

摘 要

切削用量的优选,对于保证加工质量、降低加工成本和提高劳动生产率来说,具有重要的意义。在机床、刀具和工件等条件一定的情况下,应充分发挥机床和刀具的功能和作用,以获得生产的最大效益。若切削用量选择得当,就可最大限度地挖掘出生产潜力,这就是切削用量的最优化。本论文研究的目的就在于探索将遗传算法应用于车削用量的优化,以获取最大生产率的理论和方法。

论文首先提出了本课题的来源和研究背景,指出了研究的意义、应用前景和本文的工作。文章简单介绍了优化算法的基础知识,对车削用量优化问题进行了理论分析,确定了切削用量的目标函数和其约束条件,并简单介绍了具有全局随机搜索能力的解决问题的工具——遗传算法,对遗传算法的原理、操作及应用进行了说明,为后面模型的计算打下了基础。随后,在一定的假设前提下建立了切削用量优化的数学模型,将遗传算法应用于车削用量优化的问题做了一些探索性的工作,进行了有益的尝试和分析。结果表明本文算法比较优越,实例应用和数值分析说明了该遗传算法在解决这类问题中的有效性和实用性,对生产实际有一定的指导意义,具有一定的社会和经济价值。最后,论文对全文的工作做了总结,分析了存在的问题和缺陷,并对今后的研究方向作了展望。

关键词: 车削用量; 优化设计; MATLAB; 遗传算法; 最大生产率

Abstract

To guarantee the processing quality, reduce processing costs and improve labor productivity, optimizing cutting parameters chosen is of great significance. In such condition, when the machine tools, cutting tools and work pieces conditions is in certain circumstances, we should give full play to the function of machine tools and cutting tools, to achieve the greatest production efficiency. If cutting parameter choices properly, can maximize the production potential, namely cutting parameters optimization. The purpose of this paper is to explore and research into the theory and methods of applying genetic algorithms to optimize Cutting parameters in turning to maximize productivity.

The paper first proposes the research origin and the background of this topic, have pointed out this research significance, the application prospect and the work of this article. The article briefly introduced the elementary knowledge of optimization algorithms, carried on the theoretical analysis to the question of optimization to cutting parameters in turning, Determined the objective function of Cutting parameters and its restraint conditions, and introduced a simple tool that has random search capability for solving problems ----- genetic algorithms, explained the genetic algorithm theory, operation and application, these built the foundation for the behind model calculation. Subsequently, under the certain supposition premise, established the mathematical model of the cutting parameters optimizes, did some exploratory work, carried on the beneficial attempt and the analysis that applied the genetic algorithms to optimize the Cutting parameters in turning, The result indicated the calculation method of this article will be superior, examples of the application and numerical analysis described the validity and the practicability of the genetic algorithm in solving such issues, and of a certain significance in the actual production, a certain degree of social and economic value. Finally, summed up the overall works of the papers, analyzed the problems and shortcomings, and the direction of a vision of the future research work.

Key word: Cutting parameters in turning; Optimized Design; MATLAB; Genetic algorithm Calculation; Maximum productivity

第 1 章 绪论

1.1 课题研究的背景和来源

位于川西平原、景色优美的德阳，因重大技术装备而闻名。20 世纪 50 年代，国家将德阳定位为我国的“母机工业基地”，先后在这里建成了中国第二重型机械集团公司、东方电机股份有限公司和东方汽轮机厂等大型国有骨干企业，这批企业在全国装备制造业中占有重要地位。近几年来，德阳市委、市政府抓住国家产业政策调整和西部大开发的历史机遇，重新审视传统装备业的优势和劣势，提出用高新技术和先进适用技术改造传统产业，创造德阳重大技术装备产业新优势的发展思路。经过 3 年多的努力，二重、东电、东汽“三大厂”不断发展壮大，一大批中小配套企业稳步发展，德阳重大技术装备制造业的整体实力显著增强。通过一手抓龙头、一手抓配套，德阳的重装龙头企业和中小企业得到了相得益彰的发展，形成了以三大厂为核心的德阳重装基地制造业企业集群，产业集中度和专业化水平大幅度提高，装备工业产业链已经形成。

机械制造业是德阳市工业经济中第一大主导产业。据了解，我市目前共有各类机械加工企业 200 多家，其中有 100 余家上规模企业，发展势头良好。

2002 年 1 月 30 日，市委书记、市人大常委会主任李成云在市科技顾问团召开的相关研讨会上明确提出：“要把德阳建设成为重大技术装备制造业基地。”由此拉开了我市打造重大技术装备“航空母舰”的大幕。2002 年春天，德阳市委、市政府邀请 100 多名省内外专家、学者和技术骨干，走进重装基地进行深入调研，在此基础上形成了德阳用高新技术提升传统产业的改革与发展方案。德阳市紧锣密鼓地实施了二重、东电、东汽等国有装备制造业企业内部改革，帮助企业分离社会职能，建立现代企业制度，做强主业，做大辅业。以装备制造业重点项目为纽带，德阳与清华大学、华中科技大学、西南交通大学、四川大学等国内外大专院校、科研院所合作，实施人才软性流动，开展科技攻关，建立该市装备制造业人才培养体系。德阳还从经济结构优化和加大技改投入方面入手，为可持续发展奠定了坚实的基础。同时，通过创新人才激励机制来调动各类人才的积极性和创造性，激活人才的聪明才智，进一步把德阳厚重的人才优势转化为技术优势。2003 年，省政府《关于加快我省重大技术装备制造业发展的意见》出台，德阳被纳入“1+8”工程。国

家科技部也把德阳列为“用高新技术改造提升传统装备制造业试点城市”和“制造业信息化工程重点城市”。

近两年来,在省委、省政府的领导下,我市通过整合资源,改造提升,发挥比较优势,完善和延伸产业链条,做大做强重大技术装备制造业,取得了明显成效。二重、东电、东汽分别与 60 多家中小企业签订了协作协议,成立了中小企业担保中心,为 72 家企业提供了 1.63 亿元的贷款担保。绵竹市把支持东汽发展作为地方经济发展的重大战略之一,充分利用东汽高新技术、人才资源,建立了绵竹机械加工园区。目前,绵竹市机械加工企业已发展到 49 家,仅 2003 年就新增 22 家。旌阳区以二重、东电、东汽、东电工为依托,促进机械制造、电线电缆、电子器件、工业泵和开关、汽车配件等 5 个相关产业的发展。目前,收入超过 100 万元的机械加工企业达到 90 家以上,其中有 20 余家直接为三大厂配套生产。产业群的形成,吸引了大量民间资金投入装备业,2003 年,德阳市投入到重装基地建设的民营资本超过 2 亿元。德阳依靠高新技术做大做强装备制造业,有力地促进了全市工业经济快速增长。德阳市市长方小方介绍说,2003 年,全市 102 家规模以上机械行业工业企业实现工业总产值 97.3 亿元,同比增长 29%;实现销售收入 96.2 亿元,增长 32%;实现利税 8.6 亿元,增长 46%。德阳的经济社会发展由此获得了强劲的动力。

2005 年,党和国家领导温家宝、贾庆林、黄菊到德阳视察时,对德阳重装基地建设作了很重要的指示。市委书记李成云就此提出:我们要抓住装备工业前所未有的发展机遇,把德阳建成真正意义上的中国重大装备制造业基地。政府和企业已经认识到:要靠高新技术占领市场制高点,占领了技术的制高点,就能占领市场的制高点;先进生产力中,科学技术是最重要的要素。所以近年来,德阳正在把技术创新作为谋求重装发展的源动力。

金属切削加工机床是工业生产中最重要装备设施,自从有了金属切削加工,就出现了金属切削用量的选择问题。随着社会的进步,生产力的发展,特别是金属切削机床的问世与改进和对刀具深入的研究,人们逐渐把包括切削用量选择在内的经验加以总结,形成文献,以供使用时参考。美国工程师泰勒发现了切削速度与刀具耐用度之间的关系——即著名的泰勒公式以后,人们开始用数学分析实际经验数据选择合理的切削用量。两次世界大战也大大的刺激了金属切削加工,工程师们人员对金属切削用量的选择问题更加关注。工业院校在金属切削原理课程中也专门讨论这个问题。经验法、查表法、

计算法及图解法是选择切削用量的传统方法。50 年代以来,兴起了电子计算机技术革命,人们借助于计算机进行切削用量优选并已取得了进展,更有人引入系统工程及人工智能的理论,应用专家系统来解决切削用量的选择问题,为低耗、高效、高度现代化金属切削加工开辟了一个广阔的前景。

而众所周知,德阳加工设备以重型加工、粗加工设备为主。而机械行业重型加工中,是以表皮去除,快速去切除加工余量为主要目的的,所以提高效率是关键。但是由于受到生产装备和生产任务的技术约束,往往不允许任意提高切削用量,所以切削用量的选择至关重要。《基于遗传算法的最大生产率的车削用量优化》正是在这样的情况下提出的德阳市的一个市级课题。

1.2 国内外研究状况综述

1.2.1 切削用量选择的研究现状

切削用量的优选,即选择合理的切削深度 a_p 、进给量 f 和切削速度 v 参数,对于保证加工质量、降低加工成本和提高劳动生产率都具有重要的意义。在机床、工具和工件等条件一定的情况下,切削用量的选择具有灵活性和能动性。应充分发挥机床和工具的功能,以取得生产的最大效益。如切削用量选择得当,就可最大限度地挖掘出生产潜力;倘若选择不当,则会造成很大浪费或导致生产事故。可是,迄今为止,有不少工厂在生产中选择切削用量仍处于经验选定状态,完全不能适合切削用量的最优组合,所以切削用量的优选是切削加工理论与实践的重要课题。

在 20 世纪六十年代,人们利用了数据库技术来满足提高加工效率、降低成本、提高质量和管理水平,提出了建立切削数据库。在工程材料牌号繁多,各种新材料不断出现的今天,切削数据库所起的作用尤其显著。工厂只要接通电话或传真都可向数据库咨询或索取所需数据,如装有电子计算机终端机,还可以通过电话线与数据库接通,用户可以自行查找所需的数据。工厂当接到一种新材料制成的工件,自己不必做切削实验求取切削数据,就可以很迅速地从数据库中获得所需的数据和切削用量。因此世界主要工业发达国家纷纷建立了切削数据库,并取得了巨大的技术经济效益。如美国,1964 年成立了可切削性数据中心,建立了计算机管理的切削加工数据库,如今已在磁盘储存器中储存三万多份数据文件。它所取得的社会经济效益是十分惊人的,据估计 1964-1974 年十年间就为工业部门节省了一亿四千万美元,并大大提高了美国切削加工的工艺水平。国内对此也进行了长期的研究工作,取得了

许多成果，建立了多种切削参数计算经验模型，和各种切削数据库。但是由于市场需求多样化，工件的外形越来越复杂并且更换很快，新材料不断涌现，而各个工厂的技术水平和生产习惯不同，具体到每个工件的切削条件更是千差万别，很难用一个在特定条件下建立的模型或切削数据库来覆盖所有的应用场合。在工厂实际应用时，往往需要技术人员的经验和知识来弥补这一不足，遗憾的是工厂里常常缺乏这方面的人才，这个问题在我国工厂尤为突出。以至于在生产中切削参数往往都是参考手册规范或经验公式确定，而这些规范都是在特定条件下的试验结果，因此选择参数在一定条件下使用效果好，换一个工厂甚至另一台机床往往就不好用了。另外，机床操作者的经验和水平也是关键的影响因素，在我国由于缺乏技术素质较高的工人和技术人员，因此相同的设备不同人使用的情况下生产效率会相差几倍甚至几十倍。如何优化选择切削用量成为提高机械加工质量的关键。

切削加工中，切削用量的合理选择及优化问题，关系到加工生产率、经济性、加工精度、加工表面质量以及零件的使用寿命及可靠性问题，是提高金属切削效益的主要途径之一。研究切削用量的目的就是要结合生产实际，综合运用所学金属切削的基本规律，在保证加工质量的前提下，合理的利用机床与刀具，在现有条件下突破薄弱环节，进一步提高金属切削效益。当前，随着机械制造业的迅速发展，切削用量的优化已成为一项重要的技术经济措施，是实现切削过程优化的关键。切削用量的优化，可以充分地利用现有的生产资料与设备条件，实现切削效益最优化。

总之，金属切削加工中，包括三要素：切削深度 α_p 、进给量 f 和切削速度 v ，该三要素是切削加工中最活跃的因素。

在加工任务、机床和刀具等条件确定的情况下，要想在满足零件加工质量的前提下提高加工效率、降低成本，如何确定最佳的切削用量是解决问题的关键。长期以来，切削用量数据大都是凭经验或从有关手册查取来确定的，但考虑到实际生产中所使用的机床、刀具和被加工零件等具体条件的多变性，这样确定的切削用量往往难以是某一具体情况下最佳的一组切削用量。在优化理论和计算机技术迅速发展的今天，计算机在切削加工中的应用日益增多，我们可以利用计算机方便、迅速、有效地对切削用量进行优化设计，从而解决这一问题。首先，根据具体要求，确定出切削用量的最优目标，并建立起相应的目标函数；其次，再充分考虑到生产中的各项约束条件，建立起优化

数学模型；然后，采用相应的优化方法，借助于计算机进行优化运算，得到的结果就是满足约束条件的目标函数的最优方案，也即是具体生产条件下满足加工要求的最佳的切削用量数值。

1.2.2 遗传算法研究的进展

遗传算法(Genetic Algorithms) (GA)的产生归功于 Michigan 大学的 John Holland 在 20 世纪 60 年代末期的开创性工作，其本意是在人工适应系统中设计一种基于自然演化原理的搜索机制，从试图解释自然系统中生物的复杂适应过程入手，模拟生物进化的机制来构造人工系统的模型。随后经过 30 余年的发展，取得了丰硕的应用成果和理论研究的进展，特别是近年来世界范围形成的进化计算热潮，使遗传算法受到广泛的关注。遗传算法以一种群体中的所有个体为对象，并利用随机化技术指导对一个被编码的参数空间进行高效搜索。其中，选择、交叉和变异构成了遗传算法的遗传操作；参数编码、初始群体的设定、适应度函数的设计、遗传操作设计、控制参数设定五个要素组成了遗传算法的核心内容。作为一种新的全局优化搜索算法，遗传算法以其简单通用、鲁棒性强、适于并行处理以及高效、实用等显著特点，在各个领域得到了广泛应用，取得了良好效果，并逐渐成为重要的智能算法之一。

人们对遗传算法兴趣的日益增长有两个背景，其一是工程领域，不断涌现超大规模的非线性系统，在这些系统的研究中存在着大量的经典优化方法所不能有效求解的优化问题，其二，遗传算法本身就是一种模拟自然演化这一学习过程的求解问题的方法，它能以独立或与其它方法相结合的形式用于智能机器学习系统设计。从 1985 年在美国卡耐基·梅隆大学召开的第一届国际遗传算法会议(International Conference on Genetic Algorithms: iCGA' 85)至今，遗传算法作为具有系统优化、适应和学习的高性能计算和建模方法的研究渐趋成熟。近年来，遗传算法在应用研究方面取得了重要的成果。以遗传算法为桥梁，许多原先毫不相干的学科也开始相互交叉、相互渗透，使得跨学科的研究日趋活跃。

为解决各种优化计算问题，人们提出了各种各样的优化算法，如单纯形法、梯度法、动态规划法、分枝定界法等。这些优化算法各有各的长处，各有各的适用范围，也各有各的限制。遗传算法是一类可用于复杂系统优化计算的鲁棒搜索算法，由于其运算简单和解决问题的能力而被广泛应用到

众多的领域。理论上已经证明,遗传算法能从概率的意义上以随机的方式寻求到题的最优解。但另一方面,应用实践表明,在遗传算法的应用中也会出现一些不尽人意的问題,这些问題中最主要的是它容易产生早熟现象、局部寻优能力较差等。并且一般来说,对很多问題而言,传统遗传算法的求解效果往往不是解决这个问题的最有效的方法,它比专门针对该问题的知识型启发算法的求解效率要差,虽然这种知识型启发算法并不能保证一定能够找到问题的最优解。另外,遗传算法也无法避免多次搜索同一个可行解,这也是影响遗传算法运行效率的一个因素。

另一方面,梯度法、爬山法、模拟退火算法、列表寻优法等一些优化算法却具有很强的局部搜索能力,而另一些含有问題与相关知识的启发式算法的运行效率也比较高。可以预计,在遗传算法的搜索过程中融合这些优化方法的思想、构成一种混合遗传算法(Hybrid Genetic Algorithm)是提高遗传算法运行效率和求解质量的一个有效手段。应用研究表明,将遗传算法与问题的特有知识集成到一起所构成的混合遗传算法,为继续提高遗传算法的搜索性能提供了新的思路。

我国有关遗传算法的研究,从 20 世纪 80 年代以来一直处于不断上升的趋势,特别是近年来,遗传算法的应用在许多领域取得了令人瞩目的成果,该类研究获得不同渠道的经费资助比例也在逐年上升。武汉大学的刘勇、康立山等于 1995 年出版了《非数值并行算法(第 2 册)—遗传算法》;周明、孙树栋于 1999 年出版了《遗传算法原理及其应用》;同济大学的王小平、曹立明于 2000 年出版了《遗传算法理论、应用与软件实现》。国内有关遗传算法的 BBS 电子公告牌有国家智能中心曙光站 bbs.neic.ac.cn、北京大学阳光创意站 hbs.pku.edu.cn、清华大学水土清华站 bbs.net.tsinghua.edu.cn、西安交通大学兵马俑站 bbs.xanet.edu.cn、研学论坛 <http://bbs.matwav.com> 等。

总之,遗传算法在理论上已经借鉴生物进化理论及遗传学机理形成了一套较为完整的算法体系,然而在实践上还有很多问題有待于进一步研究、探讨和完善。它们主要反映在以下方面:(1)控制参数选择问題; (2)成熟前收敛问題; (3)遗传算法的性能评价问題; (4)遗传算法的适应性问题; (5)混合算法问題; (6)从生物进化或遗传工程中不断汲取新的知识获得新的启发,从而对现有遗传算法讲行改进或提出新的新的算法等。

遗传算法当前的主要研究方向情况如下:

遗传算法自提出,特别是八十年代中期以来,已得到广泛研究与应用。

其研究的内容大体集中在以下几个方面:

1. 有关算法随机搜索机理的研究

遗传算法是通过作用于一个初始种群, 而循环执行复制、杂交、变异及选择过程的随机迭代, 故阐明如此简单的循环操作如何有效搜索整个编码空间以达到全局优化之目的有特别重要的意义。在这方面, J. H. Holland 等人发展的所谓“模式(schema)理论”引人注目。一个模式是指编码空间(即所使用的染色体的全体。当应用 L 位二进制串编码时, 该空间为一个具有相同构形(configuration)的编码的子集。所谓具有相同构形是指: 这个子集中诸编码串在某些位上具有相同的码值。例如, 在编码空间中, 给定一个模式, 一个编码称为与该模式相匹配, 如果在模式的确定位上, 此编码的值与模式的值相同, 例如, 编码(1 0 0 0 1 0 1 1 0 0)、(1 0 1 0 1 0 1 0 0 0)均与上述所指定的模式相匹配。任一给定编码必与许多模式相匹配。模式理论的核心在于: 遗传算法能够有效搜索的根本原因是, 它充分利用了模式所描述的编码之间的相似性, 虽然算法仅作用于 N 个编码组成的种群, 但这 N 个编码实际上包含阶个模式的信息。这一性质常被称作是遗传算法的隐含并行性(implicit parallism)。

模式理论可以较好地解释遗传算法的搜索机制。对于任一模式, 定义它的阶为模式中确定码值的个数, 而定义它的长度为模式中第一个确定码值位与最后一个确定码值位之间的差, 则 Holland 证明了如下的模式定理: “具有短长度的、低阶的、适应性在群体平均之上的模式将在遗传算法中以指数增长率在子代中被采样。

