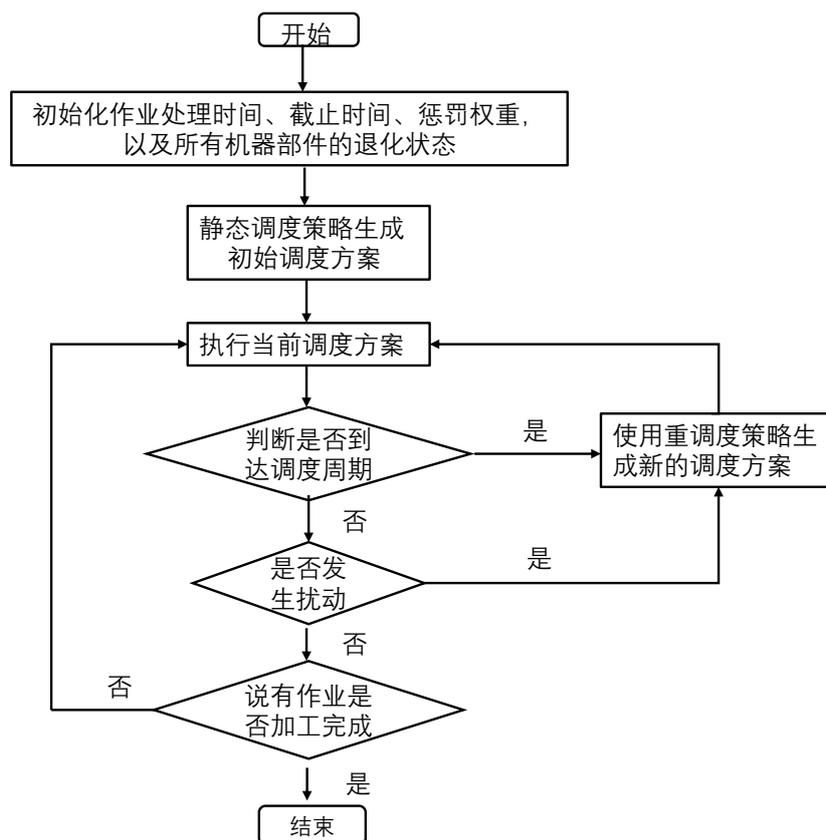


图 5.3 动态调度流程图

### 5.4.3 动态调度流程

根据上文给出的重调度驱动机制和重调度方法，可以得到在交货期变更、订单撤销和订单增加这三种扰动情况下的动态调度策略。具体流程图见图 5.3，算法流程如下：

步骤 (1)：初始化，根据订单初始化所有作业的处理时间、截止时间、惩罚权重，以及所有机器部件的退化状态。

步骤 (2)：使用并行多机系统维护与生产静态调度策略得到初始调度方案。

步骤 (3)：执行当前调度方案。

步骤 (4)：基于周期的重调度驱动，当生产时间到达设置的重调度周期时，使用静态调度策略产生新的调度方案。

步骤 (5)：基于事件的重调度驱动，若当前时刻出现扰动事件，根据扰动事件的类别执行相应的调度修复策略，跳转步骤 (3)；若当前时刻并未出现扰动事件，跳转步骤 (6)。

步骤 (6)：判断所有作业是否全部加工完成，若是，流程结束，否则跳转步骤 (3)。

## 5.5 实验结果及分析

为了验证所提出的模型的有效性，本节以一个包含3个相同的5-部件系统为例进行说明，比较在交货期变更、订单撤销和订单增加这三种扰动情况下的静态调度策略、调度修复策略和完全重调度策略的效果。其中5-部件系统的退化过程相关参数、维护与生产联合调度优化策略相关参数均与第4章中的实验设置相同。

本节的仿真实验中以18个作业为初始作业，表5.1给出了18个初始调度作业的加工时间、截止时间。使用基于周期的驱动策略的目的是消除部件的随机退化过程与期望退化轨迹的累积误差，根据所有部件退化到失效所需的时间，设置调度周期为150。

表 5.1 初始调度作业相关信息

作业编号	加工时间	截止时间	作业编号	加工时间	截止时间
1	44	60	10	30	150
2	37	60	11	44	150
3	46	60	12	37	150
4	55	60	13	46	150
5	48	60	14	55	150
6	30	60	15	45	150
7	60	60	16	30	250
8	50	60	17	60	250
9	48	60	18	50	250

