

Classified Index: TP319

Secrecy Rate: Pubized

UDC: 621.3

University Code: 10082

Hebei University of Science and Technology

Dissertation for the Master Degree

Design and Implementation of Intelligence Testing-Paper Generating System Based on General Item Bank

Candidate:	Liang Haili
Supervisor:	Prof. Wang Xiaohong
Associate Supervisor:	Lu Zhenying
Academic Degree	Master of Engineering
Applied for :	(Part-time)
Discipline:	Computer Technology
University:	Hebei University of Science and Technology



河北科技大学学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品或成果。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：梁海丽

指导教师签名：

2011年5月16日

2011年5月16日

河北科技大学学位论文版权使用授权书

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

☐ 保密，在___年解密后适用本授权书。

本学位论文属于

☒ 不保密。

(请在以上方框内打“√”)

学位论文作者签名：梁海丽

指导教师签名：

2011年5月16日

2011年5月16日

摘要

计算机网络技术、计算技术以及人工智能技术的发展推动了计算机考试系统这种崭新考试模式地不断进步。计算机考试系统摒弃了传统考试模式在时间和空间上面的局限,使考试变得更加灵活,给诸多考生带来了便利,同时减少了教师的重复劳动,提高了考试的效率。而计算机考试系统中的关键部分为智能组卷系统,即如何在已有的试题库中根据教师的要求以及教学的需要自动生成考卷的系统模块。对于智能组卷系统的研究已成为教育学和计算机科学领域研究的热点和重点。

计算机智能组卷的本质是一个基于对试卷的质量构成影响的指标求解多目标参数的优化问题,在本文中我们首先对智能组卷优化问题的各项指标进行了详细地分析并且指出了传统数学模型的不足之处,基于此我们提出了改进的智能组卷优化问题的目标函数和数学模型,该模型赋予各个分指标以相应的权重,将传统的多目标优化问题转化为单目标的优化问题,简化了问题的求解。对于分指标权重的确定,我们采用了定性与定量相结合的多目标评价决策分析方法。

目前,对于智能组卷优化问题的求解存在多种策略,在本文中一种改进的自适应遗传算法被提出并用于求解智能组卷优化问题。较之原有的自适应遗传算法,改进的自适应遗传算法通过使用混合熵来度量种群的多样性,并在交叉概率以及变异概率的确定方面做出了相应的改进,使之更有针对性地求解智能组卷优化问题,仿真实验表明改进之后的自适应遗传算法收敛速率更快以及能够更加容易地搜索到问题的全局最优解。

之后,在本文中我们利用 JSP 编程语言和 SQL Server 数据库技术开发了邢台学院《微积分》课程考试系统 HEBITPGS,智能组卷系统模块是 HEBITPGS 的核心模块。HEBITPGS 系统具有良好简捷的操作界面,可以直接生成以 Word 形式表示的试卷以及试卷的答案,并且可以编辑、修改和打印以 Word 形式表示的试卷以及试卷的答案。

最后,我们对 HEBITPGS 的性能进行了实验性的验证,最终的试验结果表明 HEBITPGS 的设计思路是可行的、实际应用是有效的,它有效地提高了组卷的成功率和试卷的质量以及有效地克服了组卷过程中的未成熟收敛现象。

关键词 智能组卷; 目标优化; 数学模型; 自适应遗传算法; 适应度函数

Abstract

The developments of the computer network technology, the computation technology as well as the artificial intelligence technology impel the progress of computer test system continuously. The computer test system has abandoned the limitation of the traditional test pattern and has brought the convenience for many examinees, simultaneously reduced teacher's redundant work. The intelligence testing-paper generating system (ITPGS), which is the key part of computer test system, has become the hot research area in the fields of pedagogy and computer science.

The essence of ITPGS is solving the optimization problems of multi-objective parameters which will impact the quality of examination paper. In this thesis, the improved objective function and the mathematical model of ITPGS have been proposed firstly. By introducing the weight for each evaluation index, the new model converts the multi-objective optimization into a single-objective optimization. In order to determine the weights, the multi-objective evaluation decision analysis method is applied.

Secondly, an improved self-adaptive genetic algorithm has been proposed to solve the mathematical model of ITPGS. Compared with the existing algorithms, the new strategy can obtain the more quickly convergence speed and find the globally optimal solution effectively and efficiently.

Then, the Calculus examination system-HEBITPGS-is designed and implemented by using the JSP programming language and SQL database technology. ITPGS is the key part of HEBITPGS. Finally, the experimental simulations have been carried out in order to validate the feasibility and effectiveness of HEBITPGS. The experimental results show that the new model can improve the quality of examination paper and overcome the premature convergence in some extent.

Key words Intelligence Testing-Paper Generating System; Objective Optimization; Education System; Mathematical Model; Adaptive Genetic Algorithm; Fitness Function

目 录

摘 要	I
Abstract	II
第1章 绪 论	1
1.1 课题研究的背景及意义	1
1.1.1 课题研究的背景	1
1.1.2 课题研究的意义	2
1.2 国内外研究现状分析	3
1.2.1 国外智能组卷系统研究现状分析	3
1.2.2 国内智能组卷系统研究现状分析	5
1.3 本文主要研究内容	6
1.4 论文结构安排	7
第2章 智能组卷系统技术研究	8
2.1 智能组卷系统的指导思想	8
2.2 智能组卷系统的体系结构	9
2.3 智能组卷系统的量度理论	13
2.3.1 真分数理论	13
2.3.2 潜在倾向理论	15
2.4 本章小结	20
第3章 智能组卷系统的改进数学模型	21
3.1 改进数学模型的目标函数	21
3.2 改进数学模型的约束条件	27
3.3 本章小结	30
第4章 改进的自适应遗传算法	31
4.1 标准遗传算法(Standard Genetic Algorithm-SGA)	31
4.2 自适应遗传算法(Adaptive Genetic Algorithm-AGA)	33
4.3 改进的自适应遗传算法(Improved AGA-IAGA)	35
4.4 本章小结	39
第5章 改进的智能组卷系统的实现与性能分析	40
5.1 使用改进的自适应遗传算法求解组卷系统的改进数学模型	40
5.2 改进的自适应遗传算法的性能验证	42
5.3 HEBITPGS 关键技术详述及代码展示	49

5.4 本章小结	57
结 论	58
参考文献	59
攻读硕士学位期间所发表的论文	63
致 谢	64
个人简历	65

第1章 绪论

1.1 课题研究的背景及意义

1.1.1 课题研究的背景

随着计算机网络技术、计算技术以及人工智能技术的发展,一种崭新的教育考试模式——无纸化自动考试模式^[1]应运而生。考试是教学过程中的一个关键环节,它可以有效地反映教育者的教学效果以及受教者的学习成果,为教育者改进教学手段、受教者转化学习方法提供了重要的参考依据。在传统的纸化考试中,教育者需要花费大量的精力和时间对考试的试卷进行收集、编制、批改以及分析,这不仅造成了大量的冗余劳动而且工作的效率往往低下。

无纸化自动考试系统^[2]是一种结合了现代计算机网络技术和人工智能技术的智能化的、科学化的以及高效化的考试模式,它集成了教育学、传播学、数学、电子科学以及计算机科学的众多优点,它将教育者的教学经验和现代科学技术进行了有效地结合,由计算机根据教育者的意愿将考试的试卷自动生成,既节省了教育者准备试卷花费的精力和时间,又能够较好地体现和反映受教者的学习水平。

相比于传统的纸化考试,无纸化考试具有如下的优点。

(1) 试题库共享和集中管理^[3]

教育者可以根据教学的经验、教学的进度以及教材的更新实时地对试题库进行扩充、更新和改进,由统一的应试机构对试题库进行维护和管理,试题库可以被获得登录权限的教育者用户统一集体使用,具有普遍和广泛的特点;

(2) 无纸化考试具有快捷和高效的特点^[4]

在传统的纸化考试中,一次考试的准备过程,教育者往往要参阅大量的参考文献、准备相当充足的考试材料以及从中选择出具有针对性的考试试题,从而形成最终的纸化试卷。在下次考试的过程中,这些步骤往往要再次被重复,造成了大量的冗余劳动;而在无纸化考试模式中,试题库可以被重复使用,以及对其进行扩充和修改,使得大量的冗余工作被节省,从而提高了出卷的效率,节省了教育者的工作时间;

(3) 组卷的过程体现了智能化和科学化^[5]

为使考试能够真正达到测试受教者学习效果的目的,教育者往往需要准备一份高质量的纸化试卷,为了获得这样一份高质量的纸化试卷,教育者需要根据自身的教学经验来对试卷进行精心地准备,以体现考试的科学性,这无形之中就对教育者提出了更高的职业要求;然而在无纸化的考试系统中,试卷的形成是基于先进的智

能组卷系统完成的,在此过程中,解放了教育者,又由于先进人工智能算法的采用,使得最终获得的考试试卷在某种程度上可以反映受教者的学习水平。

在现代化的无纸化自动考试系统中,智能组卷系统成为了当之无愧的核心构件。智能组卷是指根据教育者的教学经验在已有的试题库中自动地生成满足教学要求的考试试卷的过程。智能组卷系统由相应的指标构成,一般而言指标体系越丰富,组卷系统所生成试卷的质量越高。为了在智能技术的帮助下,生成合理、科学且有效的考试试卷,智能组卷系统需要将组卷问题转化为相应的优化问题,并且建立相应的数学模型以完成对优化问题的求解。由于评价试卷的标准有多种,例如试卷总分、章节分数、题型分数、难度系数分数、考试时间以及试卷区分度等等,因此转化之后的优化问题应为一个多目标的优化问题^[6]。简言之,智能组卷系统可以被转化为一个带有约束的多目标优化问题。

对于智能组卷系统优化问题的求解有多种策略,例如随机选择、回溯试探、补偿算法、误差补偿算法以及进化算法^[7]。前四种策略对于小规模试题库而言,在某种程度上可以生成基本可用的考试试卷,但是,对于大规模的试题库而言,这几种策略往往显得力不从心,即使能够生成质量较高的试卷也是以牺牲大量的组卷时间为代价的,在实际应用中这四种策略已渐渐被教育者忽视。采用进化算法来完成对智能组卷系统优化问题的求解已成为当今研究无纸化自动考试系统的主流技术^[8-10]。由于求解智能组卷多目标优化问题为一个 NP 难问题,即无法在线性时间内获得问题最优解的问题,因此,采用进化算法获得其最佳的近似解成为对 NP 难问题求解的最佳手段。

1.1.2 课题研究的意义

计算机科学技术以及人工智能技术的迅速发展和广泛应用,使得计算机辅助教学这一学科在教育领域得到了广泛的认可和采用。无纸化自动考试是计算机辅助教学的重要部分,它实现了依靠计算机技术建立丰富的试题库,并采用智能组卷系统生成考试所需的试卷。无纸化自动考试系统有效地实现了考试的正规化、科学化以及智能化,将教育者的教与受教者的学进行了合理的分离。

无纸化自动考试系统利用智能化组卷模块对试卷进行组合,将教育者从琐碎繁杂的试卷整理准备过程中解放出来,从而节省了教育者的时间以及提高了考试的效率。智能组卷可以有效地消除教育者掺杂在考卷中的人为主观意志,使得考试更加的正规化、客观化、真实化以及全面化,能够使得考试可以准确地反映教育者对于教学的把控情况,实时地对教学做出调整,有利于提高教学的质量。无纸化自动考试实现了教育者的教与受教者的学的分离,这将在很大程度上促进教育者不断地变更教学手段、强化专业知识以及提高个人职业素养,对整体教学质量以及教学水平

的提高起着关键的作用。

考试是一种基于抽样的统计测试,受教者得知的考试内容为测试的样本总体,而受教者真正面对的考题则是在总体中抽测的样本。在受教者接受考试之前,教育者要对考试的试卷构成进行详细周密的设计,以求最大程度地提高考试的质量。在对考试试卷进行命题的过程中,无论教育者在思想意识上的把控如何,彼此之间总会存在着形式和水平的差异。具体的差异体现在对命题的设计有些是不由自主的侧重点偏差,有些是精心设计、周密安排的考点布置。对于一次考试而言最为重要的就是命题。在传统的纸化考试中,教育者凭借自己多年的教学经验以及参阅大量的相关文献编制试题,在此过程中往往会造成对最终考试试题的编制不够完善的局面,这不是由于教育者教学经验不足或者参阅文献不够全面造成的,而是由于考试本身的特性所致:考试就是一种统计测试,而教育者在总体中选取的样本不足以代表整体样本。

在无纸化自动组卷考试模式中,主旨就是使考试更具科学化以及智能化。对考试首先进行详实地科学验证而后进行具体的周密实施,在考试的过程中引入更多的科学化元素,将考试本身与统计测试理论结合,对考试试卷的编制以及最终所得试卷的质量进行量化分析。构建科学合理、详细充足的试题库,利用现代计算机技术与人工智能技术实现的智能组卷系统生成考试所需的试卷,综合运用计算机科学、教育学以及统计学的方法和理论,结合教育者多年的教学经验,这使得考试更加的科学智能、高质高效,实现了对考试的科学管理和宏观指导,有效避免传统纸化考试取样不准确而造成考试不完善的局面。

无纸化自动考试系统中的主要模块为智能组卷系统,而智能组卷系统关键体现在如何使得生成的考卷最大限度地迎合受教者的要求以及体现受教者的真实学习水平,并使得在考卷生成过程中充分地体现科学性、智能性、随机性以及合理性。因此,在线性时间内,即较短的时间内生成高水平的试卷成为智能组卷系统研究的重点。采用人工智能技术合理有效地解决无纸化考试系统中的组卷问题,对于实现教育的现代化、教学的规范化以及考试的科学化都起着重要的作用,并且在缓解教育者的工作压力、提高教学水平和教学质量方面具有重要现实意义。本课题的研究重点即放置于此,力求更好得解决无纸化考试系统中智能组卷这一难题,利用改进的自适应遗传算法求解简化之后的智能组卷单目标优化问题将使得对无纸化考试系统的研究更具积极、实际和有效的意义。

1.2 国内外研究现状分析

1.2.1 国外智能组卷系统研究现状分析

美国学者朴来希在 20 世纪 20 年代的中期涉足程序教学和教学机器的研究,并

实际完成了一台能够同时进行考试和评分的自动无纸化考试机器，这为日后计算机教学的研究奠定了基础。

到了 20 世纪 50 年代，美国学者思津那在总结其教学经验的基础之上，在 1954 年和 1958 年分别发表了论文《学习的科学和教学艺术》和《教学机器》，并将论文中的研究成果进行了成功的运用，研制出了在美国军方教学中获得成果应用的教学机器，社会对于此机器的教学成果予以了充分的肯定并将其在整个社会教学中予以推广，从此揭开了计算机教学的新篇章。

由计算机辅助教育者完成考试最早开始于 60 年代初的美国，当时隶属于美国 IBM 公司的沃斯顿研究中心于 1958 年设计完成了世界上第一台计算机辅助教学系统，对于计算机教学系统的研究以此为起点展开。到 1960 年美国依利诺斯大学设计并开发了 Programmed Logic for Automatic Teaching Operation 辅助教学系统，之后 1963 年 International Business Machine 公司与美国斯坦福大学合作开发研制了 International Business Machine 1500 辅助教学系统。Japan 于 1963 年左右开发研制出了日本最早的计算机辅助教学系统。Canada 于 1968 年由国家研究院与国内高校联合设计实现了用于教育和考试的辅助教学考试系统。之后 70 年代，随着计算机硬件技术的迅猛发展，社会和学校对计算机的普遍配置使得计算机辅助教学与考试迅速发展起来。