2. 有关算法的全局最优性(或收敛性)研究

生物进化的“趋势向上”性似乎应蕴含遗传算法的最终收敛性, 这一研究的目的在于从理论上对这一事实给出证明。与算法收敛性紧密相关的一个问题是遗传算法的过早收敛(premature convergence)。它出现在算法还未达到全局最优情形而不再产生适应性更强的后代。研究表明: 遗传算法的过早收敛主要由杂交算子引起。在模式空间中存在大量所谓的早熟集(premature set)(即在选择与杂交算子复合作用下的不变集), 而在杂交算子作用下, 遗传算法总以非零概率“撞入”早熟集。由于早熟集的吸收性, 从而使算法产生过早收敛。

3. 有关染色体编码格式的讨论

遗传算法的作用对象是优化变量的染色体编码(即与变量的某种离散化

近似)。通常认为:采用编码方式(特别是二进制编码)有以下优点:a)可很好地指导搜索,使得有关某种结构的个体容易生存,以产生适应性更强的后代;b)使算法具有隐含并行性,使在相对少量的种群上进行的操作实质上隐含着大范围的搜索。采用编码格式的缺陷是:由于连续问题化归到组合问题求解,使算法的求解精度受到很大影响。这些优点和缺点当遗传算法用于(组合)优化问题的求解时均有明显体现。

上述有关遗传算法研究的几个方面应该说还是相当初步的。对它的进一步深入探讨正构成当前计算智能研究的“热点”。概括地说,当前有关遗传算法的研究兴趣主要集中在以下几个方面:

(1) 有关算法的理论基础:这特别包括遗传算法的收敛性、收敛速度估计、过早收敛的机理探索与预防、杂交算子的几何意义与统计解释、参数设置(如种群规模、编码长度等)对算法效率的影响等方面。我们认为:有关算法的收敛速度估计是当前特别值得花大力气加以探讨的问题(目前尚无任何结果),因为它能从理论上对遗传算法的任何修正形式,提供判别标准,以指明改进遗传算法效能的正确方向。

(2) 有关算法与其它优化技术的比较与融合:作为已知较为成熟的全局优化算法—模拟退火方法,它与遗传算法的结合已有不少研究。然而,充分利用遗传算法的大范围群体搜索性能与已知局部优化方法的快速收敛性来产生有效全局优化方法,研究仍不多见。人们感到,这种整体搜索策略与快速局部优化方法的混合是从根本上提高遗传算法计算效能的措施。这方面还需要做大量的理论分析与实验研究。

(3) 有关遗传算法的并行化研究遗传算法的群体、随机搜索特征使它有明显的可行并行化特性。设计它的各种并行化执行策略及其建立相应并行算法的理论基础均是具有重要意义的工作。

(4) 遗传算法在人工神经网络、机器学习、复杂组合优化问题求解、金融市场分析和预测、专家系统等领域已有广泛应用。一般说,根据具体应用领域来有效改进遗传算法是可能的和实际的(而相反,泛泛地对一般问题研究显得极其困难)。所以,针对具体应用问题,深化研究遗传算法(如参数设置、遗传算子构造等)是当前特别值得提倡的上作。

1.3 本文研究的主要内容和工作的

本研究课题将探索以最大生产率的车削用量优化模型为目标,试图引入

遗传算法对切削用量参数进行优化。本文还以实际生产中的切削加工为原始数据作实例进行优化计算，并分析了优化结果。

全文共分为六个部分：第一部分为绪论，主要讲述了本文研究的背景、研究目的意义、国内外研究动态和主要研究内容等；第二部分是切削用量的优化目标和目标函数，对切削用量的优化目标和目标函数进行了分析；第三部分为优化算法及 MATLAB 遗传算法理论概述，主要阐述了优化技术的发展的现状和种类，介绍了 MATLAB 遗传算法理论知识；第四部分是遗传算法的原理和实现过程，讨论了遗传算法的运算过程、主要因素、特点以及应用等，并介绍了 MATLAB 遗传算法与直接搜索工具箱 (GADS) 及使用方法；第五部分是基于遗传算法的最大生产率的车削用量优化研究与实现，阐述了建模的过程，并以实例进行运算分析；第六部分为结论及建议，对本文研究的主要成果进行总结，并对该方法的研究提出一些建议与展望。

第 2 章 切削用量的优化目标和目标函数

2.1 优化目标和目标函数分析综述

在机械加工中，当加工设备、工具和工件初步确定之后，切削用量的选择对提高生产效率、降低加工成本及提高加工质量等有直接影响，寻求较为合理的切削用量及有关参数是取得显著的技术经济效益的简便途径。对于切削用量的优化选择，需要确定一个明确的目标。根据生产实践，该目标应该与企业需要完成的经济指标相一致，常用的优化目标有以下几种：

1. 最大生产效率；2. 最低成本；3. 最大利润。

加工装备（机床、刀具、夹具等）以及加工任务（工件的要求）已定的情况下，切削加工时可以控制的变量（优化的参数）是切削速度 v ，进给量 f 和切削深度 α_p ，即切削用量的三要素。从研究切削加工的基本规律入手，找出优化目标与可控变量的函数关系，即可建立起一个目标函数。

1. 最大生产效率

如以最大加工生产率为优化的目标，首先要分析单件工时的组成。此外，生产成本的高低也决定于加工时间的长短与生产费用的大小。车削一般轴类零件外圆表面时，完成一个工序所需要的工时（单件工时） t_w 由以下几部分组成：

$$t_w = t_m + t_{cl} * \frac{t_m}{T} + t_{or} \quad (2-1)$$

式中 t_m —工序的加工时间（即机动时间）；

t_{cl} —换刀一次所需的时间；

T —刀具耐用度，目前，高速钢车刀和焊接式硬质合金车刀的耐用度一般可取 3600s(60min)；

t_{or} —除换刀时间外其他的辅助工时。

设工件加工部分长度为 l_w ，工件转速为 n ，工件直径 d ，加工余量为 h ，则：

$$t_m = \frac{l_w * h}{n * f * \alpha_p} = \frac{l_w * h * \pi * d_w}{1000v * f * \alpha_p} \quad (2-2)$$

上式中分母部分代表单位时间金属的切除量 z_w 。粗加工时，此值越大，表示生产率越高。因而写成：

$$z_v = 1000v * f * \alpha_p \quad \text{mm}^3 / \text{s} \quad (2-3)$$

在切削用量三要素中，当加工余量不太大时，切削深度 α_p 主要取决于加工余量的大小，选择余地较小，因此这时切削用量的选择与优化的重点就放在切削速度 v 与进给量 f 上。式 2-2 可以简化为：

$$t_m = \frac{L}{n * f} \quad (2-4)$$

式子中， L ——刀具行程长度。

从理论上讲， t_m 越小，对各方面都比较有利。为此，若想使 t_m 最小，必须使工件转速与进给量的乘积最大。即 $n * f$ 为最大。所以，这就是以最大生产率为目标切削用量优化的目标函数，即：

$$Z = n * f = \text{Max} \quad (2-5)$$

2. 最低成本（最小的单件加工成本）

加工一个工件的工序成本为：

$$C = M * t_m + \frac{t_m}{T} * M t_{ct} + C_t * \frac{t_m}{T} + M * t_{ct} \quad (2-6)$$

式中 C ——一个工序的加工成本；

M ——该工序单位时间内所分担的全厂费用；

t_m ——切削时间；

C_t ——每次刃磨的费用，包括刃磨费用及刀具的折旧费。

只要将式子 2-2 及刀具耐用度 T 与切削用量间的关系式带入式子 2-6 中，经过数学推导，同样可以写出以最低成本为目标切削用量优化的目标函数。

此外，如从生产单件产品得到的收益考虑，渴求出利润率最大的加工条件。

3. 最大利润

(1) 单件利润 g (元/件)

$$g = S - C \quad (2-7)$$

式中 S ——单件所收加工费用

C ——单件加工成本

(2) 单位时间利润 (即单件利润率) G (元/件)

利润率可以根据产品的价格与生产成本的差价来计算。

$$G = \frac{S - C}{t_w} \quad (2-8)$$

式中 t_w ——所花费的时间

总之, 根据需要与具体情况, 可选择不同的优化目标。当要求提高生产效率, 以及在一些特定情况下, 例如该生产工序为生产上的短线, 需求量激增, 或库存缺乏, 生产急需时, 即使加工费用增加一些, 也可选用最大生产率或最短单件工时作为优化目标。在一般正常生产的条件下, 应以最低成本作为优化目标。

2.2 限制切削用量提高的约束条件

通过式 2-3 可见, 增大切削用量, 也就是增大单位时间内金属的切除量, 可以直接提高切削加工的生产率。由此可知, 切削用量的合理优化, 就是要在保证质量和降低成本的基础上, 采取措施, 使得 $1000v * f * \alpha_p$ 的乘积不但增大, 但是, 由于前面提到的因受到生产装备和生产任务的技术约束的原因, 往往不允许任意提高切削用量。

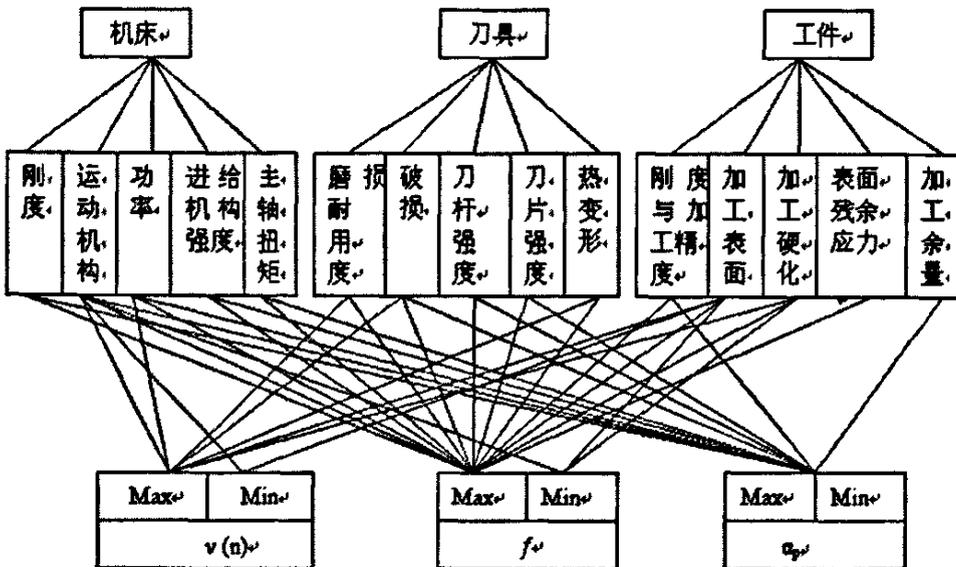


图 2-1 根据机床、刀具、工件三方面建立的约束条件

车削过程中的约束条件主要来自工艺系统 (如机床、刀具等) 的性能与

工件的技术要求（如加工精度与加工表面质量等）两个方面。技术上可能出现的约束条件很多。根据加工种类（车、铣、钻、刨、磨等）与加工性质（如粗、精加工、连续与断续加工等）的不同，主要的约束条件也可能不同，但可按机床、刀具与工件三个方面归纳为图 2-1 的关系。

由以上图形可以看出，粗加工时，金属切削量大，因此切削力大，主要的约束条件是工艺系统的刚性、机床仅给机构的强度等方面。精加工是以保证质量为主，因此主要的约束条件是加工表面的粗糙度与加工精度。其他还有一些约束条件与切削过程本身有关，顺利的切屑形成条件，积屑瘤区和颤振区的避开，保证良好的卷屑等等。可见，与切屑用量优化有关的所有约束条件，就其性质来看，广泛地涉及到切屑切屑过程物理学，机床运动学与加工系统经理学、动力学等等几方面的问题。

上述诸约束条件可以用不等式表示。以车削为例，可以列出以下主要的约束条件不等式：

1. 刀具的耐用度

车削时切削速度的计算公式为：

$$v = \frac{c_v * k_v}{60^{(1-m)} * T^m * \alpha_p^{x_v} * f^{y_v}} \quad m / s \quad (2-9)$$

当 α_p 与 f 已经选定后， v 主要受到代表刀具切削性能的刀具耐用度的限度。另一方面，切削速度 v 与工件直径 d_w 有关，因此必须满足下列不等式：

$$v = \frac{\pi d_w n}{1000} \leq \frac{c_v * k_v}{60^{(1-m)} * T^m * \alpha_p^{x_v} * f^{y_v}} \quad (2-10)$$

因此可以推导出：

$$n * f^{y_v} \leq \frac{318.31 c_v * k_v}{60^{(1-m)} * d_w * T^m * \alpha_p^{x_v}} \quad (2-11)$$

2. 机床的有效功率

切削功率：

$$P_m = F_z * v * 10^{-3} KW = 9.81 \times 60^{Z_{Fz}} * C_{Fz} * \alpha_p^{X_{Fz}} * f^{Y_{Fz}} * v^{(Z_{Fz}+1)} * K_{Fz} * 10^{-3} \quad (2-12)$$

增大切削用量三要素会增大切削功率，但后者必须小于、至少在短时内等于机床主轴上考虑了效率的传动功率，因此必须满足下列不等式：

$$P_m = F_z * v * 10^{-3} \leq P_E * \eta_m \quad (2-13)$$

式中 η_m ——机床的传动功率，一般 $\eta_m=0.75—0.85$ ，此处取 0.75；

P_E ——机床电动机的功率，查机床说明书 P_E 为 15KW。

故而可推导出：

$$n^{(Z_{Fz}+1)} * f^{Y_{Fz}} \leq \frac{101.94 * 1000^{(Z_{Fz}+1)} * P_E \eta_m}{60^{Z_{Fz}} * (\pi d_w)^{(Z_{Fz}+1)} * C_{Fz} * \alpha_p^{X_{Fz}} * K_{Fz}} \quad (2-14)$$

式中 n ——机床转速

d_w ——工件加工前直径。

X_{Fz} 、 Y_{Fz} 、 Z_{Fz} ——主切削力指数，具体数据见后面附表一。

3. 机床主轴的有效扭矩

切削时产生的扭矩必须小于或等于机床主轴系统所能传递的扭矩 M_s ， M_s 可由机床说明书中查得。它们之间的关系可一写成如下的不等式：

$$M = F_z * \frac{d_w}{2 \times 1000} = \frac{9.81}{2000} \times 60^{Z_{Fz}} * C_{Fz} * \alpha_p^{X_{Fz}} * f^{Y_{Fz}} * v^{Z_{Fz}} * K_{Fz} * d_w \leq M_s \quad (2-15)$$

因此可推导出：

$$n^{Z_{Fz}} * f^{Y_{Fz}} \leq \frac{203.87 \times 1000^{Z_{Fz}} * M_s}{60^{Z_{Fz}} * C_{Fz} * \alpha_p^{X_{Fz}} * \pi^{Z_{Fz}} * K_{Fz} * d_w^{(Z_{Fz}+1)}} \quad (2-16)$$

式中 Z_{Fz} 、 X_{Fz} ——主切削力指数。

M_s ——机床主轴系统所能传递的扭矩。

C_{F_z} —— 切削力系数。

K_{F_z} —— 切削力总修正系数。

4. 机床进给机构的强度

机床最薄弱的传动环节一般是进给机构中产生直线往复运动的齿轮轮齿的强度，如车床是拖板中与齿条相啮合的小齿轮，一般机床说明书都注明了最大容许切削力的大小。车削加工时作用于进给系统的切学力必须小于或等于该系统所容许的最大切削力。（例如：C620 型车床为 360kgf，CA6140 型车床为 500kgf）。因此可写成下面的不等式：

$$F = F_x + (F_y + F_z) * \mu_0 = KF_z \leq F' \quad (2-17)$$

式中 μ_0 为机床拖板与导轨间的摩擦系数，一般取 $\mu_0 = 0.1$ 。因此可推导出：

$$n^{Z_{F_z}} * f^{Y_{F_z}} \leq \frac{1000^{Z_{F_z}} * F'}{9.81 * 60^{Z_{F_z}} * K * C_{F_z} * \alpha_p^{X_{F_z}} * K_{F_z} (\pi d_w)^{Z_{F_z}}} \quad (2-18)$$

Z_{F_z} —— 主切削力指数。

F' —— 系统所容许的最大切削力。

X_{F_z} —— 主切削力指数，数据见后面附表一。

K_{F_z} —— 切削力总修正系数，数据见后面分析。

5. 工件的刚性

车削加工时，工件受到切削力 F_y 的作用，在半径方向产生一相对的位移 f' ，造成工件的直径误差为 $2f'$ 。

$$f' = \frac{F_y * l_0^3}{K_0 EJ} \quad (2-19)$$

式中 l_0 —— 工件的悬伸长度或装夹在两支承间的距离 (mm)。

K_0 —— 与工件装夹方式有关的系数。

E —— 工件材料的弹性模量。

J —— 工件断面惯性力矩，如直径为 d_w 的实心材料 $J = 0.05d_w^4$ 。由此可见，切削时作用在工件半径方向的切削力必须小于或

等于工件产生此先队唯一所容许的切削力。故而可写成如下不等式：

$$F_y \leq \frac{K_0 E * J * f'}{l_0^3} \quad (2-20)$$

由此可推导出：

$$n^{Z_{r'}} * f^{Y_{r'}} \leq \frac{1000^{Z_{r'}} * K_0 * E * J * f'}{9.81 * 60^{Z_{r'}} * C_{F_z} * \alpha_p^{X_{r'}} * K_{F_z} (\pi d_w)^{Z_{r'}} * l_0^3} \quad (2-21)$$

式中 X_{F_y} 、 Y_{F_y} 、 Z_{F_y} ——吃刀抗力指数。

K_0 ——工件装夹方式系数。

E ——弹性模量。

J ——工件断面惯性力矩。

f' ——车削时所容许的相对位移值。

K_{F_z} ——切削力总修正系数，数据见后分析。

l_0 ——工件的悬伸长度或装夹在两支承间的距离。

6. 刀杆的强度

切削时作用在刀具上的力应小于刀杆（刀片、刀齿）所容许的切削力。如车刀可看作一端固定的悬臂梁，受到弯曲、扭转、压缩的复合应力。如忽略 F_y 及 F_x 对车刀的作用不计时，一般可按主切削力 F_z 的弯矩计算。对矩形刀杆应满足以下不等式：

$$F_z \leq \frac{BH^2}{6l} \sigma_{bb} \quad (2-22)$$

式中 σ_{bb} 为刀杆的抗弯强度。

主切削力的大小： $F_z = 9.81 \times 60^{Z_{r_z}} * C_{F_z} * \alpha_p^{X_{r_z}} f^{Y_{r_z}} * K_{F_z}$

因此可推导出：

$$n^{Z_{r_z}} * f^{Y_{r_z}} \leq \frac{1000^{Z_{r_z}} * B * H^2 * \sigma_{bb}}{6l * 9.81 * 60^{Z_{r_z}} * C_{F_z} * \alpha_p^{X_{r_z}} * K_{F_z} (\pi d_w)^{Z_{r_z}}} \quad (2-23)$$

