

文本复制检测报告单(全文标明引文)

ADBD2018R_20180411212939423957573739

检测时间：2018-04-11 21:29:39

检测文献：71068966119975096_梁乐乐_基于多Agent的柔性生产动态调度系统研究

作者：梁乐乐

检测范围：

中国学术期刊网络出版总库
中国博士学位论文全文数据库/中国优秀硕士学位论文全文数据库
中国重要会议论文全文数据库
中国重要报纸全文数据库
中国专利全文数据库
图书资源
优先出版文献库
大学生论文联合比对库
互联网资源(包含贴吧等论坛资源)
英文数据库(涵盖期刊、博硕、会议的英文数据以及德国Springer、英国Taylor&Francis 期刊数据库等)
港澳台学术文献库
互联网文档资源
CNKI大成编客-原创作品库
个人比对库

时间范围：1900-01-01至2018-04-11

指导教师 无

检测结果

总文字复制比：7.7% 跨语言检测结果：0%

去除引用文献复制比：7.7% 去除本人已发表文献复制比：7.7%

单篇最大文字复制比：2.3% (基于遗传算法的柔性资源调度优化方法研究)

重复字数：[3978] 总字数：[51996] 单篇最大重复字数：[1206]
总段落数：[5] 前部重合字数：[149] 疑似段落最大重合字数：[3174]
疑似段落数：[5] 后部重合字数：[3829] 疑似段落最小重合字数：[127]

指标：☐ 疑似剽窃观点 ☒ 疑似剽窃文字表述 ☐ 疑似自我剽窃 ☐ 疑似整体剽窃 ☐ 过度引用

表格：0 脚注与尾注：0

1.4% (149) 71068966119975096_梁乐乐_基于多Agent的柔性生产动态调度系统研究_第1部分 (总10982字)
2.9% (267) 71068966119975096_梁乐乐_基于多Agent的柔性生产动态调度系统研究_第2部分 (总9272字)
2.8% (261) 71068966119975096_梁乐乐_基于多Agent的柔性生产动态调度系统研究_第3部分 (总9321字)
1.3% (127) 71068966119975096_梁乐乐_基于多Agent的柔性生产动态调度系统研究_第4部分 (总10120字)
25.8% (3174) 71068966119975096_梁乐乐_基于多Agent的柔性生产动态调度系统研究_第5部分 (总12301字)

(注释：无问题部分 文字复制比部分 引用部分)

1. 71068966119975096_梁乐乐_基于多Agent的柔性生产动态调度系统研究_第1部分 总字数：10982

相似文献列表 文字复制比：1.4%(149) 疑似剽窃观点：(0)

1	低碳排放约束的柔性作业车间调度研究 党世杰(导师：张国辉) - 《郑州航空工业管理学院硕士论文》 - 2017-03-01	0.8% (83) 是否引证：否
2	云制造环境下资源服务匹配研究 王成建(导师：夏芳臣) - 《南昌大学硕士论文》 - 2014-06-30	0.6% (67) 是否引证：否
3	基于全局优化和局部学习的进化多目标优化算法	0.5% (52)

原文内容

基于多Agent的柔性生产动态调度系统研究

摘要

随着全球制造化和信息化的普及,制造企业面临着技术革新和竞争加剧带来的多重压力,包括生产模式的转变、市场需求的动态变化以及设备故障等多种异常因素,企业需要迫切地提高自身的生产管理水平。柔性作业车间调度问题(Flexible Job Shop Scheduling Problem, FJSP)在生产调度问题中占据越来越重要的地位,对比传统的作业车间调度问题, FJSP在工艺规划和设备选择上具备更多的柔性,导致问题的求解难度更大。在目前生产制造领域中,企业处于复杂多变的制造环境,而动态调度系统作为生产管理的重要模块,关系着企业的核心竞争力,是企业控制生产成本,保证生产效率和质量的关键。

近年来,多Agent理论作为分布式人工智能的研究热点之一,被广泛地应用于复杂大型系统的构建中。在多Agent系统(MAS)中,每个Agent在功能和结构设计过程中被赋予了自主性、社会性和能动性,通过各个Agent之间的信息交流,能够稳定高效地进行复杂问题的求解。本文在对FJSP的研究中,综合考虑企业所处的复杂多变的生产制造环境,应用多Agent理论,构建了一种柔性生产动态调度系统,本文主要工作包括:

(1)建立了基于多Agent的柔性生产动态调度系统的结构模型。本文分析了企业生产管理的实际需求,将生产管理流程中的业务进行划分,设计了包括管理Agent、资源Agent、算法Agent和监控Agent五类Agent,其中资源Agent根据企业制造资源的分布特点以集中控制的分层式组成了资源Agent组,并与其他Agent构建了分布式系统的系统架构。

(2)定义了分布式制造环境下MAS系统中Agent间的协作方式,中心思想是通过定义Agent间信息的交互来保证系统生产调度的高效性以及生产环境的快速响应,从而定义了跨区域的生产任务分解策略以及异常调度策略。前者用于综合利用企业分布在不同地区的加工工厂的资源和技术,完成生产任务在各个区域的分解和分配过程;后者在如设备故障、紧急订单等异常因素发生时,通过Agent之间的协商和合作,实现对动态生产环境的快速响应。

(3)应用改进的蚁群算法求解柔性生产中的调度问题。本文以最小化最大完工时间为优化目标,根据基本蚁群算法在传统作业车间调度问题中的应用,结合FJSP的柔性工艺规划特点,并基于该算法收敛速度慢和易于陷入局部最优解的缺点,对基本蚁群算法中信息素更新规则、蚁群初始位置等方面进行了改进,以得到更优的调度结果。该算法封装在算法Agent中,向资源Agent组提供计算服务。

(4)开发了基于多Agent的柔性生产动态调度系统。本文根据系统的分布式结构模型,设计了系统开发的软件体系架构。其次,应用面向对象技术对Agent的属性和方法进行了封装,使用基于TCP/IP的Socket通信机制以及线程池和消息队列实现了Agent的通信模型,最后在本文所开发的调度系统上验证了改进蚁群算法、跨区域生产任务分解以及异常调度策略的有效性。

关键字: 动态调度系统, 多Agent系统, 柔性作业车间调度, 分布式资源, 改进的蚁群算法, 系统开发

Abstract

With the popularization of global manufacturing and informatization, manufacturing companies are faced with multiple pressures brought by technological innovation and increased competition, including transition of production mode, dynamic demand of market, and other abnormal factors such as equipment failures. As a result, Enterprises have to improve the quality of production management as soon as possible. Flexible Job Shop Scheduling Problem (FJSP) occupies an increasingly important position in the production scheduling problem. Compared with the traditional job shop scheduling problem, FJSP has more flexibility in process planning and equipment selection, resulting in much more difficulties among problem solving. As an important module of production management, dynamic scheduling system is the key to control production cost and ensure productivity and quality, and is helpful to improve enterprises' core competitiveness when facing complex and changeable manufacturing environment.

In recent years, multi-agent theory has become one of the research hotspots of distributed artificial intelligence and has been widely used in the construction of large complicated systems. In the multi-agent system (MAS), each agent is given autonomy, sociality, and initiative. Therefore the MAS can solve complex problems stably and efficiently through flexible information interaction among various agents. During the research of FJSP, considering the complex and ever-changing manufacturing environment in which the company is located, this paper takes aim at constructing a flexible production dynamic scheduling system with multi-agent theory, and the main work as follows:

(1) The multi-Agent based structure model of flexible production dynamic scheduling system is established. This paper analyzes the actual requirements of enterprise production management process and divides the business into five types of agents including management agent, resource agent, algorithm agent, process agent and monitor agent. The resource agents are build as resource agent group with Hierarchical control structure base on the distribution of enterprises' manufacturing resources. Also the resource agent group and other agents are connected with a distributed structure.

(2) This paper defines cooperative mode between Agents in MAS under distributed manufacturing environment. The central idea is to ensure the high efficiency of the system's production scheduling and the rapid response to the production environment by regulating the interaction of information between agents. Therefore, it defines cross-regional production task

decomposition strategy and abnormal scheduling strategy. The former strategy aims to utilize enterprise's distributed resources and technologies located in different regions through the process of decomposition and distribution of production tasks; the latter strategy defines the consultations and cooperation between Agents when abnormal factors such as equipment failures and emergency orders occur to achieve rapid response to the dynamic manufacturing environment.

(3) This paper apply an improved ant colony algorithm to solve the scheduling problem in FJSP. The objective of the scheduling system is to minimize the make . Considering there are some disadvantages when using traditional ant colony algorithm to solve FJSP such as slow convergence speed, easily to stagnation, an improved ant colony optimization based on the flexible process planning features of FJSP is proposed in this paper, and the main improvements are to redefine pheromone updating rule and uniformly distribute the initial position of the ants. This improved algorithm is encapsulated in the algorithm Agent which provides computing services to the resource agent group.

(4) The flexible production dynamic scheduling system based on multi-agent is implemented. Based on the distributed structure model, this paper designs the software architecture for the scheduling system. During the implementation of the system, this paper use object-oriented technology to encapsulate the properties and function of each Agent. This paper design communication mechanism based on TCP/IP Socket. Also the communication model of Agent is implemented by thread pool and message queue. Finally, this paper verifies the effectiveness of improved ant colony algorithm, cross-regional production task decomposition and abnormal scheduling strategy via this scheduling system.

Key Words: dynamic scheduling system, multi-agent system, flexible job shop scheduling problem, distributed resource, improved ant colony optimization, system development

目录

1 绪论.....	4
1.1 研究背景.....	4
1.2 研究目的与意义.....	4
1.3国内外研究现状.....	5
1.3.1 柔性作业车间调度问题的研究现状.....	5
1.3.2 多Agent生产调度系统的研究现状.....	6
1.4 论文的主要研究内容和组织结构.....	6
2 多Agent动态调度系统功能与结构设计.....	7
2.1 Agent与多Agent理论.....	7
2.1.1 Agent的概念及特征.....	7
2.1.2 多Agent系统的概念及特征.....	7
2.2 动态调度系统的功能分析.....	8
2.2.1 制造资源的分布式特点.....	8
2.2.2 系统的功能划分以及Agent分类.....	8
2.3 各Agent的功能设计.....	9
2.4 系统的结构分析.....	10
2.5 本章小结.....	11
3 分布式制造环境中Agent的协调机制.....	11
3.1 柔性作业车间调度问题.....	11
3.1.1问题描述.....	11
3.1.2 FJSP的数学模型.....	12
3.2 基于多Agent的生产任务跨区域分解策略.....	13
3.2.1 生产任务的形式化描述.....	13
3.2.2 基于多Agent的生产任务跨区域分解流程.....	14
3.3 基于多Agent的异常调度策略.....	16
3.3.1 紧急订单处理的异常调度策略.....	17
3.3.2 设备故障处理的异常调度策略.....	17
3.3.3 订单取消处理的异常调度策略.....	18
3.4 本章小结.....	18
4 多Agent柔性生产调度系统中的调度算法.....	18
4.1 基本蚁群算法的引进.....	18
4.2 基本蚁群算法在FJSP中的应用.....	19

4.3 基本蚁群算法在FJSP中的改进.....	22
4.4 本章小结.....	23
5 多Agent柔性生产动态调度系统的实现.....	23
5.1 制造企业的生产管理流程.....	23
5.2 柔性生产动态调度系统需求分析.....	24
5.3 多Agent系统的实现.....	25
5.3.1开发环境及工具.....	25
5.3.2 系统软件架构设计.....	25
5.3.3 多Agent系统的面向对象设计.....	26
5.3.4 Agent的通信模型设计.....	27
5.4 动态调度系统实现与界面设计.....	28
5.4.1 系统登录模块.....	28
5.4.2 系统信息管理模块.....	29
5.4.3 资源管理模块.....	30
5.4.4 工艺管理模块.....	31
5.4.5 生产调度模块.....	33
5.5 本章小结.....	35
6 仿真实验与结果分析.....	35
6.1 实验环境.....	35
6.2 实验一：基于改进蚁群算法的调度策略验证.....	35
6.3 实验二：验证基于多Agent的跨区域生产任务分解策略.....	40
6.3 实验三：验证设备故障条件下的多Agent异常调度策略.....	42
6.4 本章小结.....	45
7 总结与展望 7.1 总结.....	45
7.2 工作展望.....	46

1 绪论

1.1 研究背景

自我国改革开放以来，制造业一直是我国大力发展的产业，并为国民经济的发展和人民生活水平的提高做出了重大贡献[1]。近年来，在经济全球化的影响下，现代制造业拥有了更多的发展机遇和广泛的市场需求，但也意味着我国的制造业需要在全球化舞台上跟来自世界各地的企业竞争。全球化的浪潮对我国企业传统的生产方式和管理模式带来了诸多影响，这些影响包括[2-3]：

(1) 世界范围内科学技术日新月异，分布式计算、云计算、物联网、智能制造、人工智能等理念应运而生，为制造业生产技术的革新提供了扎实的理论支撑，也督促企业减少对人力和人为先验知识的依赖，在生产管理上加快向智能化、自动化的方向靠拢。

(2) 随着我国经济的腾飞，人们生活水平有了很大的提高，极大丰富了人民的物质文化需求，对应的，人们的消费观念和习惯发生了很大的转变，对多样化和个性化的商品给予了更多的关注，因此商家需要根据市场需求调整经营策略，不断推陈出新。丰富的产品种类要求制造企业加快转变自身的生产模式，以往的“小品种大批量”生产模式已不再适用，取而代之的则是面向订单的小批量多品种甚至是单件定制化生产，因此制造业需要摒弃传统的面向大批量的长时间跨度的静态生产方式，充分考虑市场需求的动态变化，灵活快速地响应客户的随机订单并生成满足优化目标的调度方案。

(3) 个性化的产品需求在另一方面导致了产品功能和结构的升级，因此产品在工艺规划上具备更多的柔性，单一的资源配置和生产方式无法满足柔性生产的需求，这对企业在生产调度的效率和质量上有了更高的要求。

(4) 经济全球化和发达的网络通信技术使制造业的生产服务对象的范围扩大到世界各地，企业在生产上的控制方式从集中控制式向分布式转变，对应地企业通常细化生产管理的各个流程，在世界各地建立加工工厂，满足不同地区的生产需求。

我国制造业面临的种种竞争和挑战，需要通过对生产管理进行革新，车间作为企业生产管理的最小单位，车间调度是企业保证生产质量、缩短生产周期和控制生产成本的关键。车间调度系统集成制造企业在生产计划制定、生产调度、生产管理、生产监控、质量保证等各个环节，是提高企业竞争力，实现企业智能化、敏捷化的核心所在。

1.2 研究目的与意义

车间调度是企业保证生产质量、缩短生产周期和控制生产成本的关键[6]，而其中的柔性作业车间调度占据了越来越重要的地位，这是由于产品在工艺规划上具备更多的柔性，与当前企业小批量，多品种和单件定制化的生产模式更加匹配，更能满足市场动态多变的产品需求。目前对柔性生产调度问题的算法研究已有一定的成果，但在在全球化背景下，企业所处的生产环境充满动态不确定性[4-5]，如紧急订单、订单撤销、原料紧缺、设备故障等多种异常因素，严重干扰企业正常的生产调度。其次

企业为扩大生产规模，倾向于在各个地区建立加工工厂，如何综合利用企业分散的制造资源和技术对提高企业的生产效益和效率是非常关键的[7]。因此面对企业所处的动态复杂的分布式制造环境，若仅仅通过对传统调度理论和算法进行优化，是不足以满足企业对各种异常因素的快速相应需求的，研究人员需要从生产系统的角度出发，集成企业生产管理的各个环节，对系统的功能进行合理的划分，设计满足实际生产需求的灵活系统架构，保证企业生产调度的稳定高效进行。

传统的车间调度系统属于集中控制式的静态调度系统，特点是制造资源集中、生产规模庞大、时间跨度长等，主要面向种类单一、工艺结构简单的产品制造，该形式的调度系统简单地把管理、控制、监控等功能交由人力负责，因此容错能力和稳定性低，当生产过程中发生如原料短缺或设备故障等情况时，往往需要对整个生产线进行停工处理，显然集中控制式的静态调度系统是无法满足现代制造业的要求的。针对以上不足，本文开展对柔性生产动态调度系统的研究，该研究对现代制造业具有以下现实意义：

(1) 实现企业生产管理的信息化和自动化，提高生产效率。本文立足于柔性生产调度问题，应用基于改进蚁群算法的柔性作业车间调度算法模型，同时集成企业在数据管理、生产监控等各个环节，降低对人力的依赖，为企业实现自动化的生产管理流程提供了大力支持。

(2) **提高企业对分布式资源的利用率，增强企业生产效益。企业的制造资源和技术具有分布式特征**，本文在生产调度过程中，依据企业在各地区工厂的资源配置特点进行生产任务的规划，降低企业生产设备的闲置率。

