

摘 要

作业车间调度问题(Job-shop Scheduling Problem, 简称 JSP)是一类满足任务配置和顺序约束要求的资源分配问题,是一类典型的 NP-hard 问题,至今没有找到可以精确求得最优解的多项式时间算法。有效地调度方法和优化技术的研究与应用,对于制造企业提供生产效率、降低生产成本有着重要的作用,因此越来越受学者们的关注。

遗传算法是基于“适者生存”的一种高度并行、随机和自适应优化算法,它将问题的求解表示成“染色体”的适者生存过程,通过“染色体”群的一代代不断进化,包括复制、交叉和变异等操作,最终收敛到“最适应环境”的个体,从而求得问题的最优解或满意解。

本文系统介绍了作业车间调度问题以及遗传算法的基本原理,并针对作业车间调度问题的特点,设计了一种遗传算法。最后使用 Matlab 编写程序求解 Job Shop 调度问题。并对两类典型的 FT 类问题 FT06 和 FT10 进行测试。

关键词: 车间调度, 遗传算法, Job Shop 问题

Abstract

Job shop scheduling problem (Job-shop Scheduling Problem, referred to as JSP) is a class of constraints to meet the tasks required to configure and order the allocation of resources. It is a kind of typical NP-hard problem, which has not found the optimal solution to get accurately obtained polynomial time algorithm. Effective methods and optimization techniques in scheduling research and applications, take an important role in manufacture enterprises production efficiency, reduce production costs, so more and more attention by scholars.

Genetic algorithm is based on the "survival of the fittest" in a highly parallel, random and adaptive optimization algorithm, it will solve the problem that a "chromosome" of the survival of the fittest process, through "chromosome" group of evolving from generation to generation, including reproduction, crossover and mutation operations, and eventually converge to the "best adapt to the environment," the individual, and thus obtain the optimal solution or a satisfactory solution.

This paper introduced the job-shop scheduling problem and the basic principles of genetic algorithms, to solve job shop scheduling problem, we designed a genetic algorithm, and using Matlab programming solve Job Shop Scheduling Problem. Finally test two typical kinds of problems FT06 and FT10.

Keyword: Production scheduling, Genetic algorithm, Job Shop problem

目 录

摘 要	I
Abstract	II
第一章 绪论	1
1.1 车间调度研究的目的和意义	1
1.2 车间调度的研究现状	2
1.3 本文安排	3
第二章 车间调度问题综述	4
2.1 车间调度问题	4
2.1.1 车间调度问题的描述	4
2.1.2 车间调度的性能指标	4
2.1.3 车间调度问题的分类	5
2.1.4 实际调度问题的特点	5
2.2 车间调度问题的研究方法	6
2.3 车间调度研究中存在的主要问题	9
2.4 本章小结	9
第三章 遗传算法理论综述	10
3.1 遗传算法的形成和发展	10
3.2 基本遗传算法	11
3.3 基本遗传算的实现技术	13
3.3.1 编码方法	13
3.3.2 适应度函数的确定	14
3.3.3 选择算子	14
3.3.4 交叉算子	15
3.3.5 变异算子	15
3.3.6 算法参数的选取	16
3.3.7 算法终止条件的确定	16
3.4 遗传算法的应用	17
3.5 本章小结	17
第四章 Job Shop 调度问题的遗传算法研究	18
4.1 Job Shop 调度问题描述	18
4.2 Job Shop 调度的参数设计	19
4.2.1 基本流程图	20
4.2.2 编码方式的确定	20
4.2.3 适应度函数	21
4.2.4 遗传算子的设计	21
4.2.5 算法参数的设计	22
4.3 Job Shop 调度的 Matlab 实现	22

4.4 本章小结.....	24
第五章 结论与展望.....	25
5.1 结论.....	25
5.2 展望.....	25
参考文献.....	27
致谢.....	错误！未定义书签。

第一章 绪论

本文的主要内容是运用遗传算法解决车间调度中的作业车间调度（Job Shop）问题。因此，本章首先介绍车间调度研究的目的和意义以及车间调度目前的研究现状，最后简单介绍本文的安排。

1.1 车间调度研究的目的和意义

为了适应激烈的市场竞争，多元化、多品种、小批量、高柔性生产方式正逐渐成为主流，生产的复杂性也越来越高，这对企业的管理和对生产过程的监控都提出了更高的要求。为了保证生产的高效稳定运行，以获得最大的经济效益，原来简单、局部和常规的生产计划及仅凭经验的管理已经不能满足现代生产的要求了。企业管理者面临的问题是：

如何根据市场上原材料供应和产品需求的变化进行经营决策和组织生产；

如何在生产计划改变的情况下对生产过程进行控制，以便最大限度地发挥生产的柔性；

如何在生产工艺不做大的改变前提下进行管理、决策，使企业生产最大的综合经济效益。

有限资源的合理配置与优化利用问题一直是人类社会所面临的最基本经济学问题，从一个国家、社会的宏观经济运行到企业的微观经济活动，都要受到资源条件的制约。这个问题涉及到了国民经济从宏观到微观的各个方面，人们对他的研究也就从不同的角度进行，产生了有关这类问题的不同的学科和理论，其中在生产邻域产生了重要的车间调度理论。

车间调度理论源于对制造车间生产计划与控制的研究，从诞生之日起它就引起了运筹学专业人员和应用数学家等的重视，人们在这个领域出版和发表了大量的文献著作，还有一些学者从不同的广度和深度考察了调度理论的发展状况。经过几十年的研究与探索，调度理论逐渐发展成为一个比较完整的科学理论，在企业的生产中得到了一定程度的应用，而且其研究范围已经从制造业的车间扩展到了物流、能源、交通运输和社会服务等许多行业和研究领域。

作为一个关键模块,车间调度是整个先进生产制造系统实现管理技术、运筹技术、优化技术、自动化和计算机技术发展的中心,在工厂经营管理、产品制造这两个层次上都占有极其重要的地位和作用。车间调度主要用于解决工件在机器上的调度和资源分配问题,可以大大提高生产效率和资源利用率,进而增加企业的竞争力。因此对制造企业车间作业调度问题进行研究与有重要的现实意义,它是实现车间作业调度的合理化、自动化和集成化重要环节。

然而,目前我国许多国有大中型企业生产管理技术及手段落后,缺乏系统科学的车间调度方法,生产计划和调度安排仍然依靠经验,导致产品制造周期长,资源有效利用率低,制造成本偏高,交货期长,对市场反映不够灵敏,严重影响了企业的经济效益,甚至造成许多企业亏损。

为了改变这种状况,一些企业从国外引进了一些较为先进的生产管理技术和信息系统,如 MRP II (制造资源计划) 技术和看板控制方法等,它们在一定范围内得到了应用,但都存在局限性,比如 MRP II 更多的是强调对整个企业生产计划的编制,对车间层的生产控制能力不足,特别是对生产能力的估计脱离实际,对车间生产环境变化的适应性不强,看板控制方法的应用也需要较为特殊的条件,而且国外的东西并不能都适用于国内,所以大多数制造企业的车间生产计划和调度还处在依赖车间管理人员工作经验的安排阶段。

1.2 车间调度的研究现状

调度问题的研究始于 20 世纪 50 年代,Johnson 提出了解决车间调度部分特殊问题的优化算法,代表调度理论研究的开始。60-70 年代建立了调度理论的主体(经典调度理论)并重视调度复杂性的研究。大量的研究促使车间调度领域取得了丰硕的成果,产生了很多重要的调度方法。随着 70 年代后期调度理论研究的深入及各种交叉学科的发展,又涌现出了许多新的车间调度理论与方法。