当前国外许多专业的考试部门开始使用计算机辅助考试系统对参加考试的受教者进行测试。美国 Sylvan 学习系统有限公司最主要的业务之一就是计算机无纸化考试，在华的主要业务为负责 TOEIC 考试的宣传和推广。在 Sylvan 学习系统所独有的计算机无纸化考试软件支持下，考试的试题通过计算机网络和电话通讯网络传送到世界各地获得许可的考试机构，受教者通过申请获准后参加考试，考试结束后由考试系统的评分系统对试卷进行评分，考试成绩再次通过网络传送到相应的考试机构。众所周知的 GRE (Graduate Record Examination) 考试成绩是作为进入美国知名大学读书的推荐成绩的，申请人的 GRE 成绩是作为奖学金申请的参考指标的，优异的 GRE 成绩可以反映申请人的专业修养和业务素质。这项著名的无纸化考试模式最初是由 Carnegie Foundation 承办、由 Harvard University、Yale University、Columbia University 以及 Princeton University 四所著名的大学在 1937 年联办的，GRE 考试是申请研究生入学教育的必修考试。TOEFL (Test of English as a Foreign Language) 考试是由美国教育测试服务社举办的英语能力考试，主要是为赴美国、加拿大升学的、非以英语为母语的留学生评核其英语能力的国际认可考试，其考试全部采用无纸化自动考试组织考试的试卷。除此之外，国外还有许多的无纸化考试系统的典型案例，如美国护士协会每年举办的注册护士考试、美国 Novell 公司实行的认证考试、建筑师和证券经纪人行业资格证认证考试等等这些在全球领先的计算机无纸化自动考试系统。

1.2.2 国内智能组卷系统研究现状分析

由于在中国计算机普及的相对较晚以及专业资料和研究人才的匮乏,致使国内对于无纸化考试系统的研究起步相对较晚,但在当今国外对无纸化考试系统的研究如火如荼之时,国内的无纸化考试研究也已经开始成型并取得大量具有重要的理论和实际意义的研究成果。

2001年燕山大学的金涛^[11]将模糊理论和自动反馈方法应用到智能组卷系统的设计中,并采用 Powerbuilder6.5 编程语言实现了一个基于网络的智能考试系统。2003年燕山大学的陈大平^[12]首次提出了基于当前考试系统的多目标智能组卷系统的数学模型,并创造性地将惩罚函数、权重系数以及遗传算法结合起来对此数学模型进行了求解。2004年辽宁工程科技大学的刘艺^[13]采用基于二进制编码和矩阵编码的重叠种群的遗传算法设计并实现了智能组卷系统,并对两种编码方案的优劣进行了总结。2005年华东师范大学的陈丽娜^[14]提出了智能组卷系统的相容性检测方法,并提出了具有自适应全局寻优和智能搜索技术的遗传算法,最终设计完成了一个编码较短以及适应度函数计算容易的智能组卷系统。2005年吉林大学的张月^[15]以经典的测量理论为基础,采用 B/S 结构实现了《数据库》的网上考试系统。

2006年四川大学的陈蕾^[16]采用基于分组自然数编码的改进遗传算法完成了对智能组卷系统的设计。2007年太原理工大学的张彦^[17]将智能组卷的带有约束的多目标优化问题转化为了无约束的单目标优化问题,并采用遗传算法对优化问题进行了求解。2007年中南大学的路景^[18]改进了遗传算法中的自适应调整方法和个体替换策略,并将改进之后的遗传算法应用到基于分段实数编码的智能组卷系统中。2007年大连理工大学的林海涛^[19]在对智能组卷理论进行详细分析的基础之上,对 WDM 模型进行了改进,建立了分步 WDM 模型,更加准确地体现了模型函数值最高的目标。2007年湖南师范大学的徐江涛^[20]提出了三种利用遗传算法进行智能组卷系统的方法,分析并比较了三种方法的优缺点。

2007年天津大学的宋正菊^[21]分析和评价试卷的参数与受测者能力之间的关系,并利用遗传算法设计完成了可以从海量的试题中挑选出最符合受测者当前学习状态的试题的智能组卷系统。2007年上海交通大学的杨秀梅^[22]将组卷约束条件转化为目标函数并采用二进制编码,实现了基于题型整体交叉和双位强制变异的遗传算法的组卷策略。2008年山东师范大学的初文科^[23]提出了基于自适应全局寻优搜索遗传算法的智能组卷系统构建方案,有效地解决了智能组卷系统构建过程中的相关问题。2008年湖南大学的张义良^[24]提出了基于自适应常数遗传算法的智能组卷系统,此系统建立了可以评价指标分布的成卷模式,并根据此模式建立了系统的数学模型。2008年天津大学的李军^[25]利用遗传算法实现了可以按照不同的组卷要求、对不同的课程、

知识点以及题型进行组卷的智能组卷系统,该系统在组卷成功率和试题满意度方面均有良好的表现。

2008 年华东师范大学的吴树锦^[26]设计了一个可以按照试题的各项指标进行快速搜索的、基于遗传算法的智能组卷系统,该系统解决了教学过程中组卷耗时长、试题质量低的问题。2008 年大连海事大学的陈林琳^[27]以 SQL Serve2000 为服务器、VB6.0 为开发工具实际实现了一个计算机在线考试系统。2008 年东北林业大学的王珊^[28]实现了基于《计算机应用基础》的无纸化考试系统,该系统具有良好的扩展性、安全性和可重用性。2008 年北京交通大学的宋莉^[29]应用遗传算法并结合正态分布设计了一套新的智能组卷模型。2008 年辽宁工程技术大学的张强^[30]通过分析知识点网络与试卷的关系,对组卷过程中的多约束问题进行了重新构造,建立了单套和成批试卷结构的数学模型,并采用遗传算法对模型进行了求解。

关于我国智能组卷系统的研究仍在继续,大量有意义以及具有创新性的工作^[31-35]仍在继续,在此我们不予一一详述。

1.3 本文主要研究内容

针对当前智能组卷系统存在的不足,本文的研究内容主要集中在以下四个方面:

(1) 改进智能组卷系统的数学模型

针对已有智能组卷系统数学模型独立考虑各分指标的不足,我们提出了对分指标进行加权集成从而构建组卷系统数学模型目标函数的方法,对于各分指标权值的确定我们采用了拓扑扩展的方法,使得确定的权重可以显著地消除不同的指标分量偏差由于量纲的不同对目标函数效用的影响,从而使求得的权重可以有效地衡量指标分量偏差之间的差异程度。

(2) 改进自适应遗传算法

为了有效地求解智能组卷系统的改进数学模型,我们提出了一种改进的自适应遗传算法,在改进的自适应遗传算法中,我们对引起遗传算法过早成熟的原因进行了分析,从而提出了使用混合熵衡量种群多样性、并基于混合熵对传统自适应算法中的交叉概率和变异概率的计算公式进行了重新改进。

(3) 实现智能组卷系统

在本文中我们利用 JSP 编程语言开发了邢台学院《微积分》课程考试系统 HEBITPGS,此系统具有良好简捷的操作界面,在此系统中我们实现了建库维护、查询检查、交互组卷、自动组卷、试卷打印以及成绩分析等功能。

(4) 性能验证

我们对 HEBITPGS 的性能进行了实验性的验证,最终的实验结果表明 HEBITPGS 的设计思路可行、实际应用是有效的,它有效地提高了生成试卷的质量。

1.4 论文结构安排

本文的主要内容来自于项目实施过程中所用到的知识、在系统设计过程中所采取的方法和研究实现步骤。本文的结构组织安排如下:

第1章是绪论, 主要介绍本论文的研究意义和背景, 以及本论文的研究目的和研究内容。

第2章对智能组卷系统的指导思想、体系结构以及量度理论进行了详细地总结和研究, 为智能组卷系统数学模型的建立、求解以及智能组卷系统的设计与开发提供了理论基础和技术支持。

第3章提出了智能组卷系统改进的数学模型, 将智能组卷问题表示成为一个多约束的单目标优化问题, 其中对于目标函数中权值的确定以及难度系数指标中参数的确定分别提出了相应的解决策略, 数学模型的提出为智能组卷系统的构建提供了重要的理论基础。

第4章在对标准遗传算法和自适应遗传算法进行详细地分析基础之上, 提出了使用混合熵度量种群多样性从而修改交叉概率和变异概率的改进自适应遗传算法, 改进自适应遗传算法的提出为求解智能组卷系统的数学模型提供了先进的技术手段和求解方法。

第5章对基于改进的自适应遗传算法求解改进数学模型的邢台学院《微积分》课程考试系统 HEBITPGS 的设计与实现过程进行了详细地介绍, 对 HEBITPGS 智能组卷模块的性能进行了实验性的验证, 最终的试验结果表明系统 HEBITPGS 的设计思路可行的、实际应用是有效的。

最后是本文的结论。

第2章 智能组卷系统技术研究

2.1 智能组卷系统的指导思想

智能组卷的本质就是采用特定的组卷策略,自动地在事先准备好的、可以反映教学大纲要求的试题库中抽取出合适的试题,使得组成的试卷既可以体现出教学者的教学水平又可以反映受教者的学习程度。为了设计出既满足教育者的需求又符合实际需要的高水平试卷,那么,智能组卷系统应该遵循如下的指导思想:

(1) 依据教学大纲进行组卷,使得最终的考试试卷要能够从横向和纵向全面地反映出教学大纲的要求^[36]。考试试题要体现教学大纲的要求,也就是要能够反映受教者对学过的基本概念、基本定义、基本定理以及基本技巧的理解能力和掌握程度,又要能够举一反三地诱导受教者使用所学习的知识去分析和解决新问题。为了使得考试试卷能够从横向和纵向全面地反映出教学大纲的要求,事先准备的试题库中的试题要足够的充分,试题的质量要足够的高,以满足对受教者四个基本能力的考核。

(2) 组卷之后的考试试题要能够真实地对受教者所掌握的知识水平进行考核并能够举一反三地诱导学生智力的发展^[37]。考试不仅要可以对受教者掌握的知识水平进行测试,并且通过考试还要可以提高受教者求解新问题的能力,这就要求试题的种类必须多样化,能够从不同的侧面对受教者的知识水平以及智力能力进行考核,因此,在组卷的时候就必须充分考虑试题的类型以及对特定知识点的考核力度,以求达到最佳的组卷效果。

(3) 组卷之后的考试试卷在试题难度的安排上面要层次分明,难度呈现阶梯型的分布,应该能够使考试成绩有一定的区分度^[38]。考试的试题如果过难或者过易,都会影响试题对于受教者的考核力度,以至于出现考试成绩无区分的糟糕局面:如果试题太难,超出了所有受教者的知识所能够掌控的范围,那么这将导致受教者对于学习信心的丢失,同时不能真正地反映受教者的学习水平和对知识的掌握程度;如果试题过于简单,那么所有的考生将都能够轻松地处理考试,使得考试成为形式化的测验,不能够激发受教者的学习积极性,从而势必影响受教者的学习兴趣。综上,组卷之后的考试试题应当顾忌大多数中等水平受教者的学习水平,使得试卷至少可以让大多数的中等水平受教者接受,从而使最终的考试成绩可以拉开距离,形成一定的区分度,即组卷要体现试题的难度以及成绩的区分度。

(4) 考试的试题要对受教者有积极的引导意义,也就是说考题要能够引导受教者灵活地使用所学习到的知识去解决新问题^[39]。在目前的教育中,受教者一般对考试都会相当地重视,因此考试在受教者的学习过程中就起到了某种引导性的作用,即考

试引导学生的学习,因此,如果考试的试题偏重于记忆,将造成受教者对学习过程的僵化重复,使得受教者纯粹是为了考试而学习的,这样的话将造成受教者学和考的分离,不利于教育者的教学。考试试题要能够引导受教者灵活地运用知识钻研新问题以及开阔思路,敢于创新。

(5) 考试的试题要全面的反映受教者的知识水平:这种反映既包含横向的宽度又包含纵向的深度,试卷要可以体现出受教者在知识水平和智慧能力方面的差异,因此,这就要求系统组卷时要充分地考虑教育者对知识的传达能力以及受教者对知识的接受能力。

(6) 考试的试题中要保证在试题的用语、符号、题意表达等各方面准确无误,以显示考试的庄重与严肃^[40]。组织考试试卷的过程是一个繁琐而又需要周密安排的智力劳动,试卷的命题者首先要详细地研究考试大纲以及命题安排,明确本次考试的目的、考试的对象、考试的性质、试卷中试题的形式、各种题型的数量以及分值的安排、本次考试考查的侧重点等等;另外,试卷的命题者应该根据编制的考试试题做出相应的答案,便于对考试试卷的评判;最后,考试试卷的组织者应该对最终编制好的试题进行严格详细地审查,及时发现试卷中潜在的错误,并给予及时的改正,同时,应该清楚地注明试题中每个题目的难易程度、区分度、考察的范围、考试的分数以及最终对此试题的评分标准。

2.2 智能组卷系统的体系结构

考试试题质量的好坏需要采用一些量化的指标来度量,指标用以度量考试试卷本身所含有的一些特征参数,这些量化的指标被称为考试试题的体系结构,或称之为组卷系统的体系结构,通过构建科学合理的体系结构来指导智能组卷系统正常稳定地工作。

系统的体系结构包含可以对试题的特征参数进行描述的指标集合,这些指标可以对试卷的属性、特征以及功能进行定量地描述,是构建智能组卷系统的关键。系统的体系结构越丰富,那么,将来组成试卷的质量就会越高。在现有文献中可以查实的、已用的指标集合列举如下(对于一个智能组卷系统而言,其体系结构中所包含的指标体系可以根据系统的需求呈现一定的差异性):

(1) 题型(Item Topic): 考试的试卷中所包含的试题类型大致分为如下的几种:填空题、选择题、判断题、概念论述题、简答题、分析计算题以及综合题等;

(2) 所属章节(Chapters and Sections): 考试试卷中的试题要与教学大纲中的特定章节相互对应,试题不能超出教学大纲所规定的范畴;

(3) 知识点(Knowledge Point): 即考试试卷中的考察点,是考试试卷中考试试题的内在本质,抽象的知识点以具体的考试试题的形式得以呈现;

(4) 难度系数(Degree of Difficulty): 在对考试试卷进行命题的过程中, 命题者需要根据不同的考试对象以及不同的考试形式实时地调整考试试题的难易程度, 依据组卷最终需要符合要求, 命题者应对试题的难度系数进行重复的调整, 以求得到最能够反映考试对象知识水平并且难易程度适中的考卷, 根据胡中锋和李方^[41]对难度系数的定义, 难度系数的表达式为公式(2-1)所示:

$$D_i = 1 - \frac{Average_i}{Full_i} \quad [41], \quad (2-1)$$

式中, D_i ——试卷中试题*i*的难度系数, $Average_i$ ——在本次考试中试题*i*的平均得分值, $Full_i$ ——在本次考试中试题*i*的满分分值。

同一试题的难度系数不仅依赖于特定的测试群体, 即便是同一测试群体还同样依赖于不同的测试环境, 譬如, 同样一组受测试的考生, 在刚刚学习完课程之后便进行测试与学习完课程一个学期之后再进行测试所得到的试题的难度系数将会有相当大的差异, 因此, 为了使难度系数更加客观真实、科学有效, 需要根据教育者的工作经验对难度系数加以适当地调整, 并将难度系数进行模糊化处理, 使其可以对应容易、中等难度、较高难度以及最难四个试题的难度等级, 其中, “容易”所对应的难度系数为 $D_i=0\sim0.25$, 模糊化之后为 $D_i=0.2$; “中等难度”所对应的难度系数为 $D_i=0.25\sim0.5$, 模糊化之后为 $D_i=0.4$; “较高难度”所对应的难度系数为 $D_i=0.5\sim0.75$, 模糊化之后为 $D_i=0.7$; “容易”所对应的难度系数为 $D_i=0.75\sim1$, 模糊化之后为 $D_i=0.9$ 。

一般对于一次比较正规的考试而言, 试卷的平均难度系数一般保持在 0.5 左右即可, 即使得大多数的考生可以正常地对试题进行作答, 这样最终的考试成绩才可以基本服从正态分布, 使得考试的效果发挥到最佳, 考试才最具有意义, 图 2-1 是邢台学院 2010 级旅游管理专业《微积分(上)》课程考试成绩的分布图, 可以看出考生的考试成绩分布近似于正态分布, 本次考试的所有试题难度基本保持在 0.5 左右。

(5) 试题区分度(Item Discrimination): 为了使考试达到对受教者进行有效测试的目的, 试卷的命题者在对试卷进行命题的过程中需要将试题进行合理的编排, 以便使得考试的结果可以将高水平的测试群与低水平的测试群进行区分。如果所有的考生对同一题目的作答要么全对、要么全错, 那么这样的考试便失去了意义。本文中, 在对试题的标准答案和评分原则进行详细研究的基础之上, 本文采用王友仁等^[42]提出的二列相关法对试题区分度进行计算, 计算公式如(2-2)所示:

$$ID_i = \frac{A_w - A_l}{\sigma_i} \sqrt{wl} \quad [42], \quad (2-2)$$

式中, ID_i ——试卷中试题*i*的区分度, w ——在本次测试中对试题*i*的解答通过测试的考生人数占总考生人数的百分比, l ——在本次测试中对试题*i*的解答没有通过测试的考生人数占总考生人数的百分比, A_w ——在本次测试中对试题*i*的解答通过测试