式中 X_{F_z} 、 Y_{F_z} 、 Z_{F_z} ——主切削力指数。

B ——刀杆宽度。

H ——刀杆高度。

K_{FZ} ——切削力总修正系数。

总结上述各式中各符号的系数和指数：

X_{FZ} 、 Y_{FZ} 、 Z_{FZ} ——主切削力指数。

X_{FY} 、 Y_{FY} 、 Z_{FY} ——吃刀抗力指数。

X_{FX} 、 Y_{FX} 、 Z_{FX} ——走刀抗力指数。

C_{Fz} 、 C_{Fy} 、 C_{Fx} ——切削力系数。

K_{FZ} 、 K_{FY} 、 K_{FX} ——切削力总修正系数。例如，其中：

$$K_{FZ} = K_{mFZ} * K_{\gamma FZ} * K_{kFZ} * K_{\lambda FZ} * K_{rFZ}$$

$$K_{Fy} = K_{mFy} * K_{\gamma Fy} * K_{kFy} * K_{\lambda Fy} * K_{rFy}$$

式中：

K_{mFZ} 、 K_{mFy} ——工件材料机械性质对切削力的修正系数。

$K_{\gamma F}$ ——前角对切削力的修正系数。

K_{kF} ——主偏角对切削力的修正系数。

$K_{\lambda F}$ ——刃倾角对切削力的修正系数。

K_{rF} ——刀尖圆弧半径对切削力的修正系数。

7. 加工表面粗糙度

当刀具的几何参数固定不变时，加工表面粗糙度主要与进给量有关。根据经验数据的数理统计与校验，可写出以下不等式：

$$f \leq K_R R_z^u \quad (2-24)$$

对 45 钢来说，当车刀的副偏角 $k'_r = 5^\circ \sim 15^\circ$ ，刀尖圆弧半径 r_c 与切削速度 V 不同时可确定上式中的 K_R 与 u 值。

8. 机床的最大与最小主轴转速

所选工件转速 n 应满足下列不等式：

$$n_{\min} \leq n \leq n_{\max} \quad (2-25)$$

9. 机床的最大与最小进给量

所选进给量应满足下列不等式：

$$f_{\min} \leq f \leq f_{\max} \quad (2-26)$$

10. 切削加工所容许的最大与最小切削深度

所选切削深度其最大值受到工艺系统刚度与加工余量等的限制，其最小值又受到必要的切屑形成条件与加工表面硬化层等的限制。即所选切削深度 α_p 应满足下列不等式：

$$\alpha_{p \min} \leq \alpha_p \leq \alpha_{p \max} \quad (2-27)^{[5][8]}$$

第 3 章 优化算法及 MATLAB 遗传算法理论概述

3.1 优化算法简介

3.1.1 优化技术的发展和现状

优化方法涉及的工程领域很广, 归纳而言, 最优化问题可分为函数优化和组合优化问题两大类, 其中函数优化的对象是一定区间内的连续变量, 而组合优化的对象则是解空间中离散状态。

所谓优化算法, 其实就是一种搜索过程或规则, 它是基于某种思想和机制, 通过一定的途径或规则来满足用户要求的问题的解。所以可以把最优化定义为对某些数学上定义的问题确定其最优解, 通常这些问题是一些客观现实的模型。最优化包括对各种问题最优性条件的研究, 数值求解方法的确定, 这些方法结构的研究以及方法在试验性条件下对现实问题的计算机试验。它具有非常广泛的实际应用领域, 同时最优化也可以作为纯数学的一个分枝来进行研究。

在 1940 年以前, 对多变量函数的数值最优化方法知之甚少。当时有若干最小二乘求解的方法, 以及在某些物理问题上应用的最速下降法之类的方法。

多变量问题的牛顿法是比较著名的, 对理论化学中关于变分问题的自适应域等问题也曾试图研究一些比较高级的方法。然而由于复杂, 有关的运算需要借助于大量的台式计算机。毋庸置疑, 计算机的发展极大的推进了最优化方法乃至整个数值分析领域的发展。

线性规划是最优化领域一个非常重要的分枝, 它是在本世纪 40 年代与 50 年代引入与发展起来的。然而在当时那些方法的应用领域相当窄, 在战后的一段时间内, 爬山法得到了发展, 这类方法由于不考虑所论问题的任何特殊结构而应用相当广泛。但是这后一类方法最初相当粗糙, 而且效率很低。1959 年发表的 W. C. Davidon 的一份报告可以说是最优化技术的一个转折点, 这个报告引入了后来所谓的变尺度方法。从那时起, 最优化技术有了迅速的发展, 现在对各种各样的问题都有了相应的求解方法。

随着 20 世纪 80 年代初期禁忌搜索、模拟退火、遗传算法和人工神经网络算法等优化算法的兴起, 人们对这些算法的模型、理论和应用技术等一系列问题进行了深入的研究, 并将这些算法称为现代优化算法。现代优化算法

的主要应用对象是优化问题中的难解问题。

3.1.2 优化算法的类别

就优化机制与行为而分,目前工程中常用的优化算法主要可分为:经典算法、构造型算法、改进型算法、基于系统动态演化的算法和混合型算法等。

1. 经典算法 包括线性规划、动态规划、整数规划和分枝定界等运筹学中的传统算法,其算法计算复杂性一般很大而只适于求解小规模问题,在工程中往往不实用。

2. 构造型算法 用构造的方法快速建立问题的解,通常算法的优化质量差,难以满足工程需要。譬如,调度问题中的典型构造型方法有:Johnson 法、Palmer 法、Gupta 法、CDS 法、Daunenbring 的快速接近法、NEH 法等。

3. 改进型算法,或称邻域搜索算法 它指从任一解出发,对其邻域的不断搜索和当前解的替换来实现优化。根据搜索行为,它又可分为局部搜索法和指导性搜索法。

局部搜索算法是基于贪婪思想利用邻域函数进行搜索的,它通常可描述为:从一个初始解出发,利用邻域函数持续地在当前的领域中搜索比它好的解,若能够找到如此的解,就以之为新的当前解,然后重复上述过程,否则结束搜索过程,并以当前解作为最终解,可见,局部搜索算法尽管具有通用容易实现的特点,但搜索性能完全依赖于邻域函数和初始解,邻域函数涉及不当或初值选取不合适,则算法最终的性能将会很差,同时,贪婪思想无疑会使算法丧失全局优化地能力,也即算法在搜索过程中无法避免陷入局部极值,因此,若不在搜索策略上进行改进,那么要实现全局优化,局部搜索算法采用的邻域函数必须是完全的,即邻域函数将导致解的完全枚举,而在大多数情况下是无法实现的,而且穷举的方法对于大规模问题在搜索时间上是不允许的。

鉴于局部搜索算法的缺陷,我们通常采用指导性搜索算法,即利用一些指导规则来知道整个解空间种优良解的探索,如模拟退火算法,遗传算法,禁忌搜索算法,神经优化算法和混沌算法,从不同的角度利用不同的搜索机制和策略实现对局部搜索算法的改进,来取得较好的全局优化性能。

遗传算法是进化算法的一种,目前进化算法主要包括三个研究领域:遗传算法(Genetic Algorithm),进化规划(Evolutionary Programming)和进化策略(Evolution Strategies),其中,遗传算法是迄今为止进化算法中最广

为人知的算法，自从提出后，在优化领域有着越来越广泛的应用。

4. 基于系统动态演化的方法 将优化过程转化为系统动态的演化过程。基于系统动态的演化来实现优化，如神经网络和混沌搜索等。

3.2 MATLAB 遗传算法理论概述

MATLAB 是“矩阵实验室(Matrix Laboratory)”的缩写，由美国 Mathwork 公司推出。MATLAB 是从概念设计，算法开发，建模仿真，实时实现的理想的集成环境。MATLAB 在科学研究和产品开发中起着重要的作用。

3.2.1 MATLAB 基础

从 20 世纪 80 年代起，出现了科学计算语言，也称为数学软件。因其高效、可视化和推理能力等特点，在大学教育和科学研究中，正迅速取代 FORTRAN 和 BASIC 语言。这类语言中已商品化的有 MATLAB、MATHEMATICA、MATHCAD、MAPLE 等，它们的功能大同小异，又各有所长。其中用途最广的是 MATLAB 软件。根据最近因特网上的检索，美国已有 300 多种有关 MATLAB 语言的书籍，仅 Prentice—Hall 出版社近 3 年内出版的将 MATLAB 用于各门课程的教材就超过百种，其范围包括：微积分、矩阵代数、应用数学、物理、力学、信号与系统、电子线路、电机学、机械振动、科学计算、有限元法、计算机图形学、自动控制和通信技术等。MATLAB 是一种科学计算软件，主要适用于矩阵运算及控制和信息处理领域的分析设计，它使用方便，输入简捷，运算高效，内容丰富，并且很容易由用户自行扩展。

1. MATLAB 的发展

1967 年，美国 Mathwork 公司推出的“矩阵实验室(MatrixLaboratory)”(缩写为 MATLAB)软件包；

1984 年正式推出 MATLAB1.0 版商业软件；

1988 年有了 3.1(Dos)版本；

1992 年出了 4.1(Windows)版本；

1997 年推出了 5.1(Windows)版本。

2001 年初，推出了 6.1(R12)版本；

2004 年夏又推出了他们的最新产品 MATLAB 7.0(R14)的正式版。

一方面，随着版本的升级，其内容在不断扩充，功能也更加强大。特别是在系统仿真和实时运行等方面，有很多新进展，更扩大了它的应用前景。另一方面，版本的升级对使用环境也提出了更高的要求。

MATLAB 是一种以矩阵运算为基础的交互式程序语言，专门针对科学、工程计算和绘图的需求而开发的。与其他计算机语言相比，其特点是简洁和智能化，适应科技专业人员的思维方式和书写习惯，使得编程和调试效率大大提高。它用解释方式工作，键入程序立即得出结果，人机交互性能好，为科技人员乐于接受。特别是它可适应多种平台，并且随计算机硬、软件的更新及时升级。

MATLAB 语言只有一种数据类型，一种标准的输入输出语句，不用“指针”，不需编译，比其他语言少了很多内容。MATLAB 语言的函数较多，仅基本部分就有 700 多个，其中常用的有二三百个。

2. MATLAB 的特点

MATLAB 语言有以下五个特点。

(1) 起点高

1) 每个变量代表一个矩阵，它有 $n \times m$ 个元素。从 MATLAB 名字的来源可知，它以矩阵运算而见长，在当前的科学计算中，几乎无处不用矩阵运算，这使它的优势得到了充分的体现。

2) 每个元素都看作复数。这个特点在其他语言中也是不多见的。

3) 所有的运算都对矩阵和复数有效，包括加、减、乘、除、函数运算等。

(2) 人机界面适合科技人员

1) 语言规则与笔算式相似。MATLAB 的程序与科技人员的书写习惯相近易读，易于在科技人员之间交流。

2) 矩阵行列数无需定义。要输入一个矩阵，用其他语言时必须先定义矩阵的阶数，而 MATLAB 则不必用阶数定义语句。输入数据的行列数就决定了它的阶数。

3) 键入算式立即得结果，无需编译。MATLAB 是以解释方式工作的，即它对每条语句解释后立即执行，若有错误也立即作出反应，便于编程者马上改正。这些都大大减少了编程和调试的工作量。

(3) 强大而简易的作图功能

1) 能根据输入数据自动确定坐标绘图。

2) 能规定多种坐标系(极坐标，对数坐标等)。

3) 能绘制三维坐标中的曲线和曲面。

如果数据齐全，通常只需一条命令即可出图。

(4) 智能化程度高

- 1) 绘图时自动选择最佳坐标以及自动定义矩阵维数
- 2) 作数值积分时自动按精度选择步长。
- 3) 自动检测和显示程序错误的能力强, 易于调试。

(5) 功能丰富, 可扩展性强

MATLAB 软件包括基本部分和专业扩展两大部分。基本部分包括: 矩阵的运算和各种变换, 代数和超越方程的求解, 数据处理和傅里叶变换, 数值积分等。

扩展部分称为工具箱。它实际上是用 MATLAB 的基本语句编成的各种子程序集, 用于解决某一方面的专门问题, 或实现某一类的新算法。现在已经有控制系统、信号处理、图像处理、系统辨识、模糊集合、神经网络、小波分析等 20 余个工具箱, 并且它们还在继续发展中。MATLAB 的核心内容在它的基本部分, 所有的工具箱子程序都是用它的基本语句编写的。

3.2.2 遗传算法概述

遗传算法起源于对生物系统所进行的计算机模拟研究。遗传算法以达尔文的适者生存、优胜劣态的进化原则为基础, 使用群体搜索技术, 通过对当前群体施加遗传学的基本操作, 即选择、交叉、变异等, 从而产生出新的一代群体, 并逐步使群体进化到包含或接近最优解的状态。早在 20 世纪 40 年代, 生物模拟就成为计算机科学的一个组成部分。进化计算的研究起于 20 世纪 50 年代, 当时几个计算机领域的科学家独立的开始研究进化系统, 使之将自然界的进化过程引如工程研究领域, 以解决工程进化问题。在用进化思想解决优化问题时, 使用了进化过程中的遗传、选择等概念, 并且把它们作为算子参与优化。

在 20 世纪 60 年代, R evchenberg 提出进化策略方法, 在此以后, 这一工作被 Schwefel 继续下去, 该法可以用于优化实值函数。同一时代 Fogel、Owens 和 Walsh 提出了进化规则的方法, 该方法把给定的问题描述成有限状态机制, 通过施加进化算子达到优化的目的。

Holland 在 20 世纪 60 年代运用生物遗传和进化的思想来研究自然和人工自然适应系统的生成土及它们与环境的关系, 提出在研究和设计人工自适应系统时, 可以借鉴生物的遗传的机制, 一群体的方法进行自适应搜索, 并且充分认识到交叉、变异等运算策略在自适应系统中的重要性。同时 Holland 不但发现了基于适应度的人工遗传的基于作用, 而且还对群体操作等进行了

认真的研究, 1965 年他首次提出了人工遗传操作的重要性, 并把这些应用于自然系统和人工系统中。1968 年 Holland 教授提出了遗传算法的基于定理一模拟定理, 从而奠定了遗传算法的理论基础。模式定理提出了群体中的优良个体的样本数将以指数级规律增长, 因而从理论上保证了遗传算法是一个可以用来寻求最优化可行解的优化过程。

1967 年, Bagleg 的论文中首次提出了遗传算法这一名称, 他构造的遗传算法用来搜索下棋游戏评价函数中的参数集, 这于我们现在应用的遗传算法很相似, 其中利用了类似于前面描述的复制、杂交和变异的遗传算子。他还敏锐地意识到在运算开始和结束阶段需要适当的选择率, 为此还引入了适应植比例机制, 早接近群体的起使阶段减小选择的强制性, 在后期阶段增加选择的强制性, 因而, 在接近群体收鉴时, 类似的高适应植各串之间保持了适当的竞争。

1971 年, Holland 写的关于“遗传算法在纯数学优化应用”方面的第一篇学术论文, 主要研究了物种不同的选择方法和八中交配策略。他采用 16 位二元串, 其中两个 8 位参数是用无符号而进制整数或 Gray 码整数来编码的, 群体规模为 16 个串, 通过计算机实验结果, Holland 指出由于群体规模太小 ($N=16$) 所引起的问题。

1975 年 Holland 出版了第一本《自然系统和人工系统的自适应性》著作, 书中建立了遗传算法的框架。从算法中每一代群体通过选择、交叉、变异形成而表现出不同性能。对每个染色体由于其中所含基因排列方式的不同而表现出来的不同性能。对每个性能的度量采用被称作适应度函数, 它体现了个体对环境的适应度。1975 年, De Jong 对 Holland 模式定理进行了大量的函数优化计算实验, 并得到具体指导意义结论。例如, 对与规模在 $50 \sim 100$ 的群体, 经过 $10 \sim 20$ 代的额进化, 遗传算法都能以很高的概率找到最优化或近似最优解, 同时, 定义了评价遗传算法性能的在线指标和离线指标。

20 世纪 80 年代, 遗传算法的进入了兴盛发展时期, 在理论研究和应用研究方面都成了十分热门的课题, 特别是遗传算法的应用研究显得格外活跃, 不但它的应用领域扩大, 而且遗传算法进行优化和规则学习的能力也显著提高。此外一些新的理论和方法在应用研究中得到了迅速的发展, 这些无疑地给遗传算法增添了新的活力。1983 年, Goldberg 将遗传算法应用于管道系统的优化和机器学习问题, 通过研究管道系统不仅用与实际系统相当的成本满足了供气要求, 而且也发展了一套分层容错规则。1989 年, Godberg 出版, 《搜

索、优化和机器学习的遗传算法》专著。这本书全面完整地论述了遗传算法的基本原理及应用，该书奠定了现代遗传算法的科学基础。

自 20 世纪 90 年代以来，由于进化策略、进化规则和遗传算法取得了一些令人信服的结果，所以引起了很多人的关注，进而得到了深入的研究，并在各领域快速发展和广泛的应用。在这个时期，国际遗传算法会议（ICGA）及以遗传算法的理论基础为中心的学术会议（FOGA）等的论文中反映了遗传最新发展的动向，并以《遗传算法手册》为开端涌现了一些专著。在应用研究方面更是格外活跃，主要表现在：

函数优化方面，动态参数编码初步克服了遗传算法解决连续区域问题出现的精度不高、计算效率低的弊端。

组合优化，将遗传算法用于求解 TSP 问题，以及解决运输和车间调度问题等取得满意结果，也使自身得到检验。

机器学习，主要体现在分类器系统，在很多领域中得到应用，例如遗传算法被用于学习模糊控制规则等。

控制方面，遗传算法用于工业过程、机器人等的自适应控制、最优控制。

此外遗传算法在图象处理、故障诊断、人工生命、神经网络等均有新的成果。

我国的一些科学工作者，自 20 世纪 90 年代以来，在国家 863 高技术计划和国家自然科学基金等资助下，开展了遗传算法的研究，并出版了遗传算法的专著。随继，广大科学工作者在各个领域进行了应用研究，最近几年中在机械工程领域也开展了多方面的应用，例如，在机械结构优化设计、机械零件优化、刀具优化设计和切削参数优化、设计与加工误差优化、机器人路径优化、模糊逻辑控制器、符合材料优化、故障诊断、金属成形优化及参数辨识等方面获得了广泛的应用。^[10]

第 4 章 遗传算法的原理与实现

遗传算法模拟了自然选择和遗传中发生的复制、交叉和变异等现象,从任一初始种群(population)出发,通过随机选择、交叉和变异操作,产生一群更适应环境的个体,使群体进化到搜索空间中越来越好的区域,这样一代一代地不断繁衍进化,最后收敛到一群最适应环境的个体(individual),求得问题的最优解。

4.1 遗传算法运算过程

1. 编码:解空间中的解数据 x , 作为遗传算法的表现型形式。遗传算法在进行搜索之前先将解空间的解数据表示成遗传串间的基因形串结构数据,这些串结构数据的不同组合就构成了不同的点。

2. 初始群体的生成:随机产生 N 个初始串结构数据。每个串结构数据称为一个个体, N 个个体构成了一个群体。遗传算法以这 N 个串结构作为初始点开始迭代。设置进化代数计数器 $t \leftarrow 0$; 设置最大进化代数 T ; 随机生成 M 个个体作为初始群体 $P(0)$ 。

3. 适应度值评价检测:适应度函数表明个体或解的优劣性。对于不同的问题,适应度函数的定义方式不同。

4. 选择:将选择算子作用于群体。

5. 交叉:将交叉算子作用于群体。

6. 变异:将变异算子作用于群体。

群体 $P(t)$ 经过选择、交叉、变异运算后得到下一代群体 $P(t+1)$ 。

7. 终止条件判断:若 $t \leq T$, 则 $t \leftarrow t+1$, 转到步骤 2; 若, $t > T$, 则以进化过程中所得到的具有最大适应度的个体作为最优解输出, 终止运算。

4.2 遗传算法主要因素

1. 编码方法

针对一个具体应用问题,如何设计一种完美的编码方案一直是遗传算法的应用难点之一,也是遗传算法的一个重要研究方向。由于遗传算法应用的广泛性,迄今为止人们已经提出了许多种不同的编码方法,总的来说,可以分为三大类:二进制编码方法、符号编码方法和浮点数编码方法。