Davis^[1]是最早把遗传算法(GA)应用于车间调度问题的学者之一,他在使用 GA 求解车间调度的研究中取得了近似最优解。1985 年,Davis 发表了关于把 GA 成功应用于车间调度问题的论文,充分展示了 GA 在解决车间调度问题中的前景。此后,很多学者就遗传算法在车间调度问题方面的应用作了大量研究,发表了大量卓有成效的论文,是车间调度这类 NP-hard 问题的解决展露曙光。

国内对车间调度的研究起步较晚,由于技术上的制约,基本上是依靠调度人员的经验进行车间分配和调度。随着各种智能算法尤其是遗传算法在作业调度方面的应用,国内掀起了一股应用遗传算法研究车间调度的热潮:研究工作主要集中在清华大

学的 CIMS^[2]国家重点实验室,而目前这些实验室的系统开发基本上还处在刚投入试运行阶段,距离开发出成熟的软件系统还有很长距离,实验室还在投入大量的人力和物力进行该方面的研究,特别是在开展对车间作业调度算法的研究方面。

1.3 本文安排

本文研究的框架如图所示:

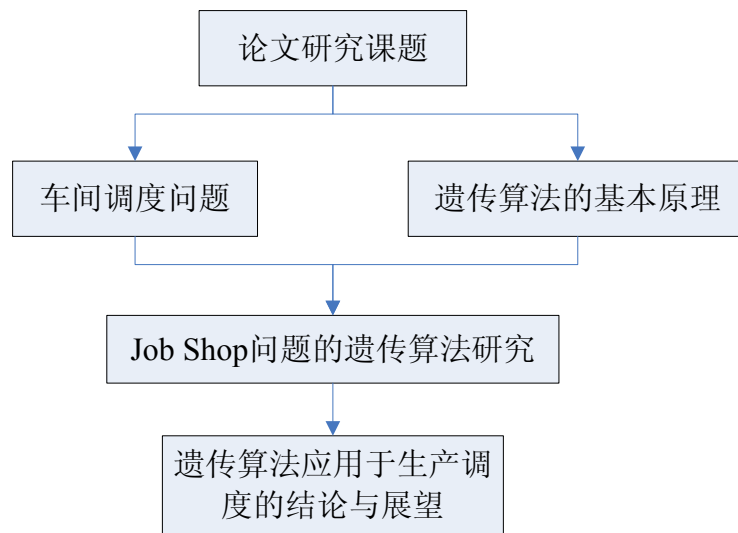


图 1-1 本文框架图

本论文共分为 5 章,各章内容如下:

第 1 章 绪论。简要介绍了本课题的目的及意义和车间调度目前的研究现状,并阐述了论文的安排。

第 2 章 车间调度问题综述。详细介绍了车间调度问题的分类、特点、研究背景,以及主要的研究方法和存在的问题。

第 3 章 遗传算法的基本原理。详细介绍了遗传算法的形成、发展、特点以及遗传算法的实现技术和应用领域,为第四章编程解决 Job Shop 调度问题提供理论基础。

第 4 章 Job Shop 问题的遗传算法研究。详细介绍了典型的 Job Shop 调度的描述和求解方法,然后对遗传算法参数进行设计,最后对两类典型的 FT 类问题 FT06 和 FT10 进行测试。

第 5 章 结论与展望。

第二章 车间调度问题综述

2.1 车间调度问题

2.1.1 车间调度问题的描述

车间调度是在满足某些约束（作业的先后关系、预定的完成时间、最早开始时间和资源能力等）的条件下对操作（作业）的排序，按照排序的次序给它们分配资源和时间，并且使某个执行目标达到最优（如总的执行时间、拖期时间和生产费用等）。

车间调度问题一般可以描述为：针对某项可分解的工作，在一定约束条件下，如何安排其组成部分（操作）所占有的资源、加工时间、先后顺序，以获得产品制造时间和成本等最优。

影响调度问题的因素很多，正常情况下有：产品的投产期、交货期、生产能力、加工顺序、加工设备和原料的可用性，批量大小、加工路径、成本限制等，这些都是所谓的约束条件。有些约束条件是必须要满足的，如交货期，生产能力等，而有些达到一定满意度就行，如生产成本等，这些约束在进行调度时可以作为确定性因素考虑。而对于设备故障，原料供应变化，生产任务等非正常情况，都是事先不能预见的，在进行调度时大都作为非确定性因素考虑。

车间调度中涉及的资源包括：原料、设备、人力、资金、能源等。车间调度的性能指标可以使成本最低、库存费用最少、生产周期最短、生产切换最少、设备利用率最高、三废最少等。

2.1.2 车间调度的性能指标

实际车间调度的性能指标大致可以归结为三类：

1. 最大能力指标 包括最大生产率、最短生产周期等，它们都可以归结为在固定或者无限的产品需求的前提下，最大化生产能力以提高经济效益。
2. 最低（高）成本指标 包括最大利润、最小运行费用、最小投资、最大收益等，其中收益指产品销售收入，运行费用包括库存成本，生产成本和缺货损失。
3. 客户满意度指标 包括最短的延迟，最小的提前或者拖期惩罚等。

在传统的调度中，一般以平均流通时间最小、制造周期最短、满足交货期为调度目标，而在实际生产中，由于提前完成的产品必须保存到交货期，而拖期产品必须交付违约金，因此，在实际调度中经常考虑“提前”或者“拖后”的惩罚。

2.1.3 车间调度问题的分类

从运筹学的角度看，车间调度可表达为在等式或不等式约束下，对一个或多个目标函数的优化。现代典型的车间调度问题是：将作业匀称地安排到各处理机上，并合理的安排作业的加工次序和开始时间，使约束条件被满足，同时优化一些性能指标。

对于车间调度问题，Graves 等人进行了分类整理，按照不同的分类标准，可分为以下四种类型^[3]：

1. 根据加工系统的复杂性，可分为单机、多台并行机、Flow Shop 和 Job Shop。

单机调度问题是所有的操作任务都在单台机器上完成，为此存在任务的优化排队问题；多台并行机的调度问题更复杂，因而优化问题更突出；Flow Shop 型问题假设所有作业都在同样的设备上加工，并有一致的加工操作和加工顺序；Job Shop 是最一般的调度类型，不同的作业具有不同的加工操作和加工顺序，并不限制作业的加工设备。现代车间调度类型往往是 Job Shop 类型的。

2. 根据性能指标，分为基于调度费用和调度性能的指标两大类。

3. 根据作业的加工特点，可将调度问题分为确定性调度和随机性调度问题。

4. 根据生产环境的特点，可将调度问题分为静态调度和动态调度。

静态调度是指所有待安排的工作均处于待加工状态，因而进行一次调度后，各作业的加工时间被确定，在以后的加工过程中就不再改变；动态调度是指作业依次进入待加工状态、各种作业不断进入系统接受加工、同时完成加工的作业又不断离开，还要考虑作业环境中不断的动态扰动，如作业的加工超时、设备的损坏等。因此动态调度要根据系统中作业、设备等的状况，不断地进行调度。实际调度的类型往往是 Job Shop 型，且是动态的。

2.1.4 实际调度问题的特点

1. 复杂性

由于装卸作业、装卸设备、库场、搬运系统之间相互影响、相互作用，每个作业又要考虑它的到达时间、装卸时间、准备时间、操作顺序、交货期等，因而相当复杂。而且调度问题是在等式或不等式约束下求性能指标的优化，在计算量上往往是 NP（多项式非确定性问题）完全问题，即随着问题规模的增大，对于求解最优化的计算量呈指数增长，使得一些常规的优化方法往往无能为力。即使对于单台机床加工问题，如

果有 n 个工件而每个工件只考虑加工时间以及与操作序列有关的安装时间, 则这个问题就和 n 个城市的 TSP (旅行商问题) 等价。对于一般加工系统, 问题更加复杂。