的考生的平均分, A_i ——在本次测试中对试题 i 的解答没有通过测试的考生的平均分, σ_i ——全体考生对试题 i 解答的总分的标准差。一般认为, 当某试题的 ID_i 值大于 0.25 时, 即认为该试题具有良好的区分度。

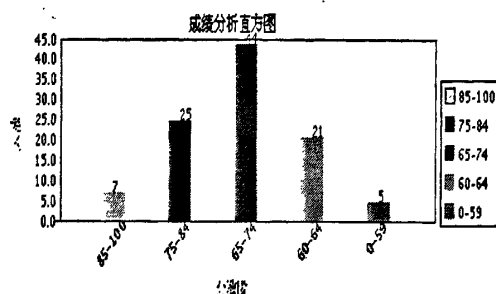


图 2-1 邢台学院 2010 级旅游管理专业《微积分(上)》课程考试成绩的分布图

Fig.2-1 The distribution of calculus scores about 2010 professional tourism management in Xingtai University

(6) 试题信度(Item Reliability): 此指标用以度量考试作为测试工具的可信程度, 此指标在很大程度上决定了考试的信用度, 如大家所熟知的国家公务员考试、研究生入学考试等这种高级别考试都是可信度相当高的测试。通常此指标依据如下的原则来确定: 选定知识水平和智力水平相当的两组学生, 在隔离的环境下使用同样一套试题对这两组考生进行测试, 检验最终考试结果的一致性。如果最终考试结果显示的一致性越高, 则说明考试试卷的信度越大, 考试越可信。试题信度可以参考试题的难度系数、试题的区分度、试题的形式以及试题的内容等指标来测量。

(7) 试题效度(Item Validity): 此指标用以衡量所实施的本次考试的成功程度, 所谓考试的成功程度即指代试卷的试题内容与教学大纲所规定的教学内容的符合程度。效度高的考试能够比较准确地对受教者所掌握的知识的深度和广度以及运用知识解决新问题的能力进行真实地度量, 效度低的考试往往会违背教学大纲的指导, 对受教者的知识水平往往不能够给出正确地测量, 造成考试资源浪费。严格地依据教学大纲的要求、教育者的教学经验以及受教者的知识掌握水平进行试卷的命题是保证试题效度的重要的基础。

(8) 试题认知层次(Item Cognitive Level): 此指标反映了考生对独立地获取知识以及掌握运用知识的能力, 考生的认知层次大致可以分为如下的六个层次: 记忆、领悟、使用、分析、汇总以及评论, 其中最低层次为记忆, 最高层次为评论, 层次越高表明考生之间的共性越少, 层次越低表明考生之间的共性越多。

(9) 试题使用率(Item Utilization Rate): 本次考试的试卷中所使用的试题在其他考试中被采用的次数以及距离上一次考试最近的时间是试题使用率主要参考的两个指标。为了保证考试试卷的质量, 诸如保密性、公平性以及安全性, 在使用智能算法

进行组卷时必须控制纸卷中试题的使用率。

(10) 答卷时间(Answer Time): 即考生完成对试卷上面所有题目的作答所需要的时间, 在这个规定的时间内并不能保证所有的考生都可以完成试题作答, 仅是对大多数的考生而言。答题时间用整数表示, 一般单位为分钟, 诸如 150 分钟、180 分钟等等, 该值的确定一般依赖于试题的命题者和组织者长期的教学经验和组织考试的经验而定并针对不同难度级别的考试加以适当地调整。

下面将上述的十项指标归纳在表 2-1 中, 并对不同的指标引入不同的符号标识, 便于后面展开论述。

表 2-1 考试试题的指标体系

Tab.2-1 The index system of examination

指标的名称		指标的表示符号	指标的说明
试题 编号	科目		考试的试卷中所包含的试题类型
	题型	IT	
	序号		
所属章节		CS	考试试卷中的试题要与教学大纲中的特定章节相互对应
知识点		KP	考试试卷中的考察点
难度系数		D	考试试题的难易程度
试题区分度		ID	使得考试可以将高水平的测试群与低水平的测试群进行区分
试题信度		R	用以度量考试作为测试工具的可信程度
试题效度		V	用以衡量所实施的本次考试的成功程度
试题认知层次		CL	反映考生独立获取知识以及掌握运用知识的能力
试题使用率		U	考试的试卷中所使用的试题在其他考试中被采用的次数以及距离上一次考试最近的时间是试题使用率主要参考的两个指标。
答卷时间		AT	考生完成对试卷上面所有题目的作答所需要的时间

上述的各指标之间并不是相互独立的, 它们之间也存在一定的关联, 譬如试题的难度系数与试题区分度之间、试题的难度系数与试题认知层次之间就会存在相互关联的关系:

(1) 难度系数与区分度之间的关联。试题的难度系数和试题的区分度都是评价考试试题质量好坏的量化指标, 试题的难度将会影响到试题的区分度, 如果试题的难度系数过于高将造成试题区分度较低的局面, 例如, 当试题的难度系数接近 1 时, 那么所有的考生对试题的作答几乎全不对, 此时便无法区别测试群体, 试题的区分度为 0; 当试题的难度系数接近 0 时, 那么所有的考生对试题的作答几乎全对, 此时

亦无法区别测试群体,试题的区分度也近乎为0;一般认为,当试题的难度系数位于0.5左右时,考生的测试成绩会被广泛接受,试题的区分度达到最高。

(2) 难度系数与认知层次之间的关联。试题的难度系数指代试题的难易程度,而试题的认知层次反映依据认知领域对考试的目标进行定位的结果,这种定位基于对诸多考试的目标外在特征和内在本质详细地分析和研究基础之上。试题的认知层次按照级别高低被依次归类,其中后一个层次较之前一个层次要包含更丰富的内涵。一般认为,试题的认知层次越高,那么试题的难度系数越大;试题的认知层次越低,试题的难度系数越小。反之亦成立。

2.3 智能组卷系统的量度理论

在智能组卷系统中,对于试题库的构建往往需要基于某种测量理论原则,其次采用计算机技术使之成为现实。上述的对试题库中试题进行评价的各种指标,例如,难度系数、区分度、效度、信度、认知层次等等,都可以被看作是这种测量理论的统计量或是项目参数,这些项目参数(指标)是需要依据测量理论计算的。测量理论不仅为试题库的评价指标提供计算的方法,同时也为试题评价指标系统的构建提供理论依据,还为考试的可信性、有效性以及权威性提供参考的标准和框架。在本文中,本文仅对以下两种重要的测量理论进行介绍:真分数理论(True Score Theory)和潜在倾向理论(Latent Trait Theory)。

2.3.1 真分数理论

真分数理论^[43]是最早使用数学对测量进行刻画的理论,它起始于十九世纪末,二十世纪30年代形成了完整的体系,50年代格李可森在其著作中对其完整的数学理论形式进行了详述,到1968年罗德和挪威科在他们的论著中实现了经典真分数理论向现代测量理论的转换。真分数(True Score=TS)是指考生在考试特征,如知识水平、能力上的真实值,当通过某种特定的测试工具或者手段对考生进行测量时,所反映出来分值被称为观察值(Observation Score=OS),由于测量手段与测量工具本身存在着测量误差(Error Score=ES),因此,观察值并不等于真实值,换言之,观察值应该由真实值和测量误差构成。而要得到真实值就必须将测量误差从观察值中分离出来。

真分数理论基于如下的三个假设来解决上述问题:

(1) 真分数的稳定性,即考生的知识水平、学习能力在特定的时间段内应该是个恒定值,应具有稳定性,即考生的知识水平在短时间内不会急剧下降或是提升;

(2) 测量误差的随机性,即测量误差的生成是完全随机的,测量误差应服从均值为0的正态分布,当测量误差大于0时,观察值大于真实值;当测量误差小于0时,观察值小于真实值,这将会导致真实值的波动,但是,当测试的次数足够多时,正负偏差将会抵消,测试误差的平均值将会趋于0。另外,测量误差与真实值之间是相

互独立的。

(3) 观察值等于真实值和测量误差之间的和, 即 $OS=TS+ES$ 。

基于上述的假设, 真分数理论推导出了两个重要的结论: 第一, 真实值等于多次观察值的平均值, 即 $TS=E(OS)$; 第二, 在多次测试试验中, 观察的方差等于真实值的方差与测量误差的方差之和, 即 $\sigma^2(OS)=\sigma^2(TS)+\sigma^2(ES)$ 。

真分数理论主要包含以下基本概念: 信度、效度、项目分析、常模以及标准化等, 下面对这些基本概念的含义进行一一地详解:

(1) 信度。指的是测试结果与真实值的一致性程度, 是测量理论中的核心概念。数学定义为真实值方差所占观察值方差的比率。由于真实值的方差根本无法计算, 因此, 信度是无法进行直接地数学运算的, 需要借助平行测试(Parallel Test)来解决这一问题, 平行测试是指可以对同一被测试的对象的同一测试特征做出相同测量的不同测试方式。可以采用平行测试的不同测试方式对同一对象的同一特征进行多次测试, 这样就会得到一个测试分数的数列, 那么这一数列的平均值就被当作是测试的真实值。基于平行测试的测量方法有多种, 其中比较著名的测试方法为重置测试信度、等效格式信度、隔半划分信度、哥伦巴赫系数、K-R20 公式和 K-R21 公式、科以特信度等。

(2) 效度。指代测试结果的有效程度, 即测试的质量与主测试者主观期望质量之间的吻合程度。效度是对测试者潜在的特征进行测量的保证, 对效度的测量同样只能采用间接的方法(行为主义 S-R 公式), S 表示测试者接受的刺激, R 表示测试者对刺激 S 所做出的反映, 而测试者的被测特征是位于刺激 S 和反映 R 之间的, 在测试者的大脑接受刺激的过程中会对刺激 S 做出处理, 处理的结果会以信号 R 的形式输出。对于效度的求解存在多种方案, 如同步时间、预先测试、表面现象、相互容纳、协同合作、事实证明等。通常为对效度研究的规范性, 需要将其分为三大类: 一是内容效度, 测试的内容与测试范围的吻合程度; 二是结构效度, 测试的结果与测试的预想之间的一致性程度; 三是效标关联效度, 测试的结果与测试外在指标之间的一致性程度。

(3) 项目分析。为了提高测试的质量, 必须保证测试的信度和效度, 因此, 在编制和研究试题的题型以及试题的内容和功能时, 就必须制定一套识别和挑选测试项目的标准, 这被称为项目分析, 项目分析中最为关键的是试题难度的分析和试题区分度的分析。试题的度量指标主要是测试的通过率, 即通过测试的人数所占全体参加测试人数的比率。试题区分度是作为试题质量评价的重要指标。

(4) 常模。仅从考试试卷的局面得分是很难鉴别考生某方面的测试特征的, 常模就是在多次测试的总体样本中抽取样本, 以抽取样本的平均数为基点, 来评定考生成绩的高低位置, 这种判定是可以通过将原始成绩转化为量化表分值来进行的。

(5) 标准化。即对测试进行严格的规范以降低测试过程中出现的测试误差,在测试的过程中需要对测试的流程、测试对象、测试环境、测试手段、测试时间以及分值规定进行统一的控制,以求获得等效的测试效果,这种思想来源于心理学中对影响测试参数指标变量的控制方法。

2.3.2 潜在倾向理论

在实际应用中真分数理论是很不足的,它无法实现项目参数和被试能力参数的不变性。潜在倾向理论^[44]就是填补了此项空白。在该理论中选出来的被试的数据组和对该项目的描述参数的估计出来的值是没有关系的。此外,应用该理论可以针对各个值提出描述准确性的指标大小,并且在测试前就可以做出预测。这种理论的这些优点可以直接应用于教学中的建设题库等工作中。

项目反应模型可以用一个单调递增的函数来描述,这个函数的自变量是被试的能力,因变量是测试结果即正确作答率。这种项目函数可以称为项目特征函数(ICF: Item Characteristic Function)或项目反应函数(IRF: Item Response Function)。它包含一定数目的项目参数(如难度参数、区分度参数、猜测参数等),现有理论可以对这些数值进行估计,显然自变量确定再利用构建出的模型,就可以得到被试能力的估计值。

对 IRF 的研究就需要先构建一种模型,一般有静态和动态的两大类型。顾名思义,静态是一种固定时刻上的描述,而动态是以时间为变量而进行的描述。目前比较成熟的是静态模型。静态模型也有多种,它们可分为单维的和多维的、二值记分的和多值记分的、正态钟型的和逻辑斯谛模型的等等。下面几种是常用的静态模型。

逻辑斯谛模型是一种基于两种假设的。即自变量只有一个,结果只有 0 和 1 两个。从这点来看这是一种很简单的假设,处理起来简单,可以回避许多多变量和多结果带来的复杂影响的问题。逻辑斯谛模型的数学表达式如(2-3)式所示:

$$P_i(\theta) = \frac{e^x}{1+e^x} = \frac{1}{1+e^{-x}} \quad [44], \quad (2-3)$$

其中, e ——自然对数的底; x ——为一个任意符号。当 x 用不同的代数式表示时,就形成了各种不同参数的逻辑斯谛模型,如双参数逻辑斯谛模型、三参数逻辑斯谛模型和单参数逻辑斯谛模型等。

(1) 双参数逻辑斯谛模型。双参数逻辑斯谛模型的数学表达式如下式(2-4)所示:

$$P_i(\theta) = \frac{1}{1+e^{-D\alpha_i(\theta-b_i)}}, i=1,2,\dots,n \quad [44], \quad (2-4)$$

其中, $P_i(\theta)$ ——某个随机选出的能力值为 θ 的考生作答正确的试题 i 的概率; α_i 和 b_i 是项目 i 的两个不同的参数; n 为项目数; D 是量表因子,通常取值为 1.7。在这里可以看出该双参数假设包含着这样一种结论:结论不受猜测因素的影响,正确

做答率只取决于自变量的大小。

(2) 三参数逻辑斯谛模型。三参数逻辑斯谛模型的数学表达式如下式(2-5)所示:

$$P_i(\theta) = c_i + \frac{1 - c_i}{1 + e^{-D \times a_i \times (\theta - b_i)}}, i = 1, 2, \dots, n^{[44]}, \quad (2-5)$$

它是在双参逻辑斯谛模型中增加了参数 c_i 后得到的。其中, c_i 是猜测函数, 即使自变量特别低, 仍然会保持的基本正确做答率。下图 2-2 所示的是一个典型的三参数逻辑斯谛模型项目特征曲线。

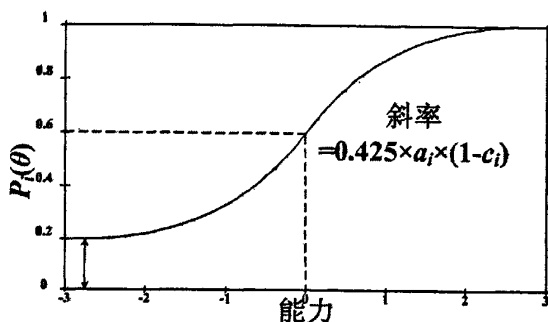


图 2-2 三参数逻辑斯谛模型曲线

Fig.2-2 The curve of Logistic with three parameters

图中横坐标表示考生的能力值, 被称为能力量表; 纵坐标表示考生的正答概率, 称为概率量表。从图中可以看出各个参数的涵义。参数 b_i 等于项目特征曲线(项目特征函数的图象称为项目特征曲线, 简称 ICC)上斜率最大处在能力量表上对应的值, 对应于 b_i 点的 ICC 的斜率为 $0.425 \times a_i \times (1 - c_i)$ 。式中 a_i 为项目区分度, a_i 值越大, ICC 越陡, 参数 c_i 是在概率量表上度量的, 它表示能力极低考生的正确做答概率。显然, 由于该模型中包含了参数 c_i , 即意味着能力极低的考生也有可能答对该项目, 因此, 这一模型适合于多项选择题的成就测验或能力测验。