根据表5.1中的信息，并行多机系统维护与生产静态调度策略和动态调度策略的调度结果的甘特图如图5.4和5.5，详细结果见表5.2。

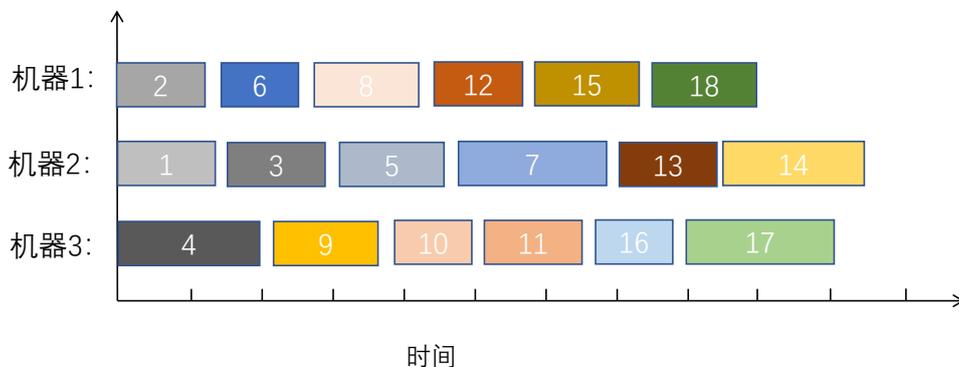


图 5.4 静态调度甘特图

根据表5.2，静态调度策略的总成本为1132.88，动态调度策略的总成本为1065.71，比静态调度策略少67.17。这说明在无扰动的情况下，使用基于周期驱

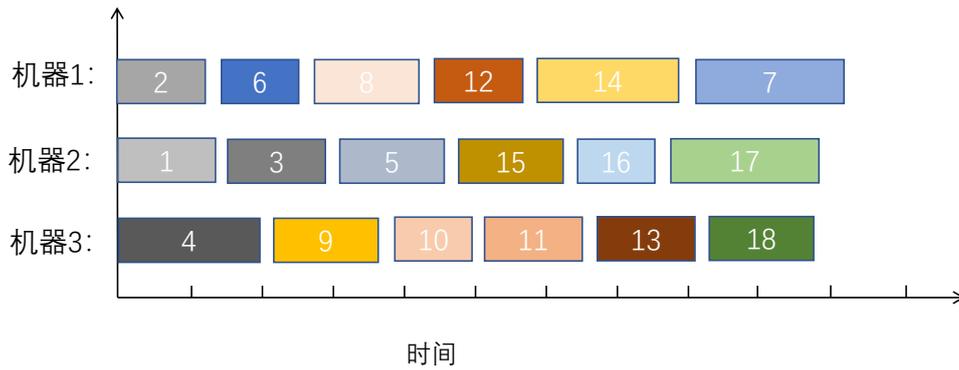


图 5.5 动态调度甘特图

表 5.2 无扰动下静态调度与动态调度结果

静态调度			动态调度		
机器编号	调度顺序	总成本	机器编号	调度顺序	总成本
机器一	2-6-8-12-15-18	1132.88	机器一	2-6-8-12-14-7	1065.71
机器二	1-3-5-7-13-14		机器二	1-3-5-15-16-17	
机器三	4-9-10-11-16-17		机器三	4-9-10-11-13-18	

动的动态调度策略比静态调度策略更优，动态调度能够在调度周期更新系统相关信息，从而更新调度结果，消除随时间累积的误差。

为了比较完全重调度策略和调度修复策略的优劣，下面给出在交货期变更、订单撤销和订单增加这三种扰动情况下的调度结果。

交货期变更：在  $t=150$  时，作业 17 的交货期提前，由 250 提前至 150，调度修复策略与完全重调度策略的调度结果的甘特图如图 5.6和图 5.7，详细结果见表 5.4；订单撤销：交货期变更：在  $t=150$  时，作业 15 的订单取消，调度修复策略与完全重调度策略的调度结果的甘特图如图 5.8和图 5.9，详细结果见表 5.5；订单增加：交货期变更：在  $t=150$  时，新作业订单达到，作业编号记为 19，其加工时间为 60，截止时间为 250，调度修复策略与完全重调度策略的调度结果的甘特图如图 5.10和图 5.11，详细结果见表 5.6。交货期变更、订单撤销和订单增加这三种扰动情况下的静态调度策略结果见表 5.3。

表 5.3 交货期变更、订单撤销和订单增加下静态调度结果

扰动类型	调度策略	总成本
交货期变更	静态调度	1244.07
订单撤销	静态调度	1011.05
订单增加	静态调度	1263.03

根据以上结果，交货期变更扰动下静态调度的总成本为 1244.07，调度修复策略的总成本为 1208.21，完全重调度策略的总成本为 1193.45；订单撤销扰动下