2. 动态随机性

在实际的车间调度系统中存在很多随机的和不确定的因素, 比如作业到达时间的不确定性、作业的加工时间也有一定的随机性, 而且生产系统中常出现一些突发偶然事件, 如设备的损坏或修复、作业交货期的改变等。

3. 多目标性

实际的计划调度往往是多目标的, 是一个多目标决策问题, 有时候这些目标间可能发生冲突。Kiran 等人将调度目标分三类: 基于作业交货期的目标、基于作业完成时间的目标、基于生产成本的目标。这种多目标性导致调度的复杂性和计算量急剧增加。基于时间和作业交货期的目标, 主要包括任务的生产周期最短, 完成日期与交货期最接近, 任务在制造系统中的总等待时间最短等; 基于成本的目标主要考虑如何减少生产过程中的资金占用和怎样合理利用资源, 使各个生产环节符合均匀, 保证生产的稳定性。

2.2 车间调度问题的研究方法

在对车间调度问题进行研究的方法^[3]上, 最初是集中在整数规划、仿真和简单的规则上, 这些方法不是调度结果不理想就是难以解决复杂的问题。随着各种新的相关学科与优化技术的建立与发展, 在调度领域也出现了许多新的优化方法, 比如神经网络法、模拟退火法、遗传算法、禁忌搜索法等、使得调度问题的研究方法向多元化方向发展。总结起来, 现有调度问题的研究方法大体上可以分为以下几种类别。

1. 基于运筹学 (OR) 的方法

OR 方法是將车间调度问题简化为数学规划模型, 采用基于枚举思想的分支定界法或动态规划算法进行解决调度最优化或近似优化问题, 属于精确方法。这类方法虽然从理论上能求得最优解, 但由于其计算复杂性的原因, 因而不能获得真正的实用。对于复杂的问题, 这种纯数学方法有模型抽取困难、运算量大、算法难以实现的弱点, 仅适合较小规模的调度问题。

2. 基于启发式规则^[4]的方法

从实用的角度来看, 启发式算法因其易于实现、计算复杂度低等原因, 在实际中得到了比较广泛的应用, 并且不断涌现出许多新的调度规则, 可将其分为三类: 简单规则、复杂规则、启发式规则。虽然启发式规则常被用于实际当中, 但他们一般不具有全局优化的特点。

3. 控制理论方法

Gershwin 等人从控制理论的角度出发,较全面地阐述了控制理论方法在制造系统的应用情况。控制论方法比较适合定量的定义基本问题,但还没有形成一套有效的调度模型表达、分析设计的技术。其缺点是模型描述能力有限,建模时不得不对真实环境进行大量的简化,求得最优解的时间随着问题规模的增大而呈指数规律增长。因此也只适合对小规模的系统求解。

4. 基于人工智能(AI)的方法

人工智能的方法是通过提高调度方法的智能来解决各类车间调度问题方法的总称。单一的数学方法和工具不足以解决实际调度问题,人工智能和专家系统的出现对解决调度的实时性和智能性提供了新的辅助手段。它们用于调度可以克服数学规划和仿真方法的不足,根据系统当前的状态和给定的优化目标,对知识库进行有效的启发式搜索和并行模糊推理机制,避开了繁琐的计算,并选择最优的调度策略,为在线决策提供支持。在基于知识和规则的调度系统中,常用的知识表示法有生产式规则、语义网络、框架等;常用的调度技术有生产规则、探拭法搜索、机会主义的推理、仿真和分级式的分解等。

专家系统作为一个较好的调度方法,存在一些不足之处:①对新的调度环境的适应性较差。②开发周期长、产品昂贵。③需要很多现场调度人员丰富的调度经验和知识,然而获取调度人员的调度经验是较困难的和有限的。同时,由于人工智能技术本身发展水平的限制,它有执行速度慢。对异步发生的外来事件不敏感,系统不通用三个明显的缺陷。

5. 基于离散事件动态系统(DEDS)的解析模型方法

对于车间类型的 DEDS,同样可用其解析模型与方法来解决车间调度问题。如极大代数法、动态规划法等。但它们适合于制造系统的性能分析。Petri 网除具有并发、动态、直观等优点外,还有能够准确快速的反应制造系统实时调度的离散型与随机性等特点,所以它与其他方法相结合在调度问题中得到了广泛的应用。目前, Petri 网用于制造系统调度存在以下问题:①节点语义的单义性,使得所携带的系统信息不够丰富。②重用性差。③当调度规则或方法复杂时,建模困难。

6. 基于模糊数学理论的方法

客观现象具有确定性与不确定性两个基本方面,经典数学表达的是现象的确定性;不确定性一方面表现为随机性,另一方面表现为模糊性。正是利用此特点,许多学者把它引入了调度领域。但与专家系统相似,这种方法同样存在开发周期长、需要丰富的调度经验和知识等缺点。

7. 拉氏松弛法

拉氏松弛法由于其在可行的时间里能对复杂的规划问题提供好的次优解,并能对

解的次优性能进行定量评估,近年来已成为解决复杂车间调度问题的一种重要方法。但不可避免的是,拉氏松弛法其对偶问题存在搜索效率低、可行调度的构造有待于进一步研究等问题。

8. 神经网络(NN)优化法

NN 用于车间调度主要有三类方式;①利用其并行计算能力,求解优化调度,以克服调度的 NP 难问题。②利用其学习能力,从优化轨迹中提取调度知识。③用 NN 来描述调度约束或调度策略,以实现生产过程的可行或次优调度。Hopfield 神经网络模型的提出为求解各种有约束优化问题开辟了一条新途径,用 Hopfield 网络解决 TSP 问题就是其在组合优化问题中的最成功的应用之一。

9. 具有计算智能的局域搜索法

近年来,一些高级局域搜索法由于具有普遍适用性和较低的经验复杂性等优点,得到了广发的重视和应用,主要有以下几种方法:

(1) 遗传算法(GA) Goldbers 首次将 GA 应用到实际的工程系统的优化当中。由于 GA 原理和操作简单,通用性强,不受限制条件的约束,且具有隐含并行性和全局解空间搜索能力,在机器学习、模式识别、控制工程、优化等领域,尤其是在车间调度领域得到广泛的应用。如何利用 GA 高效求解调度问题,一直被认为是一个具有挑战意义的难题并成为研究的热点,近年来涌现出大量论文并取得较大进展。GA 的早熟和搜索效率低问题是它的主要缺陷。

(2) 禁忌搜索(TS) TS 是解决复杂混合优化问题的一种搜索策略和方法,由 Gfove 等人提出、完善和发展。目前,TS 已在调度、交通运输、TSP 问题、电子电路设计等诸多领域中得到应用。

(3) 模拟退火(SA) SA 可求得组合优化问题的最优或此最优解。首次将 SA 用于优化领域的是 KirkPatrick,他尝试了将 SA 用于 TSP 问题的求解。此外,SA 在 Job Shop 和 Flow Shop 问题中也得到了一定的应用。

由于这些算法本质上是启发式算法且引入了随机的概念,所以存在性能稳定性的问题。

10. 组合调度方法

由于各种调度算法都不同程度地存在着这样或那样的优缺点,除了传统组合的启发式规则外,近来人们开始把各种近似算法的组合应用研究作为热点,以弥补各自的缺点,发挥各自的优势,达到高度次优化的目标。目前,这种算法的组合应用已成为解决优化调度问题很有前途的方法。