(1) 二进制编码方法

二进制编码方法是遗传算法中最主要的一种编码方法，它使用的编码符号集是由二进制符号 0 和 1 所组成的二值符号集 $\{0, 1\}$ ，它所构成的个体基因型是一个二进制编码符号串。二进制编码符号串的长度与问题所要求的求解精度有关。

(2) 格雷码编码

二进制编码不便于反映所求问题的结构特征，对于一些连续函数的优化问题等，也由于遗传运算的随机特性而使得其局部搜索能力较差。为了改进这个特性，人们提出用格雷码(GrayCode)来对个体进行编码。格雷码是这样一种编码方法，其连续的两个整数所对应的编码之间仅仅只有一个码位是不同的，其余码位都完全相同。

2. 选择算子

选择(Selection)，是在群体中选择生命力强的个体产生新的群体的过程。遗传算法使用选择算子来对群体中的个体进行优胜劣汰操作：根据每个个体的适应度值大小选择，适应度较高的个体被遗传到下一代群体中的概率较大；适应度较低的个体被遗传到下一代群体中的概率较小。这样就可以使得群体中个体的适应度值不断接近最优解。选择操作建立在对个体的适应度进行评价的基础之上。选择操作的主要目的是为了避免有用遗传信息的丢失，提高全局收敛性和计算效率。选择算子确定的好坏，直接影响到遗传算法的计算结果。选择算子确定不当，会造成群体中相似度值相近的个体增加，使得子代个体与父代个体相近，导致进化停止不前；或使适应度值偏大的个体误导群体的发展方向，使遗传失去多样性，产生早熟问题。表 4-1 所示为常用选择操作算子及特点。

表 4-1 选择操作算子

序号	名 称	特 点	备 注
1	轮盘赌选择	选择误差较大	GA 成员
2	随机竞争选择	比轮盘赌选择较好	
3	无回放随机选择	降低选择误差，操作不方便	
4	无回放式余数随机选择	误差最小	应用较广
5	均匀排序	与适应度大小差异程度正负无关	

3. 交叉算子

在生物的自然进化过程中，两个同源染色体通过交配而重组，形成新的染色体，从而产生出新的个体或物种。交配重组是生物遗传和进化过程中的一个主要环节。模仿这个环节，遗传算法中使用交叉算子来产生新的个体。交叉(Crossover)，是按较大的概率从群体中选择两个个体，交换两个个体的某个或某些位。交叉运算产生子代，子代继承了父代的基本特征。交叉算子的设计包括如何确定交叉点位置和如何进行部分基因交换两个方面的内容。遗传算法中的交叉运算，是指对两个相互配对的染色体按某种方式相互交换其部分基因，从而形成两个新的个体。交叉运算是遗传算法区别于其他进化运算的重要特征，它在遗传算法中起着关键作用，是产生新个体的主要方法。

遗传算法中，在交叉运算之前还必须先对群体中的个体进行配对。目前常用的配对算法策略是随机配对，即将群体中的 M 个个体以随机的方式组成 $M/2$ 对配对个体组。交叉操作是在这些配对个体组中的两个个体之间进行的。表 4-2 所示为常用的交叉操作。

表 4-2 交叉操作

序号	名称	特点	适用编码
1	单点交叉	标准遗传算法成员	符号
2	两点交叉	使用较多	符号
3	多点交叉	交叉点大于 2	符号
4	启发式交叉	应用领域知识	序号

4. 变异算子

变异(Mutation)是以较小的概率对个体编码串上的某个或某些位值进行改变，如二进制编码中“0”变为“1”，“1”变为“0”，进而生成新个体。在遗传算法中也引入了变异算子来产生新的个体。遗传算法中的变异运算，是指将个体染色体编码串中的某些基因座上的基因值用该基因座的其他等位基因来替换，从而形成一个新的个体。从遗传运算过程中产生新个体的能力方面来说，变异本身是一种随机算法，但与选择和交叉算子结合后，能够避免由于选择和交叉运算而造成的某些信息丢失，保证遗传算法的有效性。交叉运算是产生新个体的主要方法，它决定了遗传算法的全局搜索能力；而变异运算只是产生新个体的辅助方法，但它也是必不可少的一个步骤，因为它决定了遗传算法的局部搜索能力。交叉算子与变异算子相互配合，共同完成对搜索空间的全局搜索和局部搜索，从而使得遗传算法能够以良好的搜索性能

完成最优化问题的寻优过程。表 4-3 所示为常用变异算子。

表 4-3 变异算子。

序号	名称	特点	适用编码
1	基本位突变	标准遗传算法成员	符号
2	有效基因突变	避免有效基因缺失	符号
3	自适应有效基因突变	最低有效基因个数自适应变化	符号
4	高斯近似突变	提高对重点搜索区域的局部搜索性能力	实数

5. 约束条件的处理

在遗传算法中必须对约束条件(Constraints)进行处理,但目前尚无处理各种约束条件的一般方法,根据具体问题可选择下列三种方法,即搜索空间限定法、可行解变换法和罚函数法。

(1) 搜索空间限定法

搜索空间限定法的基本思想是对遗传算法的搜索空间的大小加以限制,使得搜索空间中表示一个个体的点与解空间中的表示一个可行解的点有一一对应的关系。对一些比较简单的约束条件通过适当编码使搜索空间与解空间一一对应,限定搜索空间能够提高遗传算法的效率。在使用搜索空间限定法时必须保证交叉、变异之后的新个体在解空间中有对应解。

(2) 可行解变换法

可行解变换法的基本思想是在由个体基因型到个体表现型的变换中,增加使其满足约束条件的处理过程,即寻找个体基因型与个体表现型的多对一变换关系,扩大了搜索空间,使进化过程中所产生的个体总能通过这个变换而转化成解空间中满足约束条件的一个可行解。可行解变换法对个体的编码方法、交叉运算、变异运算等无特殊要求,但运行效率下降。

(3) 罚函数法

罚函数法的基本思想是对在解空间中无对应可行解的个体计算其适应度时,处以一个罚函数,从而降低该个体的适应度,使该个体被遗传到下一代群体中的概率减小。可以用下式对个体的适应度进行调整:

$$F'(x) = \begin{cases} F(x) & x \text{ 满足约束条件} \\ F(x) - P(x) & x \text{ 不满足约束条件} \end{cases}$$

式中, $F(x)$ 为原适应度; $F'(x)$ 为调整后的新适应度; $P(x)$ 为罚函数。

如何确定合理的罚函数是这种处理方法难点之所在, 在考虑罚函数时, 既要度量解对约束条件不满足的程度, 又要考虑计算效率。本课题采用常用的罚函数法进行运算。

4.3 遗传算法的特点

4.3.1 概述

为解决各种优化计算问题, 人们提出了各种各样的优化算法, 如单纯形法、梯度法、动态规划法、分枝定界法等。这些优化算法各有各的长处, 各有各的适用范围, 也各有各的限制。遗传算法是一类可用于复杂系统优化计算的鲁棒搜索法, 与其他一些优化算法相比, 它主要有下述几个特点:

1. 遗传算法以决策变量的编码作为运算对象 传统的优化算法往往直接利用决策变量的实际值本身来进行优化计算, 但遗传算法不是直接以决策变量的值, 而是以决策变量的某种形式的编码为运算对象。这种对决策变量的编码处理方式, 使得我们在优化计算过程中可以借鉴生物学中染色体和基因等概念, 可以模仿自然界中生物的遗传和进化等机理, 也使得我们可以方便地应用遗传操作算子。特别是对一些无数值概念或很难有数值概念, 而只有代码概念的优化问题, 编码处理方式更显示出了其独特的优越性。

2. 遗传算法直接以目标函数值作为搜索信息 传统的优化方法不仅需要利用目标函数值, 而且往往需要目标函数的导数值等其他一些辅助信息才能确定搜索方向。而遗传算法仅使用由目标函数值变换来的适应度函数值, 就可确定进一步的搜索方向和搜索范围, 无需目标函数的导数值等其他一些辅助信息。这个特性对很多目标函数是无法或很难求导数的函数, 或导数不存在的函数的优化问题, 以及组合优化问题等, 应用遗传算法时就显得比较方便, 因为它避开了函数求导这个障碍。再者, 直接利用目标函数值或个体适应度, 也可使得我们可以把搜索范围集中到适应度较高的部分搜索空间中, 从而提高了搜索效率。

3. 遗传算法同时使用多个搜索点的搜索信息 传统的优化算法往往是从解空间中的一个初始点开始最优解的迭代搜索过程。单个搜索点所提供的搜索信息毕竟不多, 所以搜索效率不高, 有时甚至使搜索过程陷于局部最优解而停滞不前。遗传算法是从由很多个体所组成的一个初始群体开始最优解的搜索过程, 而不是从一个单一的个体开始搜索。对这个群体所进行的选择、

交叉、变异等运算，产生出的乃是新一代的群体，在这之中包括了很多群体信息。这些信息可以避免搜索一些不必搜索的点，所以实际上相当于搜索了更多的点，这是遗传算法所特有的一种隐含并行性。

4. 遗传算法使用概率搜索技术 很多传统的优化计算法往往使用的是确定性的搜索方法，一个搜索点到另一个搜索点的转移有确定的转移方法和转移关系，这种确定性往往也有可能使得搜索永远达不到最优点，因而也限制了算法的应用范围。而遗传算法属于一种自适应概率搜索技术，其选择、交叉、变异等运算都是以一种概率的方式来进行的，从而增加了其搜索过程的灵活性。虽然这种概率特性也会使群体中产生一些适应度不高的个体，但随着进化过程的进行，新的群体中总会更多地产生出许多优良的个体，实践和理论都已证明在一定条件下遗传算法总是以概率收敛于问题的最优解。当然，交叉概率和变异概率等参数也会影响算法的搜索效果和搜索效率，所以如何选择遗传算法的参数在其应用中是一个比较重要的问题。而另一方面，与其他一些算法相比，遗传算法的鲁棒性又会使参数对其搜索效果的影响会尽可能地低。

4.3.2 遗传算法与传统搜索算法的比较

1. 遗传算法与启发式算法的比较

启发式算法是指通过寻求一种能产生可行解的启发式规则，找到问题的一个最优解或近似最优解。该方法求解问题的效率较高，但是它对每一个所求的问题必须找出其特有的启发式规则，这个启发式规则一般无通用性，不适用于其他问题。但遗传算法采用的不是确定性规则，而是强调利用概率转换规则来引导搜索过程。

2. 遗传算法与爬山法的比较

爬山法是直接法、梯度法和 Hessian 法的通称。爬山法首先在最优解可能存在的地方选择一个初始点，然后通过分析目标函数的特性，由初始点移到一个新的点，然后再继续这个过程。爬山法的搜索过程是确定的，它通过产生一系列的点收敛到最优解(有时是局部最优解)，而遗传算法的搜索过程是随机的，它产生一系列随机种群序列。二者的主要差异可以归纳为如下两点：

爬山法的初始点仅有一个，由决策者给出；遗传算法的初始点有多个，是随机产生的。

通过分析目标函数的特性可知,爬山法由上一点产生一个新的点;遗传算法通过遗传操作,在当前的种群中经过交叉、变异和选择产生下一代种群。对同一优化问题,遗传算法所使用的机时比爬山法所花费的机时要多,但遗传算法可以处理一些爬山法所不能解决的复杂的优化问题。

3. 遗传算法与穷举法的比较

穷举法就是对解空间内的所有可行解进行搜索,但是通常的穷举法并不是完全穷举法,即不是对所有解进行尝试,而是有选择地尝试,如动态规划法、限界剪枝法。对于特定的问题,穷举法有时也表现出很好的性能。但一般情况下,对于完全穷举法,方法简单易行,但求解效率太低;对于动态规划法、限界剪枝法,则鲁棒性不强。相比较而言,遗传算法具有较高的搜索能力和极强的鲁棒性。

4. 遗传算法与盲目随机法的比较

与上述的搜索方法相比,盲目随机搜索法有所改进,但它的搜索效率仍然不高。一般而言,只有解在搜索空间中形成紧致分布时,它的搜索才有效。而遗传算法作为导向随机搜索方法,是对一个被编码的参数空间进行高效搜索。

经过上面的探讨,可以看到遗传算法与更多的传统优化方法在本质上有着一些不同之处,遗传算法与传统搜索算法不同,主要表现在以下几个方面:

遗传算法从一组初始点,是并行的,而不是从一个初始点,开始进行搜索;

遗传算法将参数搜索问题转化为参数空间编码的搜索问题;

遗传算法只用到目标函数,以及对应的目标函数信息,而无需其导数信息或其他辅助信息;

遗传算法采用随机变换规则,而不是确定性的规则,来指导搜索。

4.3.3 遗传算法的优点

由上述这些搜索特点,可以总结出遗传算法具有如下一些特别引人注意的优点:

应用广泛性 易于写出一个通用算法,求解不同类型的优化问题。

非线性性 大多数现行的优化算法都基于线性性、凸性、可微性等,但遗传算法没有这些假设。遗传算法只需要评价目标值的优劣,具有高度的非线性性。

适应性 若把原问题进行一些很小的改动,大多数现行的优化算法很可能完全不能使用或需要进行很大的修改,而遗传算法则需作很小的修改即可适应新的问题。

并行性 遗传算法隐含地对问题空间的许多解平面进行并行搜索,收敛速度快、稳定性强。因此,遗传算法适合不同类型、不同性质的问题的求解。

由于遗传算法所具有的不依赖问题模型的特性、全局最优性、本质并行性、高效率及解决不同非线性问题所具有的较强的鲁棒性,遗传算法的应用领域已从函数优化、模式识别等扩展到工业控制和工程优化设计等方面,并取得了很好的效果。遗传算法特别适合于复杂组合优化问题的求解,对这些问题,用传统的方法很难找到问题的最优解,或是找到优化结果需要几周甚至几年的时间,而遗传算法却能很快地找到好的结果。

4.4 遗传算法的不足之处

遗传算法作为一种优化方法,它存在自身的局限性:

1. 编码不规范及编码存在表示的不准确性。
2. 单一的遗传算法编码不能全面地将优化问题的约束表示出来。考虑约束的一个方法就是对不可行解采用或值,这样,计算机的时间必然增加。
3. 遗传算法通常的效率比其他传统的优化方法低。
4. 遗传算法容易出现过早收敛。
5. 遗传算法对算法的精度、可信度、计算复杂性等方面,还没有有效的定量分析方法。

4.5 遗传算法的应用

由于遗传算法以上的特点,决定了遗传算法可以应用到很多领域,它的主要应用领域有:

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。很多人构造了各种各样的复杂形式的测试函数,人们用这些几何特性各异的函数来评价遗传算法的性能。对于一些非线性、多模型、多目标的函数优化问题,用其它优化方法较难求解,而遗传算法却可以方便地得到较好的结果。

2. 组合优化

随着问题规模的扩大,组合优化问题的搜索空间急剧扩大,有时在目前

的计算机上用枚举法很难甚至不可能得到其精确最优解。对于这类复杂问题，人们已意识到应把精力放在寻求其满意解上，而遗传算法正是寻求这种满意解的最佳工具之一。实践证明，遗传算法对于组合优化中的 NP 完全问题非常有效。

3. 生产调度问题

生产调度问题在许多情况下所建立起来的数学模型难以精确求解，即使经过一些简化后之后可以求解，也会因简化太多而使得求解结果与实际相差甚远。遗传算法是解决复杂调度问题的有效工具，在单件生产车间调度、流水线生产车间调度、生产规划、任务分配等方面遗传算法都得到有效的应用。

4. 自动控制

在自动控制领域中有许多与优化相关的问题，遗传算法在解决这些问题时显示了良好的效果，如利用遗传算法进行航空控制系统的优化、基于遗传算法的模糊控制器优化设计、利用遗传算法进行人工神经网络的结构优化设计和权值学习等。

5. 机器人智能控制

机器人是一类复杂的难以精确建模的人工系统，而遗传算法的起源来自于对人工字适应系统的研究，所以机器人学只能控制理所当然地成为遗传算法的一个重要领域。例如，遗传算法已经在移动机器人路径规划、关节机器人运动轨迹规划、机器人逆运动学求解、细胞机器人的结构优化和行动协调等方面得到很好的应用。

6. 图像处理和模式识别

遗传算法可用于图像处理中的优化计算方面，如图像恢复、图像边缘特征提取、几何形状识别等。图象处理是计算机视觉中的一个重要领域，在图像处理过程中，如扫描、特征提取、图像分割等不可避免地会存在一些误差，这些误差会影响图像处理的效果。如何使这些误差最小是使计算机视觉达到实用化的重要要求，遗传算法在这些图像处理的油画计算方面找到了用武之地。

7. 人工生命

自组织能力和自适应能力是人工生命的两大主要特征，基于遗传算法的进化模型是研究人工生命现象的重要理论基础。可以预见，遗传算法在人工生命及复杂自适应系统的模拟与设计、复杂系统突现性理论研究中，将得到更深的发展。

8. 遗传程序设计

遗传程序设计的主要思想是使用某种编码方法,对某种结构进行遗传操作以自动生成计算机程序,虽然遗传程序设计的理论尚未成熟,但它已有一些成功的应用。

9. 机器学习

学习能力是高级自适应系统所应具备的能力之一。基于遗传算法的机器学习,特别是分类器系统,在许多领域得到了应用。例如,基于遗传算法的机器学习可用于调整人工神经网络的连接权,也可用于神经网络结构的优化设计。

10. 数据挖掘

数据挖掘(Data Mining)是指从大型数据库或数据仓库中提取隐含的、位置的、非平凡的及有潜在应用价值的信息或模式,它是数据库研究中的一个很有应用价值的新领域。

而对于本课题而言,在实际加工中人们常常希望在保证加工质量的同时,降低加工成本和提高生产率。当机床和刀具一经确定后,关键就在于确定最优的切削用量。目前,常用的优化算法,如牛顿法、混合惩罚函数法等,都是基于“爬山法”原理的单点搜索方法,其主要缺点在于:对于多峰值目标函数;算法容易收敛于局部最优;对于约束问题需构造惩罚函数;不适合于解决不连续或不可导函数的优化问题;算法搜索效率低,遗传基因算法(GA)为解决此类复杂的优化问题找到了一条全新的途径。^{[1][2][10]}

4.6 MATLAB 遗传算法与直接搜索工具箱(GADS)

最新发布的 MATLAB7.0Release14 包含一个专门设计的遗传算法与直接搜索工具(Genetic Algorithm and Direct Search toolbox)。使用遗传算法与直接搜索工具箱,可以扩展 MATLAB 及其优化工具箱在处理优化问题方面的能力,可以处理传统的优化技术难以解决的问题,包括那些难以定义或不便于进行数学建模的问题,可以解决目标函数较复杂的问题,比如目标函数不连续或具有高度非线性、随机性以及目标函数不可微的情况。

GADS 工具箱是一系列函数的集合,它们扩展了优化工具箱和 MATLAB 数值计算环境的性能。遗传算法与直接搜索工具箱包含了要使用遗传算法和直接搜索算法来求解优化问题的一些例程。这些算法使我们能够求解那些标准优化工具箱范围之外的各种优化问题。所有工具箱函数都是以 MATLAB 的

M 文件，这些文件由实现特定优化算法的 MATLAB 语句所写成。

使用语句：

```
type function_name
```

就可以看到这些函数的 MATLAB 代码。我们也可以通过编写自己的 M 文件来实现和扩展遗传算法和直接搜索工具箱的性能，也可将该工具箱与 MATLAB 的其他工具箱或 Simulink 结合使用来求解优化问题等。