(3) 单参数逻辑斯谛模型。单参数逻辑斯谛模型的数学表达式如下式(2-6)所示:

$$P_i(\theta_j) = \frac{e^{(\theta_j - b_i)}}{1 + e^{(\theta_j - b_i)}}, i = 1, 2, \dots, n^{[44]}, \quad (2-6)$$

式中, θ_j ——考生 j 的能力水平值; b_i ——项目 i 的难度, 随项目的不同, b_i 值是变化的; $P_i(\theta_j)$ ——第 j 个考生答对项目 i 的概率。

对该表达式进行分析, 可知, 当考生能力值低于项目难度值时, 即 $\theta_j < b_i$ 时, $(\theta_j - b_i)$ 为负值, 此时, 正答率小于 0.5; 能力越低, 难度越高, 该考生成功的概率越接近于零; 当考生的能力 θ_j 强于项目难度 b_i 时, 即 $\theta_j > b_i$ 时, $(\theta_j - b_i)$ 为正值, 此时, 正答率将大于 0.5 当 $(\theta_j - b_i)$ 的值越大, 考生成功的概率就越接近于 1。

在这些分析中可知单参数只不过是以上两者的特例, 不过还是有一些自己的特

性的,在实际研究中有着广泛的应用。第一,该模型的参数数量比其余的两种都要少,处理起来更加简单;第二,对参数进行估计时,所要解决的问题更加少和单一;此外,对自变量和因变量的估计是独立进行的,减少了彼此间的影响,可以更加客观有效,函数关系不会受考生或项目的影响,使用起来在评价中实现了更加公平。利用这一特性使用者就可以客观地对项目难度参数和考生能力参数分开估计。

这三种常用模型各有优缺点,基于参数设计和模型中的不同假设,它们各自适用于不同的数据处理。在使用时必须根据自己的需要选择模型。在选择时需要考虑的因素很多,通常主要是考虑要能满足模型的假设。例如,单参数和双参数中都没有猜测因素,而三参数模型则考虑了在考生能力低下时可以利用自己的猜测而有一定的正确做答率,所以三参数模型在客观题中是唯一可用的方法。还有单参数各个项目之间符合相同的函数关系,其余两个不满足这一点,所以在不能确定各个项目都符合相同的函数关系时,是不能用单参数模型的。不过在实际应用中,还应该根据具体情况进行分析,因为实际情况并不是理想的,例如在项目模型具有一定的稳定性时,即使情况和假设有偏差,仍然可以使用单参数模型。这样就可以把复杂的现实问题简单化,便于处理。

从上面的描述中可以看出每一个反应模型都有一定数目的参数,而参数值在实际中是不能直接测量得到的,往往需要进行估计。比如二值记分中,参数估计值的根据是0、1得分矩阵。设一个测验项目数为 n ,被试数为 m ,各被试对各项目的反应情况就可以用一个 $m \times n$ 的得分数据矩阵(即0、1矩阵)来描述。利用该矩阵进行参数估计的基本过程如下:

Step 1: 确定初始值。先确定一个初始值,这样在下面的运行中可以更加直观。具体方法可以根据经典测量理论计算得到项目难度和区分度的值,或假设项目参数为某个常数:

Step 2: 根据这个设定的初始值,运用数学方法得到每一个被试的能力估计值 θ ;

Step 3: 把所有被试根据估计值分组,平衡各组的估计值之和,使得每一组中的被试具有大致相近的能力估计值,把同一组内估计值的平均数作为该组估计值的代表值,用于下面的运行;

Step 4: 根据得分数据矩阵,计算出针对各个项目各个被试的正答率,每次的运算遵循相同的规律;

Step 5: 以各组被试能力估计值的代表值为横坐标,以该组被试答对某项目的正确做答率为纵坐标作图,对于每一个项目都可以做出一幅图,如图2-3中所示。图中的曲线称为项目反应函数。

Step 6: 再假设寻找一组对项目参数进行估计得出的值,并将其代入已经应用的反应模型中,使得到的答对概率值能与图中根据之前的运行得到的经验项目反应函

数可以很好地符合;

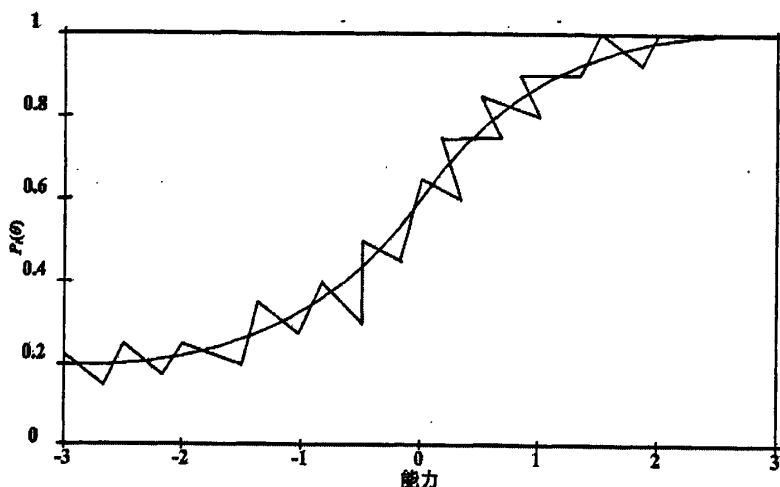


图 2-3 经验项目反映函数

Fig.2-3 The experimental item response function

Step 7: 在这组项目参数估计值得到后,反复地进行 Step 2 至 Step 6 步骤,即不停地进行尝试,并且在每次尝试中再寻找项目参数新的估计值。

从 Step 1 到 Step 7 无限次的重复,并且每次对得到的结果进行分析,根据分析对能力和项目参数的估计值修正,使之更加符合实际,直到项目参数值的大小不再变化或者变化很小。这个估计过程是很麻烦的,一般需要依赖于专门设计的运行能力强大的计算机软件。无论采用什么估计方法,基本过程还是不变的,一直按照上述的过程进行。

对参数进行估计研究比较多,得到的可行方法很多,最常用一个的方法就是极大似然法。这种方法的过程是,在对参数进行估计的过程中,运用似然函数来寻找一组项目和被试能力的参数估计值,使得在取这组估计值时,出现所观察到的实际反应数据的可能性最大。在极大似然估计中,似然函数的一般形式如下式(2-7)所示:

$$L(U|\theta, a, b, c) = \prod_{j=1}^m \prod_{i=1}^n P_{ij}^{u_{ij}} Q_{ij}^{1-u_{ij}}, \quad (2-7)$$

其中, U ——由元素 U_{ij} 组成的 $m \times n$ 项目反应矩阵; P_{ij} ——被试 j 对项目 i 的正确作答率; Q_{ij} ——被试 j 对项目 i 的答错率; a, b, c ——由各项目参数所组成的矢量; m ——被试人数, n ——测验项目数。数学运算中连乘是比较复杂的,在计算中往往会尽力避免。因此,在实际应用中会取对数进行简化计算,对数简化后得到函数式如下式(2-8)所示:

$$\ln L(U|\theta, a, b, c) = \sum_{j=1}^m \sum_{i=1}^n (u_{ij} \ln P_{ij} + (1-u_{ij}) \ln Q_{ij}), \quad (2-8)$$

利用似然参数进行估计,通常的做法是寻找使似然函数达到最大值的相应的参数值。当然对数会和似然函数在同一点取得最大值,因此只要使对数函数达到最大值即可满足似然函数最大的要求。在运算中令自然对数关于自变量的偏导数为0即可。对于有三个参数的函数模型,需要解的表达式为方程组(2-9):

$$\frac{\partial \ln L}{\partial \theta_j} = 0, \frac{\partial \ln L}{\partial a_i} = 0, \frac{\partial \ln L}{\partial b_i} = 0, \frac{\partial \ln L}{\partial c_i} = 0, \quad (2-9)$$

由于被试数为 m 、项目数为 n 、每项的参数个数为 3,该方程组应该共有 $m \times 3n$ 个方程。这些方程一般都不是线性的,为了得到方程的解,就可以采用数值计算的方法。

任何方法总会存在盲点,极大似然法估计参数时有两条缺点:首先被试技能的之前的基础知识是没有被该方法考虑的,这忽略了各种能力之间的联系和依赖性;此外,估计中是没有满分和零分的选择的,也就是会造成一定的情况遗漏。为解决这两个问题,研究人员将贝叶斯方法引进来,应用这种方法,只需要估计前所有分组中的一组的能力分布情况,就可以进行比较准确的估计,就可以满足要求。

上述的估计方法都是完全凭主观经验,为了保证估计的准确性必须要进行事后的验证,确认其可靠性。以往的理论利用信度来表示测试的可靠性,在信度的表示中,被试样组的不同会给测试的结果带来不同程度的影响,这样使用者就会增加调试的工作量。为了解决该问题,可以在信度值中加入与试样组保持相关的参数,将信度由常数变为函数,这样就不再和被试样组有关了,有了更大的适用性。此外,该函数是被试能力的函数,可以对各个水平的误差进行估计,就能在测试前估计精确度。现在,信度函数得到了广泛的应用。

信度函数常用的计算公式与极大似然估计值的性质有关,具体如式(2-10)所示:

$$I(\theta) = \sum_{i=1}^n (P_i')^2 / P_i Q_i, \quad (2-10)$$

其中, P_i ——被试对项目 i 的正答率, P_i' ——项目反应函数对于 θ 的导数; $I(\theta)$ ——测验信息函数,其值称为测验信息量; Q_i ——被试答错项目 i 的概率。从上式可以看出, $I(\theta)$ 是所有项目的和,所以对整个测验的项目信息量是可以互相叠加的,所以定义如下式(2-11)所示:

$$I(\theta, U_i) = \sum_{i=1}^n (P_i')^2 / P_i Q_i, \quad (2-11)$$

其中, $I(\theta, U_i)$ ——项目信息量; U_i ——第 i 个项目。这样,要求测验信息量只要将各项目信息量相加即可。由上式可以看出,函数的导数(P_i')和条件方差($P_i Q_i$)是项目信息函数的函数值的决定因素;导数越大或方差越小,则信息量越大。标准误差的大小可以决定测量误差,误差小说明该方法的估计越精确,可靠性越高。

通常情况下,项目信息量在能力为极大值 θ_{\max} 处达到极大值 $I(\theta, U_i)_{\max}$ 。不同的项目反应模型, θ_{\max} 和 $I(\theta, U_i)_{\max}$ 是不同的, 它们的取值受到项目参数的影响。对于最常用的三种逻辑斯谛模型^[45], θ_{\max} 和 $I(\theta, U_i)_{\max}$ 的计算表达式分别如下面的式(2-12)、(2-13)和(2-14)所示:

1) 单参数模型:

$$\theta_{\max} = b_i, I(\theta, U_i)_{\max} = \frac{1}{4} D^2; \quad (2-12)$$

2) 双参数模型:

$$\theta_{\max} = b_i, I(\theta, U_i)_{\max} = \frac{1}{4} D^2 a_i^2; \quad (2-13)$$

3) 三参数模型:

$$\theta_{\max} = b_i + \frac{1}{D a_i} \left(\ln \frac{1 + (1 + 8c_i)^{1/2}}{2} \right), I(\theta, U_i)_{\max} = \frac{D^2 a_i^2}{8(1 - c_i^2)} (1 - 2Dc_i - 8c_i^2 + (1 + 8c_i)^{3/2}). \quad (2-14)$$

在以上各式中, $D=1.7$; a_i 、 b_i 、 c_i 分别代表项目的区分度参数、难度参数和猜测参数。分析可知: 不同的函数随着参数的不同取得最大值的点也会产生变化, 当 $\theta_{\max}=b_i$ 时, 单双参数函数取得极大值; 双参数极大值和 a_i 的平方成正比, 信息量随着 a_i 值增大而增大; 三参数时, 极大值和 a_i 、 b_i 和 c_i 三个参数都有关, 其关系为: a 正相关, c 为负相关, 当 c 无穷大时, 信息量为零, 反之 $c=0$ 时, 信息量最大。

通过以上讨论, 项目信息函数受参数值的影响很明显, 明白这一点在编制测试中是很重要的。该函数反映了测试的准确程度, 由此可以进一步确定怎样去选择合适的项目, 使得将要进行的测试的准确度达到预定的要求。此外, 信息函数还具有可以相加的特性, 随着项目的增多, 测试也会变长, 函数值也就会变大, 越精确。可得结论, 为了达到一定的测试精度, 既要选择合适的项目, 又要选择合适数目的项目, 太少的话, 会减少信息量, 降低精度。

2.4 本章小结

在本章中对智能组卷系统的指导思想、体系结构以及量度理论进行了详细地总结和 research, 为后面智能组卷系统数学模型的建立、求解以及智能组卷系统的设计与开发提供了理论基础和技术支持。

第3章 智能组卷系统的改进数学模型

依据上一章节中对智能组卷系统量度理论的介绍,在本章中将对智能组卷系统数学模型的构建进行深入地探讨,并最终引入本文的智能组卷系统改进模型。

3.1 改进数学模型的目标函数

由智能组卷系统所生成的考试试卷应该满足上一章节中提及的六条试卷的基本指导原则(如第2章第2.1节所述),考试试卷应该完全遵从教学大纲的安排并且能够真实地反映考试组织者的意图。事实上,对于智能组卷系统而言,上述三点规范不是明确具体的、可以完全按照量化的要求去操控的,它们只是一种原则性的指导规范,即一种带有模糊性质的约束条件^[46]。因此,将智能组卷系统付诸实施的前提是必须将这些模糊的约束条件量化为具体的可以采用数值或是公式描述的考试试卷指标(如第2章第2.2节所述)。

通常会有众多的约束指标对考试试卷进行约束,考试专家会根据以往考试积累的丰富经验将这些指标曲线化,即将指标公式化,而后依据公式化之后的指标进行组卷,组卷时需对指标进行合理地组合搭配,不同的指标搭配所构成的集合称之为试卷模式^[47]。由公式(2-11)可得知,应试者应试能力的不同将造成试题信息量 $I(\theta, U_i)$ 的不同,当 $I(\theta, U_i)$ 取得最大值时的能力值就是适用于此试题进行测试的应试者的能力值,因此,专家会根据已掌握的应试者的能力值信息,在试题库中挑选所包含信息量最大的试题组成考试试卷以对应试者进行测试。

对于任何一款智能组卷系统而言,都必须解决四个关键问题^[48]:一是了解考试组织者的目的和意图,即本次考试的测试对象、测试环境、测试方法以及测试重要性等等;二是依据考试组织者的要求、选取适当的项目指标构建相应的试卷模式;三是根据构建好的试卷模式建立相应的智能组卷系统的数学模型;四是使用特定的优化策略对步骤(3)中生成的数学模型进行求解。经典的求解策略包括:随机选择、回溯试探、补偿算法、误差补偿算法以及进化算法。在本文中 will 详细讨论如何使用优化算法——改进的自适应遗传算法对智能组卷系统的数学模型进行求解。

由于试卷的模式反映的是考试组织者为试卷要求的定量反应,因此,试卷模式(步骤2)是生成组卷系统数学模型(步骤3)以及建立模型求解算法(步骤4)的基础。另外,组卷系统的数学模型(步骤3)复杂与否将对求解算法(步骤4)的选用提出不同的需求,较为复杂的数学模型势必将提高优化算法求解的时间以及空间复杂度,这将对智能组卷系统的实时性造成严重的影响。综上所述,要实现和完成智能组卷系统的设计就必须首先要建立合适的智能组卷系统数学模型,将模糊的约

束条件量化为具体的可以采用数值或是公式描述的考试试卷指标。

通常情况下,研究者倾向于使用确定性的参数与分布曲线来描述考试试卷的模式。分布曲线是指相应的试题指标(如第2章第2.2节所述)在考试试卷中所占的比例,用以控制和测评考试试卷的质量和效果。在本文中,考试试卷模式采用如下的方式进行描述:

(1) 试卷难度系数与试卷分数分布曲线。由第2章第2.2节所述可知考试试卷的难度被分为容易、中等难度、较高难度以及最难四个试题的难度等级,假设试卷难度系数与试卷分数的分布如式(3-1)所示:

$$\bar{D} = \{d_1, d_2, d_3, d_4\}, \quad (3-1)$$

其中, d_1 为难度系数为“容易”的试卷所对应的分数, d_2 为难度系数为“中等难度”的试卷所对应的分数, d_3 为难度系数为“较高难度”的试卷所对应的分数, d_4 为难度系数为“最难”的试卷所对应的分数。对于某套特定的考试试卷 $T = \{t_1, t_2, \dots, t_{NT}\}$, 其中 NT 为考试试卷 T 中所包含的试题题目数, t_i 为考试试卷中第 i 个考试试题, 其试卷难度系数与试卷分数的分布如下式(3-2)所示:

$$\bar{D}(T) = \{d'_1, d'_2, d'_3, d'_4\}, \quad (3-2)$$

其中, $d'_i = \sum_{j=1}^{NT} S_D(t_j), i=1, 2, 3, 4; \forall t_j \in T; S_D(t_j) = \begin{cases} 0, & d(t_j) \notin \bar{D} \\ s_D(t_j), & d(t_j) \in \bar{D} \end{cases}$, $s_D(t_j)$ 为考试

试卷中试题 t_j 的考试分数, $d(t_j)$ 为考试试卷中试题 t_j 的难度系数。试卷难度系数与

试卷分数分布偏差的计算公式,如下式(3-3)所示:

$$deviation_D = \frac{\|\bar{D}(T) - \bar{D}\|}{Score_D} = \frac{\|d'_1 - d_1, d'_2 - d_2, d'_3 - d_3, d'_4 - d_4\|}{Score_D} = \frac{\sqrt{\sum_{i=1}^4 (d'_i - d_i)^2}}{Score_D}, \quad (3-3)$$

其中, $Score_D$ 为考试试卷的期望总分值, $Score_D = \sum_{i=1}^4 SD_i$, $SD_i (i=1, 2, 3, 4)$ 为第 i 种难度的试题的期望得分。

试卷难度系数与试卷分数分布曲线反映的是考试试卷的难易程度,在一份合格的考试试卷中,不同难度系数的试题应该占有不同的比重,当考试试卷的预先难度值被设定好之后,那么,对应着不同难度级别的考试试题的分数就可以被计算出来。考试成绩的分布在很大程度上是由试卷难度系数与试卷分数分布曲线决定的,因此,试卷难度系数与试卷分数分布曲线又被称为中心曲线。

(2) 试卷区分度与试卷分数分布曲线。由第2章第2.2节所述可知考试试卷的区分度值大于 0.25 时,则认为该试卷具有良好的区分度,因此,按照区分度的分值将

区分度分为 0.01~0.15, 0.16~0.25, 0.25 以上三个等级, 假设试卷区分度与试卷分数的分布如式(3-4)所示:

$$\bar{ID} = \{id_1, id_2, id_3\}, \quad (3-4)$$

其中, id_1 为区分度为 0.01~0.15 的试卷所对应的分数, id_2 为区分度为 0.16~0.25 的试卷所对应的分数, id_3 为区分度为 0.25 以上的试卷所对应的分数。对于某套特定的考试试卷 $T=\{t_1, t_2, \dots, t_{NT}\}$ 其试卷区分度与试卷分数的分布如下式(3-5)所示:

$$\bar{ID}(T) = \{id'_1, id'_2, id'_3\}, \quad (3-5)$$

其中, $id'_i = \sum_{j=1}^{NT} S_{ID}(t_j), i=1, 2, 3; \forall t_j \in T; S_{ID}(t_j) = \begin{cases} 0, & id(t_j) \notin \bar{ID} \\ s_{ID}(t_j), & id(t_j) \in \bar{ID} \end{cases}$, $s_{ID}(t_j)$ 为考

试试卷中试题 t_j 的考试分数, $id(t_j)$ 为考试试卷中试题 t_j 的区分度。试卷区分度与试卷分数分布偏差的计算公式, 如下式(3-6)所示:

$$deviation_{ID} = \frac{\|\bar{ID}(T) - \bar{ID}\|}{Score_{ID}} = \frac{\|id'_1 - id_1, id'_2 - id_2, id'_3 - id_3\|}{Score_{ID}} = \frac{\sqrt{\sum_{i=1}^3 (id'_i - id_i)^2}}{Score_{ID}}, \quad (3-6)$$

其中, $Score_{ID}$ 为考试试卷的期望总分值, $Score_{ID} = \sum_{i=1}^4 SID_i$, $SID_i (i=1, 2, 3, 4)$ 为第 i 种区分度的试题的期望得分。

(3) 试卷知识点与试卷分数分布曲线。知识点为考试试题的内在本质, 它以试题的形式呈现, 假设考试试卷中包含的知识点的数目为 NKP , 则试卷区分度与试卷分数的分布如式(3-7)所示:

$$K\bar{P} = \{kp_1, kp_2, \dots, kp_{NKP}\}, \quad (3-7)$$

其中, $kp_i (i=1, 2, \dots, NKP)$ 为知识点 i 所对应的分数。对于某套特定的考试试卷 $T=\{t_1, t_2, \dots, t_{NT}\}$ 其试卷知识点与试卷分数的分布如下式(3-8)所示:

$$K\bar{P}(T) = \{kp'_1, kp'_2, \dots, kp'_{NKP}\}, \quad (3-8)$$

其中, $kp'_i = \sum_{j=1}^{NT} S_{KP}(t_j), i=1, 2, \dots, NKP; \forall t_j \in T; S_{KP}(t_j) = \begin{cases} 0, & kp(t_j) \notin K\bar{P} \\ s_{KP}(t_j), & kp(t_j) \in K\bar{P} \end{cases}$, $s_{KP}(t_j)$

为考试试卷中试题 t_j 的考试分数, $KP(t_j)$ 为考试试卷中试题 t_j 的知识点。试卷知识点与试卷分数分布偏差的计算公式, 如下式(3-9)所示:

$$deviation_{KP} = \frac{\|K\bar{P}(T) - K\bar{P}\|}{Score_{KP}} = \frac{\|kp'_1 - kp_1, \dots, kp'_{NKP} - kp_{NKP}\|}{Score_{KP}} = \frac{\sqrt{\sum_{i=1}^{NKP} (kp'_i - kp_i)^2}}{Score_{KP}}, \quad (3-9)$$

其中, $Score_{KP}$ ——考试试卷的期望总分值, $Score_{KP} = \sum_{i=1}^{NKP} SKP_i$, $SKP_i (i=1, 2, \dots, NKP)$

为第 i 个知识点的试题的期望得分。

(4) 试卷题型与试卷分数分布曲线。通常一份考试的试卷中会包含诸如填空题、选择题、判断题、概念论述题、简答题、分析计算题以及综合题等各种题型, 现假设考试试卷中包含的题型的数目为 NIT , 则试卷区分度与试卷分数的分布如式(3-10)所示:

$$\bar{IT} = \{it_1, it_2, \dots, it_{NIT}\}, \quad (3-10)$$

其中, $it_i (i=1, 2, \dots, NKP)$ 为题型 i 所对应的分数。对于某套特定的考试试卷 $T = \{t_1, t_2, \dots, t_{NT}\}$ 其试卷题型与试卷分数的分布如下式(3-11)所示:

$$\bar{IT}(T) = \{it'_1, it'_2, \dots, it'_{NIT}\}, \quad (3-11)$$

其中, $it'_i = \sum_{j=1}^{NT} S_{it}(t_j), i=1, 2, \dots, NIT; \forall t_j \in T; S_{it}(t_j) = \begin{cases} 0, & it(t_j) \notin \bar{IT} \\ s_{it}(t_j), & it(t_j) \in \bar{IT} \end{cases}$, $s_{it}(t_j)$ 为

考试试卷中试题 t_j 的考试分数, $it(t_j)$ 为考试试卷中试题 t_j 的题型。试卷题型与试卷分数分布偏差的计算公式, 如下式(3-12)所示:

$$deviation_{IT} = \frac{\|\bar{IT}(T) - \bar{IT}\|}{Score_{IT}} = \frac{\|it'_1 - it_1, \dots, it'_{NIT} - it_{NIT}\|}{Score_{IT}} = \frac{\sqrt{\sum_{i=1}^{NIT} (it'_i - it_i)^2}}{Score_{IT}}, \quad (3-12)$$

其中, $Score_{IT}$ 为考试试卷的期望总分值, $Score_{IT} = \sum_{i=1}^{NIT} SIT_i$, $SIT_i (i=1, 2, \dots, NIT)$ 为第 i 个题型的试题的期望得分。

(5) 试卷章节与试卷分数分布曲线。考试试卷中的考试试题中所包含的知识分布于教学大纲中的不同章节中, 重要章节中有可能包含的考题就会多于一般章节中包含的考题。假设考试试卷中包含试题所选自的章节数目为 NCS , 则试卷章节与试卷分数的分布如式(3-13)所示:

$$\bar{CS} = \{cs_1, cs_2, \dots, cs_{NCS}\}, \quad (3-13)$$

其中, $cs_i (i=1, 2, \dots, NKP)$ 为章节 i 所对应的分数。对于某套特定的考试试卷 $T = \{t_1, t_2, \dots, t_{NT}\}$ 其试卷章节与试卷分数的分布如下式(3-14)所示:

$$\bar{CS}(T) = \{cs'_1, cs'_2, \dots, cs'_{NCS}\}, \quad (3-14)$$

其中, $cs'_i = \sum_{j=1}^{NT} S_{cs}(t_j), i=1, 2, \dots, NCS; \forall t_j \in T; S_{cs}(t_j) = \begin{cases} 0, & cs(t_j) \notin \bar{CS} \\ s_{cs}(t_j), & cs(t_j) \in \bar{CS} \end{cases}$,

$s_{kp}(t_j)$ ——考试试卷中试题 t_j 的考试分数, $kp(t_j)$ ——考试试卷中试题 t_j 的章节。试卷章节与试卷分数分布偏差的计算公式, 如下式(3-15)所示:

$$deviation_{CS} = \frac{\|C\bar{S}(T) - C\bar{S}\|}{Score_{CS}} = \frac{\|cs'_1 - cs_1, \dots, cs'_{NCS} - cs_{NCS}\|}{Score_{CS}} = \frac{\sqrt{\sum_{i=1}^{NCS} (cs'_i - cs_i)^2}}{Score_{CS}}, \quad (3-15)$$

其中, $Score_{CS}$ ——考试试卷的期望总分值, $Score_{CS} = \sum_{i=1}^{NCS} SCS_i$, $SCS_i (i=1, 2, \dots, NCS)$

为第 i 个章节的试题的期望得分。

(6) 试卷认知层次与试卷分数分布曲线。由第2章第2.2节所述可知考试试卷的认知层次大致可以分为如下的六个层次: 记忆、领悟、使用、分析、汇总以及评论, 因此试卷认知层次与试卷分数的分布如式(3-16)所示:

$$C\bar{L} = \{cl_1, cl_2, cl_3, cl_4, cl_5, cl_6\}, \quad (3-16)$$

其中, cl_1 为记忆层次所对应的分数, cl_2 为领悟层次所对应的分数, cl_3 为使用层次所对应的分数, cl_4 为分析层次所对应的分数, cl_5 为汇总层次所对应的分数, cl_6 为评论层次所对应的分数。对于某套特定的考试试卷 $T = \{t_1, t_2, \dots, t_{NT}\}$ 其试卷认知层次与试卷分数分布的分布如下式(3-17)所示:

$$C\bar{L}(T) = \{cl'_1, cl'_2, cl'_3, cl'_4, cl'_5, cl'_6\}, \quad (3-17)$$

其中, $cl'_i = \sum_{j=1}^{NT} S_{CL}(t_j), i=1, 2, \dots, 6; \forall t_j \in T; S_{CL}(t_j) = \begin{cases} 0, & cl(t_j) \notin C\bar{L} \\ s_{CL}(t_j), & cl(t_j) \in C\bar{L} \end{cases}$, $s_{CL}(t_j)$ 为

考试试卷中试题 t_j 的考试分数, $cl(t_j)$ 为考试试卷中试题 t_j 的认知层次。试卷认知层次与试卷分数分布偏差的计算公式, 如下式(3-18)所示:

$$deviation_{CL} = \frac{\|C\bar{L}(T) - C\bar{L}\|}{Score_{CL}} = \frac{\|cl'_1 - cl_1, \dots, cl'_6 - cl_6\|}{Score_{CL}} = \frac{\sqrt{\sum_{i=1}^6 (cl'_i - cl_i)^2}}{Score_{CL}}, \quad (3-18)$$

其中, $Score_{CL}$ 为考试试卷的期望总分值, $Score_{CL} = \sum_{i=1}^6 SCL_i$, $SCL_i (i=1, 2, \dots, 6)$ 为第

i 个认知层次的试题的期望得分。

(7) 考试试卷作答时间。假设考试的作答时间为 $A\bar{T} = \{at_1, at_2, \dots, at_{NT}\}$, 对于某套特定的考试试卷 $T = \{t_1, t_2, \dots, t_{NT}\}$ 其试卷期望作答时间为 $A\bar{T}'(T) = \{at'_1, at'_2, \dots, at'_{NT}\}$, 检查试卷期望时间为 $\alpha \times A\bar{T}'(T)$, 则完成试卷作答的总时间为式(3-19)所示:

$$A\bar{T}(T) = (1 + \alpha) \times A\bar{T}'(T) = (1 + \alpha) \times \sum_{j=1}^{NT} at_j, \quad (3-19)$$

其中, at_j ——完成对考试试卷中试题 t_j 的作答所需要的时间。考试时间的偏差为下式(3-20)所示:

$$deviation_{AT} = \frac{\|A\bar{T}(T) - A\bar{T}\|}{Score_{AT}} = \frac{\|at'_1 - at_1, \dots, at'_{NT} - at_{NT}\|}{Score_{AT}} = \frac{\sqrt{\sum_{i=1}^{NT} (at'_i - at_i)^2}}{Score_{AT}}, \quad (3-20)$$

其中, $Score_{AT}$ ——考试试卷的期望总分值。

(8) 考试试卷的总分值。假设考试试卷的分值为 $S\bar{T} = \{st_1, st_2, \dots, st_{NT}\}$, 对于某套特定的考试试卷 $T = \{t_1, t_2, \dots, t_{NT}\}$ 其试卷期望分值为 $S\bar{T}'(T) = \{st'_1, st'_2, \dots, st'_{NT}\}$, 则试卷的总分值为式(3-21)所示:

$$S\bar{T}'(T) = \sum_{j=1}^{NT} st_j, \quad (3-21)$$

其中, st_j 为考试试卷中试题 t_j 的期望分值。考试总分值的偏差为下式(3-22)所示:

$$deviation_{ST} = \frac{\|S\bar{T}'(T) - S\bar{T}\|}{S\bar{T}} = \left| 1 - \frac{\sum_{j=1}^{NT} st_j}{S\bar{T}} \right|. \quad (3-22)$$

智能系统组卷的目标就是从预先设定的试题库中, 寻找出一个试题集合 $T = \{t_1, t_2, \dots, t_{NT}\}$ 使得这个试题集合所对应的各指标的分值偏差(3-3)、(3-6)、(3-9)、(3-12)、(3-15)、(3-18)、(3-20)和(3-22)达到最小^[49-50], 在本文中, 采用下述策略来构建组卷系统的目标函数, 目标函数的表达式如下式(3-23)所示, 之后将对目标表达式的构成进行详述:

$$F(T) = w_1 \times deviation_D + w_2 \times deviation_{ID} + w_3 \times deviation_{KP} + w_4 \times deviation_{IT} + w_5 \times deviation_{CS} + w_6 \times deviation_{CL} + w_7 \times deviation_{AT} + w_8 \times deviation_{ST}, \quad (3-23)$$

其中, $w_i (i=1, 2, \dots, 8)$ 为指标分值偏差的权重且权重满足 $\sum_{i=1}^8 w_i = 1$ 。对于目标函数权值

w_i 的确定采用如下的策略步骤:

Step 1: 基于当前试题库以往考试的结果(至少一次), 此结果记录的是上述八个指标的分值偏差, 将其记作 $DEV = \{dev_1, dev_2, dev_3, dev_4, dev_5, dev_6, dev_7, dev_8\}$;

Step 2: 以 DEV 为种子生成 ND 个指标分值偏差向量: $dev_i^{(j)} = dev_i + \delta_i^{(j)} (i=1, 2, \dots, 8; j=1, 2, \dots, ND)$, 其中 $\delta_i^{(j)}$ 为属于区间 $[-0.5, 0.5]$ 的均匀分布随机数;