2.3 车间调度研究中存在的主要问题

由于大多数调度问题属于一类 NP-hard 组合问题, 因此寻找具有多项式复杂性的最优解算法几乎是不可能的。传统的启发式算法, 智能模拟退火算法, 禁忌算法, 神经网络算法等算法其共性是对生产线优化问题寻求满足实际需要的近似解或满意解, 而非精确最优解, 在大规模生产线优化调度具有很好的应用前景, 但都是局部寻优的方法。而各种基于统计优化的方法、诸如模拟退火法、遗传算法等, 提供了一种解决调度优化问题的新途径, 但其也存在着一定程度的枚举、一般来说收敛到最优解较慢, 并且对于判断解的最优性也很困难。在这方面也需要做进一步的研究。

在调度问题的理论研究中, 大多数还是集中在针对经典的调度问题设计优化算法, 而经典调度问题与实际相差较大, 尤其在目前柔性制造环境下, 柔性车间作业调度问题将是一个研究的重点和进一步研究的方向, 而目前对于这方面研究的文献相对来说较少。在实际车间调度中, 车间计划于车间调度往往是分层进行的, 但这可能造成计划在实际调度中的不可行问题, 如何将计划与调度结合考虑, 以求总体的优化也是需要进一步研究的。

2.4 本章小结

本章主要介绍了车间调度问题的分类、特点、研究背景, 以及主要的研究方法和存在的问题, 说明使用遗传算法研究车间调度问题的必要性。

第三章 遗传算法理论综述

遗传算法（GA）是一种借鉴生物界自然选择和自然遗传机制的高度并行、随机、自适应搜索算法。它是模仿自然界生物进化过程中“物竞天择，适者生存”的原理而进行的一种多参数、多群体同时优化方法。经过 30 多年的发展，遗传算法已经在数据挖掘、车间调度、函数优化、机器学习、图像处理等领域得到成功应用，并显示出其良好的性能。

3.1 遗传算法的形成和发展

遗传算法起源于对生物系统的所进行的计算机模拟研究。早在上世纪 40 年代，就有学者开始研究如何利用计算机进行生物模拟的技术，他们从生物学的角度进行了生物的进化过程模拟、遗传过程模拟等研究工作。进入 60 年代后，美国密执安大学的 Holland^[5] 教授及其学生们受到这种生物模拟技术的启发，创造出了一种基于生物遗传和进化机制的适合与复杂系统优化计算的自适应概率优化技术——遗传算法。Holland 教授用遗传算法的思想对自然和人工自适应系统进行了研究，提出了遗传算法的基本定理——模式定理，并于 1975 年出版了第一本系统论述遗传算法和人工自适应系统的专著《自然系统和人工系统的自适应性（Adaptation in Natural and Artificial Systems）》。80 年代，Holland 教授实现了第一个基于遗传算法的机器学习系统——分类器系统（CS），开创了遗传算法的机器学习新概念。

1967 年，Holland 的学生 Bagley 在其博士论文中首次提出了“遗传算法”一词，并发表了遗传算法应用方面的第一篇论文，他发展了复制、交叉、变异、显性、倒位等遗传算子，在个体编码上使用了双倍体的编码方法。这些都与目前遗传算法中所使用的算子和方法相类似。他还敏锐的意识到了在遗传算法执行的不同阶段可以使用不同的选择率，这将有利于防止遗传算法的早熟现象，从而创立了自适应遗传算法的概念。

1975 年，De Jong 在其博士论文中结合模式定理进行了大量的纯数值函数优化计算实验，树立了遗传算法的工作框架，得到了一些重要而具有指导意义的结论。他推荐了在大多数优化问题中都较适用的遗传算法参数，还建立了著名的 De Jong 五函数测试平台，定义了评价遗传算法性能的在线指标和离线指标。

1989 年，Goldberg 出版了专著《搜索、优化和机器学习中的遗传算法（Genetic

Algorithms in Search, Optimization and Machine Learning)》。该书系统总结了遗传算法的基本原理及其应用。这本书奠定了现在遗传算法的科学基础。

1991 年, Davis 编辑出版了《遗传算法手册 (Handbook of Genetic Algorithms)》, 该书包括了遗传算法在科学计算、工程技术、和社会经济中的大量应用实例。这本书为推广科普及遗传算法的应用起到了重要的指导作用。

1992 年, Koza 将遗传算法应用于计算机程序的优化设计及自动生成, 提出了遗传编程 (Genetic Programming) 的概念, 他将一段 LISP 语言程序作为个体的基因型, 把问题的解编码为一颗树, 基于遗传和进化的概念, 对由数组成的群体进行遗传运算, 最终自动生成性能较好的计算机程序。

从遗传算法的整个发展看, 70 年代是兴起阶段, 80 年代是发展阶段, 90 年代是高潮阶段。遗传算法作为一种实用、高效、鲁棒性强的优化技术, 发展极为迅速, 已引起国内外学者的高度重视。

3.2 基本遗传算法

基本遗传算法^[6] (也称为标准遗传算法, Simple Genetic Algorithm, 简称 SGA) 是一种种群型操作, 该操作以种群中的所有个体为对象, 只使用基本遗传算子、选择算子、交叉算子和变异算子, 其遗传进化操作过程简单, 容易理解, 是其他一些遗传算法的基础, 不仅给各种遗传算法提供了一个基本框架, 同时也具有一定的应用价值。选择、交叉和变异是遗传算法的三个主要操作算子, 它们构成了所谓的遗传操作, 使遗传算法有了其他传统方法没有的特点。基本遗传算法示意图^[7]如图 3-1。

基本遗传算法的数学模型可以表示成 $SGA = (C, E, P_0, M, \Phi, \Gamma, \Psi, T)$ 。

式中:

C ——个体编码方法;

E ——个体适应度评价函数;

P_0 ——初始种群;

M ——种群大小;

Φ ——选择算子 (复制算子);

Γ ——交叉算子;

Ψ ——变异算子;

T ——遗传运算终止条件。

1. 编码: 在遗传算法中, 把一个问题的可行解从其解空间转换到遗传算法所能处理的搜索空间的转换方法称为编码。而由遗传算法的解空间向问题的转换成为解码

2. 初始种群：一组随机产生的初始解。
3. 种群大小：在中群众染色体的数量。
4. 染色体：种群中的每个个体是问题的一个解，称为染色体。染色体是一串符号，比如一个二进制字符串。这些染色体在后续迭代中不断进化，称为遗传。
5. 适应度：表示某个体对环境的适应度，用来测量染色体的好坏。
6. 后代：生成的下一代染色体。后代是由前一代染色体通过交叉或者变异运算形成的。

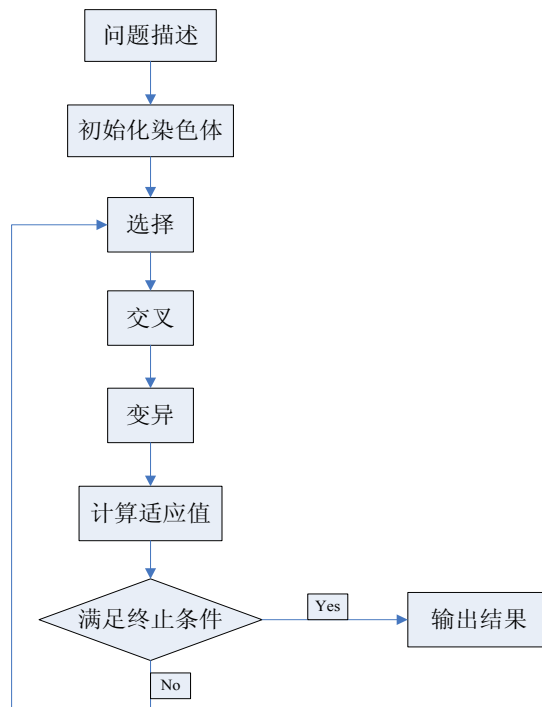


图 3-1 基本遗传算法示意图

基本遗传算法程序：

```

begin
    initialize P(0);
    t=0;
    while (t<=T) do
        for i=1 to M do
            Evaluate fitness of P(t);
        end
        for i=1 to M do
            Select operation to P(t);
        end
        for i=1 to M/2 do
            Crossover operation to P(t);
        end
        for i=1 to M do
            Mutation operation to P(t);
        end
    end
end
    
```



```

    end
    for i=1 to M do
        P(t+1)=P(t)
    end
    t=t+1;
end
end