工具箱函数可以通过图形介面或 MATLAB 命令行来访问，它们是用 MATLAB 语言编写的，对用户开放，因此可以查看算法、修改源代码或生成用户函数。

遗传算法与直接搜索工具箱可以帮助我们求解那些不易用传统方法解决的问题，譬如表查找问题等。

遗传算法与直接搜索工具箱有一个精心设计的图形用户界面，可以帮助我们直观、方便、快速地求解最新优化问题。

1. 功能特点

遗传算法与直接搜索工具的功能特点如下：

图形用户界面和命令行函数可用于快速地描述问题、设置算法选项以及监控进程。

具有多个选项的遗传算法工具可用于问题创建、适应度计算、选择、交叉和变异。

直接搜索工具实现了一种模式搜索方法，其选项可用于网络尺寸、表决方法和搜索方法。

遗传算法与直接搜索工具箱函数可与 MATLAB 的优化工具箱或其他的 MATLAB 程序结合使用。

支持自动的 M 代码生成。

2. 图形用户界面和命令行函数

遗传算法工具函数可以通过命令行和图形用户界面来使用遗传算法。直接搜索工具函数也可以通过命令行和图形用户界面来进行访问。图形用户界面可用于快速地定义问题，设置算法选项，对优化问题进行详细定义。

遗传算法和直接搜索工具箱还同时提供了用于优化管理、性能监控及终止准则定义的工具，同时还提供了大量的标准算法选项。

在优化运行的过程中，可以通过修改选项来细化最优解，更新性能结果。

用户也可以提供自己的算法选项来定制工具箱。

3. 使用其他函数和求解器

遗传算法和直接搜索工具箱和 MATLAB 及优化工具箱是紧密结合在一起的。用户可以用遗传算法或直接搜索算法来寻找最佳起始点，然后利用优化工具箱或用 MATLAB 程序来进一步寻找最优解。通过结合不同的算法，可以充分地发挥 MATLAB 和工具箱的功能以提高求解的质量。对于某些特定问题，使用这种方法还可以得到全局（最优）解。

4. 显示、监控和输出结果

遗传算法与直接搜索工具箱还包括一系列绘图函数，用来可视化优化结果。这些可视化功能直接地显示了优化的过程，并且允许在执行过程中进行修改。

该工具箱还提供了一些特殊绘图函数，它们不仅适用于遗传算法，还适用于直接搜索算法。适用于遗传算法的函数包括函数值、适应度值和函数估计。适用于直接搜索算法的函数包括函数值、分值直方图、系谱、适应度值、网格尺寸和函数估计。这些函数可以将多个绘图一并显示，可直观方便地选取最优曲线。另外，用户可以添加自己的绘图函数。

使用输出函数可以将结果写入文件，产生用户自己的终止准则，也可以写入用户自己的图形界面来运行工具箱求解器。除此之外，还可以将问题的算法选项导出，以便日后再将它们导入到图形界面中去。

5. 所需的产品支持

遗传算法与直接搜索工具箱作为其他优化方法的补充，可以用来寻找最佳起始点，然后再将通过使用传统的优化技术来进一步寻找最优解。

工具箱需要如下产品支持：

MATLAB；

优化工具箱。

6. 相关产品

与遗传算法与直接搜索工具箱相关的产品有：

统计工具箱——应用统计算法和概率模式；

神经网络工具箱——设计和仿真神经网络；

模糊逻辑工具箱——设计和仿真基于模糊逻辑的系统。；

金融工具箱——分析金融数据和开发金融算法。

7. 所需的系统及平台

遗传算法和直接搜索工具箱对于运行环境、支持平台和系统的需求,可随时通过访问网站 <http://www.mathworks.com/products/gads> 了解最新发布的信息。

这里介绍 MATLAB 7.0 Release 14 所需的最底配置是: Windows 系列操作系统, Pentium! III 500 MHz CPU, 64 MB RAM, 空闲硬盘空间 600 MB 以上。^[1]

4.7 使用 GADS 的方法

以 MATLAB 为平台所编制的遗传算法工具箱可以利用 MATLAB 强大的仿真功能,进行遗传算法的各种优化运算。

MATLAB7.0.1 的遗传算法工具有两种使用方式:(1)以命令行方式调用遗传算法函数 `ga`。(2)通过图形用户界面使用遗传算法工具。

1. 在命令行调用函数

对于在命令行使用遗传算法,可以用下列语法调用遗传算法函数 `ga`

```
[x fval]=ga(@fitnessfun, nvars, options)
```

其中, `@fitnessfun` 是适应度函数句柄;

`nvars` 是适应度函数的独立变量的个数;

`options` 是一个包含遗传算法选项参数的结构。

函数所给出的结果为: `fval` -适应度函数的最终值;

`x` -最终值到达的点。

调用函数 `ga` 时,需要提供一个参数结构体 `options`。下面为带带有缺省值的参数结构体:

```
options=  
PopulationType: 'doubleVector',  
PopInitRange: [2X1 double]  
PopulationSize: 20  
EliteCount: 2  
CrossoverFraction: 0.8000,  
MigrationDirection: 'forward'  
MigrationInterval: 20  
MigrationFraction: 0.2  
Generations: 100  
TimeLimit: Inf  
FitnessLimit: - Inf
```

```

StallLimitG: 50
StallLimitS: 20
InitialPopulation: []
InitialScores: []
PlotInterval: 1
CreationFcn: @gacreationuniform
FitnessScalingFcn: @fitscalingrank
SelectionFcn: @selectionstochunif
CrossoverFcn: @crossoverscattered
MutationFcn: @mutationuniform
HybridFcn: []
Display: final
PlotFcns: []
OutputFcns: []
Vectorized: 'off'

```

2. 通过 GUI 使用遗传算法

遗传算法工具具有一个图形用户界面 GUI，它使我们可以使用遗传算法而不用工作在命令行方式。打开遗传算法工具，可键入以下命令：

```
gatool
```

打开的遗传算法工具图形用户界面如图 4-1 所示。

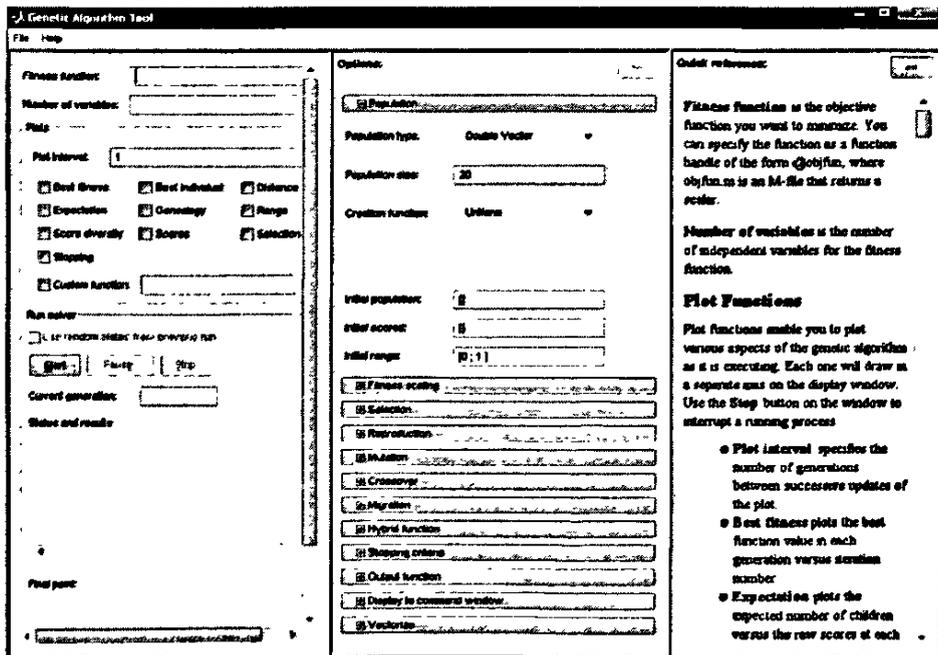


图 4-1 遗传算法工具箱图形用户初启界面

使用遗传算法工具首先必须输入下列信息:

(1)Fitnessfunction(适应度函数)——欲求最小值的目标函数。输入适应度函数的形式为@fitnessfun, 其中 fitnessfun. m 是计算适应度函数的 M 文件。符号@产生一个对于函数 fitnessfun 的函数句柄。

(2)Number variables(变量个数)--适应度函数输入向量的长度。

在“options”窗格中可以改变遗传算法的参数选项。

单击“start”按钮, 即可运行遗传算法。

由上可见, MATLAB 遗传算法与直接搜索工具箱与 MATLAB 语言进行了有机的结合, 它充分利用了 MATLAB 强大的数值分析、矩阵计算和图形处理等功能, 改变了以往的密闭性, 具有良好的人机交互性能, 易操作, 为遗传算法问题的研究与应用提供了一个很好的平台。

第 5 章 基于遗传算法的最大生产率的车削用量优化研究与实现

5.1 优化模型的建立

5.1.1 模型构造的基本思路

利用遗传算法对变异函数进行自动拟合, 首先应对变量进行编码, 然后在可行域中按编码方式随机产生一定数量的个体, 作为进化起点的第一代群体。通过计算每个个体的适应度函数值, 并依此对群体进行选择、交叉、变异等人工遗传操作, 产生新一代群体。以新产生的群体作为父本, 重复选择、交叉、变异过程, 直到终止条件得到满足为止。变量编码、初始群体生成、个体优劣评价(适应度计算)、算法控制参数、遗传算子设计、算法停止准则, 构成了遗传算法的基本要素, 对模型的构建就是对这些基本要素的操作方法进行选择。

5.1.2 总体设计流程

遗传算法优化变异函数的整个计算过程大体分为以下几步:

1. 确定决策变量及其各种约束条件, 即确定出个体的表现型 and 问题的解空间。
2. 建立优化模型, 即确定出目标函数的类型及其数学描述形式或量化方法。
3. 确定表示可行解的染色体编码方法, 也即确定出个体的基因型及遗传算法的搜索空间。
4. 确定解码方法, 即确定出由个体基因型到个体表现型的对应关系或转化方法。
5. 确定个体适应度的量化评价方法, 即确定出由目标函数值到个体适应度的转换规则。
6. 根据设定的遗传概率, 进行下述人工遗传操作, 产生新一代群体:
 - (1) 选择, 将上一代的优良个体复制后, 移入新群体中, 剔除劣质个体。
 - (2) 交叉, 将选出的个体相互配对, 交换部分染色体, 产生新个体加入新

群体中。

(3) 变异, 随机地改变某一个体的某个基因后, 形成新个体, 添入新群体中。

7. 反复执行 3—6 步, 当满足终止条件, 停止进化, 并选择最佳个体作为问题的优化结果。

所以, 具体的算法流程见下图:

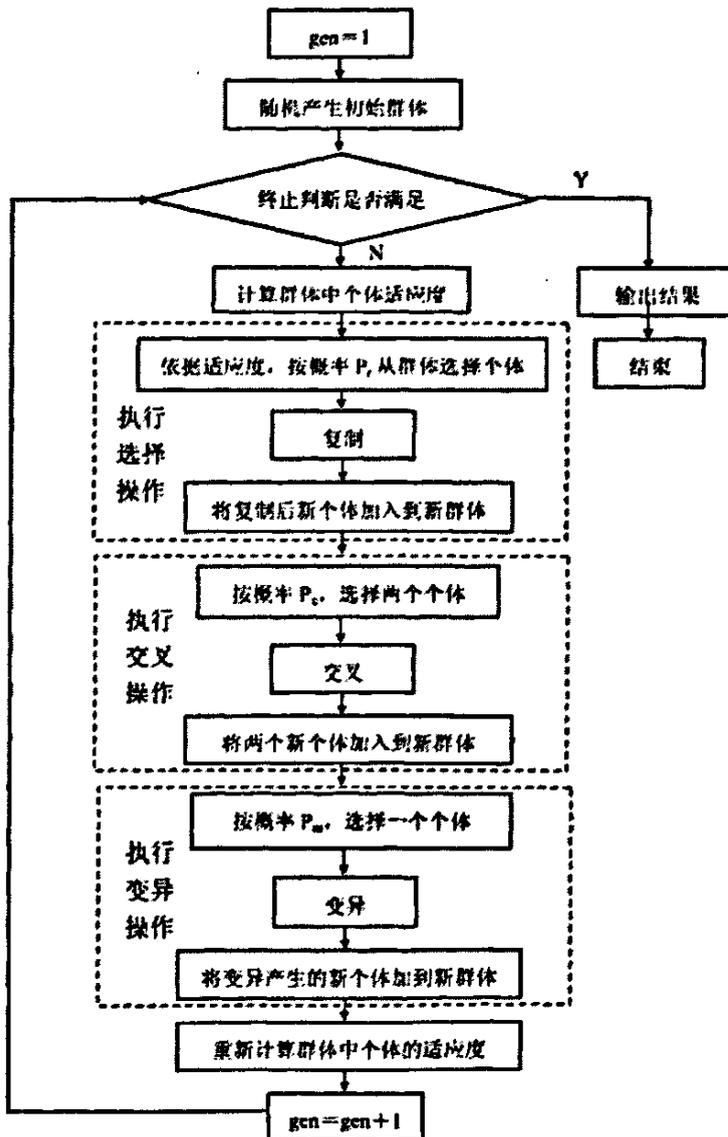


图 5-1 遗传算法流程

5.2 构建目标函数

$$\begin{aligned} \max z_w &= 1000v * f * \alpha_p \\ \text{s.t} \quad n * f^{y_v} &\leq \frac{318.31c_v * k_v}{60^{(1-m)} * d_w * T^m * \alpha_p^{x_v}} \\ n^{(Z_{Fz}+1)} * f^{y_{Fz}} &\leq \frac{101.94 * 1000^{(Z_{Fz}+1)} * P_E \eta_m}{60^{Z_{Fz}} * (\pi d_w)^{(Z_{Fz}+1)} * C_{Fz} * \alpha_p^{x_{Fz}} * K_{Fz}} \\ n^{Z_{Fz}} * f^{y_{Fz}} &\leq \frac{203.87 * 1000^{Z_{Fz}} * M_s}{60^{Z_{Fz}} * C_{Fz} * \alpha_p^{x_{Fz}} * \pi^{Z_{Fz}} K_{Fz} * d_w^{(Z_{Fz}+1)}} \\ n^{Z_{Fz}} * f^{y_{Fz}} &\leq \frac{1000^{Z_{Fz}} * F'}{9.81 * 60^{Z_{Fz}} * K * C_{Fz} * \alpha_p^{x_{Fz}} * K_{Fz} (\pi d_w)^{Z_{Fz}}} \\ n^{Z_{Fy}} * f^{y_{Fy}} &\leq \frac{1000^{Z_{Fy}} * K_0 * E * J * f'}{9.81 * 60^{Z_{Fy}} * C_{Fz} * \alpha_p^{x_{Fy}} * K_{Fy} (\pi d_w)^{Z_{Fy}} * l_0^3} \\ n^{Z_{Fz}} * f^{y_{Fz}} &\leq \frac{1000^{Z_{Fz}} * B * H^2 * \sigma_{bb}}{6l * 9.81 * 60^{Z_{Fz}} * C_{Fz} * \alpha_p^{x_{Fz}} * K_{Fz} (\pi d_w)^{Z_{Fz}}} \\ f &\leq K_R R_Z^n \\ n_{\min} &\leq n \leq n_{\max} \\ f_{\min} &\leq f \leq f_{\max} \\ \alpha_{p\min} &\leq \alpha_p \leq \alpha_{p\max} \end{aligned}$$

5.3 实验事例分析

实例：机床为 C650 车床，硬质合金刀具 YT5，工件为有表皮的锻钢件毛坯、正火处理，大进给、大切深切削外圆：

5.3.1 实验事例中所选用的硬件设施

1. 机床的用途和特点

本课题实验事例研究选用的机床为天水星火机床厂生产的 CW61100E 型普通车床。

该机床可承担各种车削工作：它能够车削各种零件的端面、外圆、内孔及公制螺纹、英制螺纹、模数螺纹、径节螺纹等，还可满足钻孔、镗孔等工艺要求，上刀架与纵向进给配合可机动车削长的锥度；同时也适用于硬质合金刀具进行强力切削，来加工各种黑色金属和有色金属。

本机床具有功率大、转速高、噪声小、刚性强、精度高和精度的保持型好等内在特点，又具有造型美观大方、布局紧凑合理等外部特点，还具有手柄集中，标牌形象直观、操作灵活方便、安全可靠、宜人性好等优点。

2. 机床的规格（摘要说明）

一般规格：

床身上最大工件回转直径： $\Phi 1000\text{mm}$

最大工件长度：5000mm

刀架上最大工件回转直径： $\Phi 650\text{mm}$

两项尖间允许支承最大工件重量 6000kg

插入车削长度 4800mm

其他主轴孔最大最大棒料直径 $\Phi 128\text{mm}$

最大规格：

主轴每分钟的额转速范围：

正转：3.15~315 r/min

反转：5~400 r/min

主轴每转刀架的进给范围：

纵向 0.1~12 mm/r

横向 0.05~6 mm/r

上刀架 0.025~3 mm/r

可加工公制螺纹的种类 44 种

可加工公制螺纹的螺距范围 1~120 mm

主轴中心距刀具安装几面的垂直距离 48 mm

刀具截面（高*宽） 45X45 mm

床鞍的最大行程 4950 mm

刀架最大横向行程（大拖板）	520	mm
上刀架最大行程	300	mm
主电机	Y180L B3 型、22KW、1500r/min	

下面为 CW61100E 型普通车床的图片：

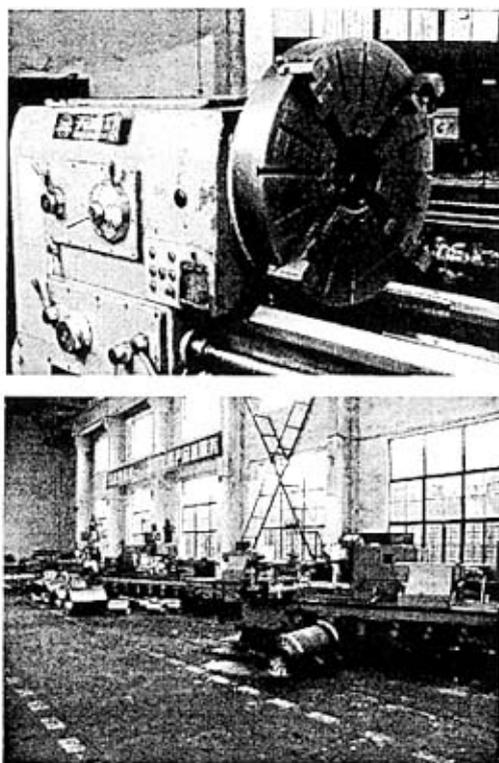


图 5-1 CW61100E 型普通车床

3. 刀具

粗加工（荒车）加工钢件，且为有表皮的锻件毛坯，所以刀具参数如下：

选用刀具为 YT5 硬质合金刀具，YT5 在钨钛合金中，强度最高，抗冲击能力和抗振动能力最强，不易崩刃，但耐磨性较差。

刀具参数：

前角 20°

主偏角 75°

刃倾角 -5°

刀尖圆弧半径：2mm

副偏角： 15°

后角 6°

倒棱前角 -15°

副偏角: 10°

5.3.2 优化运算

由以上硬件条件可确定出目标函数约束条件中的各个参数为:

$Y_v=0.45$; $C_v=235$; $K_v=0.52$; $m=0.2$; $X_v=0.15$; $\sigma_{bb}=196$; $ZFZ=-0.15$;
 $XFZ=1.0$; $YFZ=0.75$; $CFZ=270$;

$KFZ=0.81558$; $B=45$; $H=45$; $l=160$; $PE=22$; $\eta M=0.75$;

$MS=6000$; $dW=400$; $F_j=10000$; $J=1.28 \times 10^{13}$; $E=2200\text{GPa}$;