(3) 提高企业对动态不确定性的响应能力。动态调度系统在面临紧急订单、订单更改、订单撤销以及设备故障等异常状况时，能够通过系统内部各模块间的高效通信，及时反馈异常信息，同时快速给出重调度方案，减少对原生产计划的影响，保证系统的稳定性和健壮性。

1.3国内外研究现状

1.3.1 柔性作业车间调度问题的研究现状

在世界经济飞速发展的大环境下，用户需求复杂多变，产品的更新换代尤为频繁，制造业在生产调度中完成了从上世纪五十年代的单机、并行机调度到如今流水车间和作业车间调度的转变，其中柔性作业车间调度越来越受重视，这是由于工序存在机器选择，弱化了生产约束，提供了可选的加工路径，在面对客户多变的订单需求时，拥有更宽泛的生产灵活性，更适用于如今小批量多品种的生产模式。同时，由于约束更少，扩大了解空间的范围，提高了问题求解的难度，因此FJSP是比传统车间调度问题更复杂的NP-Hard问题[8]，传统车间调度问题所使用的如分支界定法、整数规划、拉格朗日松弛法等精确算法已无法适用于FJSP的求解，取而代之的是局部搜索和人工智能等新型优化方法，如局部搜索法中使用非常广泛的禁忌搜索、模拟退化和差分进化法等；人工智能算法中遗传算法、蚁群算法、神经网络、免疫算法等。这些方法在求解复杂的大规模问题中拥有更好的性能，通常会结合各种算法来进行多优化目标的FJSP求解，这样能避免单一算法的缺点，综合各类型算法的长处，以得到更优的调度结果。

随着柔性作业车间调度使用越来越广泛，FJSP一直备受国内外研究人员的关注。汪俊亮[9]等在对加工时间不确定的FJSP研究中，以最小化最大完工时间为目标，设计了一种分阶段的遗传算法，配合顺序搜索机制加快算法的求解速度，最终通过仿真实验验证了算法的鲁棒性。Zhang和Maier[10]根据FJSP的特点以及工件运输成本的问题，提出一种融合遗传算法和禁忌搜索的改进算法。王万良[11]等提出了一种改进的差分进化算法，该算法对柔性生产工件分批处理，在关键路径中使用随机搜索和自适应变异相结合的染色体编码方式来淘汰无用个体，加快收敛速度，最后通过仿真实验证明了算法能够有效缩短生产周期。刘韵[12]等为提高FJSP的求解速度以及质量，结合了禁忌搜索**算法和遗传算法的优点，提出一种元启发式求解算法，并在大规模作业**求解中验证了算法的可行性。王雷[13]等在FJSP中添加AGV运输约束，设计了一种改进的遗传算法，该算法使用多段式编码以及定义了改进的交叉变异形式，淘汰无用基因，提高进化速度。Mosle G和Mahnam M [14]提出了一种应用于多目标FJSP的改进的粒子群优化算法，并融合局部搜索方法，获得了均匀分布的Pareto解集。赵博选[15]在构建了一种两阶段混合Pareto蚁群算法应用在多目标FJSP求解中，该算法在第一阶段进行工序任务的机器选择，第二阶段对第一阶段的结果进行评估后再执行工序的排序操作，在仿真实验中证明了算法能够简化调度问题的规模和复杂性。Rossi[16]在FJSP的求解中，当蚁群陷入局部最优或过早收敛时，通过破坏路径信息素浓度以及通过路由调整蚂蚁的转移方向。Huang[17]在蚁群路径搜索过程中根据当前最优解和全局最优解的差异，动态调整信息素挥发比例以及信息素释放的浓度，在多目标FJSP仿真实验中证明了算法能有效防止蚁群停滞和早熟。Chiang[18]等提出了一种简单高效的改进多种群进化遗传算法应用于多目标FJSP中，该算法只需设置两个参数，通过有效的基因操作来保持种群的多样性，并在15个基准实例的测试中对算法进行了验证。

1.3.2 多Agent生产调度系统的研究现状

车间调度系统能有效协调生产管理的各个环节，对保证企业生产制造的稳定高效进行有着重大意义。调度系统需要统筹兼顾生产计划、库存管理、资源分配、人员调度、环境监控等诸多流程，涉及面广、难度高，因此制造企业对调度系统的灵活性和健壮性有很高的要求。分布式MAS通过Agent的自治能力以及相互之间高效灵活的通信，非常适用于复杂大型系统的构建，因此国内外众多学者尝试把多Agent理论应用到调度系统中，并取得了一定的成果。

Pereira[19]在多Agent调度系统的构建中，为每一个Agent定义自我优化的目标，指定了Agent间的竞争协商规则来达到生产目标的优化。Madureira[20]将一种基于元启发式的参数自配置方法应用在多Agent调度系统的学习模块中，提高系统的决策管理能力。Gozzi[21]在调度系统中应用了多Agent理论，通过Agent的相互协作来应对各种动态干扰事件的发生。Lim[22]构建了

基于多Agent的分层控制式的生产系统，通过层次间Agent的管理和监控，设计了分层迭代的通信机制实现了系统通信和管理的高效性。Adhau[23]使用MAS构建了分布式的调度系统，定义了Agent间的组合拍卖协商模型。杨陇苗[24]使用JADE实现了基于多Agent的MES系统，该系统应用PASSI (a Process for Agent Societies Specification and Implementation) 方法对Agent进行建模，并建立了系统的多层分布式递阶控制结构，在该基础上对多个经典调度实例进行了验证，证明了系统的实用性。王芊紫[25]在混合流水车间调度系统的构建中定义了管理Agent、工件Agent、策略Agent和机器Agent等四类Agent，并将差值排序算法封装在策略Agent中，同时定义了Agent间的协调机制实现系统的动态调度。宋娟[26]根据生产车间面临的各种动态异常因素，结合MAS和CNP (Contract Net Protocol) 协议，构建了分布式的多Agent制造系统，并通过对中小规模的机械加工实验进行仿真，验证了系统的有效性。任海英[26]针对FJSP中生产环境的各种干扰信息，构建了基于多Agent的预先/重调度系统，该系统定义了Agent间的通信协商机制来应对各种动态干扰，使用遗传算法作为系统的调度方案并完成了系统的开发，通过仿真实验验证了系统动态调度的优越性。

由以上研究现状可看出，在柔性生产调度的研究中，基于MAS的柔性生产调度系在集成高效的调度算法模型互，能够统筹兼顾企业在生产管理、数据管理、人员管理、生产监控等各方面的需求，因此在实际使用中比单纯的算法优化具有更高的实用价值。本文把多Agent理论应用在柔性生产动态调度系统的研究中，根据生产管理中各个流程的结构和功能，把其中的物理实体或算法逻辑封装为具有不同功能和结构的Agent，构建多Agent系统，通过Agent之间的交互协调，共同完成各项任务，更好地满足企业对灵活性、自适应性、分布式上的需求。

1.4 论文的主要研究内容和组织结构

本文在结构上分为六章，各章内容如下：

第一章介绍了论文的研究背景、目的和意义，展开了对柔性作业车间调度问题和多Agent车间调度系统的国内外研究现状，同时概括了本文的主要研究内容。

第二章先是介绍了关于Agent与多Agent系统的相关理论，其次根据柔性生产动态调度系统的实际需求，对系统的功能进行划分，从而设计了包括管理Agent、资源Agent、算法Agent、工艺Agent和监控Agent，同时指定了各Agent的功能职责。其次根据系统的业务流程，设计了基于多Agent的调度系统的分布式结构，其中资源Agent根据企业制造资源在各个地区的分布，构建为呈现分层控制式的资源Agent组。

第三章根据系统中Agent间的信息的传递方向，提出了基于多Agent的任务协商策略，包括跨区域生产任务分解协商策略以及重调度协商策略。

2. 71068966119975096_梁乐乐_基于多Agent的柔性生产动态调度系统研究_第2部分 总字数：9272

相似文献列表 文字复制比：2.9%(267) 疑似剽窃观点：(0)

1	<u>柔性作业车间调度方法研究</u> 张国辉(导师：高亮;李培根) - 《华中科技大学博士论文》 - 2009-08-01	1.0% (94) 是否引证：否
2	<u>改进遗传算法求解柔性作业车间调度问题的研究</u> 李晏朔;林巧; - 《机械制造》 - 2011-04-20	1.0% (91) 是否引证：否
3	<u>基于改进遗传算法求解柔性作业车间调度问题</u> 张国辉;石杨; - 《机械科学与技术》 - 2011-11-15	0.9% (85) 是否引证：否
4	<u>生物地理学算法求解柔性作业车间调度问题</u> 张国辉;聂黎;毛学港; - 《计算机应用研究》 - 2014-04-15	0.9% (85) 是否引证：否
5	<u>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</u> 宋文家;张超勇;尹勇;邵新宇; - 《中国机械工程》 - 2015-06-01 1	0.7% (68) 是否引证：否
6	<u>改进的遗传算法求解柔性作业车间调度问题</u> 刘月凡;朱星; - 《大连交通大学学报》 - 2015-08-15	0.6% (53) 是否引证：否
7	<u>两段式蚁群粒子群混合优化算法求解MFJSP</u> 李莉;周春楠; - 《计算机应用与软件》 - 2012-08-15	0.6% (51) 是否引证：否
8	<u>紧密衔接工序组联动的综合调度算法</u> 谢志强;滕宇峥;杨静; - 《自动化学报》 - 2011-03-15	0.5% (48) 是否引证：否
9	<u>柔性作业车间调度的多Agent协商策略</u> 任海英;商晓坤; - 《计算机工程》 - 2011-01-20	0.4% (36) 是否引证：否
10	<u>柔性作业车间调度问题的两级遗传算法</u> 张超勇;饶运清;李培根;邵新宇; - 《机械工程学报》 - 2007-04-15	0.3% (32) 是否引证：否
11	<u>改进遗传算法求解柔性作业车间调度问题</u>	0.3% (32)

张国辉;高亮;李培根;张超勇; - 《机械工程学报》 - 2009-07-15		是否引证：否
12	求解柔性作业车间调度问题的两级邻域搜索混合算法	0.3% (32)
赵诗奎; - 《机械工程学报》 - 2015-07-06 1		是否引证：否
13	基于遗传算法的作业车间调度问题研究	0.3% (32)
林彬彬(导师：张国基) - 《华南理工大学硕士论文》 - 2011-05-12		是否引证：否
14	多目标柔性作业车间调度问题的混合差分算法	0.3% (32)
张敬敏;张有华;李霞; - 《计算机工程与应用》 - 2011-07-11		是否引证：否
15	个性化定制下的生产调度问题研究	0.3% (30)
颜颂涛(导师：石宇强;吴双) - 《西南科技大学硕士论文》 - 2017-06-05		是否引证：否
16	用蚁群算法集成求解多加工路线柔性车间调度问题	0.3% (30)
黄学文;张晓彤;艾亚晴; - 《计算机集成制造系统 (优先出版) 》 - 2017-05-23 10:20		是否引证：否

原文内容

前者根据产品工艺特点，设计了生产任务的模型，实现了生产任务在多个资源Agent组间的任务分解流程，该过程最终得到若干满足任务需求的设备集；后者通过系统内各Agent的相互配合，完成系统对订单撤销、紧急订单和设备故障等各种异常状况的响应。

第四章基于最小化最大完工时间的优化目标，应用基本蚁群算法求解传统作业车间调度问题，并根据柔性作业车间调度问题的特点以及蚁群算法收敛速度慢和容易陷入局部最优的缺点，提出了FJSP中改进蚁群算法的模型。

第五章提出了基于多Agent的柔性生产动态调度系统的实现方式。根据系统的结构模型设计了系统的软件构架。使用基于TCP/IP的Socket通信机制以及Java的面向对象和多线程技术，设计了各个Agent的通信模型，保证了系统的通信质量和效率。同时根据各个Agent的功能设计了对应的内部模型。

2 多Agent动态调度系统功能与结构设计

2.1 Agent与多Agent理论

2.1.1 Agent的概念及特征

尽管Agent在众多研究中应用非常广泛，在理论上已渐趋成熟，但目前对Agent并未有统一的标准。Agent通常也被称为智能体，属于人工智能领域，因此人们关注的主要是Agent的自主性，从这个角度而言，Agent较为常见的定义是由Jennings和Wooldrige [28]提供的：Agent属于由硬件和软件两者综合组成的实体，能够所处的环境进行信息交换，并能够按照特定目标自主采取相应的行动。Agent通常也被认为是“可移动的”，能够根据自身的需求并在外界环境信息的刺激下作出相应的判断，改变自身的位置，还能够影响周边环境以及其他Agent。尽管现在Agent在使用上尚未大范围推广，实际的应用实例也比较少，但是Agent具有大家所公认的以下一些特点[29]：

- (1) 自主性：拥有支撑自身行为的决策库，具备学习能力，可根据与周边环境的接触扩充、修改自身的决策库，根据决策规则选择对应的行为，可连续动态地改变自身的状态。
- (2) 社交性：具有通信标识和通信语言，Agent之间具有通信的能力，遵循特定的通信规则可相互交换信息。
- (3) 反应性：能够时刻感知来自外界的信息而无需特定的信息进行唤醒，能够分析信息所表达的意图和价值，把有用的信息扩充自身数据库。
- (4) 能动性：不仅能接收外界的信息，同时还能够自主地发起对周边环境的探索，通过主动获取环境信息以及接收其他Agent的信息更改决策库以及位置。

根据以上Agent的描述，可见Agent具有学习能力，能够根据自身的目的获取、交换信息，从而更改自身的状态，包括决策库的决策规则、数据库信息、地理位置等等；同时还具有可移动性，在分布性和扩展性的环境中易于迁移[30]，因此由若干Agent组成的多Agent系统非常适合用于部署分布式系统，对于分布式系统中的各个功能模块或子系统，可以交由单独的Agent或者由若干功能相近的Agent组成的Agent组负责，通过系统内Agent的信息交互，来协同完成各项任务。

2.1.2 多Agent系统的概念及特征

多Agent系统 (MAS, Multi-Agent System) 是由Agent组成的具有一定结构和功能的分布式系统。其中的Agent需要承担系统中的部分功能，功能相似或相同的Agent可组成一个Agent组。系统需指定Agent间的通信协议和通信方式。

MAS能够对复杂问题进行分解，分配到相关的Agent，各Agent根据自身的计算能力求解子问题，通过Agent间的通信进行交互协调，最终综合各个子问题的求解结果，获取对复杂大型问题的完整解决方案。与传统集中控制式、递阶控制式[31]的系统而言，多Agent系统部署和维护所需要的成本更低，系统将计算能力和资源分散到一组Agent中，单一的Agent在子问题的计算中对硬件性能的要求较低，减少了对高性能的大型服务器的依赖。同时，Agent能独立于中央服务器运行，能并行处理消息请求以及进行问题求解，因此能有效提高系统整体的稳定性和效率。

Agent的自主性、反应性、能动性决定了多Agent系统具有更强的灵活性，对外界的变化反应更为灵敏，在环境动态变化频繁的场景中具有更大的优势。Agent作为分布式系统中的节点，由于其具有可移动性，因此对于节点的增删处理反馈更及时，保证系统的高可扩展性，再者Agent的学习能力能够调整自身的状态和行为，随着时间的推进，可以优化求解过程、提高系

统整体的求解质量，这相当于延长了系统的生命周期，减少人工干预、维护的成本。由此可见,MAS非常适用于动态车间调度系统，通过MAS对外界信息的灵敏反应以及内部Agent间的问题求解能力，可以很好地满足调度系统对动态变化的客户需求和生产环境中各种异常因素的快速响应。

2.2 动态调度系统的功能分析

2.2.1 制造资源的分布式特点

制造资源在定义上可以分为广义和狭义两类[32]，广义的制造资源可分为物料资源、设备资源、人力资源、技术资源、运输资源等，其中物料资源表示生产加工中所需的各种原材料；设备资源表示各种加工设备；人力资源表示车间管理员、维修人员、运输工人等；知识资源表示与生产管理相关的专业知识，如产品的工艺信息、生产优化技术、环境监测技术等；运输资源表示用于在各制造单元和工厂间搬运的运输工具。狭义的制造资源专门指设备资源和物料资源，本文的制造资源属于狭义的定义。

制造全球化使我国企业面临激烈竞争的同时，也为其带来了更广阔的市场，因此现代制造业具有分布式加工的特点，倾向于在各个地区建立工厂，每个工厂拥有各自的车间结构和制造资源配置，提供对应的生产服务。本文根据企业分散式资源的异地性分布结构[33]，把每个工厂封装为一个独立的资源模块，当企业收到客户的产品订单时，以网路为媒介，获取各资源模块的加工能力，进行生产任务的分解分配流程，最终从各模块的调度结果中筛选满足客户需求的最优调度方案。通过这种方式，能够把企业分布在不同地区的资源进行合理的规划，降低设备的闲置率，提供企业的生产效率。