Step 3: 计算这 $ND+1$ 个指标分值偏差向量的 $dev_i^{(j)} (i=1, 2, \dots, 8; j=1, 2, \dots, ND+1)$ 的均值 μ_i 和方差 σ_i , 令 $v_i = \sigma_i / \mu_i (i=1, 2, \dots, 8)$, v_i 表示的是第 i 个指标分量偏差的变异系数;

Step 4: 令权重 $w_i = v_i / \sum_{i=1}^8 v_i$ ($i=1,2,\dots,8$)。

采用上述方法确定的权重可以显著地消除不同的指标分量偏差由于量纲的不同对目标函数效用的影响,求得的权重可以有效地衡量指标分量偏差之间的差异程度。

3.2 改进数学模型的约束条件

在本文中,选取了上一小节中所阐述的八个指标来描述一份待生成的考试试卷:试题难度系数(D)、试题区分度(ID)、试题知识点(KP)、试题题型(IT)、试题取自的章节(CS)、试题认知层次(CL)、试题作答时间(AT)和试题的分值(ST)。基于前面引入的试卷模式,建立智能组卷系统的传统数学模型。假设现有期望生成的考试试卷 $T=\{t_1, t_2, \dots, t_{NT}\}$, 其中 NT 表示此试卷中所包含的试题数量, t_i ($i=1,2,\dots,NT$) 表示试卷 T 中的第 i 道考试试题。首先构建一个试题-指标状态矩阵 M 用以控制在组卷过程中试题的各项指标,矩阵 M 如下式(3-24)所示:

$$M = \begin{pmatrix} m_{11} & m_{12} & \cdots & m_{18} \\ m_{21} & m_{22} & \cdots & m_{28} \\ \vdots & \vdots & \ddots & \vdots \\ m_{NT,1} & m_{NT,2} & \cdots & m_{NT,8} \end{pmatrix}, \quad (3-24)$$

其中, m_{ij} 表示试卷 T 中第 i ($i=1,2,\dots,NT$) 道试题所对应的第 j ($j=1,2,\dots,8$) 个指标。矩阵 M 中的每一行代表试卷 T 中某一道试题的相关信息, M 中的每一列代表某个指标对试卷 T 中所有试题的约束情况。依据建立好的试卷模式,学要将试卷转化为试题-指标状态矩阵,也就是说根据考试组织者的要求从当前系统的试题库中挑选出一个含有 NT 个试题的组合,而这 NT 个试题的组合恰为试题-指标状态矩阵的解。基于上述的讨论,可知指标-状态矩阵 M 应该满足如下的约束条件:

(1) 矩阵 M 的第 1 列表示指标试题分值,则有下列式(3-25)成立:

$$\sum_{i=1}^{NT} m_{i1} = ST, \quad (3-25)$$

其中, ST ——本次考试的考试用卷总分值; m_{i1} ——第 i ($i=1,2,\dots,NT$) 道试题所对应的预先设定分值。 ST 和 m_{i1} 均由考试组织者指定。

(2) 矩阵 M 的第 2 列表示指标考试时间,则有下列式(3-26)成立:

$$\sum_{i=1}^{NT} m_{i2} = AT, \quad (3-26)$$

其中, AT ——本次考试的考试总时间; m_{i2} ——完成对第 i ($i=1,2,\dots,NT$) 道试题作答所需要的考试时间, AT 和 m_{i2} 均由考试组织者指定。

(3) 矩阵 M 的第 3 列表示指标章节分数,则有下列式(3-27)成立:

$$\sum_{i=1}^{NT} m_{i3} = CS_k, k=1,2,\dots,K, \quad (3-27)$$

其中, K ——本次考试的试卷中所有的试题总共出自的章节数, CS_k ——第 $k(k=1,2,\dots,K)$ 个章节中所有的试题在试卷的总分值中所占的分值, 如果当前的考试试题 $t_i(i=1,2,\dots,NT)$ 属于章节 k , 则 $m_{i3}=m_{i1}$; 否则, $m_{i3}=0$ 。

(4) 矩阵 M 的第 4 列表示指标题型分数, 则有以下式(3-28)成立:

$$\sum_{i=1}^{NT} m_{i4} = IT_k, k=1,2,\dots,K, \quad (3-28)$$

其中, K ——本次考试的试卷中所包含的题型数目, IT_k ——第 $k(k=1,2,\dots,K)$ 种题型中所含有的试题在试卷的总分值中所占的分值, 如果当前的考试试题 $t_i(i=1,2,\dots,NT)$ 属于题型 k , 则 $m_{i4}=m_{i1}$; 否则, $m_{i4}=0$ 。

(5) 矩阵 M 的第 5 列表示指标知识点分数, 则有以下式(3-29)成立:

$$\sum_{i=1}^{NT} m_{i5} = KP_k, k=1,2,\dots,K, \quad (3-29)$$

其中, K ——本次考试的试卷中所包含的知识点数目, KP_k ——第 $k(k=1,2,\dots,K)$ 个知识点中所含有的试题在试卷的总分值中所占的分值, 如果当前的考试试题 $t_i(i=1,2,\dots,NT)$ 属于知识点 k , 则 $m_{i5}=m_{i1}$; 否则, $m_{i5}=0$ 。

(6) 矩阵 M 的第 6 列表示指标认知层次分数, 则有以下式(3-30)成立:

$$\sum_{i=1}^{NT} m_{i6} = CL_k, k=1,2,\dots,K, \quad (3-30)$$

其中, K ——本次考试的试卷中所包含的认知层次数目, CL_k ——第 $k(k=1,2,\dots,K)$ 个认知层次中所含有的试题在试卷的总分值中所占的分值, 如果当前的考试试题 $t_i(i=1,2,\dots,NT)$ 属于认知层次 k , 则 $m_{i6}=m_{i1}$; 否则, $m_{i6}=0$ 。

(7) 矩阵 M 的第 7 列表示指标难度系数分数, 则有以下式(3-31)成立:

$$\frac{\sum_{i=1}^{NT} (m_{i1} \times m_{i7})}{NT} = D, \quad (3-31)$$

其中, m_{i7} ——完成对第 $i(i=1,2,\dots,NT)$ 道试题的难度系数, D ——本次考试试卷的总体难度水平, 此值一般由考试组织者根据多年的考试经验确定。 m_{i7} 的确定一般根据二项分布产生, 但在本文中, 采用服从正态分布 $N(\mu, \sigma^2)$ 的、属于区间 $[0,1]$ 之间的连续值随机数来生成, 其中, μ 和 σ 的确定参照如下的步骤:

Step 1: 设定试卷的总体难度水平 D , D 的值选定如下五个值: 0.15、0.25、0.35、0.45 和 0.55;

Step 2: 如第 2 章图 2-1 所示, 为保证最终的考试成绩可以呈现正态分布, 设定

五个成绩区间: 0~59、60~64、65~74、75~84 和 85~100(假设考试的总分为 100 分), 使得位于这五个分数区间的考生人数比例为 $1+\varepsilon_1:5+\varepsilon_2:9+\varepsilon_3:5+\varepsilon_4:1+\varepsilon_5$, 其中 ε_i ($i=1,2,3,4,5$)表示比例偏差, 系统构建时随机指定此值, 一般令 ε_i 为属于区间 $[0,0.1]$ 的均匀分布随机数;

Step 3: 求解表达式(3-32):

$$\int_0^{59} N(\mu, \sigma^2) dx : \int_{60}^{64} N(\mu, \sigma^2) dx : \int_{65}^{74} N(\mu, \sigma^2) dx : \int_{75}^{84} N(\mu, \sigma^2) dx : \int_{85}^{100} N(\mu, \sigma^2) dx \\ = (1+\varepsilon_1):(5+\varepsilon_2):(9+\varepsilon_3):(5+\varepsilon_4):(1+\varepsilon_5); \quad (3-32)$$

Step 4: 利用 Step 3 中求解出的 μ 和 σ 生成满足正态分布 $N(\mu, \sigma^2)$ 的、属于区间 $[0,1]$ 之间的连续值随机数作为试题的难度系数 $m_{i7}(i=1,2,\cdots,NT)$ 。在下表 3-1 中给出了一组关于总体难度水平 $D=0.15、0.25、0.35、0.45$ 和 0.55 所对应的 μ 和 σ 值。

(8) 矩阵 M 的第 8 列表示指标区分度分数, 则有以下式(3-33)成立:

$$ID = \frac{\sum_{i=1}^{NT} (m_{i1} \times m_{i8})}{ST}, \quad (3-33)$$

其中, m_{i8} ——完成对第 $i(i=1,2,\cdots,NT)$ 道试题的区分度, 试题库中的所有试题的区分度值一般均由考试组织者根据多年的考试经验确定, 在以后的使用过程中试题的区

表 3-1 μ 和 σ 值

Tab.3-1 The values of μ and σ

总体难度水平	ε_1	ε_2	ε_3	ε_4	ε_5	μ	σ
0.15	0.0203	0.0015	0.0419	0.0838	0.0503	0.0193	0.6979
0.25	0.0199	0.0747	0.0846	0.0020	0.0709	0.0682	0.3784
0.35	0.0604	0.0445	0.0525	0.0681	0.0429	0.0302	0.8600
0.45	0.0272	0.0932	0.0203	0.0379	0.0305	0.0541	0.8537
0.55	0.0199	0.0466	0.0672	0.0832	0.0190	0.0150	0.5936

分度还可以根据教学安排和考试情况反馈进行实时地调整; D ——本次考试试卷的总体区分度, 此值由公式(3-34)确定:

$$ID = a \times \kappa + b, \quad (3-34)$$

其中, κ ——考试总体成绩的标准差, 此值可以依据试题库多次使用的结果确定, 记录试题库每次使用时的平均成绩, 从而得出考试成绩的标准差; a 和 b 的值均为属于区间 $[0,1]$ 的均匀分布随机数。

依据上述的详细介绍本文将智能组卷系统的数学模型表示为一个带有多约束的

单目标优化表达式, 其最优解为寻找合适的试卷 $T=\{t_1, t_2, \dots, t_{NT}\}$, 其中 NT 表示此试卷中所包含的试题数量, 使得目标函数取得最小值, 将组卷系统的数学模型整理为如下(3-35)的形式:

$$\begin{aligned}
 \min F(T) &= w_1 \times deviation_D + w_2 \times deviation_{ID} + w_3 \times deviation_{KP} + w_4 \times deviation_{IT} \\
 &\quad w_5 \times deviation_{CS} + w_6 \times deviation_{CL} + w_7 \times deviation_{AT} + w_8 \times deviation_{ST} \\
 s.t. \quad &\sum_{i=1}^{NT} m_{i1} = ST; \sum_{i=1}^{NT} m_{i2} = AT; \sum_{i=1}^{NT} m_{i3} = CS_k, k=1, 2, \dots, K; \sum_{i=1}^{NT} m_{i4} = IT_k, k=1, 2, \dots, K; \\
 &\sum_{i=1}^{NT} m_{i5} = KP_k, k=1, 2, \dots, K; \sum_{i=1}^{NT} m_{i6} = CL_k, k=1, 2, \dots, K; \\
 &\sum_{i=1}^{NT} (m_{i1} \times m_{i7}) / NT = D; ID = \sum_{i=1}^{NT} (m_{i1} \times m_{i8}) / ST
 \end{aligned}$$

。 (3-35)

3.3 本章小结

在本章节中, 提出了智能组卷系统改进的数学模型, 将智能组卷问题表示成为一个多约束的单目标优化问题, 其中对于目标函数中权值的确定以及难度系数指标中参数的确定分别提出了相应的解决策略, 数学模型的提出为智能组卷系统的构建提供了重要的理论基础。

第 4 章 改进的自适应遗传算法

4.1 标准遗传算法(Standard Genetic Algorithm-SGA)

SGA 是由美国 Michigan 大学的 J.Holland 教授在模拟达尔文的遗传选择和自然淘汰生物进化过程的基础上于 1975 年首先提出的,他对遗传算法的研究主要集中在抽取解释自然系统的自适应过程和设计具有自然系统机理的人工系统两方面。SGA 是模拟生物进化过程的计算模型,它作为全局化的搜索算法具有易用、健壮和并行等显著的特点^[61, 62]。

生物遗传物质的主要载体为染色体(chromosome),DNA 是其中的最主要的遗传物质,而基因(gene)又是控制生物性状的遗传物质的功能单位和结构单位。在 SGA 中,染色体对应的是数据或是数组(通常是由一维的串结构来表示的),串结构各个位置上所取的值对应的是基因。遗传算法处理的是染色体,或称为基因个体(individual),一定数量的基因个体构成了种群(population),而个体对环境的适应程度称作适应度(fitness)。SGA 包含两个必要的数据转化操作:一个是把搜索空间中的可行解转化为遗传空间中的染色体或是个体,称之为编码操作;另一个操作是前者的相反操作,即将遗传空间中的染色体或个体转化为搜索空间中的可行解,称之为译码过程。SGA 是具有生成(generate)和检测(test)的迭代过程的遗传算法,其算法的基本流程如图 4-1 所示:

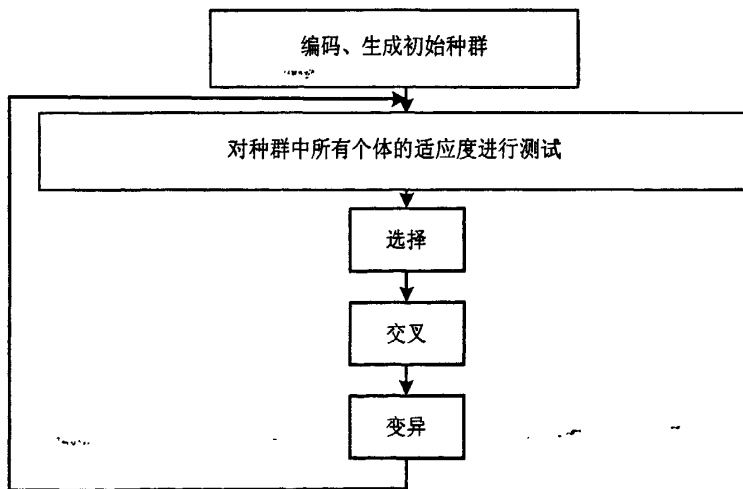


图 4-1 SGA 的流程图

Fig.4-1 The flowchart of SGA

由图 4-1 可知,SGA 是一种群体型操作,该操作是以群体中的所有个体为操作对象的,其包含三个必要的遗传算子:选择(selection)、交叉(crossover)和变异

(mutation)。SGA 包含五个基本要素：1) 参数编码；2) 初始种群的设定；3) 适应度函数的设计；4) 遗传算子的设计；5) 控制参数的设定，即种群大小以及使用遗传操作的概率等等。下面本文将结合一个具体的实例对这五个基本元素进行详细的解释，假设求解函数 $f(x)=x^2$ 在区间 $x \in [0,31]$ 上的最大值：

(1) 编码。由于 SGA 不能够直接处理空间中的数值数据，因此必须通过编码将它们转化为遗传空间中的基因型串结构数据。此例中将解空间 $[0,31]$ 的数据解 x 编码成为一个长度为 5 的二进制无符号整数，例如 $x=15$ 可以被表示为 01111；

(2) 初始种群的生成。由于 SGA 为群体型操作算法，必须为遗传操作准备一个由若干初始解构成的初始种群，本例中设定种群的大小为 4，即种群由 4 个个体构成。初始种群通过随机的方法生成(4 个属于区间 $[0,31]$ 随机值)，本例中的初始种群为： $x_1=13(c_1=01101)$ 、 $x_2=24(c_2=11000)$ 、 $x_3=8(c_3=01000)$ 和 $x_4=19(c_4=10011)$ ；

(3) 个体适应度测试。通过选择合适的适应函数计算个体的适应度，SGA 来评价种群个体的优劣，并作为后继遗传操作的依据。本例中选定目标函数 $f(x)=x^2$ 作为个体适应函数来评价个体： $f(x_1)=169$ 、 $f(x_2)=576$ 、 $f(x_3)=64$ 和 $f(x_4)=361$ ；