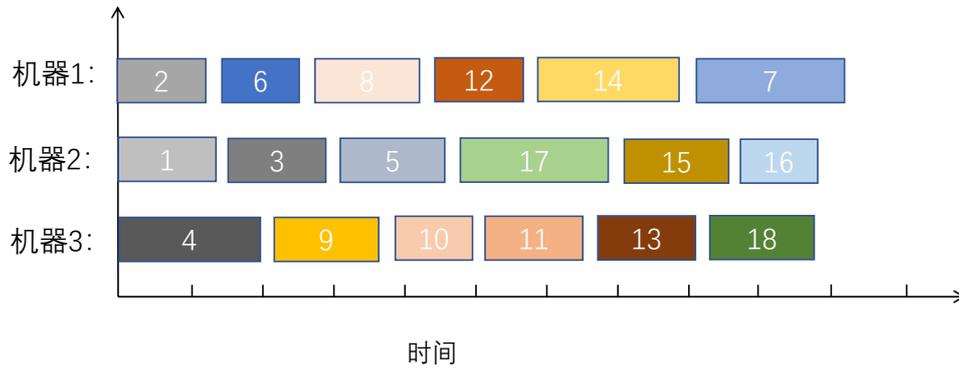


图 5.6 交货期变更扰动调度修复策略甘特图

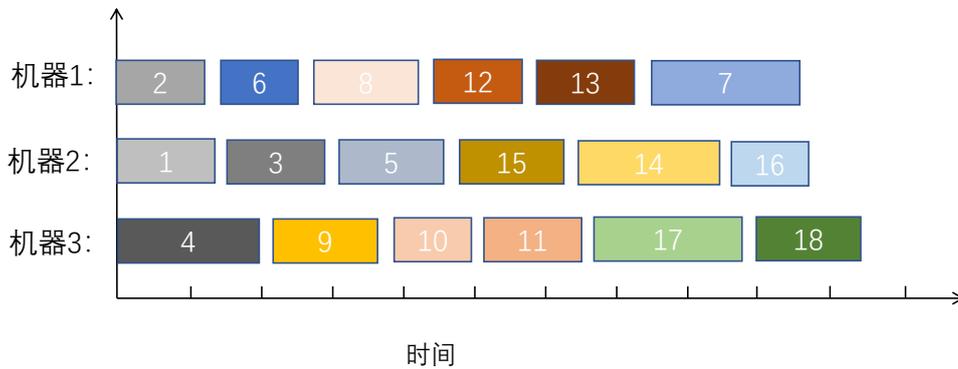


图 5.7 交货期变更扰动完全重调度策略甘特图

表 5.4 交货期变更扰动下调度修复策略与完全重调度策略结果

调度修复策略			完全重调度策略		
机器编号	调度顺序	总成本	机器编号	调度顺序	总成本
机器一	2-6-8-12-14-7		机器一	2-6-8-12-13-7	
机器二	1-3-5-17-15-16	1208.21	机器二	1-3-5-15-14-16	1193.45
机器三	4-9-10-11-13-18		机器三	4-9-10-11-17-18	

表 5.5 订单取消扰动下调度修复策略与完全重调度策略结果

调度修复策略			完全重调度策略		
机器编号	调度顺序	总成本	机器编号	调度顺序	总成本
机器一	2-6-8-12-14-7		机器一	2-6-8-12-7-18	
机器二	1-3-5-16-17	941.8	机器二	1-3-5-13-16-14	899.83
机器三	4-9-10-11-13-18		机器三	4-9-10-11-17	

静态调度的总成本为 1011.05，调度修复策略的总成本为 941.8，完全重调度策略的总成本为 899.83；订单增加扰动下静态调度的总成本为 1263.03，调度修复策略的总成本为 1243.65，完全重调度策略的总成本为 1210.54。从结果上看完全重调度策略效果最好，调度修复策略其次，这也符合预期。完全重调度策略在扰动