```

3.3 基本遗传算的实现技术

遗传算法的基本步骤^[6]是：

1. 确定问题的编码方法。由于遗传算法通常不直接作用于问题的解空间，而是利用解的某种编码表示来进行进化，因此选择合适的编码机制对算法质量和效率有很大影响。
2. 确定适应度函数。由于遗传算法通常基于适应度进行遗传操作，因此合理的适应度能够将各个体的优劣程度得以体现，并适应算法的进化过程。
3. 算法参数的选取。通常包括种群数目、交叉概率、变异概率、进化代数等。
4. 遗传算子的设计。通常包括初始化、选择、交叉、变异和替换操作等。
5. 确定算法的终止条件。终止准则应根据所求解问题的性质，在优化质量和效率方面作合理的均衡或侧重。

3.3.1 编码方法

把一个问题的可行解从其解空间转换到遗传算法所能处理的搜索空间的转换方法称为编码。遗传算法的一个显著特点是它交替地在编码空间和解空间中工作，它在编码空间对染色体进行遗传运算，而在解空间对解进行评估和选择因此编码的选择是影响算法性能与效率的重要因素。De Jong 曾提出了两条操作性较强的实用编码原则：①应使用能易于产生于所求问题相关的且具有低阶、短定义长度模式的编码方案。②应使用能使问题得到自然表示或描述的具有最小编码字符集的编码方案。

针对函数优化的编码技术主要有二进制编码、格雷码编码、实数编码、浮点数编码、符号编码、多参数级编码等

3.3.2 适应度函数的确定

适应度函数用于对个体进行评价，也是优化过程发展的依据。遗传算法在进化搜索中基本不利用外部信息，仅使用所求问题的目标函数值就可得到下一步的有关搜索信息。

最优化问题可以分为两大类，一类为求目标函数的全局最大值，另一类为求目标函数的全局最小值。由解空间中某一点的目标函数值 $f(X)$ 到搜索空间中对应个体的适应度函数值 $F(X)$ 的转换方法：

- 对于求最大值的问题，作下述转换：

$$F(X) = \begin{cases} f(X) + C_{\min}, & \text{if } f(X) + C_{\min} > 0 \\ 0, & \text{if } f(X) + C_{\min} \leq 0 \end{cases} \quad (3-1)$$

- 对于求最小值的问题，作下述转换：

$$F(X) = \begin{cases} C_{\max} - f(X), & \text{if } f(X) < C_{\max} \\ 0, & \text{if } f(X) \geq C_{\max} \end{cases} \quad (3-2)$$

式中 C_{\max} 为一个适当的相对较大的数

由于适应度度量意义下的个体差异与目标函数值度量意义下的个体差异有所不同，因此若适应度函数设计不当，将难以体现个体的差异，选择操作的作用就很难体现出来，从而造成早熟收敛等缺点。对适应度进行调节是常用的改进方法。此外在车间调度问题的求解中，目前已开发了许多基于工序的选择操作，它基于目标值的好坏分配选择概率，无需设计适应度函数，一定程度上缓解了适应度函数定义的困难。

在选择操作时出现以下问题：

1. 在进化初期，通常会产生一些超长的个体，若按照比例选择法，这些异常个体因为竞争突出而控制了选择过程，影响算法的全局优化性能。
2. 在遗传进化的后期，即算法接近收敛时，由于种群中个体适应度差异较小时，继续优化的潜能降低，可能获得某个局部最优解。

上述问题通常称为遗传算法的欺骗问题。适应度函数设计不当可能出现这种问题，因此适应度函数的设计是遗传算法设计的一个重要方面。

3.3.3 选择算子

遗传算法中的选择操作就是用来确定如何从父代群体中按某种方法选取哪些个体遗传到下一代群体中的一种遗传运算。选择操作的目的是为了避免有效基因的损失，是高性能的个体的以更大的概率生存，从而提高全局收敛性和计算效率。判断染色体优良与否的准则就是各自的适应度值，个体适应度值越大，其被选择的机会就越

多。选择操作体现了达尔文的优胜劣汰、适者生存原则。选择的方法根据不同的问题，采用不同的方案，最常用的方法是比例选择，此外还有最优保存策略、确定式采样选择、无回放随机选择、排序选择和随机联赛选择。

比例选择又称轮盘赌，是一种回放式随机采样的方法。其基本思想是：设群体大小为 M ，个体 i 的适应度为 F_i ，则个体 i 被选中的概率 p_{is} 为

$$p_{is} = F_i / \sum_{i=1}^M F_i \quad (i=1,2,\dots,M) \quad (3-3)$$

由式（3-3）可见，适应度越高的个体被选中的概率也越大；反之，适应度越低的个体被选中的概率也越小。…

3.3.4 交叉算子

遗传算法中的所谓交叉运算，是指对两个相互配对的染色体按某种方式相互交换器部分基因，从而形成两个新的个体。交叉运算是遗传算法区别与其他进化算法的重要特征，它在遗传算法中起着关键作用，是产生新个体的主要方法。最常用的交叉算子是单点交叉算子，此外还有双点交叉、多点交叉、均匀交叉和算术交叉。

单点交叉是指在个体编码串中只随即设置一个交叉点然后在该点相互交换两个配对个体的部分染色体。单点交叉的重要特点是：若邻接基因座之间的关系能提供较好的个体性状和较高的个体适应度的话，这种单点交叉操作破坏这种个体性状和降低个体适应度值的可能性最小。

3.3.5 变异算子

当交叉操作产生的后代适应值不再进化且没有达到最优时，就意味着算法的早熟收敛。这种现象的根源在于有效基因的损缺，变异操作一定程度上克服了这种情况，有利于增加种群的多样性。变异首先在群体中随机地选择一个个体，以一定的概率随机改变基因串中某个字符的值。

变异操作是按位进行的，对于采用二进制编码的个体来说，就是将 1 变为 0 或将 0 变为 1，变异发生的概率极低，一般取值为 0.0001-0.1 之间，它本身是一种随机搜索，但与选择、交叉算子结合在一起，就能避免由复制和交叉算子引起的某些信息的永久丢失，从而保证了遗传算法的有效性。

3.3.6 算法参数的选取

遗传算法中需要选择的运行参数主要有个体编码长度 l 、群体大小 M 、交叉概率 p_c 、变异概率 p_m 、终止代数 T 、代沟 G 。这些参数对遗传算法的运行性能影响较大。

1. 编码串长度 l 。使用二进制编码来表示个体时，编码串长度 l 的选取与问题所要求的求解精度有关；使用浮点数编码来表示个体时，编码串长度 l 与决策变量的个数 n 相等。

2. 群体大小 M 。种群大小是影响算法最终优化性能和效率的因素之一。通常，种群太小时，不能提供足够的采样点，以致算法性能很差，甚至得不到问题的可行解；种群太大时，尽管可增加优化信息以阻止早熟收敛的发生，但无疑会增加计算量，从而使收敛时间太长。一般地，种群大小取值为 $20 \sim 100$ 。

3. 交叉概率 p_c 。交叉概率用于控制交叉操作的频率。交叉概率太大时，种群中串的更新很快，进而会使高适应度值的个体很快就被破坏掉，并且会因搜索不必要的解空间而耗费大量的计算时间；概率太小时，交叉操作很少进行，搜索的解空间太小，增加停在非最优解上的机会。交叉概率一般取值为 $0.4 \sim 0.99$ 。