2.2.2 系统的功能划分以及Agent分类

车间调度系统最基本的功能在于满足生产任务在时间和经济效益上的要求，尽可能地给出最优的调度方案，并按照方案把任务安排到对应的加工设备上。目前市场需求动态变化、商品更新迭代的周期不断缩短，制造环境也充满诸多不确定性，因此系统的结构要能够兼顾灵活性和稳定性，当生产任务发生变化时，能够动态调整系统结构；当发生各种异常情况时，系统结构中的各单元能够灵活配合，快速反馈。

完整的车间动态调度系统应当具备以下功能：一是能够接收并管理来自客户的订单任务，包括对任务合法性和优先级的判定；二是对订单任务进行分解分配，把任务分解成工件的工序集合，根据工序在设备选择和加工路径的柔性特点计算调度方案，并向生产设备下达加工任务；三是能够快速响应如订单撤销、紧急订单、设备故障等异常因素，开启重调度，尽量降低原有调度方案对交货期、加工成本和设备负荷的影响；四是对系统数据的高效管理，包括工艺信息、制造资源信息、历史工作信息、设备故障信息等等。根据以上功能，本文把动态调度系统生产管理涉及的工作流程总结为：订单的输入和处理、产品柔性工艺信息的管理、柔性生产任务规划以及异常调度[34]。对应以上流程，本文基于多Agent理论，分别设计了五类Agent，包括管理Agent、资源Agent、工艺Agent、算法Agent、和监控Agent，其中管理Agent负责订单任务的处理以及协调系统内的其他Agent的通信和行动；资源Agent分为车间Agent和设备Agent，设备Agent对应系统内的加工设备，负责模拟设备的加工序列，车间Agent负责管理分布其中的设备Agent，根据企业资源的分布式特点，同属一个工厂的资源Agent构成一个资源模块，封装为资源Agent组；工艺Agent负责管理系统内产品的工艺信息，为其他Agent提供数据查询接口并响应系统管理人员对产品工艺信息的修改；算法Agent是对系统与柔性作业车间调度相关的算法的封装，即基于特定的设备集合和任务集合进行调度方案的计算；监控Agent对应系统的监控设备，通过分析监控数据，反馈系统内设备的工作状况，通知系统开启重调度流程。调度系统把其中的物理实体、数据或算法逻辑封装为以上五类Agent，构建了多Agent动态调度系统，其中的Agent各司其职的同时，通过Agent间灵活自由的通信，共同完成生产管理的完整流程。

2.3 各Agent的功能设计

调度系统中各Agent的具体功能设计如下：

管理Agent：分为全局管理Agent和子管理Agent。全局管理Agent相当于系统内部的管理员，协调Agent间的通信和任务配合。每个Agent在启动运行后需在全局管理Agent注册自身的通信标识，通信标识是Agent间通信和任务协调的重要依据。其次全局管理Agent作为系统对外的通信接口，负责接收并审核来自客户的生产订单，根据交货期对订单任务进行优先级排序，协调资源Agent和算法Agent对任务进行分解分配工作。同属一个地区工厂的资源Agent将构成一个资源Agent组，而子管理Agent是系统分配给一个资源Agent组的管理员，作为Agent组与全局管理Agent通信的主要接口。当子管理Agent接收到全局管理Agent传递的订单任务后，根据其下车间Agent的设备配置和加工能力来判断能否完成该任务。

资源Agent：资源Agent代表系统内的制造资源，分为车间Agent和设备Agent。设备Agent对应一个加工设备，记录设备的各项信息，包括设备名称、编号、工作状态、加工能力等；车间Agent对应系统内的一个车间，负责对该车间中的设备Agent进行管理，当有新设备添加或旧设备删除时，车间Agent需要与对应的设备Agent建立或移除关联信息。如图2-1所示，同属一座工厂内的车间Agent和设备Agent构成一个资源Agent组，由子管理Agent负责管理和控制,当企业接受到客户的产品订单时，根据各个资源Agent组的生产能力完成订单任务的分解以及资源的分配。

图2-1 资源Agent组结构

工艺Agent: 负责对系统的产品工艺信息进行管理，为其他Agent提供数据查询服务。该工艺信息用于表示系统所能生产加工的产品种类、产品需要的原材料、产品中各工件的加工工艺流程以及其中各个工序的设备选择等信息。工艺Agent负责为全局管理Agent审核订单任务提供依据，当任务对设备、原材料、加工精度的要求能够与工艺Agent记录的工艺信息一一对应时，表示任务的合法性通过，可进一步进行任务的分解分配。

算法Agent: 封装了调度系统内的算法逻辑, 基于特定的设备集合和生产任务为资源Agent组提供调度方案的计算服务。本文的算法Agent选择了基于改进蚁群算法的柔性作业车间调度算法。当全局管理Agent把生产任务传递到各个资源Agent组时, 资源Agent根据其中设备Agent所记录的设备加工能力和加工序列, 由算法Agent执行调度算法计算调度方案, 最终全局管理Agent从各个资源Agent组返回的调度结果中筛选完工时间最短的调度方案。

监控Agent: 代表系统内的检测设备。监控Agent读取监控设备采集的设备工作状态信息, 当判定设备故障后, 通知故障设备所属的资源Agent组, 由对应的子管理Agent回收分配到故障设备的工序任务, 根据任务对设备的要求, 重新分配到正常设备的加工序列。

2.4 系统的结构分析

多Agent调度系统一般存在三种较广泛使用的系统结构, 分别是层次型、分布型和混合型[35]:

(1) 层次型: 该结构强调的是节点间的从属关系, 属于递阶控制式, Agent间通过分层的形式进行管理, 上层Agent负责对下层Agent进行管理和控制, 不同层次的Agent间通信是必须通过层次间逐层传递, 位于同一层次的Agent之间可平等自由地通信, 因此系统主要通过上层Agent对下层Agent的控制管理来完成任务。该结构具有很多优点, 首先是易于扩展, 层次的增删或者在某一层次中Agent的增删对系统其它部分影响很小。其次易于系统功能的划分和管理, 层次间的从属关系能够规范下层Agent的行为, 相同层次的Agent间易于数据共享。但该结构的缺点同样明显, 首先是通信效率较低, 当系统规模扩大, 层次加深时, 顶层和底层之间Agent的通信需要跨越多个层次, 延时明显。而且当同一层次的Agent数量过多时, 会加重上层节点控制和协调的负担, 最严重的问题是, 当上层Agent失效时, 层次间的通信将被阻断, 因此一般通过增加功能相近的可替换节点来预防该情况的发生, 可以一定程度上提高系统的稳定性。

(2) 分布型: 该结构中各节点间不存在从属关系, 不需要经过消息中转, 相互间可以平等自由通信, 系统通过各Agent之间的协调完成任务。对比层次型结构, 分布型结构拥有更高的灵活性和扩展性, 由于Agent间自由通信, 信息可直达, 因此效率更高, 并且不存在由于某一Agent失效导致局部或全局通信瘫痪的情况。该结构同样存在缺点, 由于不存在Agent间的控制从属关系, 因此Agent之间数据共享、状态同步更困难, 当任务对数据一致性要求很高时, 分布型系统效率反而会下降, 因此通常会添加专门用于进行数据同步和共享操作的Agent。再者由于信息在任意的Agent之间可达, 提高通信效率的同时也可能导致通信风暴[25], 这是由于Agent具有自主性和能动性, 可以发起对环境以及Agent的探测, 若不以约束, 可能占用过多的通信带宽, 导致网络阻塞。

(3) 混合型: 通常情况下, 可以综合两种结构的优点, 兼顾两者在控制管理和通信效率上的性能, 因此就有了混合型的多Agent系统结构, 该结构为功能相近以及依赖数据共享的Agent组建Agent组, 实施局部的控制管理, 并且由组内具有管理职能的Agent发起对环境的探索, 能够约束系统整体的通信量, 同时功能职责分明的Agent之间不设置从属关系, 可自由通信, 保证系统的灵活性和高效性。

为了保证系统生产管理中各流程能够平滑衔接, 信息高效传递, 本文的调度系统的结构设计采用的是混合型, 如下图2-2所示。根据现代企业制造资源的分布式特点, 同属一个资源Agent组的子管理Agent、若干车间Agent以及对应的设备Agent以层次型的结构进行呈现。由图2-2可知, 全局管理Agent与各资源Agent组存在从属关系, 上层Agent对下层Agent具有控制协调的功能, 而全局管理Agent、算法Agent、工艺Agent和监控Agent间是平等的关系, 能够自由通信, 因此本文调度系统的体系结构属于典型的混合型结构, 柔性生产动态调度系统应用该结构具有如下优点:

(1) 易于资源管理和系统扩展。资源Agent组的结构符合现代制造系统的资源分布特点, 制造业的资源结构基本是以“工厂-车间-设备”的方式进行呈现, 一个加工工厂内设置若干车间, 每个车间根据所生产加工的产品、零件来配备对应的生产设备, 而企业可能在多个地理位置分布有加工工厂, 制造业资源的这种分布特点是与层次型结构一致的。系统是基于各设备Agent的仿真数据对订单任务执行分解分配过程的, 该过程涉及诸多数据和状态的共享, 通过上层Agent对下层Agent控制管理的能力, 能够更有效率协调资源Agent间的工作。

(2) 系统运行效率高, 稳定性强。资源Agent组属于层次型结构, 有全局管理Agent进行统一的管理。全局管理Agent、算法Agent、工艺Agent和监控Agent之间是松耦合关系, 属于分布式结构, 各Agent之间不存在从属关系, 消息直接可达, 无需经过层次间或中央服务器的消息中转, 通信效率高, 在协同解决大型复杂的调度任务时更灵活高效。

(3) 系统的成本降低。车间调度系统把生产管理的各个功能进行划分并分配到对应的Agent, 相当于把系统所需的计算量 and 控制量分散到各个Agent中, 降低了系统对大型高性能服务器等硬件的依赖。

图2-2系统结构图

2.5 本章小结

本章首先在Agent和MAS的相关理论基础上, 根据动态车间调度系统的业务需求, 定义了基于MAS的调度系统中Agent的职责划分, 其次通过分析各Agent中信息的输入和输出特点, 设计了系统的体系结构, 而后为了发挥Agent的优越性, 为各种类型的Agent进行了决策库的设计。

3 分布式制造环境中Agent的协调机制

3.1 柔性作业车间调度问题

3.1.1 问题描述

柔性作业车间调度问题可总结为把n个工件分配到m台机器上进行加工, 每个工件从加工到完成需要经历若干道工序, 工序

之间有顺序约束，并且每道工序存在一种或多种设备选择，分别对应不同的完成时间，通过定义各工序的机器选择、工序在机器上的加工顺序以及开始加工时间，来达到一个或多个优化目的[36]。在传统的作业车间调度问题中，设备的加工能力、工艺的加工路线确定且唯一，即工序间的加工顺序、工序的加工时间和所需的加工设备类型是确定的。而柔性作业车间调度问题中，设备能对不同的工序进行加工，工序也有多种机器选择，。如下表呈现了两个工件中工序的设备选择以及对应的加工时间。

工件工序设备选择和对应的加工时间

m1 m2 m3
p1 o1,1 - 5 3
o1,2 9 - 7
p2 o2,1 7 6 -
o2,2 3 - 2

图3-1 2*2调度实例的析取图

图3-1是该调度实例对应的析取图。图中的实线箭头表示的是同一个工件中工序的顺序约束关系，虚线箭头表示不同工件的加工是相互独立的。由于FJSP降低了工序对机器的选择约束，当工件数目或工序的设备选择增多时，对应析取图的模型将更加复杂，对应更大的解空间，这时候需要的计算量将急剧增加，因此FJSP是比传统的作业车间调度问题难度更大的NP-Hard问题。

3.1.2 FJSP的数学模型

柔性作业车间调度问题的研究中通常作出以下几点假设[37]：

- (1) 每台设备在任一时刻只能加工一道工序。
- (2) 每个工件在任一时刻只能有一道正被加工的工序，且工序加工过程中不得中止。若由于设备故障等原因导致必须中止加工，则工件报废，需重新安排另外的同种工件进行加工。
- (3) 不同工件间的加工互不影响，工件与工件之间没有顺序约束，不同工件间的工序之间也不存在顺序约束。
- (4) 初始时刻各设备为空闲状态，均可投入生产。
- (5) 工序在同属一个地区的设备上加工时忽略转移时间。

数学模型中的各个变量定义如下：

- (1) n 工件数量
- (2) m 设备数量
- (3) $P=\{p_1, p_2, \dots, p_i, \dots, p_n\}$ 工件集合， p_i 表示第 i 个工件
- (4) $M=\{m_1, m_2, \dots, m_i, \dots, m_m\}$ 设备集合， m_i 表示第 i 个设备
- (5) $O_i=\{o_{i,1}, o_{i,2}, \dots, o_{i,k}\}$ 工件 i 的工序集合， $1 \leq i \leq n$ ，其中 $o_{i,k}$ 表示工件 i 的第 k 道工序
- (6) $M C_{i,j}=m c_{i,j,k}$ 工序 $o_{i,j}$ 的设备选择序列， $m c_{i,j,k}$ 表示工序 $o_{i,j}$ 能在设备 m_k 上进行加工
- (7) $T_{i,j}=t_{i,j,k}$ 工序 $o_{i,j}$ 的加工时间序列，与设备选择集合 $M C_{i,j}$ 中元素一一对应， $t_{i,j,k}$ 表示工序 $o_{i,j}$ 在设备 m_k 上的加工时间。
- (8) $T S_i=\{t s_{i,1}, t s_{i,2}, \dots, t s_{i,k}\}$ 设备 i 的加工序列， $t s_{i,k}$ 表示设备 i 上的第 k 道加工工序
- (9) $S_{i,j}=s_{i,j,k}$ 工序 $o_{i,j}$ 的开始加工时间， $s_{i,j,k}$ 表示工序 $o_{i,j}$ 在设备 k 的加工序列上的开始加工时间
- (10) $E_{i,j}=e_{i,j,k}$ 工序 $o_{i,j}$ 的加工结束时间， $e_{i,j,k}$ 表示工序 $o_{i,j}$ 在设备 k 的加工序列上的加工结束时间
- (11) $C=c_i$ 工件 i 的完工时间
- (12) $D=d_i$ 工件 i 的交货期

FJSP约束条件的数学表达如下：

- (1) 工件相邻工序间存在顺序约束，若 $o_{i,j+1}$ 必须在 $o_{i,j}$ 完成后才能开始加工，则有 $s_{i,j+1,k} \leq e_{i,j,k'}$
- (2) 每个设备在任一时间只能加工一道工序，若工序 $o_{i,j}$ 和工序 $o_{i',j'}$ 为设备 k 加工序列中的相邻工序，则有 $e_{i,j,k} \leq s_{i',j',k}$
- (3) 工序在加工过程中不可中止： $t_{i,j,k} = e_{i,j,k} - s_{i,j,k}$

FJSP的优化目标通常是对完工时间、机器负荷、拖期时间和加工成本中的一个或多个的结合。本文在FJSP中的调度算法的优化目标设置为最小化最大完工时间，因此目标函数如下： $f = \min C_{\max} = \min(\max(C_i)), 1 \leq i \leq n$

3.2 基于多Agent的生产任务跨区域分解策略

在全球化背景下，制造企业在管理上具有社会性，在生产上具有异地性，倾向于调整资源结构和技术分布，在世界各地均设有加工工厂，各工厂根据自身的技术和设备配置生产对应的产品。当企业接收到客户的生产订单时，根据生产任务的需求以及各个工厂的生产能力决定任务的分配，若单独的工厂无法完成完整的生产任务，则企业需将任务作进一步的分解，对于其中的每一个子任务，重新进行任务的分配。本文以通信网络为媒介，通过全局管理Agent对多个资源Agent组进行管理和控制，完成生产任务在分布式工厂车间中的分解分配流程。

3.2.1 生产任务的形式化描述

在本文的调度系统中，客户的生产订单通过合法性判定后将被封装为对应的生产任务MT (Manufacturing Task)，MT可分为三种粒度的任务，分别是产品级任务PRT (product task)、工件级任务PT (part task) 和工序级任务WPT (working procedure task)，与产品的构造相似，一个产品由若干工件组成，一个工件由一组工序以及工序间约束组成。PRT、CPT、

WPT分别对应产品、工件和工序，工艺Agent中所记录产品、工件和工序在制定生产约束后均可封装为对应生产任务。PRT、CPT和WPT。

MT的形式化描述如下：

MT={MTInfo, Constraint, Construction, Resource}

MTInfo(Manufacturing Task Information)描述了生产任务的基本信息,使用一个四元组 (ID,NAME,TYPE,DESCR) 表示，其中的ID是MT的唯一标识；NAME是MT的名称，概括了生产对象的基本特征；TYPE指定了MT的粒度，即MT属于PRT、PT或WPT。