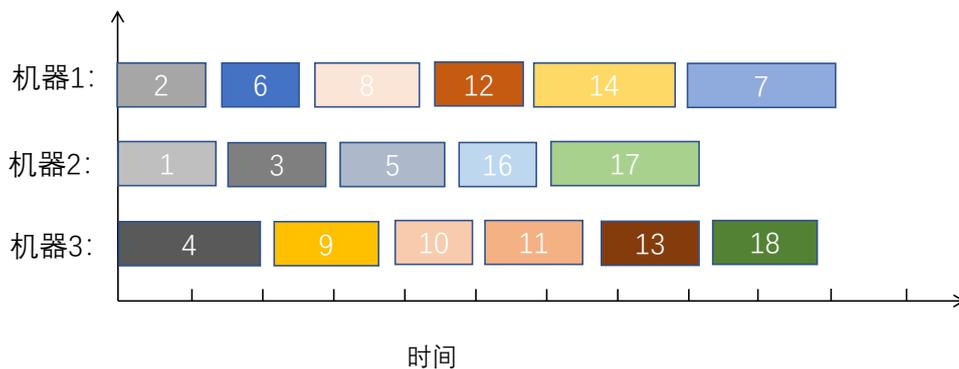


图 5.8 订单取消扰动调度修复策略甘特图

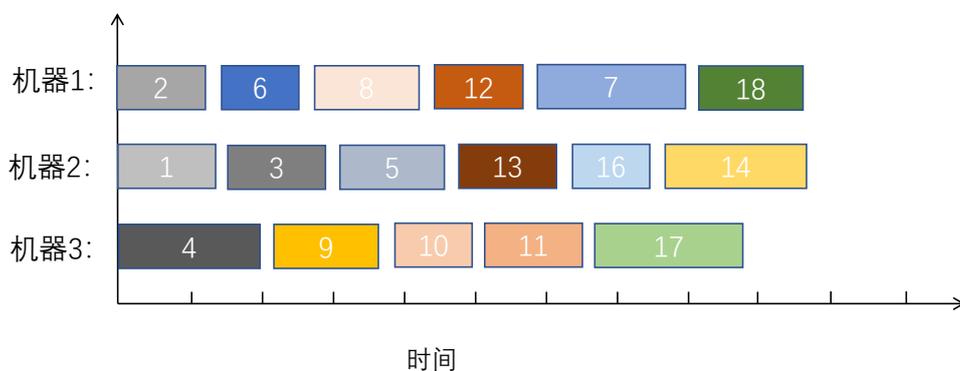


图 5.9 订单取消扰动完全重调度策略甘特图

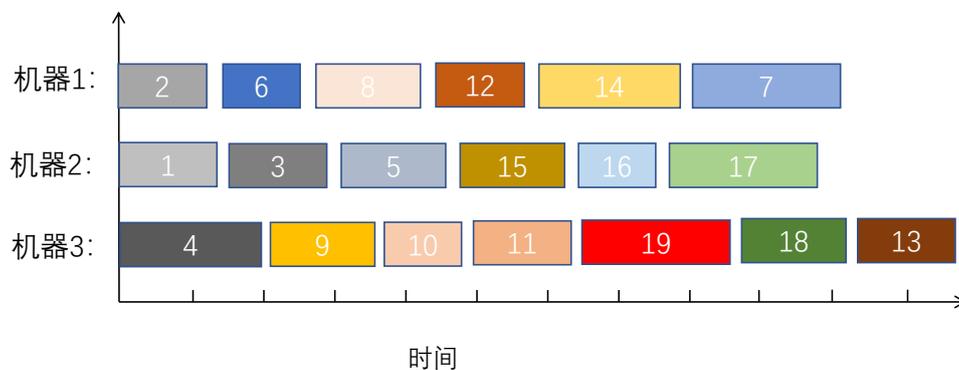


图 5.10 订单增加扰动调度修复策略甘特图

表 5.6 订单增加扰动下调度修复策略与完全重调度策略结果

调度修复策略			完全重调度策略		
机器编号	调度顺序	总成本	机器编号	调度顺序	总成本
机器一	2-6-8-12-14-7	1243.65	机器一	2-6-8-12-16-19-17	1210.54
机器二	1-3-5-15-16-17		机器二	1-3-5-13-18-7	
机器三	4-9-10-11-19-18-13		机器三	4-9-10-11-15-14	

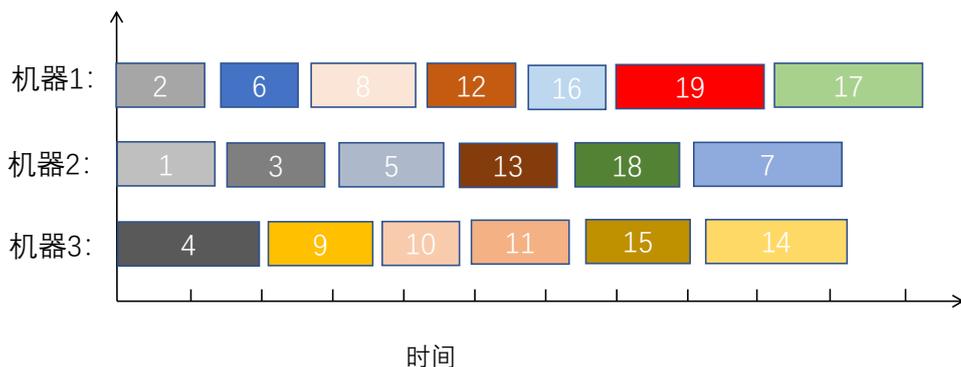


图 5.11 订单增加扰动完全重调度策略甘特图

发生时在总成本上能达到最优解，但对比原调度结果有较大改动，这对生产过程的稳定性造成一定影响。而调度修复策略在扰动发生时仅更改部分调度结果，就能够很好地处理扰动事件，综合来看，调度修复策略更适用于实际生产环境。

## 5.6 本章小结

本章针对生产计划扰动下并行多机系统的动态调度问题，提出了一种基于周期和事件混合驱动的调度修复方法。对并行多机系统的调度模型建模，并在模型中考虑了扰动事件。分别针对交货期变更、订单撤销和订单增加这三种扰动事件设计了相应的调度修复策略。最终通过实验表明，所提出的动态调度方法在无扰动情况以及有扰动情况下均优于静态调度策略。

## 第6章 总结与展望

### 6.1 工作总结

论文以多部件系统为研究对象,设计了产品质量、维护决策与生产调度的联合优化方法。优化过程中将产品质量退化纳入到维护决策中,考虑维护决策与生产调度之间的相互关系,兼顾维护成本、质量损失和交货期,实现整体效益的最优。论文的主要研究工作总结如下:

(1) 针对固定生产计划下多部件系统的产品质量和维护决策联合优化问题,对部件的退化过程进行建模,并分析了多部件系统退化状态与产品质量之间的关系,以及生产计划对维护决策的影响,提出了一种基于强化学习的维护决策方法。将维护决策建模为马尔可夫决策过程,在状态空间引入当前生产批次剩余时间的概念,在奖励函数中加入了与健康状态相关联的质量损失以及与生产计划相关的停机成本,然后使用深度强化学习算法求解。实验表明所提维护方法能够实现降低维护成本的同时减少质量损失,并能够根据生产计划变化动态调整维护策略。

(2) 针对多部件系统的产品质量、维护决策和生产调度联合优化问题,设计了一种两阶段联合优化方法。第一阶段以维护决策为主,在维护中考虑生产计划以及产品质量,利用强化学习实现能够根据生产计划变化动态调整的维护策略。第二阶段利用第一阶段得到的维护策略,计算生产过程中的维护时间、维护成本和质量损失,在调度模型中计算质量损失、维护成本和延误成本。实验表明所设计的两阶段联合优化方法能够同时优化产品质量、维护决策和生产调度。

(3) 针对生产计划扰动下并行多机系统的动态调度问题,提出了一种基于周期和事件混合驱动的调度修复方法。对并行多机系统的动态调度问题建模,并在模型中考虑了交货期变更、订单撤销和订单增加这些扰动事件。为了解决静态调度随时间累积的误差以及扰动事件对生产带来的影响,设计了基于周期和事件混合驱动策略。分析了两种不同的重调度方法,并针对交货期变更、订单撤销和订单增加这三种扰动事件分别设计了相应的调度修复策略。实验表明所提动态调度方法在保持生产过程的稳定性的同时能够及时处理生产扰动事件。