$f' = 0.7$; $l=1000$; $n_{\min}=3.15$; $n_{\max}=315$; $f_{\min}=0.1$; $f_{\max}=12$; $a_{p\min}=1$;

$a_{p\max}=25$; $Zfy=-0.30$; $Xfy=0.90$; $Cfy=199$; $Yfy=0.60$; $Kfy=0.82$;

$T=3600$; $KR=0.0176$; $u=1.083$; $K_o=140$; $R_z=125$

$n=(1000 \cdot x(1))/(\pi \cdot dw)$;

调用主程序文件, 将上述各参数输入到下面的各个输入窗口界面中:

工件的刚性

工件表面弹性力矩 切削力总体修正系数 C_{fy}

工件的悬伸长度或夹紧在两支承间的距离 工件材料的弹性模量

车削时所容许的相对位移值 工件装夹方式系数

加工工件

工件直径 R_z K_r u

切削加工所容许的最大与最小切削深度(单位: mm)

最小切削深度(单位: mm) 最大切削深度(单位: mm)

吃刀抗力指数

X轴 Y轴 Z轴

确定 取消

机床

机床的最大与最小进给量(单位: mm/秒)		机床进给机构的强度	
最小进给量	最大进给量	系统所允许的最大切削力	
最大与最小主轴转速(单位: 转/分钟)		主切削力指数	
最小主轴转速	最大主轴转速	1轴	2轴
机床电动机的功率(单位: kW)		切削力总修正系数	
机床主轴系统所能传递的扭矩(单位:)		切削力总修正系数 ??????	
机床的传动效率(单位:)		切削力系数	
说明			

确定 取消

fig_dj

刀杆的强度			
刀杆宽度			
刀杆的抗弯强度			
刀杆高度			
刀杆伸出长度			
刀具的耐用度			
根据切削用量手册查得	Y_v	C_v	X_v
切削速度的修正系数			
刀杆指数			
刀具耐用度			

确定 取消

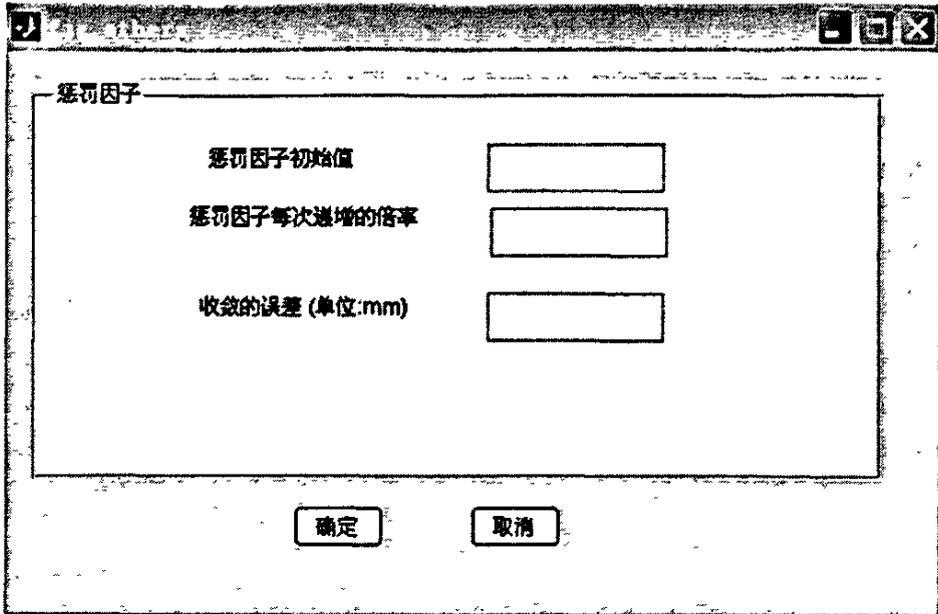
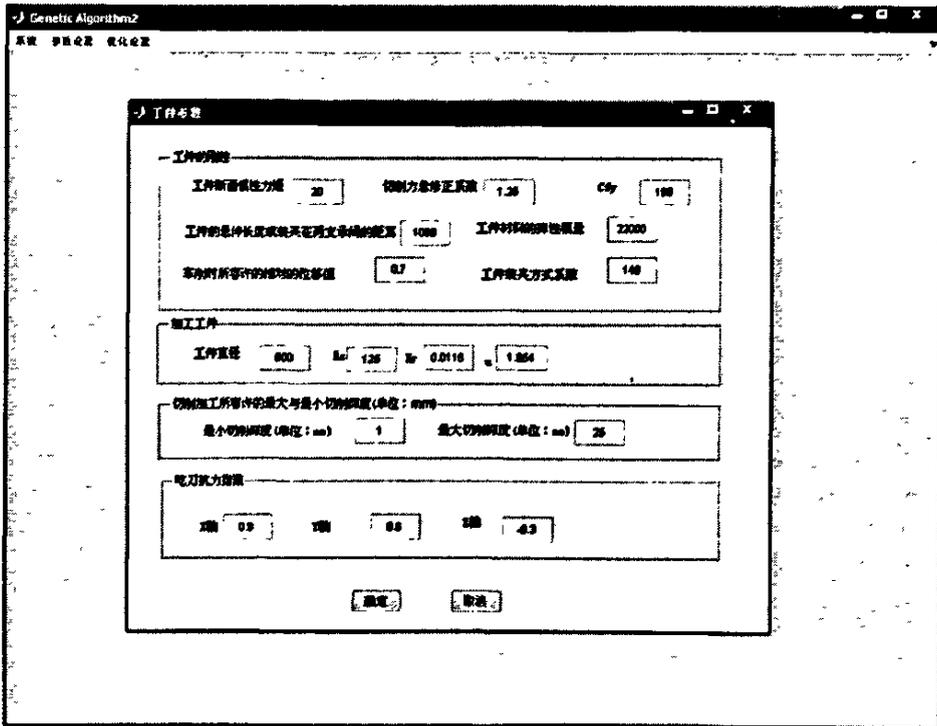


图5-2 各输入参数窗口

运行 main.m 主程序，调用各个参数主界面，输入本实例的各个参数，



机床参数

机床

机床的最大与最小进给量(单位: mm/min)		机床进给机构的强度	
最小进给量	0.15	最大进给量	0.7
		系统所容许的最大切削力	
		800	
最大与最小主轴转速(单位: 转/分钟)		主切削力指数	
最小主轴转速	10	最大主轴转速	1200
		X轴	1
		Y轴	0.75
		Z轴	0.15
机床电动机的功率(单位: kW)		切削力总修正系数	0.7637
15			
机床主轴系统所能传递的扭矩(单位:)		切削力总修正系数 (Z向)	1.2
2			
机床的传动功率(单位:)		切削力系数	270
0.75			
说明			
确定 取消			

刀具参数

刀杆的强度

刀杆宽度	50
刀杆的抗弯强度	0.568
刀杆高度	80
刀杆伸出长度	160

刀具的耐用度

根据切削用量手册查得	V_f	0.45	C_v	235	X_v	0.15
切削速度的修正系数		0.67				
刀杆指数		0.25				
刀具耐用度		30				

确定 取消

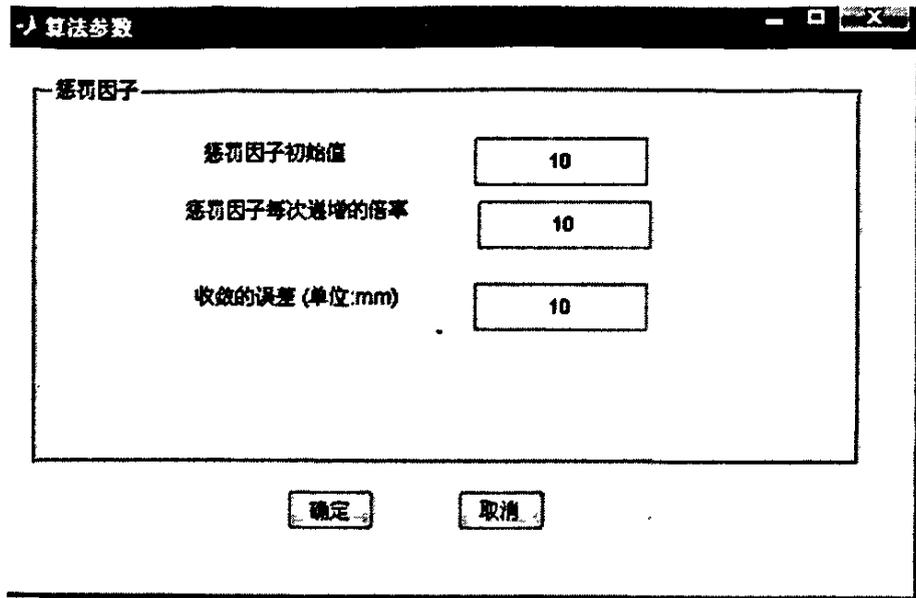


图5-3 调用各输入参数的主界面

确定后，得到优化结果如下：

Optimization terminated: maximum number of generations exceeded.

-----GA results-----

fval =

-4.5569e+005

x =

33.5687 0.7041 19.2799

NumIterations =

7

n =

26.7131

运行优化结果的界面如下：

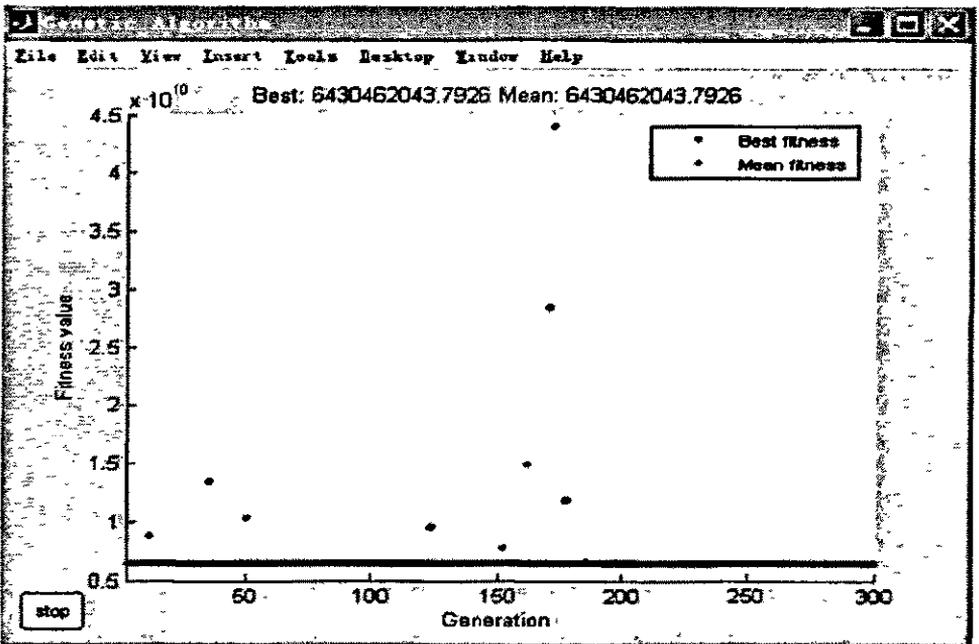
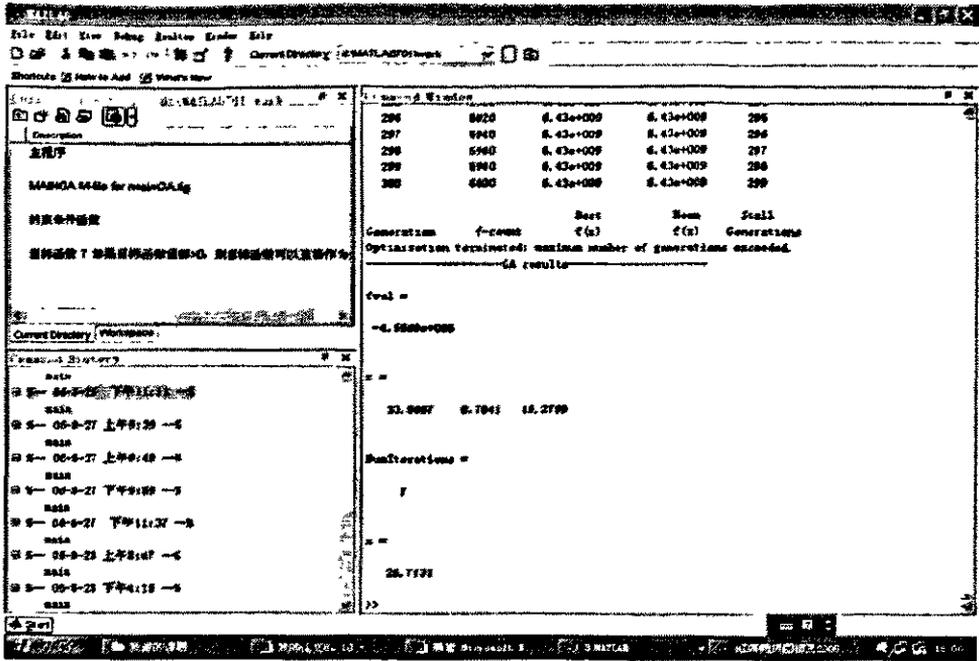


图5-4 优化结果的界面

5.3.3 事例分析

1. 应用 MATLAB 遗传工具箱, 选用适当的参数结构体很重要, 针对具体的实验项目和事例, 需要进行调试和试验, 本课题的程序运行经过多次测试, 选用以下参数结构体较恰当:

```
options=struct('PopulationType', 'doubleVector', ...
'PopInitRange', [vmin 0.15 1;vmax 1.2 25], ...
'PopulationSize', 20, ...
'EliteCount', 8, ...
'CrossoverFraction', 0.8, ...
'MigrationDirection','forward', ...
'MigrationInterval',20, ...
'MigrationFraction',0.2, ...
'Generations', 300, ...
'TimeLimit', inf, ...
'FitnessLimit', -inf, ...
'StallGenLimit', 300, ...
'StallTimeLimit', 30, ...
'InitialPopulation', [], ...
'InitialScores', [], ...
'PlotInterval',1, ...
'CreationFcn',@gacreationuniform, ...
'FitnessScalingFcn', @fitscalingrank, ...
'SelectionFcn', @selectionstochunif, ...
'CrossoverFcn',@crossoverscattered, ...
'MutationFcn',@mutationuniform, ...
'HybridFcn', [], ...
'Display', 'iter', ...
'PlotFcns', @gaplotbestf, ...
'OutputFcns', [], ...
'Vectorized','off');
%fminuncOptions = optimset('Display','iter', 'LargeScale','off');
%options = gaoptimset(options,'HybridFcn',{@fminunc, fminuncOptions});
%options = gaoptimset('Display','off');
%options = gaoptimset('PlotFcns',
{@gaplotbestf,@gaplotbestindiv,@gaplotexpectation,@gaplotstopping})
```

2. 建立了以生产效率为目标函数, 以刀具耐用度、加工精度、表面质量

和机床功率等等为约束条件的优化数学模型，所以综合考虑了诸多因素，为合理选择切削用量提供了理论依据，避免了用经验选择切削用量时可能产生的局限性。

3. 通过实验证明，该优化结果是切实可行的，可用以指导生产。对于本事例中，操作者切削用量的经验选取值为：

切深 (a_p) : 15mm; 进给量 (f) : 0.6mm, 转速 (n) : 25r/min

$v = \pi \cdot d_w \cdot n / 1000$, 此处机床转速单位为 m/min

$v = \pi \times 400 \times 25 \div 1000 = 31.416 \text{ m/min}$

所以，单位时间金属的切除量

$$z_{w1} = 1000v \cdot f \cdot \alpha_p = 1000 \times 31.416 \times 0.6 \times 18 = 3.3929 \times 10^5 \text{ mm}^3 / \text{min}$$

而优化结果为：

$$x(1) = 33.5687$$

$$x(2) = 0.7041$$

$$x(3) = 19.2799$$

Best : 6.43e+009

Mean : 6.43e+009

$$\text{即: } z_{w2} = 1000v \cdot f \cdot \alpha_p = 4.5569 \times 10^5 \text{ mm}^3 / \text{min}$$

对比两切削效果，可得出：

加工效率提高了：

$$\sigma = \frac{z_{w2} - z_{w1}}{z_{w1}} = \frac{4.557 \times 10^5 - 3.3929 \times 10^5}{3.3929 \times 10^5} \times 100\% = 34.3\%$$

4. 对机床和刀具的影响：

采用经验值切削时，机床运行正常，机床噪声小，刀具磨损较小；采用优化参数切削时，机床运行正常，但机床振动加大，噪声较明显，刀具磨损加大。在切削完成后，用精加工试验切工件，能达到精度要求，说明采用该参数加工，对机床未产生无破坏性影响作用。

总之，综合以上分析，可得出以下结论：该优化结果可以用于指导实际加工，在实际生产中较大程度地提高了生产效率；当加工环境和工件改变时，

只须改变输入到主参数界面的数据，就可得到相应的切削用量优化参数，方法简便易行，所以通用性较强。

第 6 章 结论及建议

6.1 结论

切削用量的优化在机械工程领域是一个十分重要的、研究价值较高的一个课题,并一直是实际金加工方面的一个较难深入,亟待解决的问题,是切削加工理论与实践的重要课题。它的选择影响到生产率、生产成本和加工质量。车床是一种主要的切削生产设备,如何确定合理的切削用量以充分发挥它的加工潜力,是一项很有重要的工作。在实际的加工中,当机床和刀具一经确定后,关键就在于确定最优的切削用量。在加工实际中,操作者一般根据日常的工作经验选用切削用量的经验值,完全不能适合切削用量的最优组合,故而难于发挥出机床等硬件设施的功能和效率。本文在切削用量优化中引入了一种新的手段——GA,给传统的切削用量的优化问题注入了新的活力,其优越性在于:执行效率大大提高,有效避免了局部最优解,且实现方法简单可靠。遗传基因算法(GA)为确定最优的切削用量这样的优化问题找到了一条全新的途径。

本论文围绕遗传算法的理论和方法对切削用量的优化问题进行了探讨。建立了目标函数的优化数学模型,约束条件里包括了所有制约加工效率的主要因素,所求结果能应用于生产,为在车削时选择最佳切削用量提供了理论依据,避免了经验选择的局限性。本课题的约束条件的定义具有柔性的思路,设计了输入参数的主界面,当加工条件——比如机床、工件、刀具条件改变时,只须改变输入的参数就可以得到对应生产条件下的最优切削用量,所以该研究成果可用于不同的实际生产场合,通用性较强。总之,本课题的研究结果对于指导生产,充分发挥机床和刀具的功能和效率,降低制造企业的生产周期和成本,提高生产率有着重要的意义,并具有潜在的经济效益和社会效益。

该研究成果已应用于重型机械标准件厂等单位的生产实际中,具用户反应,效果很好,保证了产品质量,提高了生产率,深受用户的好评。用户使用证明能见附件。

该方法可类似地推广到其他类型的金属切削加工中切削用量的优化控制中去。

6.2 建议

由于时间、资料及自身能力的限制,本文在分析探索,论证推理,开拓创新等方面都有许多不尽人意之处,已做的研究也不够深入,更留下了许多没有来得及探讨的问题,以下几个问题有待开展深入的研究和探讨:

1. 对于现代企业来讲,机械制造工业正朝着高精度、高效率 and 低成本的方向发展。本文只侧重讨论了机加工中单工序切削用量的优化分析与计算方法,随着计算机的普及。

本课题侧重研究了“基于遗传算法的最大生产率的车削用量优化”问题,根据生产实践,优化目标应该与企业需要完成的经济指标相一致,常用的优化目标有以下几种:最大生产效率;2、最低成本;3、最大利润。今后的研究可针对以上三个优化目标分别进行研究。