指 标		
疑似剽窃文字表述		
<div>1. 加工路线确定且唯一，即工序间的加工顺序、工序的加工时间和所需的加工设备类型是确定的。而柔性作业车间调度问题中，设备能对不同的工序进行加工，</div> <div>2. 析取图。图中的实线箭头表示的是同一个工件中工序的顺序约束关系，虚线箭头表示不同工件的</div>		
3. 71068966119975096_梁乐乐_基于多Agent的柔性生产动态调度系统研究_第3部分		总字数：9321
相似文献列表 文字复制比：2.8%(261) 疑似剽窃观点：(0)		
1	蚁群算法在公交路径查询中的应用 李娜;刘天时;杨莹;-《计算机时代》-2009-02-15	0.5% (44) 是否引证：否
2	基于蚁群算法的最短路径问题的研究和应用 黄贵玲;高西全;靳松杰;谈飞洋;-《计算机工程与应用》-2007-05-01	0.4% (37) 是否引证：否
3	智能停车诱导系统的设计 秦澄;徐雪峰;-《科技视界》-2013-07-15	0.4% (34) 是否引证：否
4	现代大型停车场车位诱导优化算法及仿真 王一军;陶杰;-《计算机仿真》-2007-11-15	0.4% (34) 是否引证：否
5	基于网边缘控制的QoS路由算法 樊秀梅;林闯;崔逊学-《清华大学学报(自然科学版)》-2004-01-30	0.3% (32) 是否引证：否
6	基于群集智能技术的网络路由算法研究 马立肖;赵占芳;-《现代计算机》-2006-11-25	0.3% (32) 是否引证：否
7	蚁群算法研究及前景 王焱伟;-《电脑知识与技术》-2016-12-15	0.3% (31) 是否引证：否
8	100606311_李国玉_基于蚁群算法的路径优化应用 李国玉-《大学生论文联合比对应》-2014-05-10	0.3% (31) 是否引证：否
9	一种基于CVRP的改进蚁群算法 王书勤;黄茜;-《甘肃科技》-2011-02-15	0.3% (29) 是否引证：否
10	一种求解军事指派问题的改进蚁群算法 王书勤;黄茜;-《兵工自动化》-2013-06-15	0.3% (29) 是否引证：否
11	自适应蚁群算法在流水车间调度问题上的应用 朱益江;-《连云港职业技术学院学报》-2007-12-30	0.3% (29) 是否引证：否
12	自适应蚁群算法在Flow Shop调度问题上的应用研究 朱益江;-《常州工学院学报》-2007-12-15	0.3% (29) 是否引证：否
13	基于蚁群算法的船舶航线设计方法的研究 朱青(导师：郝燕玲)-《哈尔滨工程大学硕士论文》-2010-12-01	0.3% (29) 是否引证：否
原文内容		

DESCR (Description) 是对生产对象的详细描述，如CPT中DESCR描述了工件的精度和规格等信息，PT中描述了工序加工的注意事项等。

Constraint描述了MT的生产约束，使用三元组 (TIME，COST，QUALITY) 来表示，其中TIME为MT对完工时间，即交货期的约束；COST为MT对生产成本的约束；QUALITY为MT对生产质量的约束，如工件在规格、精度上的要求。由于PT是

PRT的子任务，WPT为PT的子任务，因此TIME由上级任务指定。

Construction描述了生产任务的构成以及子任务的约束关系，例如PRT由若干PT组成；CPT由一组WPT按照一定的约束关系组成。Construction可以描述为：

Construction={COMP,PROCESS}

COMP={MT1, MT2,..., MTi, ..., MTn}

COMP (Composition) 是生产任务MT的子任务集合，其中MTi是MT的第i个子任务。当MT的粒度为WPT时，任务无法再分解，因此COMP为空。PROCESS是MT的生产工艺，呈现了子任务完成的顺序约束，例如当MT的粒度为PRT时，PROCESS指定了工件级子任务的生产对象的组装顺序；当MT粒度为CPT时，PROCESS指定了工序级子任务加工的顺序约束。

Resource描述了生产任务MT对资源的需求，表示如下：

Resource={M,HR,E,T}

其中M (Material) 表示生产任务MT对原材料的需求；HR (Human Resource) 表示MT在生产过程中对技术人员的需求；E (Equipment) 表示可用于完成MT的设备序列；T (Time) 为加工时间序列，序列中的元素与E中元素一一对应，表示每一个机器选择对应的加工时间。

3.2.2 基于多Agent的生产任务跨区域分解流程

目前主流的生产任务的调度规划通常是根据任务对设备、原材料、精度、尺寸等要求以及产品的工艺约束，建立数学模型，使用如局部搜索、人工智能算法等进行单目标或多目标的调度方案求解，以上的调度过程是基于资源集中配置的前提下进行的，即资源的分布局限在一间工厂的一个或多个车间，因此对于资源具有异地性的企业通常无法适用。

本文的基于多Agent的生产任务的跨区域分解策略的中心思想是把任务的分解和分配过程进行一定程度的分离，首先全局管理Agent以网络为媒介，把生产任务MT广播至各个资源Agent组，资源Agent组和算法Agent相互协调，根据设备的加工能力计算MT对应的调度方案；而后全局管理Agent从各个资源Agent的调度结果中筛选最优的调度方案，若不存在资源Agent组独立完成生产任务，则全局管理Agent需对MT进一步分解，对于每一个子任务重新执行跨区域分解策略。

全局管理Agent与资源Agent、资源Agent组内的各个Agent间均是分层式结构，在生产任务分解过程中，通过上层Agent对MT的分解，协调下层子Agent对子任务的匹配，最后从各个匹配结果中筛选最优的调度方案，因此当管理Agent或资源Agent接收到MT，且根据MT的COMP进行分解得到子任务MT1, MT2,..., MTn,其下第i个子Agent和算法Agent对子任务MTk的调度方案定义为Ti,k，则跨区域生产任务分解过程中，各Agent间协商的目标函数可表示如下： $f=\min_{1 \leq i \leq m} \max_{1 \leq k \leq n} T_{i,k}$

其中n为MT的子任务集合COMP中任务数量，m为上层Agent的子Agent数量。

本文的基于多Agent的订单任务的跨区域分解策略的执行过程遵循以下原则：

(1) 生产任务MT的分解顺序需符合产品结构，按照产品级任务PRT、工件级任务PT、工序级任务WPT从上往下逐层分解，其中PRT是若干PT的集合，PT是若干WPT的结合。

(2) 生产任务MT分解过程中的传递顺序需符合全局管理Agent与资源Agent组之间的分层式结构，即按照全局管理Agent、子管理Agent、车间Agent的顺序从上往下逐层传递。

(3) MT的粒度与资源Agent组的结构需一一对应。PRT、PT、WPT分别与子管理Agent、车间Agent、设备Agent对应。当MT粒度大于Agent级别时，资源Agent需对TM进一步分解；当TM粒度小于自身粒度时，直接把TM发布至其下的各个资源Agent；当两者级别相同方可进行能力判定，即判断自身的设备配置能否满足MT对制造资源的需求。

(4) 尽量保证生产任务MT的粒度。随着生产任务MT分解程度的加深，最终参与分解的任务粒度将越来越小，极端情况下会导致一个工件级别的生产任务PT包含的多个WPT被分配到不同地区的生产设备上，在实际生产中，这需把工件在多个地区工厂车间之间进行转移，从而大大提高了各种费用和成本。因此，当某个粒度的任务分解所得到的设备组合能够满足任务的交货期时，不再进行进一步的分解。其次，除非征得客户的同意，传递到各个资源Agent组的生产任务的最小粒度为工件级任务，即同属一个工件的工序任务不可被分散到不同地区的工厂中。

基于多Agent生产任务跨区域分解策略的执行流程如下图所示，可以细分为以下步骤：

步骤一：封装生产任务。全局管理Agent接收到客户的生产订单后，进行订单合法性判定。合法性通过后，根据产品结构按照工序、工件、产品的顺序从下往上进行封装，得到生产任务MT。

步骤二：任务发布。全局管理Agent把MT广播至其下各个资源Agent组。

步骤三：资源Agent组内进行粒度判定以及任务分解。上层Agent接收到MT后，若MT粒度与Agent级别对应，进行能力判定，执行步骤四；若MT粒度小于Agent的级别，直接把TM广播至其下各个子Agent，由子Agent进行能力判定；若MT粒度大于资源Agent的级别，需对MT的子任务集合COMP进一步分解，得到若干子任务MTk，并逐一把子任务广播至其下的子Agent，即执行步骤二。

步骤四：能力判定。当资源Agent级别与MT粒度相同时，进行能力判定，即根据自身的原材料和设备配置与MT对资源的需求进行对比，若匹配成功执行步骤五，否则向上层Agent返回MT无法完成的结果，执行步骤六。

步骤五：计算调度方案。资源Agent把MT和设备集发送至算法Agent，由算法Agent基于MT的加工约束和设备的加工能力以及工作状态计算并返回调度方案到上层Agent。

步骤六：结果筛选。上层Agent收集到下层子Agent对MT的判定结果，分两种情况进行处理：

(a) 所有子Agent均返回无法完成的判定结果。若MT可进一步分解，则对分解得到的子任务集，逐个发布到下层子Agent，即执行步骤二；若MT无法再分解（粒度为WPT），则向上层Agent返回无法完成的判定结果，若此时Agent位于最上层，即为全局管理Agent，表明系统无法完成任务，向客户返回判定结果，结束分解流程。

(b) 存在子Agent能够完成任务，从返回的结果中选择用时最少的调度方案，返回到上层Agent，若Agent为全局管理Agent，进行交货期的判定。若该方案满足任务的交货期，则结束分解流程，该方案作为最终的调度方案；若该方案不满足MT的交货期且MT可进一步分解，分解任务后执行步骤二，若MT不可分解，表明系统无法完成任务，向客户返回判定结果，结束分解流程。

图3-2 生产任务跨区域分解流程

3.3 基于多Agent的异常调度策略

当生产过程出现异常因素时，系统需要及时、快速地进行响应并采取对应的措施，进行调度方案的调整，即开启异常调度过程。系统面临的异常因素通常包括设备故障、订单取消、紧急订单、原材料短缺等等。由于全局管理Agent在订单任务审核中对任务所要求的原材料和系统库存原材料进行比对，因此本文主要对设备故障、紧急订单以及订单取消三种情况的重调度策略进行说明。

3.3.1 紧急订单处理的异常调度策略

当生产订单被标记为紧急订单时，对应的生产任务拥有最高优先级，本文把该最高优先级的作用范围分为两种：即局部最高优先级和全局最高优先级，系统根据两种优先级从而采用以下的重调度策略：

(1) 局部最高优先级是指紧急订单对应的生产任务在全局管理Agent的任务队列中具备最高优先级。紧急任务被插入到任务队列的最前端，按照跨区域的生产任务分解、算法Agent执行调度策略计算等流程最终筛选出最优的调度方案，该过程是基于系统内设备Agent原有的加工序列进行的，不对之前完成任务分解分配的生产订单造成影响。

(2) 全局最高优先级是指紧急订单在系统所有生产任务中具有最高优先级，包括之前已完成任务分解分配的生产任务。该情况下系统先进行一次打赌，即按照局部最高优先级来对紧急任务进行分解分配，若此过程汇总筛选的调度方案满足紧急订单的交货期，表明能够兼顾紧急任务和之前已完成分配的生产任务两者的需求。若打赌失败，对系统中所有设备Agent所记录的加工序列进行备份后清空，而后系统在各设备空闲的前提下进行跨区域生产任务分解并由算法Agent计算、筛选最优的调度方案。最后把对设备Agent原有的生产序列备份的工序任务，按照相对顺序放回到对应的工序序列中。

3.3.2 设备故障处理的异常调度策略

当设备由于零部件磨损、断裂等导致设备故障无法运行时[38]，将导致原本安排给该设备的生产任务无法进行加工，同时由于工件中工序存在顺序约束，同属一工件的安排在正常设备上的工序任务也无法按照原生产计划进行，因此必须调整原有的调度方案。

在对设备故障过程中作出如下设定：

(1) 工序加工过程不可中断。若当设备正在加工某道工序时发生故障从而必须停工时，则对应的工件报废，不可继续加工，需安排工件重走完整的加工流程。

(2) 工件任务由所属地区的工厂优先处理。当工件任务执行过程中由于设备故障导致后续工序无法按照原本的分配方案加工时，先查看该工件所处工厂的设备配置是否满足后续工序的机器选择，若满足，把工件安排到替换设备上加工，否则交由全局管理Agent重新进行工件任务的分解分配流程。

(3) 若重调度后调整所得的若干调度方案均无法满足生产任务的交货期，从中选择延迟时间最小的方案，并提交到客户说明原因。

设备故障处理的重调度策略执行过程如下：

(1) 监控设备检测到设备故障，对应的监控Agent读取故障信息并进行封装，发送到故障设备所属资源Agent组的子管理Agent。

(2) 工序回收与工件任务封装。首先子管理Agent通知车间管理Agent把故障设备Agent未执行加工的工序任务回收，并回收同属一工件的安排在正常设备上的工序任务，而后把这些任务重新封装为新的工件任务，封装过程为：首先确定安排在故障设备上的工序所属的工件任务集合 $\{p_1, p_2, \dots, p_k\}$ ，然后把集合内的工件剩余的后续未加工工序按照工序约束顺序重新封装，得到新的工件任务集合 $P=\{p_1, p_2, \dots, p_k\}$ 。

(3) 车间Agent移除故障设备对应的设备Agent，子管理Agent根据其下各车间Agent的设备配置与集合P中各工件任务的设备需求进行对比，若存在可替换设备，则该资源Agent组按照生产任务分解策略重新计算调度方案。若不存在可替换设备，则子管理Agent需把工件任务集合P发送至全局管理Agent，由全局管理Agent和其他资源Agent组中进行任务集合分解协商，求解新的调度方案。

(4) 若求解的调度方案无法满足生产任务的交货期，需向客户返回延迟时间并进行原因说明。

3.3.3 订单取消处理的异常调度策略

当全局管理Agent接收到取消生产任务的请求后，若该任务处于全局管理Agent的任务队列中，尚未进行任务的分解分配过程，则可直接从任务队列中删除该任务。若该任务已经完成任务分解流程，其中的各工序任务已经安排到不同设备Agent的加

工序列中，则全局管理Agent需向各个资源Agent的发送撤除请求，由子管理Agent通过层次间信息传递来通知其中的设备Agent移除属于该订单任务的工序任务，此时各设备Agent的工序队列中将出现空闲的时间片，可在满足工序顺序约束的前提下移动各生产任务，能够减小各生产任务的交货期。

3.4 本章小结

本章通过定义各类型Agent间信息的传递和反馈，来提高系统对外界信息的灵敏度，其中重点描述了生产任务的数学表达模型，以及生产任务在多个资源Agent组之间的分解流程，同时在异常状况发生条件下Agent之间的合作流程。

4 多Agent柔性生产调度系统中的调度算法

在基于多Agent的生产任务跨区域分解过程中，每个资源Agent组对应为一个子调度系统，根据自身的设备配置与生产任务对制造资源的需求进行匹配，当匹配成功后，交由算法Agent执行调度算法计算调度方案，最终由全局管理Agent从各个资源Agent组筛选完工时间最短的调度结果。本文使用蚁群算法作为动态调度系统中的调度算法，由算法Agent进行封装，为资源Agent组提供计算服务。

4.1 基本蚁群算法的引进

M.Dorigo等意大利学者在观察蚁群觅食过程中发现，蚁群在预先没有与食物相关信息指引的情况下，总能找到一条连接食物源和蚁巢的路径[39]。学者们通过分析蚂蚁觅食行为以及相互间的作用关系，总结了影响蚂蚁行动决策的因素，并由此提出了蚁群算法。蚁群算法属于仿生学的启发式算法[40]，多用于图论中最短路径的搜索问题，算法通过模仿蚁群觅食过程中的行为以及蚂蚁间、蚂蚁与环境间的相互作用规律，使问题求解能够逐渐收敛，得到最优解。

蚁群在觅食过程中存在如下的行为和规律[41]：

(1) 蚂蚁从蚁巢出发寻找食物时，会沿途洒下一种化学物质，即信息素。信息素对其他蚂蚁有一定的指引作用，信息素浓度越高的方向，蚂蚁选择的概率越大。

(2) 蚂蚁的行为具有一定的随机性。当周围环境没有信息素指引时，蚂蚁会概率性地选取觅食方向；当环境中存在信息素指引时，蚂蚁并不完全遵循信息素浓度最高的方向，而是存在一定的概率探索信息素浓度较低的方向。