### 6.2 研究展望

论文面向多部件生产系统,研究了产品质量、维护决策与生产调度联合优化方法,包括质量损失建模、维护决策与生产调度关系分析、联合优化模型的建立和优化算法的实现,并设计了并行多机系统的动态调度方法,但论文的研究仍然

有值得继续深入的地方：

(1) 论文在探讨多部件维护决策过程中，考虑了多部件之间的经济相关性。而在当前制造车间中，也有存在随机相关性和结构相关性的多部件系统，而这方面的深入研究尚显不足。因此，可在论文研究基础之上，进一步针对那些具有结构相关性和随机相关性的多部件系统开展进一步的研究。

(2) 论文在构建维护决策模型时，将产品质量作为目标函数的一部分，实现降低维护成本的同时降低质量损失。然而，文中并未探究不同质量控制方法在维护决策制定过程中的应用，以及其如何与生产调度相互作用以实现全局优化。因此，未来的研究工作可以重点探索质量控制手段、维护策略以及生产调度之间的协同优化方法。

(3) 在论文的研究框架中，针对维护决策问题采用了强化学习方法，动态地寻找最佳的维护时机和策略；在生产调度模块运用了遗传算法，以求解复杂生产环境下最优的作业排序问题。值得注意的是，近期已有部分文献开始尝试利用强化学习解决生产调度相关难题，揭示了强化学习在处理动态、实时、多目标生产调度决策上的强大潜力。鉴于此，未来的研究方向可着眼于融合强化学习方法，对生产调度与维护决策进行联合优化。

## 参 考 文 献

- [1] 张晓红. 多部件系统维修决策及维修与备件库存联合决策研究[D]. 太原科技大学, 2015.
- [2] KUMAR P, MAITI J, GUNASEKARAN A. Impact of quality management systems on firm performance[J]. *International Journal of Quality & Reliability Management*, 2018, 35(5): 1034-1059.
- [3] SHAHIN A. A comprehensive approach for maintenance performance measurement[C]// *Proceedings of the 2nd National Conference on Maintenance & Repair, Tehran University of Science and Technology*. 2003.
- [4] PARENTE M, FIGUEIRA G, AMORIM P, et al. Production scheduling in the context of industry 4.0: review and trends[J]. *International Journal of Production Research*, 2020, 58(17): 5401-5431.
- [5] 成国庆, 周炳海, 李玲. 劣化系统的生产, 质量控制与视情维护联合建模与优化[J]. *计算机集成制造系统*, 2019, 25(7): 1620-1629.
- [6] TIAN Z, LIAO H. Condition based maintenance optimization for multi-component systems using proportional hazards model[J]. *Reliability Engineering & System Safety*, 2011, 96(5): 581-589.
- [7] ZHU Q, PENG H, VAN HOUTUM G J. A condition-based maintenance policy for multi-component systems with a high maintenance setup cost[J]. *Or Spectrum*, 2015, 37: 1007-1035.
- [8] POPPE J, BOUTE R N, LAMBRECHT M R. A hybrid condition-based maintenance policy for continuously monitored components with two degradation thresholds[J]. *European Journal of Operational Research*, 2018, 268(2): 515-532.
- [9] BAKIR I, YILDIRIM M, URSAVAS E. An integrated optimization framework for multi-component predictive analytics in wind farm operations & maintenance[J]. *Renewable and Sustainable Energy Reviews*, 2021, 138: 110639.
- [10] LU B, ZHOU X. Quality and reliability oriented maintenance for multistage manufacturing systems subject to condition monitoring[J]. *Journal of Manufacturing Systems*, 2019, 52: 76-85.
- [11] DONG E, GAO T, CHENG Z, et al. Opportunistic maintenance strategy for complex equipment with a genetic algorithm considering failure dependence: A two-dimensional warranty perspective[J]. *Sensors*, 2022, 22(18): 6801.
- [12] ZHANG N, SI W. Deep reinforcement learning for condition-based maintenance planning of multi-component systems under dependent competing risks[J]. *Reliability Engineering & System Safety*, 2020, 203: 107094.