(4) 选择。选择的目的是从当前的种群中选择优良的个体使它们作为父代繁衍下一代个体，判断个体优劣与否的标准就是通过参考个体的适应度值：个体的适应度越高，其被选择的机会就越大。选择操作的实现方法有多种，在本例和本文中采用计算适应度概率的方法，即首先计算种群中所有个体的适应度总和 $\sum_{i=1}^4 f(x_i)$ ，之后

计算每个个体的适应度值比例 $\frac{f(x_i)}{\sum_{i=1}^4 f(x_i)}$ ($i=1,2,3,4$)，并以此作为相应的选择概率，

则： $p(x_1)=0.14$ 、 $p(x_2)=0.49$ 、 $p(x_3)=0.06$ 、 $p(x_4)=0.31$ 。假设需要选择出 4 个个体，则个体 1 和个体 4 将被复制 1 次、个体 2 将被复制 2 次、而个体 3 将会被淘汰，这就满足了最优秀的个体获得了更多的繁殖机会、最差的个体被淘汰的原则，将选择的优秀个体存储在匹配池(mating pool)中以备其他遗传操作使用。常用的选择操作还包括轮盘选择、随机抽样、局部选择和锦标赛选择等；

(5) 交叉。本例中对一点交叉进行介绍：首先，对匹配池中的个体进行随机匹配；其次，在进行配对的个体中随机地选择交叉点。图 4-2 展示了个体 2 与个体 1 在交叉

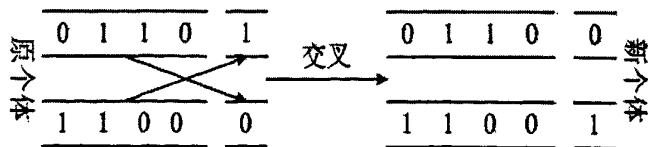


图 4-2 交叉操作

Fig.4-2 The operation of crossover

点4处进行配对的示意图这样就会得到新个体01100和11001。同理,个体2与个体4在交叉点2处进行配对又会得到新个体11011和10000。这四个新个体就构成了当前的新种群。事实上,会发现交叉操作是SGA最终的操作,这是因为 $c_1=01100(x_1=12)$ 、 $c_2=11001(x_2=25)$ 、 $c_3=11011(x_3=27)$ 和 $c_4=10000(x_4=16)$ 所对应的适应度的平均值和最大值都有了明显地提高(293 vs. 439 和 576 vs. 729),由此可见,种群已经开始向着目标函数最优解的方向开始移动了;

(6) 变异。变异的操作是按照位进行的,即把某一位的内容进行变异,对于二进制编码而言,所谓的变异就是将二进制编码的个体中的0变异为1,1变异为0。变异操作也是随机进行的,一般而言,变异的概率 p_m 都取得极小,在本例中的变异概率都取 $p_m=0.001$,变异操作克服了SGA容易陷于局部解的缺点。

结合上述的实例,给出SGA的算法流程:

Step 1: 随机地生成初始种群,并根据构建的适应度函数计算初始种群中每个个体的适应度值;

Step 2: 判断停机准则是否被满足,若满足,则算法结束;否则,执行下述步骤;

Step 3: 根据种群个体的适应度执行选择操作;

Step 4: 按照交叉概率 p_c 进行交叉操作;

Step 5: 按照变异概率 p_m 进行变异操作;

Step 6: 返回 Step 2。

基于上述的介绍,不难发现SGA具有如下的优点:(1)遗传算法从问题解的串集开始搜索,而不是从单个解开始。这是遗传算法与传统优化算法的极大区别。传统优化算法是从单个初始值迭代求最优解的;容易误入局部最优解。遗传算法从串集开始搜索,覆盖面大,利于全局择优。(2)许多传统搜索算法都是单点搜索算法,容易陷入局部的最优解。遗传算法同时处理群体中的多个个体,即对搜索空间中的多个解进行评估,减少了陷入局部最优解的风险,同时算法本身易于实现并行化。(3)遗传算法基本上不用搜索空间的知识或其它辅助信息,而仅用适应度函数值来评估个体,在此基础上进行遗传操作。适应度函数不仅不受连续可微的约束,而且其定义域可以任意设定。这一特点使得遗传算法的应用范围大大扩展。(4)遗传算法不是采用确定性规则,而是采用概率的变迁规则来指导他的搜索方向。(5)具有自组织、自适应和自学习性。遗传算法利用进化过程获得的信息自行组织搜索时,硬度大的个体具有较高的生存概率,并获得更适应环境的基因结构。

4.2 自适应遗传算法(Adaptive Genetic Algorithm-AGA)

在SGA中,交叉操作和变异操作是影响算法性能的关键,因而,交叉概率 p_c 和变异概率 p_m 的确定就显得尤为重要, p_c 和 p_m 的取值将会直接影响到遗传算法的收

敛性、成功率以及效率等。由前面的介绍可知,标准的遗传算法 SGA 采用固定的交叉概率 p_c 和变异概率 p_m , 不能够针对遗传算法运行的不同阶段适应性地做出自我调整以适应算法的进化, 同时, 针对不同的优化问题, 交叉概率 p_c 和变异概率 p_m 的确定也要根据问题的特征反复的实验才可以根据经验而确定适合的值, 这本身就是一个困难和繁琐的问题。如果交叉概率 p_c 越大, 则新个体产生的速度就越快, 优秀个体被破坏的可能性也就越大; 如果交叉概率 p_c 越小, 则新个体产生的速度就越慢, 搜索过程就越慢, 致使算法停滞。而对于变异概率 p_m 如果其过于小则不利于新个体的生成, 如果其过大, 则遗传算法就会退化成为纯粹的随机搜索算法。

在 1994 年 Srinivas 教授提出了基于交叉概率 p_c 和变异概率 p_m 的自适应遗传算法被视为经典的自适应遗传算法(AGA)。1995 年 Hinterding 等人在他们的著作中对遗传算法的适应性进行了定义并对其分类, 他们将遗传算法的适应性分为三类: 1) 确定性的适应性(deterministic adaptation): 根据确定性的规则确定遗传算法中的控制参数; 2) 有适应能力的适应性(adaptive adaptation): 根据进化过程中获取的信息确定控制参数的方向和大小; 3) 自适应的适应性(self-adaptive adaptation): 控制参数在进化的过程中能够被编码并经历交叉和变异而自行的进化。Srinivas 教授提出的自适应遗传算法就是基于第三种自适应性的, 其主要思想为:

交叉概率 p_c 和变异概率 p_m 可以根据适应度的不同而进行自动的改变, 当种群中各个个体的适应度值比较接近时, 则增加交叉概率 p_c 和变异概率 p_m ; 当种群的个体适应度值比较分散时, 则减少交叉概率 p_c 和变异概率 p_m 。同时, 对于适应度值高于群体平均适应度值的个体, 给予其较小的交叉概率 p_c 和变异概率 p_m , 使得此个体可以以较大的可能性进入下一代的繁殖; 对于适应度值低于群体平均适应度值的个体, 给予其较大的交叉概率 p_c 和变异概率 p_m , 使得此个体在进入下一代的繁殖的过程中被淘汰。

在自适应遗传算法 AGA 中, 交叉概率 p_c 按照如下(4-1)式进行自我地调整和改进:

$$p_c = \begin{cases} p_c^{(1)} + \frac{(p_c^{(2)} - p_c^{(1)}) \times (f - f_{\min})}{f_{\text{avg}} - f_{\min}}, & \text{if } f \leq f_{\text{avg}} [61], \\ p_c^{(2)}, & \text{if } f > f_{\text{avg}} \end{cases} \quad (4-1)$$

其中, p_c 为当前种群的交叉概率; $p_c^{(1)}$ 和 $p_c^{(2)}$ 分别为当前种群中个体交叉概率的下限和上限; f 为种群中个体的适应度值; f_{\min} 为当前种群中所有个体的适应度值的最小值; f_{avg} 为当前种群中所有个体适应度值的平均值。

在自适应遗传算法 AGA 中, 变异概率 p_m 按照如下(4-2)式进行自我地调整和改进:

$$p_m = \begin{cases} p_m^{(1)} + \frac{(p_m^{(2)} - p_m^{(1)}) \times (f - f_{\min})}{f_{\text{avg}} - f_{\min}}, & \text{if } f \leq f_{\text{avg}} [61], \\ p_m^{(2)}, & \text{if } f > f_{\text{avg}}. \end{cases} \quad (4-2)$$

其中, p_m 为当前种群的变异概率; $p_m^{(1)}$ 和 $p_m^{(2)}$ 分别为当前种群中个体变异概率的下限和上限; f 为种群中个体的适应度值; f_{\min} 为当前种群中所有个体的适应度值的最小值; f_{avg} 为当前种群中所有个体适应度值的平均值。

在上式(4-1)与(4-2)中, 对于 $p_c^{(1)}$ 和 $p_c^{(2)}$ 以及 $p_m^{(1)}$ 和 $p_m^{(2)}$ 值的确定, 在 Srinivas 教授提出的自适应遗传算法中分别按照下式(4-3)设定:

$$p_c^{(1)} = 0.6, p_c^{(2)} = 0.9; p_m^{(1)} = 0.001, p_m^{(2)} = 0.1. \quad (4-3)$$

根据上述的公式(4-1)与(4-2)绘制了个体的交叉概率 p_c 和变异概率 p_m 与适应度 f 之间的曲线, 如下图 4-3 所示:

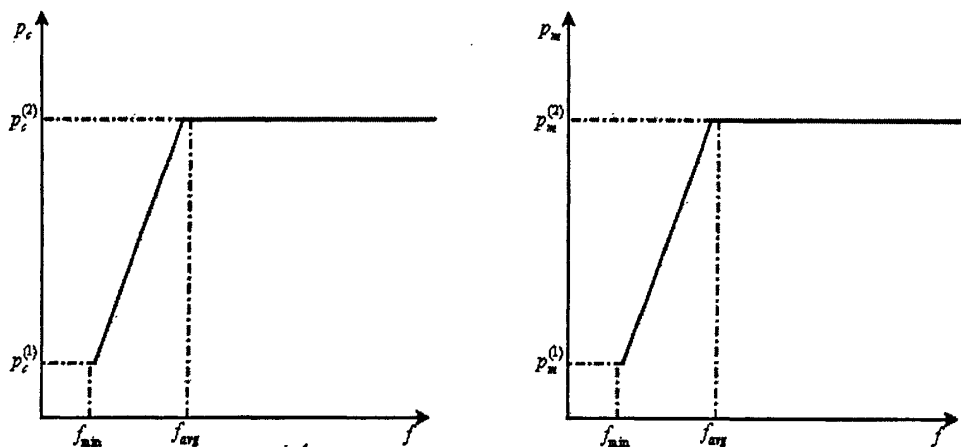


图 4-3 交叉概率 p_c 和变异概率 p_m 与适应度 f 之间的曲线

Fig.4-3 The relationships between p_c , p_m and f

由上图 4-3 可知, 当适应度值 f 越接近最小适应度值 f_{\min} 时, 交叉和变异的概率 p_c 和 p_m 就越小; 当适应度值较高时, 交叉和变异的概率 p_c 和 p_m 就越大。通过采用上述的方法, 自适应遗传算法能够自动地调节 p_c 和 p_m 以使遗传算法能够较快收敛, 并在很大程度上可以防止遗传算法的过早收敛。

4.3 改进的自适应遗传算法(Improved AGA-IAGA)

标准的遗传算法以及自适应遗传算法均面临着算法过早收敛的问题, 即早熟现象, 在没有搜索到问题的最优解之前或者仅搜索问题的局部最优解时算法就过早地停止。相当的文献均已证明遗传算法以及其变体在一定的条件下可以收敛到问题的最优解, 但是在众多的实际问题中由于理论上的条件不能够被满足而致使算法往往达不到理想的效果, 过早收敛是遗传算法及其变体最为突出的问题之一。在本章节

中试图对遗传算法的这一问题进行解释,分析其发生的原因并给出改进措施,将改进的算法称之为改进的自适应遗传算法(IAGA)。

为展开讨论,首先给出如下的定义1:

定义 1 设 $S = \{\bar{s}_1, \bar{s}_2, \dots, \bar{s}_N\}$ 表示当前种群,其中 N 为当前种群所含有的个体的个数, \bar{s}_i 表示当前种群的第 i 个个体,其可以被表示为 $\bar{s}_i = \{s_{i1}, s_{i2}, \dots, s_{iL}\}$, L 为个体 \bar{s}_i 的长度(二进制编码的长度)。种群 S 的多样性 $\alpha(S)$ 被定义为 $\sum_{i=1}^N s_{ij} \neq 0$ 或是 $\sum_{i=1}^N s_{ij} \neq N$ 的个体个数,因此,种群的成熟度 $\beta(S)$ 被定义为 $\beta(S) = L - \alpha(S)$ 。

基于上述的定义1,可以推导出:种群的多样性严格单调递减,并收敛至0。事实上,可以从直观上给出对此结论的解释:随着种群的进化,优秀的个体不断地繁衍自己的后代而致使种群中的个体逐渐地在基因上而趋于一致,从而使种群的多样性逐渐的减少,以至于最终收敛至0。在理论上,是期望优秀的个体基因在下一代的繁衍中被保存,但是在真正的实际问题中,由于受到计算机硬件水平的限制,通常设置的初始种群规模有限,以至于在算法的运行过程中发生近亲繁殖的现象。在遗传算法中,交叉操作是获取下一代个体的一个重要操作,在种群规模足够大的前提下,交叉操作是可以搜索到问题的最优解的,然而对于小规模种群而言,由于种群多样性的丧失,致使交叉操作的能力被不断的减弱以至于最终失去其功用。由以上分析可知,种群多样性的丢失是遗传算法过早收敛的根本原因。

交叉操作在遗传算法中具有两个功能:第一,能够繁殖下一代,即对问题空间展开搜索;第二,导致算法的收敛,或收敛到问题的全局最优解,或收敛到问题的局部最优解、亦或收敛到问题的非全局非局部最优解。下面给出通过保持种群的多样性而防止算法过早收敛的改进措施,在这之前首先应该可以准确地度量当前种群的多样性。本文中,采用信息熵来度量遗传算法种群的多样性。经典 Shannon 熵的定义如下式(4-4)所示:

$$H(X) = - \sum_{x \in X} (p(x) \log p(x)), \quad (4-4)$$

其中, X ——状态空间,对于 $\forall x \in X$, $p(x)$ —— x 的概率分布。

对于当前的种群是具有不确定性的,因此为使用熵来度量种群的多样性提供了可能,那么这种种群的不确定性表现在哪些方面呢?首先将问题的解空间划分为 n 个不相交的区域 $X = \{X_1, X_2, \dots, X_n\}$,在当前种群 $S = \{\bar{s}_1, \bar{s}_2, \dots, \bar{s}_N\}$ 中任意地挑选一个个体 $\bar{s}_i (i=1, 2, \dots, N)$,设个体 \bar{s}_i 属于区域 $X_k (k=1, 2, \dots, n)$ 的概率为 p_k ,其中 $p_k \geq 0$ 且

$\sum_{k=1}^n p_k = 1$,这样的话每一个个体隶属于哪一块区域就具有了一定的不确定性。因此,

这就为采用信息熵来度量种群多样性提供了可能性。下面介绍两种利用信息熵来度

量算法种群多样性的方法：第一种，由种群个体的适应度出发的适应度熵 H_1 ；第二种，由种群个体的编码串出发的编码串熵 H_2 。

适应度熵 H_1 ：

Step 1: 假设解空间 $X = [X_{\min}, X_{\max}]$ ，其长度为 L_X 。设当前种群 $S = \{\bar{s}_1, \bar{s}_2, \dots, \bar{s}_N\}$ 中个体适应度的最小值和最大值分别为 f_{\min} 和 f_{\max} ，则下述公式(4-5)成立：

$$L_X = X_{\max} - X_{\min}; \quad (4-5)$$

Step 2: 将区间 $[X_{\min}, X_{\max}]$ 进行 N 等分，则每个小区间可以被表示为下式(4-6)所示：

$$\left[X_{\min} + L_X \times \frac{(i-1)}{N}, X_{\min} + L_X \times \frac{i}{N} \right], i=1, 2, \dots, N, \quad (4-6)$$