(3) 信息素会自行挥发。随着时间的进行，信息素会挥发直至完全消失。因此离蚁巢越远，信息素浓度越低。

因此，当一只蚂蚁沿某条路径寻找到食物后，沿途洒下的信息素会吸引其他蚂蚁，这些蚂蚁在行动过程中同样会洒下信息素，吸引更多的蚂蚁选择该路径，随着时间的推移，路径上的信息素浓度将越来越高，这是一个正反馈的过程，因此蚁群总能找到一条连接蚁巢和食物源的路径。又由于蚂蚁的行动具有随机性，存在少数蚂蚁独辟蹊径，选择信息素浓度低甚至没有信息素的方向，当蚂蚁寻找到一条通往食物源更短的路径，吸引其他蚂蚁的同时，由于路径短，所花时间少，残留信息素更多，因此选择该路径的蚂蚁会逐渐增多，最终蚁群会收敛到更短的路径。由此可看出，蚁群觅食过程中信息素的正反馈作用能够指引蚁群寻找到食物，而蚂蚁之间行动中所具备的一定随机性可以使蚁群聚集到最优的路径。根据蚁群算法的原理，可以总结出该算法具有如下的优点[42]：

(1) 逻辑清晰简洁、易于实现

蚁群在自然环境中的觅食规则非常简单，所受干扰和限制少，遵循“初始化——状态转移——洒下信息素——目标判定——更新信息素”的简洁行为模式，同时对应的数学模型简单，计算量较小，易于实现。

(2) 较强的鲁棒性

蚁群算法最终的输出结果是通过多次循环中对每一只蚂蚁的全局搜索结果进行比较、筛选来得到的。因此即使存在蚂蚁在较低概率下求解得到较差的结果，对当前的全局最优结果影响较小，同时蚂蚁间信息素的相互指引作用能有效降低搜索过程中接受坏选择的几率。

(3) 正反馈

蚁群觅食过程中依靠信息素相互作用、配合，呈现出很强的社会性。每只蚂蚁在路径搜索过程中洒下的信息素会对全体蚂蚁当前以及未来的行动具有指引作用。受到指引的蚂蚁同时也会洒下信息素，进一步增强信息素浓度，循环反复，将导致越来越多的蚂蚁集中到某条路径。因此对应的算法中，蚂蚁根据状态转移概率公式决定可达范围的位置的同时，也会更新环境信息素相互影响，直至达到共同的优化目标。

4.2 基本蚁群算法在FJSP中的应用

基本蚁群算法遵循蚁群觅食过程中体现的社会性以及蚂蚁和环境间的相互作用，其执行流程如图。可以分为以下几个步骤

(1) 定义觅食环境以及初始化蚁群

初始化环境中的起始信息素浓度，定义蚁群的规模、蚂蚁的搜索规则、各蚂蚁的起始位置等。

(2) 定义搜索目标

定义蚁群的搜索目标，如蚁巢距离食物的路径最短、单位距离中食物量最大等等。

(3) 蚂蚁位置移动

蚂蚁根据环境中各个方向的信息素浓度以及当前环境约束（如障碍物）等概率性地选择自身下一步的位置。

(4) 撒下信息素

蚂蚁每进行一次位置转移都需要在所经过的路径中洒下一定量的信息素。

(5) 终止判定

蚁群终止搜索的判断依据一般有两个：是否达到搜索目标以及搜索时间。当蚁群搜索时间达到一定值时，终止搜索。搜索时间在算法中一般使用迭代次数表示。

(6) 信息素挥发

随着时间的推移，蚂蚁洒下的信息素会按照一定的比例挥发直至完全消失。

基本蚁群算法在根据搜索目标、搜索规则以及信息素浓度等进行概率性位置转移的特点非常适用于旅行商问题 (Traveling Salesman Problem, TSP) 的求解[39]。TSP定义了n座城市，其中任意两座城市之间均可达，推销员需要从某个起点城市出发，仅访问每个城市一次后回到起点，要求推销员所经过的回路路程最短[43]。TSP等同于在一各完全图中寻找一条路径最短的哈密顿回路，这是典型的NP完全问题，随着完全图中节点数目增加，经典的精确算法无法在多项式时间内进行求解。

FJSP (以最小化最大完工时间为单优化目标) 与TSP非常相似：两者均属于图论中的最短路径搜索问题，其中TSP等同于要求在完全图中寻找一条最短回路，FJSP要求在若干工件的工艺规划网络图中寻找一条用时最少的加工路径；TSP中每个城市只能被访问一次，FJSP中每个工序只能被加工一次；FJSP和TSP同为NP完全问题，精确算法难以求解。

FJSP和TSP之间存在如下几点差异：

(1) TSP中任意两个城市之间可达，若干城市构成了一个完全图。FJSP中同属一个工件的工序之间存在顺序约束，工序的加工顺序必须严格按照约束。

(2) TSP求解的路径是一个回路，达到最后一个城市时需返回到起点城市。FJSP所求的加工路径要求包括每个工件工艺规划网络图的起始节点S和终止节点E，不要求是回路。

(3) TSP求解的路径要求包含每一个城市，FJSP中蚂蚁同样要求包含每一个工序，但工序存在机器选择，因此在进行状态转移，即从一个节点转移到另一个节点时，蚂蚁需要面临工序的机器选择问题。

为阐述基本蚁群算法在FJSP的执行过程，先对其中的数学符号进行说明： w : 蚂蚁数量 N : 算法循环次数 Q : 每轮循环中蚂蚁洒下的信息素总量 $\tau_{ij}(t)$: t 时刻信息素浓度矩阵, $1 \leq i, j \leq n$ 。 τ_{ij} 表示从工序 i 转移到工序 j 路径上的信息素浓度。 τ_0 : 各工序间初始信息素浓度 ρ : 信息素挥发因子 $\Delta\tau_{ij}(t)$: t 时刻在工序 i 和工序 j 的路径间洒下的信息素增量 $\eta_{ij}(t)$: t 时刻蚂蚁从工序 i 转移到工序 j 的启发式，由工序 j 的完工时间决定 $P_{ij,x}(t)$: t 时刻蚂蚁 x 从工序 i 转移到工序 j 的概率 Tabu_x : 蚂蚁 x 在工序 i 时的禁忌池，记录了蚂蚁 k 选择工序 i 时走过所有工序 allowed_x : 蚂蚁 x 在工序 i 时的可选池，记录了蚂蚁 k 完成工序 i 时下一步可选的工序

下面根据FJSP和TSP中的差异对上述符号表示进行细节的说明：

(1) FJSP中每个工序存在机器选择，每个选择对应一种加工时间。例如假设存在两个工序，工序1和工序2，以及三个设备，设备 m_1 、设备 m_2 和设备 m_3 。两个工序的机器选择如下表。由表可知工序1可在设备1和设备3上进行加工，对应的加工时间是3和6；工序2可在设备1、2和3上加工，对应的加工时间分别是2、5和1。因此工序1的设备选择序列 $\text{MC}_1=(m_1, m_3)$ ，加工时间序列为 $T_1=(3, 5)$ 。同理可得 $\text{MC}_2=(m_1, m_2, m_3)$, $T_2=(2, 5, 1)$ 。

设备 m_1 设备 m_2 设备 m_3

工序1 3 -- 6

工序2 2 5 1

(2) TSP和FJSP中禁忌池 Tabu 都是用于记录蚂蚁已经走过的节点，但两者中的可选池 allowed 有所差异。TSP中由于任意两个节点间均可达，因此可选池包含除禁忌池中节点外的所有节点；FJSP中由于工件的工序间存在顺序约束，因此可选池中包含的节点只能是禁忌池节点中的后续节点。假设FJSP需要对如下图4-1中两个工件进行调度，算法执行过程中某只蚂蚁已走过节点 $o_{1,1}$, $o_{1,2}$ 和 $o_{2,1}$ ，即 $\text{tabu}=\{o_{1,1}, o_{1,2}, o_{2,1}\}$ ，则根据工序间的约束关系此时 $\text{allowed}=\{o_{1,3}, o_{2,1}\}$ 。

图4-1 两个工件的工艺网络图

(3) 蚂蚁在经过的路径需要洒下信息素，随着时间的推移，信息素会自行挥发直至完全消失，因此需要对信息素矩阵进行更新，更新公式如下： $\tau_{ij}(t+1)=1-\rho\tau_{ij}(t)+\Delta\tau_{ij}(t)$ $\Delta\tau_{ij}(t)=\sum_{x=1}^m \Delta\tau_{ij,x}(t)$

其中， $\Delta\tau_{ij,x}(t)$ 为 t 时刻蚁群 x 在工序 i 和工序 j 之间洒下的信息素浓度

蚂蚁洒下信息素的时机有两种，一种是每次进行状态转移时洒下信息素，另一种是在每轮结束时更新信息素，本文选择后者，如下： $\Delta\tau_{ij,x}(t)=Q/L_k$, & 蚂蚁 x 经过路径 (i,j) , 0, & 蚂蚁 x 不经过 (i,j)

其中 Q 是定值， L_k 是蚂蚁经过的路径总长度，在FJSP中 L_k 代表蚂蚁走过的路径对应的完工时间。因此完工时间越长的路径，单位长度撒下的信息素量就越少。

(4) 蚂蚁 x 从工序 i 转移到工序 j 的概率计算公式为： $P_{ij,x}(t)=\tau_{ij,\alpha}(t) \times \eta_{ij,\beta}(t) / \sum_{s \in \text{allowed}_x} \tau_{is,\alpha}(t) \times \eta_{is,\beta}(t)$, & $j \in \text{allowed}_x$, 0, & otherwise

其中， η_{ij} 是蚂蚁从工序 i 转移到工序 j 的启发式，由工序 j 的完工时间决定，由于工序 j 存在机器选择序列 MC_j ，因此 η_{ij} 的计算表达式如下： $\eta_{ij}(t)=k/\text{MC}_j \eta_{ij,k}(t)$, & $j \in \text{allowed}_x$, 0, & otherwise $\eta_{ij,k}(t)=1/c_{i,j,k}$, $j \in \text{allowed}_x$, $k \in \text{MC}_j \eta_{ij,k}$ 为蚂蚁在工序 i 时转移到工序 j 的设备选择 $\text{mc}_{j,k}$ 的启发式， $c_{i,j,k}$ 是蚂蚁选择 $\text{mc}_{j,k}$ 后工序 j 的完工时间。

1. 分析蚂蚁觅食行为以及相互间的作用关系，总结了影响蚂蚁行动决策的因素，并由此提出了蚁群算法。
2. 信息素，吸引更多的蚂蚁选择该路径，随着时间的推移，路径上的信息素浓度将越来越高，

4. 71068966119975096 梁乐乐 基于多Agent的柔性生产动态调度系统研究 第4部分

总字数：10120

相似文献列表 文字复制比：1.3%(127) 疑似剽窃观点：(0)

1	<u>基于多核的并行粒子滤波运动目标跟踪</u> 王爱侠;李晶皎;王青;王骄; - 《计算机科学》 - 2012-08-15	0.5% (51) 是否引证：否
2	<u>面向智能交通的路径规划相关技术研究</u> 王俊(导师：谢俊元) - 《南京大学硕士论文》 - 2013-05-01	0.4% (39) 是否引证：否
3	<u>基于MOSS的网上订餐系统的设计与实现</u> 徐凤;邓楠;王鹤翔;赵博雅; - 《计算机光盘软件与应用》 - 2013-03-15	0.3% (31) 是否引证：否
原文内容		

根据以上数学模型，基本蚁群算法在FJSP中的执行流程总结如下：

(1) 初始化。定义蚂蚁数量 m 、初始信息素浓度、优化目标(完工时间等)、最大循环次数 N ,为每只蚂蚁设置可选池和禁忌池、当前最优路径等。

(2) 执行N次迭代。在每次迭代开启前, 清空所有蚂蚁的可选池和禁忌池, 把每个工件的起始节点添加到每只蚂蚁的可选池中。蚂蚁根据环境信息素浓度和状态转移概率公式从可选池中选择下一个移动的节点。选择某个节点后, 把该节点加入到禁忌池中, 同时把其后续节点添加到可选池中。当且仅当蚂蚁遍历所有工序后, 结束循环。循环结束前需把本轮最优路径与全局最优路径进行对比, 若前者更优, 则进行替换。

(3) 结束一轮循环后, 进行终止条件判定。若当前最优路径达到优化目标或者循环次数到达N, 可结束算法流程; 否则进行信息素更新, 并开启下一轮循环。

(4) 结束算法流程, 输出全局最优路径。

4.3 基本蚁群算法在FJSP中的改进

基本蚁群算法的缺点主要有两个：收敛速度较慢和较容易偏离最优解，陷入局部最优。算法初始由于迭代次数少，环境信息素初始浓度低，各路径信息素浓度无差异，对蚂蚁的指引作用弱，蚂蚁更多是根据可达范围内工序的完工时间来概率选择转移的工序，而后需要经过多次迭代累积信息素，当不同工序节点的信息素浓度存在较大差异时，才能指引大多数蚂蚁集中到某一条路径，因此算法初始收敛速度较慢，需要的执行时间较长；蚁群间信息素的相互指引、作用使算法具备了正反馈特性，保证算法最终能收敛到某个可行解，但同时FJSP中工件的工序间存在顺序约束，导致蚂蚁的能见度低，在初始信息素浓度较低的情况下容易受可见范围内工序的完工时间影响而选择较差的路径，而后信息素的释放又会彼此相互吸引，从而蚂蚁搜索过程中易陷入局部最优解。若算法迭代次数较少，可能导致蚂蚁没有机会搜索其他路径，无法跳出局部最优解。

目前最大最小蚂蚁系统 (Max-Min Ant System, MMAS) 是使用较为广泛的对基本蚁群算法的改进策略, 经众多研究人员的实验, MMAS在TSP的求解有着理想的优化性能[44]。MMAS主要改进了信息素更新规则, 在每次迭代中, 只对全局或本轮迭代的最佳路径进行信息素的更新, 该方法能够更大程度地提高蚁群算法的正反馈, 加快收敛速度, 本文在参考MMAS的改进策略的基础上, 提出如下的信息素更新公式: $\tau_{i,j}(t+1) = 1 - \rho \tau_{i,j}(t) + \Delta \tau_{i,j}(t)$ $\Delta \tau_{i,j}(t) = x = 1/m \Delta \tau_{i,jx}(t) \Delta \tau_{i,jx}(t) = Q/Lx \times e/Lx - L_{best}$, & 蚂蚁 x 为最佳蚂蚁且 $(i,j) \in Lx0$. & otherwise

其中, L_{best} 为蚁群的全局最优路径, 假设蚂蚁 x 为本次迭代的最佳蚂蚁, 对应的最佳搜索路径为 L_x , 由上式可知, 当 L_x 优于 L_{best} , 且优于的程度越高, 信息素增量越大, 这有助于提高蚁群路径搜索的收敛速度; 若 L_x 不如 L_{best} , 说明 L_x 路径上信息素浓度偏高, 且偏离最优解, 因此需要降低该路径上的信息素增量, 能降低蚁群陷入局部最优的概率。同时, 为了避免某一路径上信息素过度积累, 导致蚁群无法探索其他的路径, 过于早熟, 为节点间的信息素浓度设置一个最大值 τ_{max} 和最小值 τ_{min} , 这样可以避免某条路径通过正反馈不断积累信息素导致蚁群无法跳出局部最优解, 也可避免某些路径信息素挥发导致信息素浓度过低, 缩小蚁群的搜索空间。

在蚁群进行路径搜索过程中会记录全局的最佳路径，因此若存在蚂蚁当前的路径时间大于全局最优路径时，可提前结束该蚂蚁的搜索，减少算法的计算量。同时，基本蚁群算法初始化时是把所有工件的起始工序添加到各只蚂蚁的可选池中，此时各节点的信息素浓度相同，对蚂蚁没有指引作用，因此大部分蚂蚁可能选择同一个工序作为搜索路径的起点，将较大概率集中于某一条局部最优路径中。同时由于初始信息素浓度低，蚂蚁搜索过程中主要受可选池中工序的完工时间的影响进行状态转移，当某条路径信息素与其他路径产生较大差异时，同样容易导致算法早熟。因此，本文在前10%的迭代中，把蚁群平均分配到各个工件的起始工序的各个设备选择中，同时忽略信息素的指引作用，即信息素权重系数满足以下公式： $\alpha=0$, $\&count \leq 0.1 \times N \&count0$. &else

其中 α_0 为算法设置的信息素权重系数，count为算法当前的迭代次数。随着迭代次数的增加，环境信息素逐渐积累并产生差异，对蚂蚁的行动具有更强的指导意义，能够扩大蚂蚁的搜索范围，降低算法早熟的概率。