- [13] ROCCHETTA R, BELLANI L, COMPARE M, et al. A reinforcement learning framework for optimal operation and maintenance of power grids[J]. *Applied energy*, 2019, 241: 291-301.
- [14] HUANG J, CHANG Q, ARINEZ J. Deep reinforcement learning based preventive maintenance policy for serial production lines[J]. *Expert Systems with Applications*, 2020, 160: 113701.
- [15] WEIS, BAO Y, LI H. Optimal policy for structure maintenance: A deep reinforcement learning framework[J]. *Structural Safety*, 2020, 83: 101906.
- [16] YOUSEFI N, TSIANIKAS S, COIT D W. Reinforcement learning for dynamic condition-based maintenance of a system with individually repairable components[J]. *Quality Engineering*, 2020, 32(3): 388-408.
- [17] HU Y, MIAO X, ZHANG J, et al. Reinforcement learning-driven maintenance strategy: A novel solution for long-term aircraft maintenance decision optimization[J]. *Computers & industrial engineering*, 2021, 153: 107056.
- [18] ANDRIOTIS C P, PAPAKONSTANTINOOU K G. Deep reinforcement learning driven inspection and maintenance planning under incomplete information and constraints[J]. *Reliability Engineering & System Safety*, 2021, 212: 107551.
- [19] YANG H, LI W, WANG B. Joint optimization of preventive maintenance and production scheduling for multi-state production systems based on reinforcement learning[J]. *Reliability Engineering & System Safety*, 2021, 214: 107713.
- [20] HAO S, ZHENG J, YANG J, et al. Deep reinforcement learning for joint optimization of condition-based maintenance and spare ordering[J]. *Information Sciences*, 2023, 634: 85-100.
- [21] WANG L, LU Z, REN Y. Joint production control and maintenance policy for a serial system with quality deterioration and stochastic demand[J]. *Reliability Engineering & System Safety*, 2020, 199: 106918.
- [22] FARAHANI A, TOHIDI H. Integrated optimization of quality and maintenance: A literature review[J]. *Computers & Industrial Engineering*, 2021, 151: 106924.
- [23] RASAY H, TAGHIPOUR S, SHARIFI M. An integrated maintenance and statistical process control model for a deteriorating production process[J]. *Reliability Engineering & System Safety*, 2022, 228: 108774.
- [24] TAMBE P P, KULKARNI M S. A reliability based integrated model of maintenance planning with quality control and production decision for improving operational performance[J]. *Reliability Engineering & System Safety*, 2022, 226: 108681.
- [25] JIN J, SHI J. State space modeling of sheet metal assembly for dimensional control[Z]. 1999.
- [26] JIN J, CHEN Y. Quality and reliability information integration for design evaluation of fixture system reliability[J]. *Quality and Reliability Engineering International*, 2001, 17(5): 355-372.
- [27] CHEN Y, JIN J. Quality-oriented-maintenance for multiple interactive tooling components in

- discrete manufacturing processes[J]. *IEEE Transactions on Reliability*, 2006, 55(1): 123-134.
- [28] BOX G E, WILSON K B. On the experimental attainment of optimum conditions[M]// *Breakthroughs in statistics: methodology and distribution*. Springer, 1992: 270-310.
- [29] TAGUCHI G. *Introduction to quality engineering: designing quality into products and processes*[M]. 1986.
- [30] HADIDI L A, AL-TURKI U M, RAHIM A. Integrated models in production planning and scheduling, maintenance and quality: a review[J]. *International Journal of Industrial and Systems Engineering*, 2012, 10(1): 21-50.
- [31] 朱传军, 冯诗健, 张超勇, 等. 考虑设备预防性维护的开放车间调度问题[J]. *中国机械工程*, 2023, 34(14): 1693.
- [32] LI J Q, PAN Q K, TASGETIREN M F. A discrete artificial bee colony algorithm for the multi-objective flexible job-shop scheduling problem with maintenance activities[J]. *Applied Mathematical Modelling*, 2014, 38(3): 1111-1132.
- [33] 宋文家, 张超勇, 尹勇, 等. 基于多目标混合殖民竞争算法的设备维护与车间调度集成优化[J]. *中国机械工程*, 2015, 26(11): 1478-1487.
- [34] SHEIKHALISHAHI M, ESKANDARI N, MASHAYEKHI A, et al. Multi-objective open shop scheduling by considering human error and preventive maintenance[J]. *Applied Mathematical Modelling*, 2019, 67: 573-587.
- [35] LU S, LIU X, PEI J, et al. A hybrid abc-ts algorithm for the unrelated parallel-batching machines scheduling problem with deteriorating jobs and maintenance activity[J]. *Applied Soft Computing*, 2018, 66: 168-182.
- [36] CHENG T, YANG S J, YANG D L. Common due-window assignment and scheduling of linear time-dependent deteriorating jobs and a deteriorating maintenance activity[J]. *International Journal of Production Economics*, 2012, 135(1): 154-161.
- [37] 周元辉, 周晓军, 陆志强. 带缓冲区的三设备串行生产系统动态机会维护策略研究[J]. *现代制造工程*, 2011(7): 6-10.
- [38] HAO L, LIU K, GEBRAEEL N, et al. Controlling the residual life distribution of parallel unit systems through workload adjustment[J]. *IEEE Transactions on Automation Science and Engineering*, 2015, 14(2): 1042-1052.
- [39] SUN Q, YE Z S, ZHU X. Managing component degradation in series systems for balancing degradation through reallocation and maintenance[J]. *IIE transactions*, 2020, 52(7): 797-810.
- [40] XIA T, JIN X, XI L, et al. Production-driven opportunistic maintenance for batch production based on mam-apb scheduling[J]. *European Journal of Operational Research*, 2015, 240(3): 781-790.
- [41] DO VAN P, BÉRENGUER C. Condition based maintenance model for a production deteri-