2. 生产中还会经常会遇到既要实现最高生产率(最短加工时间)同时又要达到最低加工成本或其他技术经济指标要求的多目标情况,因此,这时应根据实际需要建立多目标模型。现在的多目标优化方法往往将其转化为各目标之加权和,然后采用单目标的优化技术。但是,这样做存在几大缺点(1)不同性质的目标之间单位不一致,不易作比较;(2)各目标加权值的分配带有较大的主观性;(3)优化目标仅为各目标的加权和,优化过程中各目标的优化进展不可操作;(4)各目标之间通过决策变量相互制约,往往存在相互矛盾的目标,致使加权目标函数的拓扑结构十分复杂。基于传统数学规划原理的多目标优化方法在实际工程优化问题中往往表现出一定的脆弱性,因此,当前有必要研究高效实用的多目标优化与决策的算法及理论。而正是由于多目标问题的广泛存在性与求解的困难性,该问题一直是富有吸引力与挑战性的。

2. 由于实际的生产装备有较多的因素是较难预知和量化的,在本文提出的算法中如果加入模糊逻辑的概念,也许将使切削用量的优化系统更加具有实用性。

3. 通过对国内外发展现状的研究发现,遗传算法本身还有许多可改进之处。遗传工具箱使用内部函数和程序进行运算,界面不是很理想,今后的研究可以在这方面做一些工作,使得计算方法与该课题更能有机的结合。

4. 遗传算法在各个领域已有广泛应用。一般说,根据具体应用领域来有效改进遗传算法是可能的和实际的(而相反,泛泛地对一般问题研究显得极其

困难)。所以,针对具体应用问题,深化研究遗传算法(如参数设置、遗传算子构造等)是当前特别值得提倡的上作。

5. 深入研究在切削用量的优化系统中如何发挥人的作用,如何完善人机交互能力,将人的智能与机器的智能有效的集成,这也将成为一个重要的研究方向。

6. 主界面还需更加完善和友好,提高人机的交互性和对话性,更加适应现场。

展望未来,随着科学技术的不断加速进步,软件技术和测试技术等的高速发展,优化技术在工程中的广泛应用,已成为当今制造业中提高产品质量和劳动生产率,降低劳动成本的有效措施。人们对切削用量的优化工作必将更加深入和细致,研究工作将会取得更大的进展,这对于生产实际中,指导生产、在保证加工中质量的基础上,充分发挥机床和刀具的功能和效率,减少生产成本和周期,具有重大的现实意义。

致 谢

本文的研究工作是在导师黄松和教授悉心指导下完成的，作为黄松和老师的学生，我感到骄傲、自豪，黄松和教授学识渊博，治学严谨，认真负责，态度和蔼，为人亲切，是位难得的好老师。而他扎实的专业知识、执着的科学追求、务实的工作作风更是我一生学习的榜样。几年来，黄松和教授对我的学习和工作给予了很大的关心和支持，倾注了大量的心血，使我在个人修养、工作素质、业务能力等各个方面都得到了很大的进步和提高，在此向导师黄松和教授表示最诚挚的谢意。

论文的完成期间得到了李柏林教授的亲切指导和热心帮助，在此向他表示由衷的感谢。

感谢交大机械系的所有领导和老师，感谢研究生院的全体老师和领导，他们为我们的学业付出了大量的劳动。各位老师一直以来的辛勤工作和教诲，使我能够顺利完成研究生学习，并在专业理论知识和实际科研、工作等各方面受益匪浅！

感谢我亲爱的家人们在生活和精神上给予我的一如既往的关怀、照顾和支持，感谢我亲爱的朋友和同事们给予我的鼓励和无私的帮助。

最后，感谢单位和西南交通大学的对我的培养，这些年的学习生活将成为我的美好的记忆。

参考文献

- [1] 雷英杰等编. MATLAB 遗传算法工具箱及应用. 中国西安: 西安电子科技大学出版社, 2005. 4 7
- [2] 梁尚明 殷国富编. 现代机械优化设计方法. 北京: 化学工业出版社, 2005.
- [3] 孙祥等编. MATLAB 7.0 基础教程. 北京: 清华大学出版社, 2005. 5
- [4] 王小平 曹利民编. 遗传算法——理论、应用与软件实现. 西安: 西安电子科技大学出版社, 2005. 1
- [5] 孟少农等编. 机械加工工艺手册第一卷. 北京: 机械工业出版社, 1991. 9
- [6] 南京工学院等编. 金属切削原理. 福建: 福建科学技术出版社, 1984. 8
- [7] 金属机械加工工艺人员手册修订组编. 金属机械加工工艺人员手册. 上海: 科学技术出版社, 1982
- [8] 许香谷 肖诗纲等编, 金属切削原理与刀具. 重庆: 重庆大学出版社, 1992. 10
- [9] 陈立周编, 机械优化设计方法. 北京: 冶金工业出版社, 2005. 3
- [10] 陈伦军等编, 机械优化设计遗传算法. 北京: 机械工业出版社, 2006. 1
- [11] 云庆夏, 黄光球, 王战权编. 遗传算法与遗传规划. 北京: 冶金工业出版社, 1997. 3
- [12] 刘植义等. 遗传学. 北京: 人民教育出版社, 1982. 6
- [13] 项毅等. 机床电气控制. 南京: 东南大学出版社, 2001. 2
- [14] 王沫然编. MATLAB 与科学计算. 北京: 电子工业出版社, 2005. 1
- [15] 杨荣福 董申编. 金属切削原理. 北京: 机械工业出版社, 1988. 3
- [16] 陆剑中 孙家宁编. 金属切削原理与刀具. 北京: 机械工业出版社, 1992. 3
- [17] 刘鸿文编. 材料力学. 北京: 北京: 高等教育出版社, 1990. 8
- [18] 张维纪编. 金属切削原理与刀具. 浙江: 浙江大学出版社, 1991. 8
- [19] 艾兴 肖诗钢. 金属切削用量. 北京: 高等教育出版社, 1985. 1

[20] 机械加工技术手册编写组. 机械加工技术手册. 北京: 北京出版社, 1991. 7

[21] 中国石油物资装备总公司 石油勘探开发可学研究院 上海工业大学清华大学编. 常用金属材料切削用量数据手册. 北京: 石油工业出版社, 1995. 6

[22] 吴善元, 张永标, 史维明. 金属切削原理与刀具. 北京: 机械工业出版社, 1995. 8

[23] De Jong K A, sarma J. Generation Gaps Revisited. In Foundations of Genetic Algorithms 2, Whitley LD(Ed), Morgan Kaufmann Publishers, 1993

[24] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, MA: Addison-Wesley, 1989.

D. E. Goldberg. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, Massachusetts. 1989

[25] 苏金明 阮沈勇 王永利编. MATLAB 工程数学 北京: 电子工业出版社, 2005. 8

[26] 王启平. 机械制造工艺学 [M]1 哈尔滨: 哈尔滨工业大学出版社, 1995. 2

[27] 周明, 孙树栋. 遗传算法原理及应用. 北京: 国防工业出版社, 1999. 3.

[28] 黄众华. 切削力学. 机械工业出版社, 1992

[29] 张文修编. 梁怡遗传算法的数学基础. 西安交通大学出版社, 2000

[30] 陈国良 王煦法 庄镇泉 王东生. 遗传算法及其应用. 邮电出版社, 1996

[31] 周泽华. 金属切削理论. 上海科学技术出版社, 1992

[32] 张铃, 张钱. 遗传算法机理的研究. 软件学报. 2000, 11 (7): 945-952

[33] 王大力 鲍莉等. 效率最高和成本最低的双目标切削用量优化设计. 洛阳大学学报 1999. 14(4): 57-62

[34] 郭兴. 基于遗传基因算法的切削用量优化. 中南民族学院学报. 1999. 18(2): 27-31

[35] 王晓华. 切削用量的优化选择. 汽车工艺与材料. 1999. 10 (2): 12-13

- [36]徐宗本, 陈志平, 章祥荪. 遗传算法基础理论研究的新近发展. 数学进展. 2000, 29(2): 97-113
- [37]牛小铁. 数控车切削用量的优化. 煤炭技术 2003. 22(8):22-23
- [38]王雅虹. 切削时切削用量的优化设计 洛阳大学学报. 1996. 11(4):33-37
- [39]罗隆福 李勇 周立华. 基于MATLAB平台的遗传算法工具箱的优化设计. 湖南工业职业技术学院学报. 2005. 5(1): 9-11
- [40]Z. Khan, B. Prasad, T. Singh. Machining Condition Optimization by Genetic Algorithms and Simulated Annealing. Computer & Operations Research, 1997 24(71) 647-657
- [41]Kumar algorithm for A. R., M. Pathak, Y. P. Gupta, and H. R. Paraei, a genetic algorithm for distributed system topology design, Computers and industrial Engineering vol. 28, no. 3, pp. 659-670 1995.
- [42]刘勇 刘宝坤 李光泉. 基于MATLAB平台的遗传算法工具. 天津大学学报 2001. 34(4) 490-494
- [43]郭强, 赵晓华, 孙亮. MATLAB 遗传工具箱的设计 北京工业大学学报 2001. 27(3)
- [44]胡义刚, 侯新志. 基于遗传算法的切削用量优化. 机械工程师. 2005. (3) 20-21
- [45]毛明来. 遗传算法在物流系统中的应用研究. 天津大学管理学院 [硕士学位论文] 2004. 6
- [46]官东. 基于遗传算法的物流配送优化研究. 湖南大学 [硕士学位论文]. 2003. 8
- [47]刘昊杨. 遗传算法研究及遗传算法工具箱开发. 天津大学软件学院 [硕士学位论文]. 2005. 5
- [48]罗隆福. 基因控制遗传算法的理论与应用研究. 湖南大学 [博士学位论文]. 2001. 6
- [49]李修飞. 基于遗传算法的生产组织优化研究. 浙江工业大学 [硕士学位论文]. 2003. 12
- [50]王鲁. 基于遗传算法的多目标优化算法研究. 武汉理工大学 [硕士学位论文] 2006. 3

附 录

附录 1: 几个重要的参数表^{[5][7][19]}

附表 1 切削力和切削功率等公式中的系数及指数:

加工材料	刀具材料	加工型式	公式中的系数及指数											
			主切削力 F_Z				吃刀抗力 F_Y				走刀抗力 F_X			
			C_{FZ}	X_{FZ}	Y_{FZ}	Z_{FZ}	C_{FY}	X_{FY}	Y_{FY}	Z_{FY}	C_{FX}	X_{FX}	Y_{FX}	Z_{FX}
结构钢及铸钢 $\sigma_b = 0.637 GP_a$	硬质合金	外圆纵车、横车及镗孔	27 0	1 .0	0 .75	-0. 15	19 9	0 .90	0 .6	-0. 3	29 4	1 .0	0 .5	-0. 4
		切槽及切断	36 7	0 .72	0 .8	0	14 2	0 .73	0 .67	0	--	--	--	--

附表 2 加工钢及铸铁时刀具集合参数改变时切削力的修正系数

参数		刀具材料	修正系数			
名称	数值		名称	切削力		
				F_z	F_y	F_x
前角 γ_0°	-15	硬质合金	$K_{\gamma F}$	1.25	2.0	2.0
	-10			1.2	1.8	1.8
	0			1.1	1.4	1.4
	10			1.0	1.0	1.0
	20			0.9	0.7	0.7
主偏角 k_r°	30	硬质合金	K_{kF}	1.08	1.30	0.78
	45			1.0	1.0	1.0
	60			0.94	0.77	1.11
	75			0.92	0.62	1.13
	90			0.89	0.50	1.17
刃倾角 λ_s°	5	硬质合金	$K_{\lambda F}$	1.0	1.75	1.07
	0				1.0	1.0
	-5				1.25	0.85
	-10				1.5	0.75
	-15				1.7	0.65
刀尖圆弧半径 r_ϵ	0.5	高速钢	K_{rF}	0.87	0.66	1.0
	1.0			0.93	0.82	
	2.0			1.0	1.0	
	3.0			1.04	1.14	
	5.0			1.1	1.33	

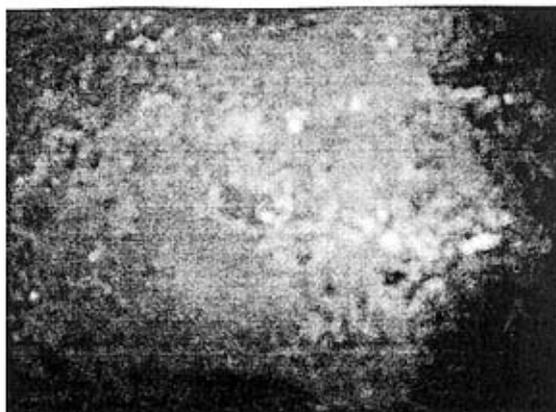
附表 3 刀尖圆弧半径与切削速度不同时 K_R 与 u 值

刀尖圆弧半径 r_ε (mm)	切削速度 $v(m/min)$	K_R	u
0.5	≤ 50	0.0116	1.054
	> 80	0.0244	0.878
1.0	≤ 50	0.0145	1.063
	< 80	0.0362	0.822

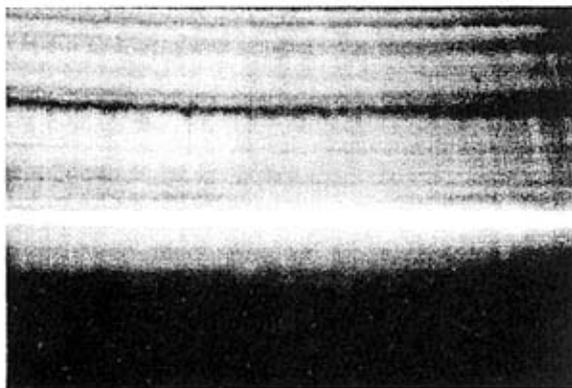
附录 2: 部分实验照片



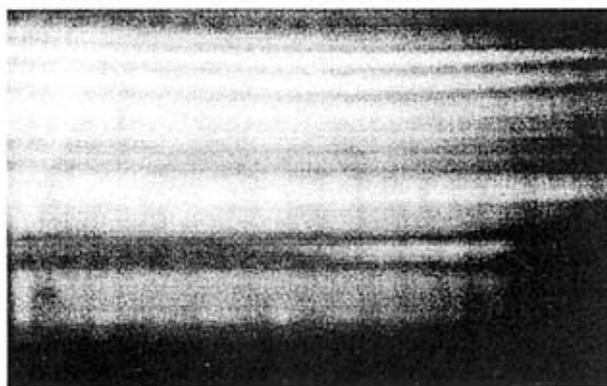
附录图 1 CW61100E 机床加工生产图



附录图 2 毛坯图



附录图 3 经验参数切削加工表面



附录图 4 优化参数切削加工表面

附录 3: 用户使用证明:

证 明

我单位“德阳市重型机械备件厂”是德阳重大技术装备制造基地的配套企业,长期以来,主要承接中国二重集团公司等大型企业的大锻件、盘类零件的加工。由于生产任务重,时间要求紧,长期寻求一种技术支撑来解决大型铸、锻件加工中,在保证加工质量的前提下提高生产效率的途径。经联系,与四川工程职业技术学院合作,开展了“基于遗传算法的最大生产率的车削用量优化”课题的研究,该项目完成后,经在我单位试用,证明主要有下列优点:

在未采用该系统以前,主要靠工人经验选取切削参数,来保证切削质量,有时为了提高生产效率,参数过大,导致出现振动大、粗糙度大,残余应力高等影响加工和产品质量的缺陷。另一方面,选取的参数有时又过小,这就会极大地影响生产效率,导致不能按时交工,影响了企业重装基地的配套能力。

采用了该系统以后,根据不同的加工材料和刀具,输入相关参数,可以很快地寻求到用于该加工的最佳切削参数。经过对比实验后发现,应用该系统加工的产品,振动波纹没有明显加大,粗糙度未提高,生产周期相对缩短,既保证了加工质量,又提高了生产效率。



附录 4: 部分优化程序

1. 主程序

```
%主程序
% clear all;    %清除内存中变量
% clc;         %清除命令窗口
% *****设置初始值*****
%机床的最大与最小主轴转速 (单位:转/分钟)
global nmin;
global nmax;
global dw; %工件直径 (单位:mm)

vmin=pi*dw*nmin/1000 %
vmax=pi*dw*nmax/1000
% reply = inputdlg('输入工件直径')

numberOfVariables = 3;
%输入参数个数
%x(1)切削速度v
%x(2)进给量f
%x(3)切削深度
% *****设置遗传算法的options, 如没有, 则使用ga.m中定义的默认值*****
options=struct('PopulationType', 'doubleVector', ...
'PopInitRange', [vmin 0.15 1;vmax 1.2 25], ...
'PopulationSize', 20, ...
'EliteCount', 8, ...
'CrossoverFraction', 0.8, ...
'MigrationDirection','forward', ...
'MigrationInterval',20, ...
'MigrationFraction',0.2, ...
'Generations', 300, ...
'TimeLimit', inf, ...
'FitnessLimit', -inf, ...
'StallGenLimit', 300, ...
'StallTimeLimit', 30, ...
'InitialPopulation', [], ...
'InitialScores', [], ...
'PlotInterval',1, ...
'CreationFcn',@gacreationuniform, ...
```

```

'FitnessScalingFcn', @fitscalingrank, ...
'SelectionFcn', @selectionstochunif, ...
'CrossoverFcn', @crossoverscattered, ...
'MutationFcn', @mutationuniform, ...
'HybridFcn', [], ...
'Display', 'iter', ...
'PlotFcns', @gaplotbestf, ...
'OutputFcns', [], ...
'Vectorized', 'off');
%fminuncOptions = optimset('Display','iter', 'LargeScale','off');
%options = gaoptimset(options,'HybridFcn',{@fminunc, fminuncOptions});
%options = gaoptimset('Display','off');
%options = gaoptimset('PlotFcns',
(@gaplotbestf,@gaplotbestindiv,@gaplotexpectation,@gaplotstopping))

..
clc;
% *****优化的参数*****
global mu pStep pTol;
% mu = 10;      %惩罚因子
% pStep = 10;   %惩罚因子每次递增的倍率
% pTol = 10;    %收敛的误差 (单位:mm)

ctr = 0;        %初始化反复计数器, 表示代数
flag = true;    %收敛标记
Fval=[];

while flag

    ctr = ctr + 1;    %代数加1
    %优化, xf3_myfun是添加了惩罚的适应度函数
    [x , Fval(ctr) , e , o , p , s] = ga(@xf3_myfun,mu , numberOfVariables , options);
    %计算使用当前的各项的约束值 penalties
    [c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12,c13] = xf3_mycon(x);
    %在当前x值下计算约束值

    %Objective function 添加了惩罚项, 去掉惩罚项就是实际结果
    fval(ctr) = Fval(ctr) - mu*sum(c1(c1<0).^2); %去掉罚函数 (ctr)
    fval(ctr) = fval(ctr) - mu*sum(c2(c2<0).^2); %去掉罚函数
    fval(ctr) = fval(ctr) - mu*sum(c3(c3<0).^2);