本文改进的蚁群算法的执行流程可总结如下图，详细过程如下：

- (1) 初始化算法参数，包括环境初始信息素浓度，迭代次数、信息素权重系数、能见度权重系数等。
- (2) 初始化蚁群，包括蚂蚁数量、蚂蚁的禁忌池和可选池。
- (3) 判断迭代次数 $\text{count} < 0.1N$ 是否成立，若成立，均匀分布蚂蚁的初始位置，即把工件的初始工序的及其选择均匀地分配到每一只蚂蚁的禁忌池中，同时把该工序的后续工序以及其他工件的起始工序添加到蚂蚁的可选池中；若不成立，把所有工件的起始工序添加到每只蚂蚁的可选池中。
- (4) 状态转移概率计算。通过状态转移公式从可选池的工序中选择下一个工序。
- (5) 把当前已选择的工序添加至禁忌池，并把其后续工序添加到可选池。
- (6) 计算蚂蚁当前路径的时间是否大于全局最佳路径，若是，结束该蚂蚁的搜索。
- (7) 判断蚂蚁的可选池是否为空，若不为空，转至步骤(4)。
- (8) 从蚁群中筛选本轮迭代中的最佳路径，并与全局最佳路径进行对比，若本轮最佳路径更优，则进行替换。
- (9) 信息素更新。判断当前的迭代次数 count 是否满足 $\text{count} < 0.1N$ ，若满足，忽略信息素更新，否则只对本轮最佳路径进行信息素更新。
- (10) 判断迭代次数已达最大值 N ，若是，输出全局最优路径并结束算法，否则跳转至步骤(2)。

4.4 本章小结

本章应用了蚁群算法作为系统的调度算法，提出了基本蚁群算法在FJSP中的数学模型，并根据算法收敛速度慢，易于陷入局部最优的缺点，通过定义信息素更新方式以及蚂蚁的均匀分布来对算法进行改进。

5 多Agent柔性生产动态调度系统的实现

5.1 制造企业的生产管理流程

调度系统的实现需要符合制造企业实际的生产管理需求，制造企业传统的生产管理流程如下图5-1所示，当客户下达生产订单时，由销售部负责接收并处理，销售部根据订单内容提炼客户对产品数量、精度、规格、交货期等要求，并转交到工艺部；工艺部根据产品的技术指标，从数据库中获取符合要求的工件加工工艺信息，而后由生产计划部转接；生产计划部的车间调度员根据工件的加工工艺、车间的设备配置以及订单的交货期，完成工序的机器分配和顺序的确定工作，然后统计生产过程所需的物料清单并提交到采购部；采购人员检查库存，确定物料充足后，通知生产计划部安排生产。

图5-1 生产管理流程

5.2 柔性生产动态调度系统需求分析

制造企业传统的生产管理流程高度依赖人力，生产计划的制定依靠个人的先验知识，在生产订单频繁变更、工艺复杂的产品生产中难以保证生产计划的质量和效率。其次信息由人为传递，可能出现传递不及时甚至传递错误信息的情况，当出现设备故障时，信息人为传递的滞后性会干扰正常的生产流程。针对以上不足，柔性生产动态调度系统应能满足企业的自动化和信息化需求，集成生产管理的各个环节，提供先进的调度机制，保证高质量和高效率的生产计划的同时能够实时监测系统的生产环境，快速响应各种动态变化。因此本文的调度系统设置如图5-2所示的几个功能模块：

登录模块：提供用户的登录和注册功能，根据用户权限提供对应的系统操作。

信息管理模块：提供对员工信息和客户信息的增删查改，当用户是系统管理员时，可以修改其他用户的使用权限。

资源管理模块：可用于对各个地区的工厂中车间结构以及设备配置进行查看和修改，同时可查看设备的工作状态和进度。

工艺管理模块：对系统内产品的工艺信息进行管理，可查看产品的组成，工件的加工工艺等。

生产调度模块：用于设置参与调度的生产任务并制定生产计划，提供对改进蚁群算法的参数设置以优化调度结果。

系统帮助模块：提供系统操作指南帮助用户快速掌握系统的使用流程。

图5-2 生产调度系统功能模块划分

5.3 多Agent系统的实现

5.3.1 开发环境及工具

操作系统：Microsoft Windows 10 Professional

开发环境：JDK1.8.0_144

开发平台：Eclipse4.7.0

开发语言：Java

数据库：MySQL5.6.17

5.3.2 系统软件架构设计

本文的调度系统属于混合型的结构。系统根据企业各地区工厂的数量建立对应的资源Agent组，每个资源Agent组中包含一个子管理Agent，负责管理和协调其中的车间Agent和设备Agent。其次各个子管理Agent之间通过全局管理Agent相互联系，由此可见全局管理Agent和各个资源Agent组之间组成了分层式的结构，上层Agent通过协调下层Agent的通信来完成生产任务的分解分配。根据以上特点，全局管理Agent和各资源Agent组之间非常适合使用C/S架构，即客户端/服务器模式。全局管理Agent、算法Agent、工艺Agent和监控Agent之间相对独立，以分布式结构联系在一起，在通信上处于平等的地位，因此适合使用点对点通信。因此，根据系统中各Agent之间的合作模式，最终把系统软件架构分为三层：表现层、应用层、数据层，如

图5-3所示。

(1) 表现层：提供友好人机交互界面，是管理员下达指令、监控生产流程的窗口。

(2) 应用层：完成调度系统生产管理的全流程，如制定生产计划、任务分解、资源分配、数据管理、异常检测等，是实现自动化、信息化和敏捷化的关键。具体表现为各类型Agent独立运行，承担由系统所分配的各项职能和任务，各司其职相互协商配合，维持系统生产调度工作的稳定有序进行。

(3) 数据层：为应用层中各Agent提供数据支持，当Agent响应管理员的指令以及相互配合达到共同目标的过程中，数据层需要提供数据的增删查改操作，该层存储管理的数据主要包括：各设备的工作状态、工作序列、历史工作数据、产品工艺信息、库存信息、设备监控信息等等。

图5-3 系统体系结构图

5.3.3 多Agent系统的面向对象设计

本文的调度系统是通过多个Agent的相互协调配合来实现生产管理的各个流程，Agent是对系统内物理实体或算法逻辑的封装，被赋予了特定的功能职责，彼此间相对独立的同时，能够通过消息传递相互关联，这些特点与面对对象设计中继承、封装和多态的特征非常匹配。Agent可看作是对对象进行更深层次的发展和完善，是一种粒度更大，具有智能性的对象[45]。因此Agent适合使用面对对象的思想来进行功能的封装和结构的设计，可使用任意的面对对象的编程语言实现，如C++、Java、C#等。考虑到Java语言具有平台无关性，易于移植，Java程序能够在任意一台安装JRE环境的主机上运行，这与Agent的移动性以及MAS易扩展的分布式特点相符，因此本文使用Java的面向对象技术来进行各类Agent的实现。如下图5-4是Agent的系统类图。

图5-4 系统类图

下面对系统中各Agent的属性定义、方法封装以及继承关系进行说明：

(1) Agent是基类，定义了Agent标识 (AID)、名称 (name)、类型 (type) 属性，同时定义了Agent初始化、启动、注册/注销以及通信相关的方法。

(2) MA (Manger Agent) 为管理Agent，继承自Agent。管理Agent和各资源Agent之间是分层式架构，因此subAgentList记录了管理Agent的子Agent，broadcast()方法用于把生产任务广播至下层子Agent，filter()方法用于筛选下层Agent返回的调度方案。

(3) GMA (Global Manager Agent) 为全局管理Agent，继承自MA，维护一个生产任务队列，队列中任务按照优先级进行排序。当接收到客户的订单时，GMA使用checkLegality()审核订单，而后封装为生产任务，使用computePriority()确定任务优先级，最后插入到生产队列中。

(4) SMA (Sub-Manager Agent) 为子管理Agent，继承自MA，接收到GMA的生产任务时，广播至其下的各个车间Agent，当接收到监控Agent的设备故障信息时，使用recycleTasks()回收故障设备上的工件任务。

(5) RA (Resource Agent) 为资源Agent，继承自Agent。FactoryID定义Agent所属的工厂，ResourceAgent由有两个子类，分别是车间Agent (JA) 和设备Agent (MA)。JA接收到生产任务时，使用evaluate()判断其下的设备集是否满足任务对制造资源的需求，使用requestCompute()通知算法Agent进行调度方案的计算；MA记录对应设备的名称、加工能力、加工序列、工作状态等。当GMA确定最终的调度方案后，MA需要在加工序列中添加对应的工序任务。

(6) ALA (Algorithm Agent) 为算法Agent，继承自Agent，内部封装了与调度算法相关的参数，使用compute()提供调度方案的计算服务。

(7) MTA (Monitor Agent) 为监控Agent，继承自Agent，使用analyze()分析采集数据，判断是否存在设备故障；当设备发生故障时，使用notify()通知SMA。

(8) PA (Process Agent) 为工艺Agent，继承自Agent，processList记录了系统工件的工艺信息，使用checkOrder()为GMA审核订单的合法性，checkNewProcess()为工艺部新增工件的合法性进行审核。

5.3.4 Agent的通信模型设计

多Agent系统作为分布式的系统，对复杂大型问题的求解依赖其内Agent的相互配合，因此对Agent间通信效率有很高的要求，尤其当频繁发生订单更改、设备故障等动态变化时，将出现大量的消息请求，为了能及时处理反馈，需要保证系统的通信质量。本系统使用基于Socket的通信机制完成Agent间的信息交互，其中选择底层通信协议为TCP/IP。Agent的通信模型由线程池和同步阻塞队列实现。

Socket通信属于消息传递的通信方式。Agent通信双方需事先确定请求消息的格式，发送方在进行消息的封装时需严格遵循规则约束，接收方接收到消息后同样按照约定的方式对消息的字段逐个进行解析，从中分析请求方的意图并采用对应的处理方案。本系统采用基于TCP/IP的Socket通信，底层的TCP协议是基于字节流的安全可靠通信协议，不保留传输数据的边界，为了能够让通信双方接收到字节流时能够正确识别数据，本系统使用Java的序列化和反序列化技术实现对象和字节流之间的双向转换。顾名思义，序列化能够把Java对象转换为字节序列，之后可以存储在数据库或在网络上进行传输；反序列化能够把序列化后的字节流还原为原本的对象。通过该方法，接收方Agent获得反序列化的对象后，通过获取对象的属性信息便可解析出发送方的请求意图以及请求数据。

为保证系统的通信效率，Agent需要能够并发处理来自其他Agent的请求，因此通常采用多线程技术，即每当接收到一个消

息请求，Agent就新建线程建立Socket连接。该方式能够充分利用多核CPU的性能，并发处理大量请求，提高Agent的数据处理能力，从而保证了系统的通信效率。该方式同时也存在一个明显的缺点，当过多的请求蜂拥而至时，Agent需要在极短时间内创建大量的线程。虽然线程的粒度较小，创建和销毁的开销远比线程小，但过多的线程十分占用CPU资源，导致Agent无法响应后续的请求，因此本系统使用线程池缓存线程，实现线程的复用，具体过程是：设定线程池和消息队列，线程池初始线程数目 n_0 ，最大线程数目 n_{max} ，消息队列的容量为 c 。如图5-5所示。当Agent接收到一个请求时，从线程池中分配一个空闲线程进行处理；若无空闲线程，检查当前线程池内线程数目 c ，若 $c < c_{max}$ ，新建线程，否则请求进入消息队列，直至线程池中出现空闲线程。若Agent消息队列为空且当前线程数目大于 c_0 时，销毁多余的线程。若有新的请求到来且Agent消息队列已满时，Agent将丢弃该请求并向发送方发送消息，请求发送方延后重新发送。

图5-5 线程池和消息队列

5.4 动态调度系统实现与界面设计

5.4.1 系统登录模块

登录模块用于向用户展示登录界面，如图5-6所示。用户需输入用户名和密码，经系统验证身份成功后方可进入系统，登录成功后进入系统的主页面，如图4-2所示。

图5-6 系统登录界面

图5-7 系统主界面

5.4.2 系统信息管理模块

系统信息管理模块用于管理用户信息以及企业的物料信息。下图5-8是系统的用户信息展示界面。用户的权限分为三个等级，从高到低排列分别是系统管理员、子系统管理员和工艺管理员。系统管理员可查看各个工厂的资源配置，并下达面向各工厂的生产计划。子系统管理员对应于一间工厂的管理员，只能查看并制定所属工厂的生产计划。工艺管理员负责管理系统的产品工艺，能够对工艺信息进行修改。图5-8中，系统管理员可以修改其他用户的个人信息，包括姓名、年龄、性别以及操作权限。图5-9呈现了企业的物料信息，包括物料名称以及对应的数量。

图5-8 用户信息管理界面

图5-9 物料管理界面

5.4.3 资源管理模块

工厂管理模块可以为系统管理员提供资源配置的功能，点击主界面的工厂资源配置选项，可进入如图4-5所示界面，该界面呈现了系统的工厂分布以及各工厂的车间结构，由图5-10可知，系统包含两间工厂，分别为工厂一和工厂二。其中工厂一有两个车间，分别为车间1和车间2。

图5-10 工厂资源配置页面

如图5-11，点击管理按钮后进入管理模式，可进行车间的添加或者删除操作。在非管理模式下点击车间，可查看车间的设备配置，如图5-12表明车间1中包含三个设备 m_1 、 m_2 和 m_3 ，管理员同样能对设备进行增删查改等操作。

图5-11 资源管理界面

图5-12 车间1的设备配置

5.4.4 工艺管理模块

在主界面点击工艺管理选项后可进入工艺管理模块，该模块为工艺管理员提供对产品工艺信息的管理接口，如图5-13呈现了系统的工件种类，可见系统目前可生产加工的工件共有六种。

图5-13 工艺管理界面

如图5-14是点击了“管理工件”选项后进入的管理模式，此时可对工件信息进行增删查改。在非管理模式下，选择工件 p_1 后将呈现工件的加工工艺信息，如图5-15所示，从中可看出工件 p_1 包含的工序以及工序的机器选择和加工时间。

图5-14 工艺信息管理界面

图5-15 工件 p_1 的加工工艺信息

5.4.5 生产调度模块

在主界面选择生产调度选项后可进入生产调度模块，当系统接收到客户订单时，该模块用于计算并呈现分配给各工厂的生产计划，同时为系统管理员提供生产计划的制定，如图5-16为生产调度的界面，在该界面上管理员可以设置参与调度的工厂、工件任务以及与调度算法相关的参数。

图5-16 生产调度主界面

选择生产调度主界面的参数配置按钮，可以对本文的调度算法的参数进行设置，下图5-17呈现了关于蚁群算法的参数，同时还可设置设备故障触发概率用于测试系统对生产过程中的由于设备故障引起的异常调度。

图5-17 算法参数设置

当选择生产调度主界面的任务配置按钮时，可以选择参与调度的工厂以及工件种类和数量，如图5-18所示，管理员选择了工厂一接收生产计划，生产计划包括工件 p_1 、 p_2 和 p_3 三类工件，每类工件数量均为一个。完成参数设置和任务配置后，点击启动按钮，系统最终得到的调度方案如图

5-19所示，该结果呈现各工件的工序任务在工厂一的三个设备中的分配情况以及开始加工时间。

图5-18 任务配置

图5-19 调度结果

5.5 本章小结

本章主要介绍了基于多Agent的柔性生产动态调度系统的实现，先是介绍了企业传统的生产管理流程，针对其中的不足提出了调度系统的需求，并根据系统的Agent划分设计了系统的体系结构，同时使用Java的面向对象技术封装了Agent的属性和功能，使用线程池和消息队列构建了Agent的通信模型，最后介绍了调度系统的界面设计。

6 仿真实验与结果分析

6.1 实验环境

本文的实验环境为：Intel(R) Celeron(R) G530 CPU @ 2.40GHz，RAM 2.0GB，操作系统 Win 10 32位, JDK版本 jdk1.8.0_144

6.2 实验一：基于改进蚁群算法的调度策略验证

该实验用于验证本文的基于改进蚁群算法的调度策略在FJSP中的可行性，优化目标为最小化最大完工时间。文献[46]提出了一种基于多种群蚁群算法用于FJSP求解中，该文献对从某航空发动机公司提取的6*10柔性生产调度实例对算法进行了验证，该实例包含6种工件，每种工件各包含6道工序，调度的设备集共有10个设备。各工件的工序组成、工序的设备选择以及对应的加工时间如下表所示：