- orating system[C]//2010 Conference on Control and Fault-Tolerant Systems (SysTol). IEEE, 2010: 424-429.
- [42] LIU Q, CHEN S, DONG M, et al. Optimal condition-based maintenance strategy model for multi-component batch production system under dynamic demand[J]. *The International Journal of Advanced Manufacturing Technology*, 2022, 122(11): 4293-4307.
- [43] 陆志强, 牟小涵. 考虑质量衰退的批量调度与预防性维护的联合优化[J]. *同济大学学报(自然科学版)*, 2021, 49(4): 535-543.
- [44] XU S, DONG W, JIN M, et al. Single-machine scheduling with fixed or flexible maintenance [J]. *Computers & Industrial Engineering*, 2020, 139: 106203.
- [45] GHALEB M, TAGHIPOUR S, SHARIFI M, et al. Integrated production and maintenance scheduling for a single degrading machine with deterioration-based failures[J]. *Computers & Industrial Engineering*, 2020, 143: 106432.
- [46] DUFFUAA S, KOLUS A, AL-TURKI U, et al. An integrated model of production scheduling, maintenance and quality for a single machine[J]. *Computers & Industrial Engineering*, 2020, 142: 106239.
- [47] LIU Q, DONG M, CHEN F. Single-machine-based joint optimization of predictive maintenance planning and production scheduling[J]. *Robotics and Computer-Integrated Manufacturing*, 2018, 51: 238-247.
- [48] CUI W. Approximate approach to deal with the uncertainty in integrated production scheduling and maintenance planning[J]. *Journal of Shanghai Jiaotong University (Science)*, 2020, 25: 106-117.
- [49] PAPROCKA I. The model of maintenance planning and production scheduling for maximising robustness[J]. *International Journal of Production Research*, 2019, 57(14): 4480-4501.
- [50] SHARIFI M, GHALEB M, TAGHIPOUR S. Joint parallel-machine scheduling and maintenance planning optimisation with deterioration, unexpected breakdowns, and condition-based maintenance[J]. *International Journal of Systems Science: Operations & Logistics*, 2023, 10(1): 2200888.
- [51] CUI W, LU Z, LI C, et al. A proactive approach to solve integrated production scheduling and maintenance planning problem in flow shops[J]. *Computers & Industrial Engineering*, 2018, 115: 342-353.
- [52] FEI Y, MA H. Multi-objective joint optimization of batch-discrete hybrid flow shop scheduling integrated with machine maintenance[C]//2018 5th International Conference on Industrial Engineering and Applications (ICIEA). IEEE, 2018: 247-253.
- [53] MIYATA H H, NAGANO M S, GUPTA J N. Integrating preventive maintenance activities to the no-wait flow shop scheduling problem with dependent-sequence setup times and makespan

- minimization[J]. *Computers & Industrial Engineering*, 2019, 135: 79-104.
- [54] DAI M, TANG D, GIRET A, et al. Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints[J]. *Robotics and Computer-Integrated Manufacturing*, 2019, 59: 143-157.
- [55] 齐二石, 孔海宁, 何曙光. 钢铁企业副产煤气多周期动态优化调度[J]. *系统工程理论与实践*, 2010(11): 2071-2079.
- [56] OUELHADJ D, PETROVIC S. A survey of dynamic scheduling in manufacturing systems[J]. *Journal of scheduling*, 2009, 12: 417-431.
- [57] PICKARDT C W, HILDEBRANDT T, BRANKE J, et al. Evolutionary generation of dispatching rule sets for complex dynamic scheduling problems[J]. *International Journal of Production Economics*, 2013, 145(1): 67-77.
- [58] NASIRI M M, AHMADI N, KONUR D, et al. A predictive-reactive cross-dock rescheduling system under truck arrival uncertainty[J]. *Expert Systems with Applications*, 2022, 188: 115986.
- [59] TIGHAZOUI A, SAUVEY C, SAUER N. Predictive-reactive strategy for identical parallel machine rescheduling[J]. *Computers & Operations Research*, 2021, 134: 105372.
- [60] LI K, DENG Q, ZHANG L, et al. An effective mcts-based algorithm for minimizing makespan in dynamic flexible job shop scheduling problem[J]. *Computers & Industrial Engineering*, 2021, 155: 107211.
- [61] 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.
- [62] SHAHGHOLI ZADEH M, KATEBI Y, DONIAVI A. A heuristic model for dynamic flexible job shop scheduling problem considering variable processing times[J]. *International Journal of Production Research*, 2019, 57(10): 3020-3035.
- [63] GAO K, YANG F, LI J, et al. Improved jaya algorithm for flexible job shop rescheduling problem[J]. *IEEE Access*, 2020, 8: 86915-86922.
- [64] JOSEBECK G, GOWTHAM A. Demystifying the pf curve & augmenting machine learning for maintenance optimization[C]//2022 Annual Reliability and Maintainability Symposium (RAMS). IEEE, 2022: 1-5.
- [65] NAKAGAWA T. A summary of block replacement policies[J]. *RAIRO-Operations Research*, 1979, 13(4): 351-361.
- [66] THOMAS D S, THOMAS D S. The costs and benefits of advanced maintenance in manufacturing[M]. US Department of Commerce, National Institute of Standards and Technology ..., 2018.