3. 变异概率 p_m 。变异概率太大，即随机搜索的变换太多，那么后代就可能时从双亲继承下来的好特性，这样算法就失去从过去的搜索中学习的能力；变异概率太小则不会产生新个体，一些有用的基因就不能进入选择。变异概率一般取值为 $0.0001 \sim 0.1$ 。

4. 终止代数 T 。表示遗传算法运行到指定的进化代数之后就停止运行，并将当前群体中的最佳个体作为所求问题的最优解输出。一般的取值范围是 $100 \sim 1000$ 。

5. 代沟 G 。代沟用于控制每代中群体被代替的比例，即每代有 $N \times (1-G)$ 个父代个体被选中进入下一代种群。 $G=100\%$ 意味着所有个体将被替换； $G=50\%$ 意味着一半种群将被替换。

3.3.7 算法终止条件的确定

由于实际应用 GA 时不容许让其无停止的进行搜索，同时问题的最优解通常也未知，因此必须设计一些近似收敛准则来终止算法进程。常用的方法包括：给定一个最大进化代数；给定个体评价总数；给定最佳搜索解的最大滞留代数等。一般遗传运算的终止进化代数取值为 $100 \sim 500$ 。

3.4 遗传算法的应用

遗传算法提供了一种求解复杂系统优化问题的通用框架,它不依赖于问题的具体领域,对问题的种类有很强的鲁棒性,所以广泛应用于许多领域^[6]。

1. 函数优化。函数优化是遗传算法的经典应用领域,也是对遗传算法进行性能评价的常用算例。可以用各种各样的函数来验证遗传算法的性能。对于一些非线性、多模型、多目标的函数优化问题,使用遗传算法可得到较好的结果。

2. 组合优化。随着问题规模的增大,组合优化问题的搜索空间急剧扩大,有的在目前计算机上用枚举法很难甚至不能求出问题的最优解。对这些问题,遗传算法可有效求出满意解。

3. 车间调度问题。采用遗传算法能够解决复杂的车间调度问题。在单件车间调度、流水车间调度、生产规划、任务分配等方面,遗传算法都得到了有效应用。

4. 自动控制。自动控制领域有很多与优化相关问题需要求解,使用遗传算法取得了良好的效果。

5. 图像处理。图像处理的扫描过程、特征提取过程等不可避免的会存在一些误差,这些误差会影响图像处理的效果。遗传算法在这些图像处理的优化计算方面已有应用。

6. 遗传编程。Koza 发展了遗传程序设计的概念,它使用了以 LISP 语言所表示的编码方法,算法基于对一种树形结构所进行的遗传操作来自动生成计算机程序。

7. 机器学习。基于遗传算法的及其信息,特别是分类器系统,在很多领域中得到了应用。

8. 数据挖掘。数据挖掘是数据库研究中的一个很有应用价值的新领域,由于遗传算法的特点,遗传算法可用于数据挖掘中的规则开采。

3.5 本章小结

本章主要介绍了遗传算法的形成、发展、特点以及遗传算法的实现技术和应用领域,为第四章编程解决 Job Shop 调度问题提供理论基础。

第四章 Job Shop 调度问题的遗传算法研究

Job Shop 调度问题^[3]是许多实际车间调度问题的简化模型，是最困难的组合优化问题之一，其研究具有重要的理论意义和工程价值。

4.1 Job Shop 调度问题描述

作业车间调度问题 (Job Shop Scheduling problem, 简称 JSP) 一般可以描述^[8]为： m 个工件在 n 台机器上加工，一个工件可以分为 k 道工序 (本文研究的 FT 类问题中，工序数等于机器数)，每一台机器在每个时刻只能加工某个工件的某道工序，只能在上道工序加工完成后才能开始下一道工序的加工，前者称为占用约束，后者称为顺序约束。所以 JSP 问题的研究内容包括工件的加工顺序和工件各工序的加工时间以及工件工序的加工设备分配。常见的 Job Shop 调度问题应满足如下约束：

1. 一个工件不能同时在不同机器上加工；
2. 对整个工件来说，在加工过程中采用平行移动方式，即当上一道工序完工之后，立即送往下道工序加工；
3. 不容许中断，当一个工件一旦开始加工，必须一直进行到完工，不容许中途停下来，插入其他工件；
4. 每台机器可以执行多种工序的加工任务，但每台机器同时只能加工一个工件；
5. 工件数、机器数、加工时间已知，且加工时间与加工顺序无关；
6. 容许工件在工序之间等待，容许机器在工件未到达时闲置；
7. 工件加工技术上的约定事先给定。

下面是 $m/n/G/C_{\max}$ 调度问题的常用数学描述： $i = 1, 2, \dots, n; k = 1, 2, \dots, m$

$$\min \max_{1 \leq k \leq m} \{ \max_{1 \leq i \leq n} c_{ik} \} \quad (4-1)$$

$$\text{s. t. } c_{ik} - p_{ik} + M(1 - a_{ihk}) \geq c_{ih}, \quad i = 1, 2, \dots, n; k = 1, 2, \dots, m \quad (4-2)$$

$$c_{jk} - c_{ik} + M(1 - x_{ijk}) \geq p_{jk}, \quad i, j = 1, 2, \dots, n; k = 1, 2, \dots, m \quad (4-3)$$

$$c_{ik} \geq 0, \quad i = 1, 2, \dots, n; k = 1, 2, \dots, m \quad (4-4)$$

$$x_{ijk} = 0 \text{ 或 } 1, \quad i, j = 1, 2, \dots, n; k = 1, 2, \dots, m \quad (4-5)$$

其中, 式(4-1)表示目标函数, 即要求最大完成时间最小化; (4-2)表示工艺约束条件决定的各种工件各操作的先后加工顺序; 公式(4-3)表示加工各工件的机器的先后顺序。式中, 符号 c_{ik} 和 p_{ik} 分别为 i 工件在机器 k 上的完成时间和加工时间; M 是一个足够大的整数; a_{ihk} 和 x_{ijk} 分别为指示系统和指示变量, 其意义为:

$$a_{ihk} = \begin{cases} 1, & \text{若机器 } h \text{ 先于机器 } k \text{ 加工工件 } i \\ 0, & \text{非上述情况} \end{cases} \quad (4-6)$$

$$a_{ijk} = \begin{cases} 1, & \text{若工件 } i \text{ 先于工件 } j \text{ 在机器 } k \text{ 上加工} \\ 0, & \text{非上述情况} \end{cases} \quad (4-7)$$

JSP 的求解远远复杂于旅行商问题 (TSP) 和流水车间调度问题 (Flow Shop)^[3], 其原因如下:

1. 调度解的编码复杂且多样化, 算法的搜索操作多样化;
2. 解空间容量巨大, n 个工件、 m 台机器的问题包含 $(n!)^m$ 种排列;
3. 存在工艺技术约束条件的限制, 必须考虑解的可行性;
4. 调度指标的计算相对算法的搜索比较费时;
5. 优化超曲面缺少信息, 通常复杂且存在多个分布无规则甚至彼此相邻极小。