工件工序加工设备集

m1 m2 m3 m4 m5 m6 m7 M8 m9 m10

p1 o1,1 -- 4 - 10 -- 12 - 15

o1,2 - 6 - 8 ----- 11

o1,3 ----- 9 -- 4 -

o1,4 --- 7 -----

o1,5 ----- 6 ----

o1,6 ----- 13 -- 14

p2 o2,1 10 ---- 15 ----

o2,2 -- 8 ---- 6 --

o2,3 --- 5 -- 7 ---

o2,4 --- 4 - 8 -- 3 -

o2,5 - 10 ---- 12 ---

o2,6 --- 4 -- 7 ---

p3 o3,1 -- 9 ----- 14

o3,2 ---- 8 ----

o3,3 - 9 ----- 6 -

o3,4 -- 4 -- 6 ---

o3,5 -- 7 ----- 9

o3,6 ----- 6 -- 9

p4 o4,1 --- 5 -- 4 --

o4,2 -- 6 --- 5 --

o4,3 5 -- 8 -- 11 -- 9

o4,4 - 3 -- 11 12 - 5 - 10

o4,5 ----- 8 -- 8

o4,6 1 -- 5 -- 7 -- 8

p5 o5,1 6 ---- 3 ----

o5,2 3 ---- 6 --- 8

o5,3 -- 9 ----- 11

o5,4 - 7 -- 1 ---- 9

o5,5 5 -- 14 -----

o5,6 ---- 5 -- 8 --

p6 o6,1 ---- 10 8 ----

o6,2 --- 3 ----- 3

o6,3 ---- 10 -----

o6,4 ----- 9 --- 10

o6,5 --- 4 --- 7 --

o6,6 -- 13 -- 15 -- 14

本文的基于改进蚁群算法调度策略的参数设置如下：

参数名称参数值

工件数量 6

工序初始设备选择数 75

蚂蚁数量m 75

迭代次数N 100

忽略信息素影响迭代次数 10

信息素权重系数 2

能见度权重系数 3

初始信息素浓度 1

每轮循环释放信息素总量 20

节点间信息素浓度最小值 0.1

节点间信息素浓度最小值 8

信息素挥发因子 0.1

根据该调度实例，本文的基于蚁群算法的调度策略在最小化最大完成时间的单优化目标下执行，在本文的调度系统中的工厂一中设置十个设备对六个工件进行调度方案的计算，所得最优解52，而献[27]得到的最优解为53，可见本文算法具有较强的求解性能。

指 标		
疑似剽窃文字表述		
1. 环境及工具		
操作系统：Microsoft Windows 10 Professional		
开发环境：JDK1.		
5. 71068966119975096_梁乐乐_基于多Agent的柔性生产动态调度系统研究_第5部分		总字数：12301
相似文献列表 文字复制比：25.8%(3174) 疑似剽窃观点：(0)		
1	基于遗传算法的柔性资源调度优化方法研究 赵诗奎(导师：顾新建;方水良) - 《浙江大学博士论文》 - 2013-10-01	9.8% (1206) 是否引证：否
2	柔性机器人制造单元调度算法研究及应用 陈业(导师：杨煜俊) - 《广东工业大学硕士论文》 - 2017-05-01	7.3% (895) 是否引证：否
3	考虑能耗优化的柔性作业车间动态调度方法研究 何院生(导师：闫纪红) - 《哈尔滨工业大学硕士论文》 - 2016-07-01	5.6% (690) 是否引证：否
4	船舶企业虚拟制造单元瓶颈缓冲区问题研究 郭婷芳(导师：吕洁) - 《江苏科技大学硕士论文》 - 2016-06-06	5.1% (626) 是否引证：否
5	12040203004_黄歆雨_基于混合遗传算法的柔性作业车间动态调度问题理论与应用研究 黄歆雨 - 《大学生论文联合比对库》 - 2016-04-12	4.9% (598) 是否引证：否
6	蚁群算法及其在分布式柔性作业车间调度中的应用 何怡(导师：凌海峰) - 《合肥工业大学硕士论文》 - 2016-04-01	4.6% (569) 是否引证：否
7	制造系统柔性资源调度问题及其优化算法研究 鲁陈璐(导师：方水良) - 《浙江大学硕士论文》 - 2016-03-01	4.6% (564) 是否引证：否
8	多因素多工艺集成环境下生产调度问题研究 巴黎(导师：李言;杨明顺) - 《西安理工大学博士论文》 - 2016-06-30	4.1% (507) 是否引证：否
9	机械加工制造过程能耗优化方法研究 张中伟(导师：唐任仲;陶俐言) - 《浙江大学博士论文》 - 2016-01-01	3.6% (448) 是否引证：否
10	面向模糊车间调度问题的智能算法研究 郑友莲(导师：李元香) - 《武汉大学博士论文》 - 2011-11-01	3.4% (422) 是否引证：否
11	基于改进量子粒子群算法的柔性作业车间调度研究	3.1% (376)

	周恺(导师：纪志成) - 《江南大学硕士论文》 - 2016-06-01	是否引证：否
12	基于重调度需度驱动机制的柔性作业车间多目标动态调度研究 陈鸿海(导师：赵韩) - 《合肥工业大学博士论文》 - 2015-08-01	3.0% (368) 是否引证：否
13	柔性车间调度问题中的智能优化算法研究 张庭(导师：徐华) - 《江南大学硕士论文》 - 2016-06-01	2.8% (340) 是否引证：否
14	资源不确定条件下项目调度多目标优化研究 何立华(导师：张连营) - 《天津大学博士论文》 - 2013-05-01	2.8% (339) 是否引证：否
15	孙国翔_数据驱动的机械制造车间调度方法研究 孙国翔 - 《大学生论文联合比对库》 - 2017-05-16	2.0% (249) 是否引证：否
16	基于改进遗传算法的多目标柔性作业车间调度问题研究 何浩(导师：赵小强;梁宗辉) - 《兰州理工大学硕士论文》 - 2016-04-18	1.9% (234) 是否引证：否
17	不确定环境下钢结构多项目生产优化调度研究 袁玲玲(导师：段鹰) - 《重庆大学硕士论文》 - 2016-05-01	1.8% (220) 是否引证：否
18	基于组合拍卖机制的分布式柔性车间优化调度的研究 李冬(导师：王艳红) - 《沈阳工业大学硕士论文》 - 2015-02-28	1.7% (215) 是否引证：否
19	离散制造企业资源弹性约束下作业车间调度问题研究 林仁(导师：周国华) - 《西南交通大学博士论文》 - 2016-11-03	1.7% (212) 是否引证：否
20	液压缸基础零件制造车间MES的研究与设计 王博(导师：陆宝春) - 《南京理工大学硕士论文》 - 2016-12-01	1.7% (203) 是否引证：否
21	基于关键链的多项目优化调度问题研究 苏慧(导师：彭武良) - 《沈阳理工大学硕士论文》 - 2013-12-20	1.6% (191) 是否引证：否
22	基于改进的混合免疫算法的车间调度问题研究 郝倩(导师：黄明) - 《大连交通大学硕士论文》 - 2014-06-30	1.4% (176) 是否引证：否
23	柔性加工系统PLC程序设计研究 曹著明(导师：殷际英;顾春光) - 《北方工业大学硕士论文》 - 2011-02-10	1.4% (169) 是否引证：否
24	基于虚拟现实技术的城市轨道交通车站仿真平台设计与实现 陈昂扬 - 《大学生论文联合比对库》 - 2016-05-23	1.1% (138) 是否引证：否
25	推拿治疗小儿便秘的古代文献与实证研究 李柱烨 (LEE JUYEON) (导师：顾一煌) - 《南京中医药大学博士论文》 - 2017-05-21	0.3% (39) 是否引证：否
26	时间不确定的job shop问题的一种改进遗传算法 刘胜辉;张晓欢; - 《自动化技术与应用》 - 2011-04-25	0.3% (36) 是否引证：否

原文内容

图6-1为本文调度系统的计算的最优解，对应的甘特图如图6-2所示。文献[46]的最优解对应甘特图如图6-3所示。

图6-1 100次迭代所得的最优解

图6-2 本文调度系统所求最优解对应的设备-时间甘特图

图6-3 文献[46]的最优解对应的设备-时间甘特图

本文使用基本蚁群算法（ACO）在相同的参数条件下进行20次试验，每次试验迭代次数为100，所得结果求平均值，与本文改进的调度算法比较结果如下表6-1所示。20次试验中平均数值的收敛曲线如下图6-4所示。

Table 6-1

算法最优解前100次迭代的平均解最差解

ACO 53 53.45 55

本文的算法 52 52.789 54

图6-4 20次实验的平均收敛曲线

由以上数据和图表可知，本文的算法在初始收敛速度上稍慢于ACO，但具有更好的求解性能，能收敛得到更优的最优解。由此可见，本文对于基本蚁群算法的改进是有效的，能够避免算法早熟的同时，能保证较好的求解质量。

为了进一步验证本文调度算法的求解质量，需要与其他文献进行对比。本文应用了文献[42]提供的4*6柔性调度实例，如下表所示，该实例包含4个工件，每个工件有三道工序，安排在4个设备上调度。

工件工序机器选择

m1 m2 m3 m4 m5 m6

p1 o1,1 2 3 4 ---
o1,2 - 3 - 2 4 -
o1,3 1 4 5 ---
p2 o2,1 3 - 5 - 2 -
o2,2 4 3 -- 6 -
o2,3 -- 4 - 7 11
p3 o3,1 5 6 ----
o3,2 - 4 - 3 5 -
o3,3 -- 13 - 9 12
p4 o4,1 9 - 7 9 --
o4,2 - 6 - 4 - 5
o4,3 1 - 3 -- 3

该实例在本文的调度系统中设置蚂蚁数量为32，迭代次数为100，其他参数不变，系统中工厂一的设备数量为6，对上表4个工件的运行结果的最优解为17，如下图6-5所示，对应甘特图如图6=6所示，20次实验中的平均收敛曲线如图6=7所示。文献[42]的最优解同样为17，甘特图如图所示

图6-5 本文调度系统的运行结果

图6-6 本文算法所得最优解对应的甘特图

图6-7 本文算法在20次实验中的平均收敛曲线

图6-8 文献[42]最优解对应的甘特图

文献[42]基于该实例与其他文献的调度算法进行了对比，在20次实验，每次实验迭代次数为100的情况下，与本文改进的蚁群算法的调度结果对比如下：

文献最优解平均收敛代数最差解平均计算时间/S

文献[47] 17 26.7 19 24.4

文献[48] 17 67.3 18 -

文献[49] 18 ---

文献[42] 17 10.1 17 0.23

本文算法 17 2.3 17 0.285

根据上表数据可知，本文算法能够在在较短时间内求得最优解，实验中如果把蚂蚁数量设置为20只，平均计算时间降低为0.198秒，而最优解同样为17。通过与以上文献的对比，可以看出本文的算法在柔性生产调度问题中具有较强的求解能力。

6.3 实验二：验证基于多Agent的跨区域生产任务分解策略

本文在车间调度过程中把生产任务的分解和分配进行一定程度的分离，其中通过跨区域生产任务分解流程，在保持工件级任务粒度的前提下，得到若干满足任务需求的设备集，每个设备集中的设备配置均有能力完成对完整任务加工的能力。该过程根据任务粒度和制造资源的地区分布特性，减少了调度设备集中设备的规模，取而代之的是多个规模较小的设备集。而后通过本文改进的蚁群算法基于每个设备集进行求解，最终从中筛选完工时间最短的调度方案。因此该实验用于跨区域分解策略配合本文改进的蚁群算法能否完成生产任务在各地区工厂中的最优分配，从而合理利用企业的分布式制造资源。

文献[50]根据云制造环境下资源的分布式特征，建立资源虚拟化框架，并根据FJSP的调度特点建立工序和设备选择约束，以加工时间和最小为目标，使用启发式-蚁群算法进行资源的配置。该文献基于两家企业的制造资源配置，提供了一个柔性作业车间调度实例，该实例包含3个工件，3个设备，其中每个工件包含三道工序，工序的设备选择及对应的加工时间如下表

工件 工序设备选择和对应的加工时间

m1 m2 m3
p1 o1,1 2 3 2.5
o1,2 1 1 0.5
o1,3 3 1.5 0.5
p2 o2,1 1 1.5 2
o2,2 1 2 0.5
o2,3 3 1.5 0.5
p3 o3,1 2 1 1
o3,2 1 0.5 2
o3,3 0.5 1 1

该实例中，两家企业的资源配置相同，均配备完整的三个设备，其中企业1的设备m1和设备m3分别在[2,2.5]和[2,3]的时间段被占用，如图6-9所示。企业2的设备均为空闲，加工序列为空。

图6-9 企业1设备原始加工序列甘特图

在本实验中，基于以上调度实例把两家企业对应为位于不同地区的两间加工工厂，对应本文调度系统中的工厂一和工厂二，两间工厂各包含三个设备m1,m2和m3，其中对工厂一的设备设置如图6-5的初始工序的占用。实验初始化时，把三个工件任务封装为一个完整的产品任务，由全局管理Agent广播至两个子管理Agent。由于两间工厂设备配置相同且均有三个设备，因此在跨区域分解过程中两个资源Agent组均会得到设备集，而后把设备集中Agent模拟的设备加工序列以及生产任务发送到算法Agent，算法Agent计算调度方案后返回到对应的资源Agent组。工厂一和工厂二最终所得的最优解分别如图6-10和图6-11所示，对应最优解甘特图如图6-12和6-13所示。其中工厂一的最优解为4，工厂二的最优解为3.5；由于工厂一的设备存在工序占用，调度方案在完工时间上大于工厂二，因此最终全局管理Agent从中选择工厂二执行生产计划。

图6-10 工厂一的调度结果

图6-11 工厂二的调度结果

图6-12 工厂一的最优解甘特图

图6-13 工厂二的最优解甘特图

由此可见，本文的基于多Agent的跨区域生产任务分解策略能够根据生产任务对制造资源的要求以及企业跨地区的资源分布式特点得到对应的设备集合，配合本文改进的蚁群调度算法，能够提高企业分布式资源的利用率。

6.3 实验三：验证设备故障条件下的多Agent异常调度策略

本文的基于多Agent的动态调度系统是通过各Agent之间的协作来完成共同的任务，在异常状况发生，Agent接收异常信息后，通过Agent间高效的通讯，信息在各类型Agent中进行传递以及采取对应的行为，从而达到快速及时响应异常状况的目的。该实验通过比较在异常情况发生时，根据系统的反应时间、重调度所需的时间以及跟正常调度方案的差异来验证系统的重调度协商策略的可行性。

本实验基于文献[51]提供的6*5车间调度实例，该实例的设备集包含5个设备，工序的设备选择和对应的加工时间如下表所示。文献[51]的作者在基于多Agent的动态制造系统中，定义了资源Agent、工件Agent和管理Agent三类Agent，当故障发生时，通过基于合同网的Agent间协商机制完成故障资源Agent上工序任务的转移。

工件工序设备选择以及对应的加工时间

m1 m2 m3 m4 m5

p1 o1,1 3 4 ---

o1,2 - 8 - 9 -

o1,3 - - 9 - 7

o1,4 - 5 - 4 -

o1,5 - 3 - - 3

p2 o2,1 6 - - 8 -

o2,2 - 8 - 6 -

o2,3 5 - 4 - -

o2,4 - - 2 - 3

o2,5 - 3 - - 4

p3 o3,1 - - 4 - 6

o3,2 5 - 7 - -

o3,3 - 7 9 - -

o3,4 5 5 - - -

o3,5 - - - 6 6

p4 o4,1 - 9 - - 7

o4,2 - 3 5 - -

o4,3 - 4 - 6 -

o4,4 - - - 3 -

o4,5 - 3 - - 7

p5 o5,1 6 4 - - -

o5,2 - - 7 5 -

o5,3 - - - 7 9

o5,4 8 - 7 - -

o5,5 - - 5 5 -

p6 o6,1 - 3 7 - -

o6,2 9 - - 7 -

o6,3 - - 6 - 9

o6,4 - - 6 7 -

o6,5 - 4 - 4 -

根据该实例，该实验在本文的调度系统工厂一的车间1中设置五个设备，分别为m1,m2,m3,m4和m5，用于对该实例的六个工件进行调度方案的计算，调度结果如图6-14，对应的甘特图如图6-15所示。

图6-14 系统调度结果

图6-15 系统调度结果对应的甘特图

该实验中，设备故障由监控Agent触发，而后把故障信息发送到子管理Agent，子管理Agent通知其下的车间Agent进行故障设备Agent的移除，而后进行工序任务的回收工作，最终重新封装为新的工件任务在资源Agent组间进行分解分配，得到重调度方案。本文把故障发生到新的工件任务封装完成定义为系统的反应时间，反应时间的大小体现了系统对设备故障是否灵敏；新工件任务封装完成到重调度方案生成定义为异常调度时间，体现系统调度算法的效率；异常调度方案与正常调度方案的完工时间差值用于衡量系统重调度方案的有效性。

根据3.1中对FJSP中车间调度的假设，工序加工过程中不可中断，若由于设备故障原因导致设备停工，工序加工不完整，则对应的工件报废，原有的工件任务作为加急工件任务参与重调度，因此该实验在设备故障触发上进行如下设置：

(1) 仅在设备空闲时间触发设备故障，若工序 o_{ij} ， $o_{i'j'}$ 为某一设备 k 加工工序序列 Tsk 中的相邻工序，当时间 t 满足以下条件时，可触发设备故障： $e_{ij}, k \leq t \leq s_{i'j'}, j', k$