```

```

fval(ctr) = fval(ctr) - mu*sum(c4(c4<0).^2);
fval(ctr) = fval(ctr) - mu*sum(c5(c5<0).^2);
fval(ctr) = fval(ctr) - mu*sum(c6(c6<0).^2);
fval(ctr) = fval(ctr) - mu*sum(c7(c7<0).^2);
fval(ctr) = fval(ctr) - mu*sum(c8(c8<0).^2);
fval(ctr) = fval(ctr) - mu*sum(c9(c9<0).^2);
fval(ctr) = fval(ctr) - mu*sum(c10(c10<0).^2);
fval(ctr) = fval(ctr) - mu*sum(c11(c11<0).^2);
fval(ctr) = fval(ctr) - mu*sum(c12(c12<0).^2);
fval(ctr) = fval(ctr) - mu*sum(c13(c13<0).^2);

if ctr>1 && abs(fval(ctr)-fval(ctr-1))<pTol
    %大于第一代，且上下两代的误差小于pTol，则表示进化结束
    %修改收敛标记
    flag = false;
else
    %增大惩罚因子，每次增大pStep倍，但最大不超过1e8，超过就判定非收敛的或无解
    mu = min(mu*pStep, 1e8);
end

%修改'InitialPopulation'参数
%修改'InitialScore'参数
options = gaoptimset(options, 'InitialPopulation', p, 'InitialScore', s);
end %While循环结束

%-----现实优化的结果-----
disp('-----GA Results-----');
display('最后结果: ')
fval = fval(end)
display('切削速度v , 进给量f, 切削深度')
x
msgbox(sprintf('切削速度v : %f\n 进给量f :%f \n 切削深度 :%f', x(1), x(2), x(3)))
display('代数')
NumIterations = ctr
%-----

```

2. 约束条件函数文件

```
function [c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, c11, c12, c13] = xf3_mycon(x)
```

```
%约束条件函数
```

```
%输入参数
```

```
% x(1) 切削速度v
```

```
% x(2) 进给量f
```

```
% x(3) 切削深度ap
```

```
%-----
```

```
%
```

```
% 机床、刀具、工件三方面的参数
```

```
% 共13个约束条件
```

```
% 所有不等式约束全部转换成 ，如果约束条件成立，不等式结果>0的格式
```

```
%
```

```
%-----
```

```
% dw=400; %工件直径(单位: )
```

```
global dw;
```

```
global n;
```

```
n=(1000*x(1))/(pi*dw); % n机床转速, x(1)切削速度v
```

```
%-----
```

```
% 与机床相关的参数
```

```
%-----
```

```
global Zfz; %主切削力指数
```

```
global Xfz; %X轴 主切削力指数
```

```
global Yfz; %Y轴 主切削力指数
```

```
global Cfz; %切削力系数
```

```
global Kfz; %切削力总修正系数
```

```
global Pe; %机床电动机的功率(单位: KW)
```

```
global nitm; %机床的传动功率(单位: )
```

```
global Ms; %机床主轴系统所能传递的扭矩(单位: )
```

```
global Fj; %系统所容许的最大切削力
```

```
global nmin; %最小主轴转速
```

```

global nmax;    %最大主轴转速

global fmin;    %最小进给量
global fmax;    %最大进给量

% Zfz=-0.15; %主切削力指数
% Xfz=1.0;    %X轴 主切削力指数
% Yfz=0.75;    %Y轴 主切削力指数
% Cfz=270;    %切削力系数
% Kfz=0.81558; %切削力总修正系数
%
%
% Pe=22;      %机床电动机的功率(单位: KW)
% nitm=0.75;  %机床的传动功率(单位: )
% Ms=6000;    %机床主轴系统所能传递的扭矩(单位: )
% Kxfz=1.2;   %切削力总修正系数
%
% Fj=10000;   %系统所容许的最大切削力
%
% nmin=1.35;  %最小主轴转速
% nmax=135;   %最大主轴转速
%
% fmin=0.1;   %最小进给量
% fmax=1.2;   %最大进给量

%机床的有效功率
c3=(101.94*1000.^(Zfz+1)*Pe*(nitm))/(60.^(Zfz)*(pi*dw).^(Zfz+1)*(Cfz)*x(3).^(Xfz)
*(Kfz))-n.^(Zfz+1)*x(2).^(Yfz);
%机床主轴的有效扭矩
c4=(203.87*1000.^(Zfz)*Ms)/(60.^(Zfz)*(Cfz)*x(3).^(Xfz)*(Kfz)*pi.^(Zfz)*(dw).^(Zf
z+1))-n.^(Zfz)*x(2).^(Yfz);
%机床进给机构的强度
c5=(1000.^(Zfz)*2*Fj)/(9081*60.^(Zfz)*(Cfz)*x(3).^(Xfz)*(Kfz)*(pi*dw).^(Zfz))-n.^(
Zfz)*x(2).^(Yfz);
%机床的最大与最小主轴转速(单位: 转/分钟)
c6=nmax-n;
c7=n-nmin;
%机床的最大与最小进给量(单位: mm/转)

```

```
c8=fmax-x(2);
c9=x(2)-fmin;
%-----
% 与工件相关的参数
%-----
global J;          %工件断面惯性力矩
global E;          %工件材料的弹性模量
global ff;         %车削时所容许的相对的位移值
global l;          %工件的悬伸长度或装夹在两支承间的距离
global Cfy;
global Kfy;        % 切削力总修正系数
global Xfy;        % X轴 吃刀抗力指数
global Yfy;        % Y轴 吃刀抗力指数
global Zfy;        % Z轴 吃刀抗力指数
global Ko;         %工件装夹方式系数

global apmin;      %最小切削深度
global apmax;      %最大切削深度

global Kr;
global u;
global Rz;         % 微观不平度十点平均高度

% J=128;           %工件断面惯性力矩
% E=2200;          %工件材料的弹性模量
% ff=0.7;          %车削时所容许的相对的位移值
% l=1000;          %工件的悬伸长度或装夹在两支承间的距离
% Cfy=199;
% Kfy=1.25;        % 切削力总修正系数
% Xfy=0.90;        % X轴 吃刀抗力指数
% Yfy=0.60;        % Y轴 吃刀抗力指数
% Zfy=-0.30;       % Z轴 吃刀抗力指数
% Ko=140;          %工件装夹方式系数
%
% apmin=1;         %最小切削深度
% apmax=25;        %最大切削深度
%
% Kr=0.0176;
% u=1.054;
```

```

% Rz=125;      % 微观不平度十点平均高度

c10=(1000.^(Zfy)*(Ko)*E*J*ff)/(9.81*60.^(Zfy)*(Cfy)*x(3).^(Xfy)*(Kfy)*(pi*dw).^(Zfy)*1.^(3)-n.^(Zfy)*x(2).^(Yfy);
%切削加工所容许的最大与最小切削深度(单位: mm)
c12=apmax-x(3);
c13=x(3)-apmin;

%加工表面粗糙度
c11=Kr*Rz.^(u-x(2));
%-----
% 与刀具相关的参数
%-----
global Yv;      %根据切削用量手册查得
global Cv;      %
global Xv;      %
global Kv;      %切削速度的修正系数
global m;       %指数
global T;       %刀具耐用度, 目前, 高速钢车刀和焊接式硬质合金车刀的耐用度一般可取
3600s (60min)

global bb;      %刀杆的抗弯强度
global B;       %刀杆宽度
global H;       %刀杆高度
global L;       %刀杆伸出长度

% Yv=0.45;     %根据切削用量手册查得
% Cv=235;      %
% Xv=0.15;     %
% Kv=0.67;     %切削速度的修正系数
% m=0.2;       %指数
% T=3600;      %刀具耐用度, 目前, 高速钢车刀和焊接式硬质合金车刀的耐用度一般可取
3600s (60min)
%
% bb=196; %刀杆的抗弯强度
% B=45;       %刀杆宽度
% H=45;       %刀杆高度
% L=160;      %刀杆伸出长度

```

%刀具的耐用度

```
c1=(318.31*Cv*Kv)/(60.^(1-m)*dw*T.^m*x(3).^(Xv)) - n*x(2).^(Yv);
```

%刀杆的强度

```
c2=(1000.^(Zfz)*bb*B*H.^2)/(6*L*9.81*60.^(Zfz)*Cfz*x(3).^(Xfz)*(pi*dw).^(Zfz)*(Kfz))-n.^(Zfz)*x(2).^(Yfz);
```

3. 目标函数文件

```
function f = xf3_myfun(x, mu)
```

% 目标函数

%? 如果目标函数值都>0, 则目标函数可以直接作为适应度函数; 否则需要进行尺度变换

% x设计变量

% mu 是惩罚因子 (penalty factors)

%首先找一个值

```
f=-1000*x(1)*x(2)*x(3);
```

```
[c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, c11, c12, c13] = xf3_mycon(x);
```

```
f = f + mu*sum(c1(c1<0).^2); % Add inequality penalty
```

```
f = f + mu*sum(c2(c2<0).^2); % Add inequality penalty
```

```
f = f + mu*sum(c3(c3<0).^2);
```

```
f = f + mu*sum(c4(c4<0).^2);
```

```
f = f + mu*sum(c5(c5<0).^2);
```

```
f = f + mu*sum(c6(c6<0).^2);
```

```
f = f + mu*sum(c7(c7<0).^2);
```

```
f = f + mu*sum(c8(c8<0).^2);
```

```
f = f + mu*sum(c9(c9<0).^2);
```

```
f = f + mu*sum(c10(c10<0).^2);
```

```
f = f + mu*sum(c11(c11<0).^2);
```

```
f = f + mu*sum(c12(c12<0).^2);
```

```
f = f + mu*sum(c13(c13<0).^2);
```

%如果C1>0, 则表示不违反约束条件, 不给予惩罚; 否则加上c1.^2, 以示惩罚。

%如c>=0, c<0返回0, c(c<0) 返回[], c(c<0).^2返回[], sum(c(c<0).^2) 返回0;

%如c<0, c<0返回1, c(c<0) 返回c值, 负数, c(c<0).^2返回c平方值,

4. 部分主界面程序:

工件参数主界面文件:

```
function varargout = fig_gj(varargin)
% FIG_GJ M-file for fig_gj.fig
%   FIG_GJ, by itself, creates a new FIG_GJ or raises the existing
%   singleton*.
%
%   H = FIG_GJ returns the handle to a new FIG_GJ or the handle to
%   the existing singleton*.
%
%   FIG_GJ('CALLBACK', hObject,eventData,handles,...) calls the local
%   function named CALLBACK in FIG_GJ.M with the given input arguments.
%
%   FIG_GJ('Property','Value',...) creates a new FIG_GJ or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before fig_gj_OpeningFunction gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to fig_gj_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Copyright 2002-2003 The MathWorks, Inc.

% Edit the above text to modify the response to help fig_gj

% Last Modified by GUIDE v2.5 21-Aug-2006 09:04:15

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @fig_gj_OpeningFcn, ...
                  'gui_OutputFcn',  @fig_gj_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
```

```
if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

%-----
% --- Executes just before fig_gj is made visible.
function fig_gj_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to fig_gj (see VARARGIN)

% Choose default command line output for fig_gj
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes fig_gj wait for user response (see UIRESUME)
% uiwait(handles.figure1);
set(hObject, 'Name', '工件参数')
if exist('ParamGJ.mat', 'file')==2
    load('ParamGJ')
    SetTextBoxValue('edit_J', ParamGJ.J)
    SetTextBoxValue('edit_E', ParamGJ.E)
    SetTextBoxValue('edit_ff', ParamGJ.ff)
    SetTextBoxValue('edit_l', ParamGJ.l)
    SetTextBoxValue('edit_Cfy', ParamGJ.Cfy)
    SetTextBoxValue('edit_Kfy', ParamGJ.Kfy)
    SetTextBoxValue('edit_Xfy', ParamGJ.Xfy)
    SetTextBoxValue('edit_Yfy', ParamGJ.Yfy)
    SetTextBoxValue('edit_Zfy', ParamGJ.Zfy)
    SetTextBoxValue('edit_Ko', ParamGJ.Ko)
    SetTextBoxValue('edit_apmin', ParamGJ.apmin)
```

```
SetTextBoxValue('edit_apmax', ParamGJ.apmax)
SetTextBoxValue('edit_Kr', ParamGJ.Kr)
SetTextBoxValue('edit_u', ParamGJ.u)
SetTextBoxValue('edit_Rz', ParamGJ.Rz)
SetTextBoxValue('edit_dw', ParamGJ.dw)
end
%-----
% --- Outputs from this function are returned to the command line.
function varargout = fig_gj_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function edit1_Callback(hObject, eventdata, handles)
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of edit1 as text
% str2double(get(hObject, 'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
set(hObject, 'BackgroundColor', 'white');
end
```

end

```
function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a double
```

% --- Executes during object creation, after setting all properties.

```
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit_J_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to edit_J (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

% Hints: get(hObject,'String') returns contents of edit_J as text

% str2double(get(hObject,'String')) returns contents of edit_J as a double

% --- Executes during object creation, after setting all properties.

```
function edit_J_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to edit_J (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
%         See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit_Kfy_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to edit_Kfy (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of edit_Kfy as text
```

```
%         str2double(get(hObject,'String')) returns contents of edit_Kfy as a double
```

```
% --- Executes during object creation, after setting all properties.
```

```
function edit_Kfy_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to edit_Kfy (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
%         See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit_l_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to edit_l (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_1 as text
% str2double(get(hObject,'String')) returns contents of edit_1 as a double
```

```
% --- Executes during object creation, after setting all properties.
function edit_1_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit_1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
```

```
function edit_ff_Callback(hObject, eventdata, handles)
% hObject handle to edit_ff (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_ff as text
% str2double(get(hObject,'String')) returns contents of edit_ff as a double
```

```
% --- Executes during object creation, after setting all properties.
function edit_ff_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit_ff (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit_E_Callback(hObject, eventdata, handles)
% hObject    handle to edit_E (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_E as text
%         str2double(get(hObject,'String')) returns contents of edit_E as a double
```

% — Executes during object creation, after setting all properties.

```
function edit_E_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_E (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit_apmin_Callback(hObject, eventdata, handles)
% hObject    handle to edit_apmin (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

% Hints: get(hObject,'String') returns contents of edit_apmin as text

% str2double(get(hObject,'String')) returns contents of edit_apmin as a double

```
% --- Executes during object creation, after setting all properties.
function edit_apmin_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_apmin (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
%     See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit_apmax_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to edit_apmax (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of edit_apmax as text
```

```
%     str2double(get(hObject,'String')) returns contents of edit_apmax as a double
```

```
% --- Executes during object creation, after setting all properties.
```

```
function edit_apmax_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to edit_apmax (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
%     See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit_Yfy_Callback(hObject, eventdata, handles)
% hObject    handle to edit_Yfy (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_Yfy as text
%        str2double(get(hObject,'String')) returns contents of edit_Yfy as a double

% --- Executes during object creation, after setting all properties.
function edit_Yfy_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_Yfy (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit_Xfy_Callback(hObject, eventdata, handles)
% hObject    handle to edit_Xfy (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_Xfy as text
%        str2double(get(hObject,'String')) returns contents of edit_Xfy as a double

% --- Executes during object creation, after setting all properties.
function edit_Xfy_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_Xfy (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
% See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
```

```
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
```

```
end
```

```
function edit_Zfy_Callback(hObject, eventdata, handles)
```

```
% hObject handle to edit_Zfy (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of edit_Zfy as text
```

```
% str2double(get(hObject,'String')) returns contents of edit_Zfy as a double
```

```
% --- Executes during object creation, after setting all properties.
```

```
function edit_Zfy_CreateFcn(hObject, eventdata, handles)
```

```
% hObject handle to edit_Zfy (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
% See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
```

```
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
```

```
end
```

```
function edit_Cfy_Callback(hObject, eventdata, handles)
```

```
% hObject handle to edit_Cfy (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of edit_Cfy as text
%         str2double(get(hObject,'String')) returns contents of edit_Cfy as a double
```

```
% — Executes during object creation, after setting all properties.
```

```
function edit_Cfy_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to edit_Cfy (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
%         See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
```

```
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
```

```
end
```

```
function edit_Ko_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to edit_Ko (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of edit_Ko as text
```

```
%         str2double(get(hObject,'String')) returns contents of edit_Ko as a double
```

```
% — Executes during object creation, after setting all properties.
```

```
function edit_Ko_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to edit_Ko (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
%         See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
```

```
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
```

end

```
function edit_Rz_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to edit_Rz (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of edit_Rz as text
```

```
%          str2double(get(hObject,'String')) returns contents of edit_Rz as a double
```

```
% --- Executes during object creation, after setting all properties.
```

```
function edit_Rz_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to edit_Rz (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
%          See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
```

```
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
```

```
end
```

```
function edit_Kr_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to edit_Kr (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of edit_Kr as text
```

```
%          str2double(get(hObject,'String')) returns contents of edit_Kr as a double
```

```
% --- Executes during object creation, after setting all properties.
```

```
function edit_Kr_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to edit_Kr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit_u_Callback(hObject, eventdata, handles)
% hObject    handle to edit_u (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_u as text
%         str2double(get(hObject,'String')) returns contents of edit_u as a double

% --- Executes during object creation, after setting all properties.
function edit_u_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_u (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function figure1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
```

```
%-----
```

```
% --- Executes on button press in btn_cancel.
function btn_cancel_Callback(hObject, eventdata, handles)
% hObject handle to btn_cancel (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
selection = questdlg('关闭当前窗口?',...
                    '请确认',...
                    'Yes', 'No', 'Yes');
```

```
switch selection,
    case 'Yes',
        delete(gcf)
    case 'No'
        return
```

```
end
```

```
%-----
```

```
function [value]= GetEditValue(ObjectTag)
%自定义函数
    EditString=get(findobj('Tag', ObjectTag), 'String');
    value=EditString
    %value=str2num(EditString);
```

```
function SetTextBoxValue(TagName, num)
```

```
    set(findobj('Tag', TagName), 'String', num2str(num))
```

```
%-----
```

```
% --- Executes on button press in btn_ok.
function btn_ok_Callback(hObject, eventdata, handles)
% hObject handle to btn_ok (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
% J=128;          %工件断面惯性力矩
% E=2200;        %工件材料的弹性模量
% ff=0.7;        %车削时所容许的相对的位移值
% l=1000;        %工件的悬伸长度或装夹在两支承间的距离
% Cfy=199;
% Kfy=0.82; % 切削力总修正系数
% Xfy=0.90; % X轴 吃刀抗力指数
% Yfy=0.60; % Y轴 吃刀抗力指数
% Zfy=-0.30; % Z轴 吃刀抗力指数
% Ko=140;       %工件装夹方式系数
%
% apmin=1;      %最小切削深度
% apmax=25;     %最大切削深度
%
% Kr=0.0176;
% u=1.054;
% Rz=125; % 微观不平度十点平均高度
%将参数保存在结构数组中
ParamGJ.J=GetEditValue('edit_J')
ParamGJ.E=GetEditValue('edit_E')
ParamGJ.ff=GetEditValue('edit_ff')
ParamGJ.l=GetEditValue('edit_l')
ParamGJ.Cfy=GetEditValue('edit_Cfy')
ParamGJ.Kfy=GetEditValue('edit_Kfy')
ParamGJ.Xfy=GetEditValue('edit_Xfy')
ParamGJ.Yfy=GetEditValue('edit_Yfy')
ParamGJ.Zfy=GetEditValue('edit_Zfy')
ParamGJ.Ko=GetEditValue('edit_Ko')
ParamGJ.apmin=GetEditValue('edit_apmin')
ParamGJ.apmax=GetEditValue('edit_apmax')
ParamGJ.Kr=GetEditValue('edit_Kr')
ParamGJ.u=GetEditValue('edit_u')
ParamGJ.Rz=GetEditValue('edit_Rz')
ParamGJ.dw=GetEditValue('edit_dw')
save('ParamGJ')
setappdata(0,'ParamGJ',ParamGJ)
%
```

攻读硕士学位期间发表的论文及科研成果

- | | | |
|-----------------------------|---------|--------------|
| 《HS-WEDM 加工工艺指标的主要影响因素及其控制》 | 机床与液压 | 2004 年 9 期 |
| 《水压传动的特性及应用分析》 | 机械制造 | 2004 年 11 期; |
| 《液压与气压传动》 | 东南大学出版社 | 2004、8 |
| 《机电专业英语》 | 化工出版社 | 2003、6 |
| 《数控专业英语》 | 机械工业出版社 | 2002、6 |