4.2 Job Shop 调度的参数设计

4.2.1 基本流程图

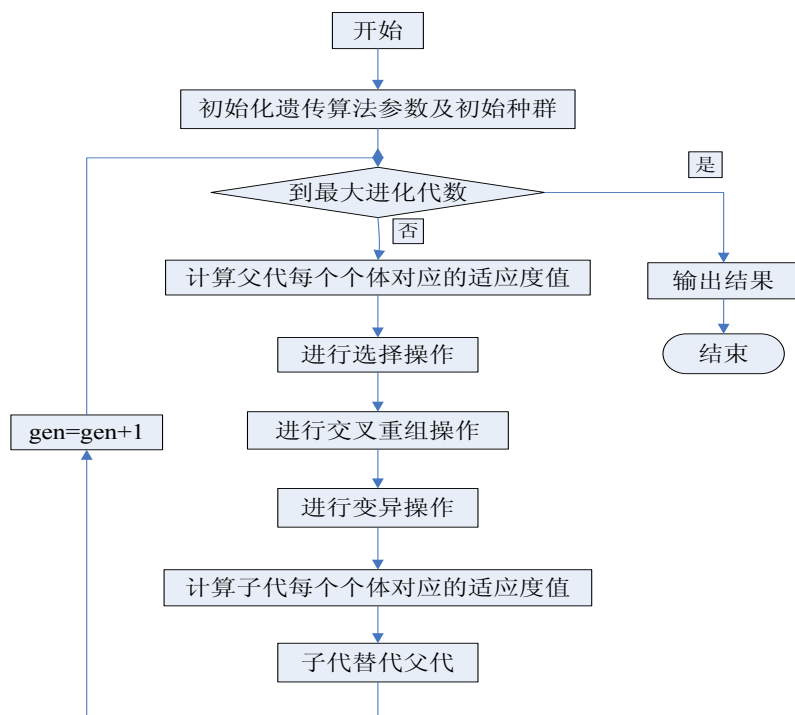


图 4-1 JSP 遗传算法流程图

4.2.2 编码方式的确定

编码问题是设计遗传算法的首要 and 关键问题。遗传算法的编码必须考虑“染色体”的合法性、可行性、有效性以及对问题解空间表征的完全性，求解 JSP 同样如此。鉴于 JSP 的组合问题特性以及工艺约束性，染色体的 Lamarkian 特性、解码的复杂性、编码的空间特性和存储量的需求是设计遗传算法编码通常要考虑的问题。目前提出的 Job Shop 调度编码方法有基于工序的编码、基于工件的编码、基于优先表的编码、基于优先规则的编码、基于完成时间的编码。本文采用基于工序的编码方法^[9]。

对于 m 个工件 n 台机器的 Job Shop 调度问题，基于供需的编码方法将每个染色体用 $m \times n$ 个代表工件的基因组成，是所有工件的一个排列，其中各工件均出现 n 次。一条染色体可以表示为 $X = \{x_1, x_2, \dots, x_{m \times n}\}$ ，其中 $x_k = i (1 \leq i \leq m, 1 \leq k \leq m \times n)$ 。如果 x_k 是染色体 X 中从 x_1 到 x_k 的第 $j (j \leq m)$ 个 i ，则表示工件 i 的第 j 个工序，个体 X 表示各工件各工序基于作业加工顺序的一个排列。

先介绍程序中的符号。

1. 加工时间矩阵 T ： $m \times n$ 矩阵，存储 m 个工件 $m \times n$ 个工序加工时间。
2. 调度方案编码矩阵 X ： $m \times n$ 矩阵，储存 m 个工件在 n 台机器上的加工先后顺序。

3. 遗传进化迭代次数 M : 取 50。
4. 种群规模 N (取偶数): 取 100。
5. P 向量: $1 \times n$ 的向量, n 个工序中, 每一个工序所具有的机床数目, 本论文描述的问题中每个工件都要经过 n 道工序的加工。

6. 交叉概率 P_c : 取 0.8。

7. 变异概率 P_m : 取 0.05。

编码算法如下。

步骤 1: 令 $k = 1$ 。

步骤 2: 按下述方法选择一个工件 I 。

①随机产生 $[1, N]$ 内的整数 I 表示一个工件;

②由 X 得到的工件 I 目前尚未加工的优先权最高工序对应的机器 $J_N(I, 1)$ 。如果 $J_N(I, 1) \neq 0$, 则转向步骤 3, 否则执行步骤 2。

步骤 3: 令 $s[k] = I, k = k + 1$ 。

步骤 4: 将 X 的第 I 行个元素依次左移一位, 尾部空出其位置填 0。

步骤 5: 如果 X 中有一个元素不为 0 则返回步骤 2, 否则结束。

解码算法如下。

步骤 1: 初始化, 即当前时刻所有机器上已加工工件的个数置 0, 各个工件从第一个工序开始, 令 $k(i) = 1, i = 1, 2, \dots, n$ 。

步骤 2: 依次取染色体中的基因 x_i , 计算工件 x_i 所对应的 $k(x_i)$ 道工序的最早开工时间。

4.2.3 适应度函数

对于 Job Shop 调度问题, 我们仍然采用目标函数的最小化最大完工时间作为适应度函数, 即令 C_{\max}^k 表示第 k 个染色体的最大完工时间, 那么适应值为:

$$Fit(v_k) = 1/C_{\max}^k$$

对于上述的调度问题, 全部工件的加工排序方式共有 $(n!)^m$ 。调度任务就是从这么多个排序中选择最优的或是“好”的一种调度, 大规模生产过程中工件的加工调度总数将会非常大。而遗传算法由于其隐含并行性能够有效收敛于目标解, 用于大规模调度问题。

4.2.4 遗传算子的设计

1. 选择算子

设由 N 个染色体组成群体 $C=\{x_1, x_2, \dots, x_i\}$, 选择算子中主要解决的问题是群体中染色体的选择概率 p_r

本文采用轮盘赌选择方法, 这在上一章中已经介绍过, 该方法根据每个染色体适应度值的比例来确定该染色体的选择概率或生存概率, 适应度值高的染色体被选入下一代的概率比较高, 因此, 容易继承比较好的染色体。

$$p_r(x_i) = \frac{f(x_i)}{\sum_{i=1}^n f(x_i)} \quad (4-8)$$

其中 $f(x_i)$ 为染色体 x 的适应值。

2. 交叉算子

在上一章已介绍过了常用的交叉算子, 本论文采用单点交叉, 即在染色体中随机产生一个交叉位置, 后代染色体继承父代 1 前半部分基因, 继承父代 2 的后半部分基因, 这样破坏个体性状和降低个体适应度值的可能性较小。

3. 变异算子

当交叉操作产生的后代适配值不再进化且没有达到最优时, 就意味着算法的早熟收敛。这种现象的根源在于有效基因的缺损, 变异操作在一定程度上克服了这种情况, 有利于增加种群的多样性, 本论文采用基本位变异法。

4.2.5 算法参数的设计

1. 根据编码方式选择二进制编码方法;
2. 迭代次数 $M=50$, 种群规模 $N=100$, 交叉概率 $P_c=0.8$, 变异概率 $P_m=0.05$;
3. 目标函数为最小化所有工件最大加工时间, 即所有工件加工完成的总时间;

4.3 Job Shop 调度的 Matlab 实现

本论文利用 Matlab 软件^[10]实现遗传算法设计。MathWorks 公司推出的一套高性能的数值计算和可视化软件, 它集数值分析、矩阵运算和图形显示于一体, 构成一个方便的、界面友好的用户环境。本论文对两个典型的 FT 类问题^[3]FT06 和 FT10 进行测试。

表 4-1 FT 类问题的研究

工件	机器	问题	最优 Maksepan	花费时间/s	最早取得最优值得研究者
6	6	FT06	55	41	Balas(1969)
10	10	FT10	930	6420	Lageweg(1984)

设置参数如下：进化迭代次数 $M=50$ ，种群规模 $N=100$ ，交叉概率 $P_c=0.8$ ，变异概率 $P_m=0.05$ 。对每种问题运行五次。

表 4-2 6 个工件 6 台机器的加工工序和时间

—	1	2	3	4	5	6
J_1	M_3 1	M_1 3	M_2 6	M_4 7	M_6 3	M_5 6
J_2	M_2 8	M_3 5	M_5 10	M_6 10	M_1 10	M_4 4
J_3	M_3 5	M_4 4	M_6 8	M_1 9	M_2 1	M_5 7
J_4	M_2 5	M_1 5	M_3 5	M_4 3	M_5 8	M_6 9
J_5	M_3 9	M_2 3	M_5 5	M_6 4	M_1 3	M_4 1
J_6	M_2 3	M_4 3	M_6 9	M_1 10	M_5 4	M_3 1