(2) 任意时刻只有一台设备触发设备故障。

(3) 当设备加工序列为空时，即完成最后一道工序的加工后，不触发设备故障。

本实验基于甘特图6-11对应的调度方案进行，以50%的概率随机触发空闲设备故障，在100次故障中异常调度的反应时间、重调度时间，时间差异的平均值如下表所示。由表中数据可知，故障发生时间越早，所需的重调度时间越长，这是由于后续未开始加工的工序较多，时间越早封装所得的新工件越接近原始工件任务，调度算法所需的计算量越大。其次，反应时间与故障发生时间相关性小，这是由于反应时间主要由Agent之间信息的传递决定，依赖Agent之间的通信质量和效率。

时间段故障次数平均反应时间/s 平均异常调度时间/s 平均时间差

0-5 11 0.638 2.764 13.32

5-10 15 0.569 2.863 12.80

10-15 12 0.471 2.368 10.51

15-20 5 0.587 2.472 9.36

20-25 18 0.483 1.806 7.43

25-30 16 0.431 1.342 8.26

30-35 11 0.472 0.879 7.71

35-40 12 0.433 0.203 5.25

40-45 0 - - -

根据表中数据可知，该实验平均反应时间为0.511s,平均异常调度时间为1.837s,平均时间差为9.25。当故障时间发生越早，异常调度方案与正常调度时间差越大，这是由于回收的工序多，且故障设备无法用于任务分配，需由其他可替换设备用于工序任务的转移，因此异常调度结果对应的时间较长，所需要的计算时间也更长；若故障发生时间接近调度尾声，由于回收工序少，计算时间大幅降低，时间差对应也降低；从中还可看出异常调度过程中反应时间与故障发生时间无相关性，这是由于反应时间主要是由Agent间通信所占用的时间决定，工序回收以及新工件任务的封装所需时间基本是固定的，影响小。根据以上分析，最终可得到如下结论：系统在故障设备发生时反应时间和异常调度时间短，异常调度方案与正常调度时间差保持在较小的范围内，说明系统对异常状况反应灵敏，异常执行时间短，对原生产计划影响小，具有实用价值，从而验证了在故障条件下的基于多Agent的异常调度策略的有效性。

6.4 本章小结

本章主要设计了三个实验用于验证基于多Agent的柔性生产调度系统的有效性，其中实验一验证本文改进的蚁群算法在FJSP中的可行性，实验二验证在各资源Agent组以及算法Agent的相互配合下，能够实现生产任务的跨区域分配，实验三验证在设备故障条件下，各Agent之间通过信息交互，能够快速完成重调度流程，调整原调度方案。最后根据各个实验的实验结果和数据，证实了系统在分布式环境下能够合理地分配资源，快速响应异常状况，具有实用价值。

7 总结与展望

7.1 总结

制造业是国民经济的重要组成部分，为我国的经济的腾飞作出了巨大的贡献。如今市场需求复杂多变，制造业的生产模式已经逐渐向小批量多品种甚至是单件定制化生产转变，所生产加工的产品通常具备柔性，这需要企业具备高效高质量的车间调度算法。同时还需要注意到企业所处的生产环境通常充满诸多不确定性，包括客户订单的频繁更改、撤销以及设备故障等等，这些具有不可预料性的异常因素严重干扰企业正常的生产管理流程，为了保证企业生产的平稳高效进行，仅仅通过研究高效的调度算法是不足够的，还需从系统的角度出发，集成高效的调度算法，设计灵活的系统架构，提高系统对异常因素的响应速度，衔接企业生产管理的各个流程，实现企业生产管理的自动化与信息化。

本文在研究柔性生产调度系统中，考虑到企业所处的动态变化环境以及FJSP的复杂性，应用了多Agent理论构建柔性生产动态调度系统，所做工作如下：

(1) 对FJSP的特点、研究现状以及现代制造企业所处的生产环境进行了深入的分析，对企业生产管理过程中的功能需求进行了划分，同时根据多Agent理论在复杂大型问题求解的优越性，构建了基于多Agent的柔性生产动态调度系统，该系统由管理Agent、资源Agent、算法Agent、监控Agent以及工艺Agent组成，五类Agent根据企业资源的分布式特点构成了混合式的结构，在保证生产调度高效的同时，提高系统对异常信息的响应速度。

(2) 设计了企业分布式制造环境下Agent的协作流程，通过定义了生产任务跨区域分解策略来帮助企业对异地性资源和技术整合；定义了基于多Agent的异常调度策略来保证系统对异常因素的快速响应，维持企业生产管理的稳定性。

(3) 把蚁群算法作为系统的调度算法，同时根据FJSP的特点以及基本蚁群算法收敛慢易早熟的特点进行了改进，主要通过更改基本蚁群算法的信息素更新规则以及蚂蚁的均匀分布来提高蚁群算法的适用性。

(4) 开发了基于多Agent的柔性生产动态调度系统。根据企业生产管理的需求构建了系统的体系架构，使用Java的面向对象技术和多线程技术进行Agent模型的实现，最终提出系统各模块的设计与实现。该系统能够实现对用户信息、物料信息、工艺信息等的管理，同时提供生产调度模块进行柔性生产调度的仿真验证。

7.2 工作展望

在基于多Agent的柔性生产动态调度系统的研究中，系统仍然存在许多有待完善的地方：

(1) 由于条件所限，本文的调度系统是在单机环境下运行的，通过面向对象和多线程技术对Agent进行封装来模拟分布式环境，今后需加深对多Agent理论的研究，以互联网为媒介，实现真正的分布式多Agent系统。

(2) 本文应用的是基于最小化最大完工时间的单目标优化的蚁群算法作为系统的调度算法，实际生产环境中还需考虑成本、设备负载等各种因素，因此今后可结合多种调度算法对多目标优化的柔性生产调度问题开展研究，提高动态调度系统的实用性。

(3) 需要把本文的调度系统真正用于实际的生产管理中，方可从企业实际生产面临的难题中获取需求，提高系统的实用性，例如工厂车间中的设备故障所造成的影响应当是多样的，需要联合实际采取对应的措施。

文献引用

- [1] 徐梦周, 贺俊. 第三次工业革命的特征及影响[J]. 政策瞭望, 2012(10):46-47.
- [2] 吕铁, 邓洲. 第三次工业革命的技术经济特征[J]. 中国党政干部论坛, 2013(10):6-10.
- [3] 纪建强等. 第三次工业革命:特征、影响及应对战略[J]. 管理现代化, 2015, 35(1):127-129.
- [4] 黄群慧, 贺俊. 中国制造业的核心能力、功能定位与发展战略——兼评《中国制造2025》[J]. 中国工业经济, 2015(06):5-17.
- [5] 左乐. 不确定环境下柔性作业车间的多目标动态调度研究[D]. 北京交通大学, 2015.
- [6] 刘想德. 作业车间实时调度若干关键问题研究[D]. 重庆大学, 2013.
- [7] 陶永, 王田苗, 李秋实, 赵罡. 基于“互联网+”的制造业全生命周期设计、制造、服务一体化[J]. 科技导报, 2016, 34(04):45-49.
- [7] 苏凯凯. 云制造环境下的制造资源优化配置方法研究[D]. 北京交通大学, 2017.
- [8] 周恺, 纪志成. 关于柔性作业车间调度问题的仿真研究[J]. 计算机仿真, 2016, 33(03):282-287+375.
- [9] 汪俊亮, 张洁, 秦威, 银莉, 陈定方. 加工时间不确定的柔性作业车间鲁棒调度方法[J]. 中国机械工程, 2015, 26(05):627-632.
- [10] Zhang Q, Manier H, Manier M A. A genetic algorithm with tabu search procedure for flexible job shop scheduling with transportation constraints and bounded processing times[J]. Computers & Operations Research, 2012, 39(7):1713-1723.
- [11] 王万良, 范丽霞, 徐新黎, 赵燕伟, 张静. 多目标差分进化算法求解柔性作业车间批量调度问题[J]. 计算机集成制造系统, 2013, 19(10):2481-2492.
- [12] 刘韵, 胡毅, 房超, 罗企. 解决柔性车间作业调度问题的侦查包围搜索算法[J]. 组合机床与自动化加工技术, 2015(11):124-128.
- [13] 王雷, 蔡劲草, 唐敦兵, 李明. 基于改进遗传算法的柔性作业车间调度[J]. 南京航空航天大学学报, 2017, 49(06):779-785.
- [14] Moslehi G, Mahnam M. A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search[J]. International Journal of Production Economics, 2011, 129(1):14-22.
- [15] 赵博选, 高建民, 陈琨. 求解多目标柔性作业车间调度问题的两阶段混合Pareto蚁群算法[J]. 西安交通大学学报, 2016, 50(07):145-151.
- [16] Rossi A, Dini G. Flexible job-shop scheduling with routing flexibility and separable setup times using ant colony optimisation method[J]. Robotics and Computer-Integrated Manufacturing, 2007, 23(5):503-516.
- [17] Huang R H, Yang C L, Cheng W C. Flexible job shop scheduling with due window—a two-phormone ant colony approach[J]. International Journal of Production Economics, 2013, 141(2):685-697.
- [18] Chiang T C, Lin H J. A simple and effective evolutionary algorithm for multiobjective flexible job shop scheduling[J]. International Journal of Production Economics, 2013, 141(1):87-98.
- [19] Pereira I, Madureira A. Self-Optimizing A Multi-Agent Scheduling System: A Racing Based Approach[J]. 2016.
- [20] Pereira I, Madureira A, Oliveira P M. Case-based Reasoning for Meta-heuristics Self-Parameterization in a Multi-Agent Scheduling System[C]// International Symposium on Computational Intelligence for Engineering Systems. 2013.

- [21] Gozzi A, Paolucci M, Boccalatte A. A multi-agent approach to support dynamic scheduling decisions[J]. 2002:983-988.
- [22] [Lim M K, Zhang Z. A multi-agent based manufacturing control strategy for responsive manufacturing\[J\]. Journal of Materials Processing Technology, 2003, 139\(1-3\):379-384.](#)
- [23] [Adhau S, Mittal M L, Mittal A. A multi-agent system for distributed multi-project scheduling: An auction-based negotiation approach\[J\]. Engineering Applications of Artificial Intelligence, 2012, 25\(8\):1738-1751.](#)
- [24] 杨陇苗,王玉洁.基于多Agent的MES系统调度[J].信息系统工程,2017(02):120-121+124.
- [25] 王芋博,张文新,王柏琳,吴子轩.基于Agent的混合流水车间动态调度系统[J].计算机应用,2017,37(10):2991-2998.
- [26] 宋娟.多Agent分布式车间动态调度仿真系统研究[J].制造业自动化,2011,33(24):57-58+74.
- [27] 任海英,邹艳蕊.基于多Agent的柔性作业车间预先/重调度系统[J].武汉理工大学学报(信息与管理工程版),2012,34(01):69-73.
- [28] [Wooldridge M, Jennings N R. Agent theories, architectures, and languages: A survey\[M\]// Intelligent Agents. Springer Berlin Heidelberg, 1994:408-431.](#)
- [29] Jennings N R, Wooldridge M J. Agent Technology: Foundations, Applications and Markets.[J]. Renewable Energy Technologies, 1998, 2(1):43-51.
- [30] 刘勇. 多Agent系统理论和应用研究[D]. 重庆大学, 2003.
- [31] 赵龙文, 侯义斌. 多Agent系统及其组织结构[J]. 计算机应用研究, 2000, 17(7):12-14.
- [32] 李成海, 黄必清. [基于属性描述匹配的云制造服务资源搜索方法\[J\]. 计算机集成制造系统, 2014, 20\(6\):1499-1507.](#)
- [33] 张执南, 李响, 陈斌,等. 企业分布式资源环境的概念设计[J]. 机械设计与研究, 2013, 29(5):1-3.
- [34] 张利平. [作业车间预反应式动态调度理论与方法研究\[D\]. 华中科技大学, 2013.](#)
- [35] 杨娜. 多agent生产调度系统的设计与实现[J]. 信息技术与信息化, 2014(9):191-192.
- [36] [Ponsich A, Coello C A C. A hybrid Differential Evolution-Tabu Search algorithm for the solution of Job-Shop Scheduling Problems\[M\]. Elsevier Science Publishers B. V. 2013.](#)
- [37] 张国辉, 王永成, 张海军. 多阶段人机协同求解动态柔性作业车间调度问题[J]. 控制与决策, 2016, 31(1):169-172.
- [38] 刘轩, 尚璠, 白翱. 设备故障驱动的作业车间生产任务重调度方法研究[J]. 制造业自动化, 2016, 38(12):26-30+60.
- [39] [Dorigo M, Gambardella L M. Ant colony system: a cooperative learning approach to the traveling salesman problem\[J\]. IEEE Trans on Ec, 1997, 1\(1\):53-66.](#)
- [40] [田松龄, 陈东祥, 王太勇, 刘晓敏. 一种异步蚁群算法求解柔性作业车间调度问题\[J\]. 天津大学学报\(自然科学与工程技术版\), 2016, 49\(09\):920-928.](#)
- [41] 王硕. 基于改进蚁群算法的作业车间调度研究[D]. 华东理工大学, 2013.
- [42] 刘志虎. 基于改进蚁群算法的柔性车间调度研究[D]. 安徽工程大学, 2016.
- [43] 袁豪. 旅行商问题的研究与应用[D]. 南京邮电大学, 2017.
- [44] [黄厦. 基于改进蚁群算法的柔性作业车间调度问题研究\[D\]. 昆明理工大学, 2015.](#)
- [45] 王媛. 多agent生产调度系统的设计与实现[D]. 大连理工大学, 2005.
- [46] [薛宏全, 魏生民, 张鹏, 杨琳. 基于多种群蚁群算法的柔性作业车间调度研究\[J\]. 计算机工程与应用, 2013, 49\(24\):243-248+261.](#)
- [47] [张维存, 郑丕谔, 吴晓丹. 基于主-从遗传算法求解柔性调度问题\[J\]. 计算机集成制造系统, 2006, 12\(8\):1241-1245.](#)
- [48] [杨晓梅, 曾建潮. 遗传算法求解柔性job shop调度问题\[J\]. 控制与决策, 2004, 19\(10\):1197-1200.](#)
- [49] 王小蓉, 李蓓智, 周亚勤,等. 基于混合遗传算法的柔性作业车间调度研究[J]. 现代制造工程, 2015(5):39-42.
- [50] 何林燕. 云制造环境下柔性作业车间调度算法的研究[D]. 哈尔滨理工大学, 2017.
- [51] 宋娟. 基于多Agent的制造车间动态调度系统研究[D]. 沈阳工业大学, 2004.

致谢

三年的研究生学习生涯即将结束了, 回想其中的点点滴滴, 不禁感慨万千。首先, 我由衷地感谢我的导师, 张平教授, 这篇论文以及各种相关实验都是在张老师的悉心指导下完成的。张老师严谨的学习态度, 丰富的知识储备让我在科研学习中少走了很多弯路。在这三年中, 我一直比较任性, 给张老师添了很多麻烦, 但老师一直不厌其烦地纠正我的学习态度, 教导了我很多为人处世的道理, 在我日常生活出现困难的时候给予了我很多鼓励和关怀, 我心怀愧疚的同时又无比感恩, 必将珍惜这份宝贵的人生经验。

同时还需要感谢实验室的李方老师和杜广龙老师。两位老师积极向上、治学严谨, 对我的科研工作[给予了很多指导, 给我的生活带来了很大帮助, 在此表示由衷的敬意。](#)

此外, 我还要感谢我的室友何子平和徐伟杰。两位室友非常优秀, 非常善良, 三年来我们一起上课, 一起在实验室开展工作, 没有出现一次矛盾, 他们平时分享我的快乐, 分担我的忧愁, 一起成长, 能够遇到两位志同道合的好朋友, 我感到非常幸运。



另外我还要感谢我的师姐陈昕叶，我平时的科研工作以及论文的完成少不了师姐的功劳，师姐教会了我很多宝贵的学习方法，在算法理论和实验设计上给予了很多指导，在此我同样感到无比的感恩。同时，我还要感谢我的师弟庄焕嘉、赵俊宇、邵亨康，在团队实验过程中，他们给予了我很多技术支持。

感谢我的爸爸、妈妈、姐姐，他们一直是我成长路上的避风港，为我付出了很多，承担了很多，他们也一直包容我的任性，我的缺点，成长至今，我唯有通过不断的努力来回报家人对我无微不至的关怀。

最后，向各位论文评审专家致以深深的敬意，感谢你们在百忙之中抽出宝贵的时间对本文进行审阅，给我提出宝贵的修改意见！

指 标
疑似剽窃文字表述
1. 各位论文评审专家致以深深的敬意，感谢你们在百忙之中抽出宝贵的时间对本文进行审阅，

说明：1.指标是由系统根据《学术论文不端行为的界定标准》自动生成的。

- 2.红色文字表示文字复制部分; 黄色文字表示引用部分。
- 3.本报告单仅对您所选择比对资源范围内检测结果负责。
- 4.Email：amlc@cnki.net  <http://e.weibo.com/u/3194559873>  http://t.qq.com/CNKI_kycx