- [67] BUDAI-BALKE G, DEKKER R, NICOLAI R P. A review of planning models for maintenance and production.[J]. Report/Econometric Institute, Erasmus University Rotterdam, 2006(EI 2006-44).
- [68] LEVRAT E, THOMAS E, IUNG B. Odds-based decision-making tool for opportunistic production-maintenance synchronization[J]. International Journal of Production Research, 2008, 46(19): 5263-5287.
- [69] ALASWAD S, XIANG Y. A review on condition-based maintenance optimization models for stochastically deteriorating system[J]. Reliability engineering & system safety, 2017, 157: 54-63.
- [70] GORJIAN N, MA L, MITTINTY M, et al. A review on degradation models in reliability analysis[C]//Engineering Asset Lifecycle Management: Proceedings of the 4th World Congress on Engineering Asset Management (WCEAM 2009), 28-30 September 2009. Springer, 2010: 369-384.
- [71] GRALL A, BÉRENGUER C, DIEULLE L. A condition-based maintenance policy for stochastically deteriorating systems[J]. Reliability Engineering & System Safety, 2002, 76(2): 167-180.
- [72] VAN NOORTWIJK J M. A survey of the application of gamma processes in maintenance[J]. Reliability Engineering & System Safety, 2009, 94(1): 2-21.
- [73] THOMAS L C. Replacement of systems and components in renewal decision problems[J]. Operations research, 1985, 33(2): 404-411.
- [74] SUTTON R S, BARTO A G. Reinforcement learning: An introduction[M]. MIT press, 2018.
- [75] BHAGAT S, BANERJEE H, HO TSE Z T, et al. Deep reinforcement learning for soft, flexible robots: Brief review with impending challenges[J]. Robotics, 2019, 8(1): 4.
- [76] CUAYÁHUITL H. Simpleds: A simple deep reinforcement learning dialogue system[J]. Dialogues with Social Robots: Enablements, Analyses, and Evaluation, 2017: 109-118.
- [77] MARUGÁN A P. Applications of reinforcement learning for maintenance of engineering systems: A review[J]. Advances in Engineering Software, 2023, 183: 103487.
- [78] COLLEDANI M, TOLIO T. Integrated quality, production logistics and maintenance analysis of multi-stage asynchronous manufacturing systems with degrading machines[J]. CIRP annals, 2012, 61(1): 455-458.
- [79] WANG L, LU Z, HAN X. Joint optimisation of production, maintenance and quality for batch production system subject to varying operational conditions[J]. International Journal of Production Research, 2019, 57(24): 7552-7566.
- [80] SHI H, ZHANG J, ZIO E, et al. Opportunistic maintenance policies for multi-machine production systems with quality and availability improvement[J]. Reliability Engineering & System

- Safety, 2023, 234: 109183.
- [81] MODRAK V, SOLTYSOVA Z. Batch size optimization of multi-stage flow lines in terms of mass customization[J]. Int. J. Simul. Model, 2020, 19(2): 219-230.
- [82] GHALEB M, ZOLFAGHARINIA H, TAGHIPOUR S. Real-time production scheduling in the industry-4.0 context: Addressing uncertainties in job arrivals and machine breakdowns[J]. Computers & Operations Research, 2020, 123: 105031.

## 致 谢

文章到此已至尾声，而我的求学生涯也即将结束，这一路经历了很多，也收获了很多。研究生的三年，收获了很多知识，也遇见了很多优秀的人，他们是我生活、学习和工作上的榜样。

感谢我的导师康宇教授和赵云波教授。感谢康老师给我转专业保研的机会，让我能够尝试一条新的道路从头开始。康老师对科研工作认真严谨，在生活中待人随和，从老师身上学到的科研思想和待人处事态度使我受益良多。感谢赵老师对我科研方面的悉心指导，在赵老师的耐心引导下，我慢慢形成了科研思维体系，对科研工作有了自己的理解，提升了发现问题、研究问题、解决问题的能力。赵老师严谨的科研态度和对学术的追求是我的榜样。

感谢实验室的同学们，在生活和科研上给予我许多帮助。特别地，感谢李鹏飞师兄对我的悉心指导，带我做项目、找研究方向、修改论文。鹏飞师兄在科研上指导我的学术研究，在生活上也给我很多帮助，很幸运在研究生期间能有鹏飞师兄的指导。感谢王涛师兄，涛哥对自己严格的要求、对科研严谨的态度，是我学习的目标，激励着我不断前行。感谢课题组的伙伴们：李鹏飞师兄、王涛师兄、许畅师兄、赵昀升、汪洋、王若山师弟、黄康杰师弟、刘金伟师弟，在科研上我们互相讨论一起学习，共同进步，在生活上一同聚餐，一起团建，让我的科研生活丰富多彩。感谢我的室友们：陈少军、张天浩、刘朝虎，我们一起聊科研、聊生活、聊未来，在三年的时光中给了我很多帮助，少军的热爱生活，天浩的乐观开朗，朝虎的勤勉奋斗，都是我学习的目标。祝愿实验室的同学们未来科研顺利，在各自的领域放光发热。

感谢我的父母和哥哥，你们的支持是我前行路上最大的动力。走过十余载的求学之路，从孩童时的懵懂无知，到现在虽不敢说学有所成，但也收获颇多，也将担起自己的责任继续前行，未来的路上愿一直有你们的陪伴。祝愿我的父母身体健康，我的哥哥工作顺利。

最后感谢自己，这一路走来磕磕绊绊，有过放弃，有过坚持。曾今的我豪情万丈过，经历现实的打击后也懦弱颓废过，也尝试鼓起勇气突破自我。现在的我也许没有当初想象的那么优秀，但也接受普通的自己，一个不沉溺过去、不逃避现实、不害怕未来，有着自我追求的自己。在未来的路上也将追寻自己的内心，书写自己的人生。

## 在读期间发表的学术论文与取得的研究成果

### 已发表论文

1. Ming Chen, Yu Kang, Kun Li, Pengfei Li. "Joint Optimization of Maintenance and Production Scheduling for Multi-component Systems", 2024 5th International Conference on Mechatronics Technology and Intelligent Manufacturing (ICMTIM 2024) (accept)

### 待发表期刊论文

1. Ming Chen, Yu Kang, Kun Li, Pengfei Li, Yunbo Zhao. "Deep reinforcement learning for maintenance optimization of multi-component considering quality and production plan", Quality Engineering (under review)