表 4-3 6×6 Job Shop 调度问题的实验结果

次数	1	2	3	4	5
完工时间	56	57	55	58	58
电脑运行时间/s	40	39	45	39	40

表 4-4 10 个工件 10 台机器的加工工序和时间

—	1	2	3	4	5	6	7	8	9	10
J_1	M_1 29	M_2 78	M_3 9	M_4 36	M_5 49	M_6 12	M_7 62	M_8 56	M_9 44	M_{10} 21
J_2	M_1 43	M_3 90	M_5 75	M_{10} 11	M_4 69	M_2 28	M_7 46	M_6 46	M_8 72	M_9 30
J_3	M_2 91	M_1 85	M_4 39	M_3 74	M_9 90	M_6 10	M_8 12	M_7 89	M_{10} 45	M_5 33
J_4	M_2 81	M_3 95	M_1 71	M_5 99	M_7 9	M_9 52	M_8 85	M_4 98	M_{10} 22	M_6 43
J_5	M_3 14	M_1 6	M_2 22	M_6 61	M_4 26	M_5 69	M_9 21	M_8 49	M_{10} 72	M_7 53
J_6	M_3 84	M_2 2	M_6 52	M_4 95	M_9 48	M_{10} 72	M_1 47	M_7 65	M_5 6	M_8 25
J_7	M_2 46	M_1 37	M_4 79	M_3 31	M_7 88	M_6 74	M_{10} 48	M_9 19	M_8 36	M_5 32
J_8	M_3 31	M_1 86	M_2 46	M_6 74	M_5 32	M_7 88	M_9 19	M_{10} 48	M_8 36	M_4 79
J_9	M_1 76	M_2 69	M_4 76	M_6 51	M_3 85	M_{10} 11	M_7 40	M_8 89	M_5 26	M_9 74
J_{10}	M_2 85	M_1 13	M_3 61	M_7 7	M_9 64	M_{10} 76	M_6 47	M_4 90	M_5 52	M_8 45

表 4-5 10×10 Job Shop 调度问题的实验结果

次数	1	2	3	4	5
完工时间	1043	1029	1061	1028	1032
电脑运行时间/s	135	165	151	166	161

表 4-2 和表 4-4 中, (M_i, j) 表示在第 i 个机器上加工 j 时间。表 4-3 和表 4-5 是实验结果, 由于每次运行程序得到的调度排序可能不同, 所以对于规模较小的问题容易得到最优或次优值, 对于规模较大的问题, 一般不易得到最优结果, 而且最后的完工时间也可能不同。

4.4 本章小结

本章首先介绍了典型的 Job Shop 调度的描述和求解方法, 然后对遗传算法参数进行设计, 最后对两类典型的 FT 类问题 FT06 和 FT10 进行测试说明算法的有效性和可行性。

第五章 结论与展望

遗传算法是模拟生物在自然环境中的遗传和进化过程而形成的一种自适应全局优化概率搜索算法,能处理传统搜索方法难于解决的复杂和非线性问题,可广泛应用于函数优化、组合优化、车间调度问题、图像处理、机器学习、自适应控制、规划设计和人工生命等领域,是 21 世纪有关智能计算中的关键技术之一,但遗传算法在理论与应用技术上还存在一些不足和缺陷,其主要表现为易于陷入局部最优解。

5.1 结论

本论文对车间调度(Job Shop)相关问题进行了详细地描述,建立数学模型,使用 Matlab 强大的数值计算能力和众多的库函数编写算法,对两类典型的 FT 类问题 FT06 和 FT10 进行了测试,得出以下结论:

1. 车间调度问题尤其是 Job Shop 调度问题是一类 NP 完全问题,目前还未找到可以精确求得最优解的多项式时间算法,因而以遗传算法为主的邻域搜索算法在车间调度领域受到了广泛重视和研究,本论文就是对最一般的 Job Shop 问题进行研究。
2. 车间调度问题使用遗传算法研究,有别于其他问题的特点,对参数的设置不同,得到的调度结果也不相同,尤其交叉概率、迭代次数、种群规模对结果影响较大。

5.2 展望

Job Shop 是最一般的调度问题,是车间调度领域的典型代表,而遗传算法是邻域搜索智能算法的代表,因此基于遗传算法的 Job Shop 具有重要的理论意义和工程价值。然而仍有大量工作需要深入和完善:

1. 对遗传算子的特性分析与结构设计有助于更好利用遗传算法解决 Job Shop 问题,而这方面的研究目前还不够深入,因此对于有很强工程背景的 Job Shop 问题,这无疑是一个很重要的研究方向。
2. 将 Job Shop 的数学模型现实化,动态调度的研究尤其值得重视,并将已有的理论成果推广到实践工程领域,用实践来检验技术的潜力,这不仅能够推动先进生产

的发展,而且将促使优化技术的进一步完善。同时,鉴于遗传算法对生产技术人员较为陌生,开发相关的实用化集成化调度平台和优化技术,无疑具有很强的工程意义。

3. 鉴于国际学术界对混合系统与算法的日益重视,混合优化策略以及相应仿真环境的开发无疑是一个值得关注的研究方向,尤其是将已有各种调度方法的优点与遗传算法有效融合^[11]。

总之,随着计算技术、优化技术、数学理论和计算机硬件的发展,遗传算法、Job Shop 本身以及他们的结合研究将得到更完善的解决,研究与应用领域将更宽广,理论研究将更系统化,更接近于实际,并朝集成化、实用化^[12]、最优化^[13]、多目标化、动态化、规模化发展。

参考文献

- [1] Davis L D. Handbook of Genetic Algorithms. Van Nostand Reinhold, 1991.
- [2] 刘飞等编著. 《CIMS 制造自动化》. 机械工业出版社, 1997.
- [3] 王凌编著. 《车间调度及其遗传算法》. 清华大学出版社, 2003 年 5 月.
- [4] 应保胜, 张华, 杨少华. 敏捷制造车间布局优化的启发式算法. 计算机集成制造系统. 2004. 10 (8) .
- [5] Holland J H. Adaptation in Nature and Artificial Systems. MIT Press, 1992
- [6] 周明, 孙树栋编著. 《遗传算法原理及应用》. 国防工业出版社, 1999 年 6 月.
- [7] 何燕. 基于遗传算法的车间调度优化及其仿真, [硕士论文]. 武汉理工大学 2006 年 5 月.
- [8] 花植, 郭剑毅, 王海雄, 安鑫. 基于遗传算法的车间调度原型系统研究及应用. 云南大学学报 (自然科学版), 2009.
- [9] 王龙生, 叶文华. 一种新型的作业车间调度算法的研究与实现. 机械科学与技术出版社. 2004 年第 23 期.
- [10] 王沫然编著. 《MATLAB 与科学计算》. 电子工业出版社, 2003 年 9 月.
- [11] 黄孔亮, 雍正止. 一种求解作业车间调度问题的协同进化算法. 深圳大学学报理工版. 2004 年 10 月, 第 8 期
- [12] 吴序一, 伍晓宇. 非量产模式下车间调度的改进遗传算法. 深圳大学学报理工版, 2006 年 7 月, 第 23 卷第 3 期.
- [13] 潘全科, 孙志俊, 朱剑英. 基于遗传算法的作业车间调度优化. 南京航空航天大学. 2002 年 6 月, 第 31 卷第 3 期.