统计每个小区间中属于当前种群的个体的个数，将其记为 $N_i (i=1, 2, \dots, N)$ ；

Step 3: 计算种群 $S = \{\bar{s}_1, \bar{s}_2, \dots, \bar{s}_N\}$ 中个体出现在第 $i (i=1, 2, \dots, N)$ 小区间的概率 p_i ， p_i 的计算表达式为下式(4-7)所示：

$$p_i = N_i / N, (i=1, 2, \dots, N); \quad (4-7)$$

Step 4: 这样的话种群中的个体则会对应着一个分布序列 $\{p_1, p_2, \dots, p_N\}$ ，将这个分布序列带入公式(4-4)即可得到当前种群的多样性，如公式(4-8)所示：

$$H_1(S) = - \sum_{i=1}^N p_i \log p_i. \quad (4-8)$$

由公式(4-8)可知，当种群中的个体分布越分散时，种群的多样性越大，种群的熵越大，当每个小区间分布着 1 个个体时， $H_1(S)$ 的值达到最大为 $H_1(S)_{\max} = \log N$ ；当种群中的个体分布越集中时，种群的多样性越小，种群的熵越小，所有的个体均分布于同一个小区间时， $H_1(S)$ 的值达到最小为 $H_1(S)_{\min} = 0$ 。

编码串熵 H_2 ：

Step 1: 根据当前种群 $S = \{\bar{s}_1, \bar{s}_2, \dots, \bar{s}_N\}$ 个体编码串的不同将种群划分为 M 类：

$$S = \{S_1, S_2, \dots, S_M\}, \text{ 其中 } \bigcup_{i=1}^M S_i = S \text{ 且 } S_i \cap S_j = \emptyset (\forall i, j=1, 2, \dots, M, i \neq j);$$

Step 2: 统计 $S_i (i=1, 2, \dots, M)$ 类中包含的个体的个数，将其记作 M_i ，显然

$$\sum_{i=1}^M M_i = N;$$

Step 3: 计算种群个体出现在第 $S_i (i=1, 2, \dots, M)$ 类的概率 $p_i = M_i / N$ ；

Step 4: 这样的话种群中的个体又会对应着一个分布序列 $\{p_1, p_2, \dots, p_M\}$ ，将这个分布序列带入公式(4-4)即可得到当前种群的多样性，如公式(4-9)所示：

$$H_2(S) = - \sum_{i=1}^M p_i \log p_i. \quad (4-9)$$

当种群中的所有个体完全不相同,此时种群的多样性达到最大,种群的熵取得最大值 $H_2(S)_{\max}=\log N$;当种群的所有个体完全相同时,此时种群的多样性达到最小,种群的熵取得最小值 $H_1(S)_{\min}=0$ 。这就说明,种群中个体的类型越多,多样性越大,种群的熵也就越大。

上述两种熵(适应度熵 H_1 和编码串熵 H_2)在一定的程度上可以较好地衡量种群的多样性,但是,适应度熵 H_1 是根据种群中个体的适应度值计算种群的多样性,当种群中所有个体的适应度保持一致时,此时适应度熵 H_1 最小,但是,若此时种群中所有个体在编码串熵不尽相同,说明种群仍然保持着相当的多样性,此时编码串熵 H_2 会达到最大;反之,编码串熵 H_2 根据种群中个体的编码计算种群的多样性,当种群的个体在编码上几乎相同时,编码串熵 H_2 会达到最小,但此时如果个体的适应度不尽相同,那么又会造成适应度熵 H_1 很大的情况。这说明上述的两种熵均有着自己的不足之处,若仅仅采用单一熵衡量种群的多样性将会有失偏颇,在很大程度上将造成种群过早停滞、失去继续进化的现象。在本文中,考虑将这两种熵进行结合,使用混合熵来衡量种群的多样性,即将适应度熵 H_1 和编码串熵 H_2 进行合并来度量种群的多样性,以克服上述两种熵在度量多样性时的不足,改进之后的混合熵为下式(4-10)所示:

$$H(S)=w_1 \times H_1(S)+w_2 \times H_2(S), 0 \leq w_1, w_2 \leq 1, w_1 + w_2 = 1. \quad (4-10)$$

其中,关于权重 w_1 和 w_2 的确定,给出了如下的标准:为保证遗传算法的随机性, w_1 为属于区间 $[0,1]$ 之间的服从均匀分布的随机数, $w_2=1-w_1$ 。

在给出了度量种群多样性的混合熵 $H(S)$ 之后,本文就可以根据此改进的混合熵来改进经典的自适应遗传算法,以提高种群的多样性,避免种群进化过早地停滞。根据前述可知交叉操作和变异操作对遗传算法的性能有着重要的影响作用,因此,本文基于(4-10)给出的混合熵对交叉概率 p_c 和变异概率 p_m 的表示方法中 $p_c^{(1)}$ 和 $p_c^{(2)}$ 以及 $p_m^{(1)}$ 和 $p_m^{(2)}$ 的表达式进行了改进(在传统的自适应遗传算法中, $p_c^{(1)}$ 和 $p_c^{(2)}$ 以及 $p_m^{(1)}$ 和 $p_m^{(2)}$ 的值为静态的、根据经验人为指定的),改进之后的公式分别如下式(4-11)和(4-12)所示:

$$\begin{cases} p_c^{(1)}(t) = p_c^{(1)} - p_c^{(2)} / N^2 + (p_c^{(2)} - p_c^{(1)}) / \exp(H(S)) \\ p_c^{(2)}(t) = p_c^{(2)} \end{cases}, \quad (4-11)$$

$$\begin{cases} p_m^{(1)}(t) = p_m^{(1)} - p_m^{(2)} / N^2 + (p_m^{(2)} - p_m^{(1)}) / \exp(H(S)) \\ p_m^{(2)}(t) = p_m^{(2)} \end{cases}, \quad (4-12)$$

其中, $p_c^{(1)}(t)$ 和 $p_c^{(2)}(t)$ 分别表示遗传算法在第 t 次迭代的过程中交叉概率的下限和上

限; $p_m^{(1)}(t)$ 和 $p_m^{(2)}(t)$ 分别表示遗传算法在第 t 次迭代的过程中变异概率的下限和上限; $p_c^{(1)}$ 和 $p_c^{(2)}$ 以及 $p_m^{(1)}$ 和 $p_m^{(2)}$ 为算法第一次迭代时赋予交叉概率的下限和上限以及变异概率的下限和上限的初始值,如表达式(4-3)所示。由于 $p_c^{(2)}$ 和 $p_m^{(2)}$ 的表达式为定值,所以在下图4-4给出了 $p_c^{(1)}$ 和 $p_m^{(1)}$ 随种群熵 $H(S)$ 变化的示意图:

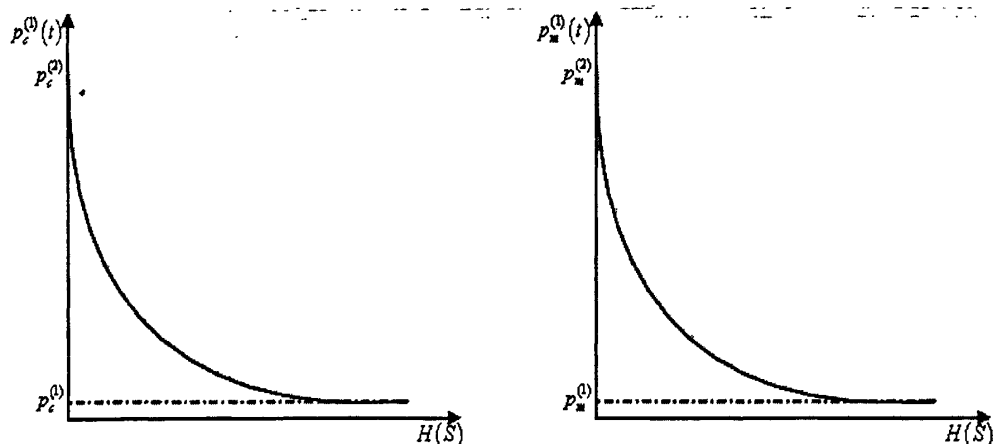


图4-4 交叉概率 $p_c^{(1)}(t)$ 和变异概率 $p_m^{(1)}(t)$ 与种群熵 $H(S)$ 之间的曲线

Fig.4-4 The relationships between $p_c^{(1)}(t)$, $p_m^{(1)}(t)$ and $H(S)$

由上图4-4可知,随着 $H(S)$ 的增大,即种群多样性的增大, $p_c^{(1)}(t)$ 和 $p_m^{(1)}(t)$ 的值逐渐的减小,在第 t 次迭代的过程中交叉概率和变异概率的变化区间(由下限和上限构成)分别为: $p_c \in [p_c^{(1)}(t), p_c^{(2)}(t)]$ 和 $p_m \in [p_m^{(1)}(t), p_m^{(2)}(t)]$ 将会逐渐的变宽,交叉概率 p_c 和变异概率 p_m 的取值就会越大;反之,随着 $H(S)$ 的减小,即种群多样性的减小, $p_c^{(1)}(t)$ 和 $p_m^{(1)}(t)$ 的值逐渐的增大,在第 t 次迭代的过程中交叉概率和变异概率的变化区间分别为: $p_c \in [p_c^{(1)}(t), p_c^{(2)}(t)]$ 和 $p_m \in [p_m^{(1)}(t), p_m^{(2)}(t)]$ 将会逐渐的变窄,交叉概率 p_c 和变异概率 p_m 的取值就会越小。

4.4 本章小结

在本章节中,在对标准遗传算法和自适应遗传算法进行详细地分析基础之上,提出了使用混合熵度量种群多样性从而修改交叉概率和变异概率的改进自适应遗传算法,改进自适应遗传算法的提出为求解智能组卷系统的数学模型提供了先进的技术手段。

第 5 章 改进的智能组卷系统的实现与性能分析

5.1 使用改进的自适应遗传算法求解组卷系统的改进数学模型

在第 3 章给出了智能组卷系统的数学模型，其数学表达式如 (3-35)所示，下面将给出使用在第 4 章中提出的改进自适应遗传算法求解公式(3-35)的详细步骤。智能组卷系统操作的对象为大学课程中《微积分》考试试题的生成，本文所探讨的组卷系统中总共包含的题型数目为填空题(G_1)、单项选择题(G_2)、多项选择题(G_3)、判断题(G_4)、概念论述题(G_5)、简答题(G_6)、分析计算题(G_7)以及综合题(G_8)八种。在生成的一份《微积分》考试试卷中每种题型的具体数量和分值如下表 5-1 所示。

表 5-1 一份《微积分》考试试卷中各种题型的数量和分值

Tab.5-1 The number and score of various questions in a calculus exam paper

题型	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8
试题数量	10	10	5	10	5	5	4	2
单位分值	1	1	2	1	2	2	5	10
总分值	10	10	10	10	10	10	20	20

基于上述的《微积分》考试试卷，本文给出了使用改进的自适应遗传算法求解改进数学模型(3-35)的具体流程，如下：

(1) 编码方式。由于所探讨的组卷系统中总共包含的题型数目为填空题(G_1)、单项选择题(G_2)、多项选择题(G_3)、判断题(G_4)、概念论述题(G_5)、简答题(G_6)、分析计算题(G_7)以及综合题(G_8)八种，并且在预先设定好的试题库中每种题型的具体数量如下表 5-2 所示：

表 5-2 《微积分》试题库中各种题型的数量

Tab.5-2 The number of various questions in a calculus dataset

题型	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8
试题数量	6648	4745	3057	8950	2915	2071	1451	216

因此，本文对试卷的编码方式为：

$\{g_{11} \ g_{12} \ \dots \ g_{1L_1}\} \ \{g_{21} \ g_{22} \ \dots \ g_{2L_2}\} \ \dots \ \{g_{81} \ g_{82} \ \dots \ g_{8L_8}\}$

其中， $\{g_{i1}, g_{i2}, \dots, g_{iL_i}\}$ 为试卷中第 $i(i=1,2,\dots,8)$ 种类型的题目所对应的编码段， $L_i(i=1,2,\dots,8)$ 表示试题库中第 $i(i=1,2,\dots,8)$ 种类型的试题的个数，具体的值如表 5-2 所示。 $g_{ij}(i=1,2,\dots,8; j=1,2,\dots,L_i)$ 的具体取值如下所示：

$$g_{ij} = \begin{cases} 1, & \text{第 } i \text{ 种类型题目中的第 } j \text{ 道题被试卷选中} \\ 0, & \text{第 } i \text{ 种类型题目中的第 } j \text{ 道题没有被试卷选中} \end{cases}$$

$g_{ij} (i=1,2,\dots,8; j=1,2,\dots,L_i)$ 满足如下的表达式: $\sum_{j=1}^{L_i} g_{ij} = l_i, l_i (i=1,2,\dots,8)$ 为《微积分》试卷中第 $i (i=1,2,\dots,8)$ 种类型的试题的个数, 具体的值如表 5-1 所示。

上述编码的主要思想为: 1) 在可行解的二进制编码串中, 每一个编码段代表着一种类型的试题, 编码段的长度由试题库中此类型试题的个数决定; 2) 每种类型的试题的代码段内独立编码, 即题型之间的编码相互独立、互不影响; 3) 可行解编码串的长度由题型的数目以及试题库中的试题总数决定。

(2) 选择操作。选择操作确定的具体方式为计算适应度概率的方法, 此方法简单易行、有利于减少组卷过程中的时间消耗, 并且此方法还可以确保适应度较高的个体肯定能够被保留到下一代种群中, 在求解(3-35)的过程中, 选择操作的具体实现方法为:

Step 1: 计算当前种群中每一个个体在繁衍的下一代种群中的期望保留数目 N_i , 如式(5-1)所示:

$$N_i = N \times \frac{f(\bar{s}_i)}{\sum_{i=1}^N f(\bar{s}_i)}, (i=1,2,\dots,N), \quad (5-1)$$

其中, N 为当前种群中所包含的个体的个数, $f(\bar{s}_i) (i=1,2,\dots,8)$ 为种群中个体 \bar{s}_i 的适应度;

Step 2: 每个个体在下一代种群中的生存数目为不大于 $N_i (i=1,2,\dots,N)$ 的最大整数 $\lfloor N_i \rfloor (i=1,2,\dots,N)$, 经历了此步, 则共有 $\sum_{i=1}^N \lfloor N_i \rfloor$ 个个体被保存到了下一代的种群中;

Step 3: 对当前种群中的 N 个个进行排序, 排序规则为按照 $N_i (i=1,2,\dots,N)$ 的小数部分降序排序, 之后, 取出排序序列中的前 $N - \sum_{i=1}^N \lfloor N_i \rfloor$ 个个体再次保留到下一代种群中。上述三步操作完成之后, 即可得到下一代含有 N 个个体的种群。

(3) 交叉操作。交叉操作是决定遗传算法搜索能力的关键所在, 在使用改进的自适应遗传算法求解(3-35)式的过程中, 本文采用了单点交叉以及均匀交叉, 下面一一介绍这两种方法:

I. 单点交叉: 对于含有 N 个个体的种群需要进行 $\frac{\lfloor N \rfloor}{2}$ 次的两两之间的配对。对

于每一次配对的两个个体之间,在 $\sum_{i=1}^8 L_i$ 个交叉点中随机地选取一个交叉点进行配对。

根据第4章的(4-1)和(4-11)公式确定的交叉概率 p_c 确定个体之间相互交换的染色体,之后完成配对。

II. 均匀交叉: 随机地生成长度为 $L' = \sum_{i=1}^8 L_i$ 的二进制代码串 $B = \{b_1, b_2, \dots, b_{L'}\}$,

之后对于种群中的两个个体 $G_p = \{g_1^{(p)}, g_2^{(p)}, \dots, g_{L'}^{(p)}\}$ 和 $G_q = \{g_1^{(q)}, g_2^{(q)}, \dots, g_{L'}^{(q)}\}$: 如果 $b_i = 0 (i=1, 2, \dots, L')$, 则新个体 G_p' 和 G_q' 中的第 $i (i=1, 2, \dots, L')$ 个基因位分别继承 G_p 和 G_q 中的第 i 个基因位上的基因; 如果 $b_i = 1 (i=1, 2, \dots, L')$, 则新个体 G_p' 的第 i 个基因位上继承 G_p 的第 i 个基因位上的基因, G_q' 的第 i 个基因位上继承 G_q 的第 i 个基因位上的基因。

(4) 变异操作。交叉操作决定了遗传算法的全局搜索能力, 而变异操作是在局部出发对种群中的个体进行调整, 它可以有效地保持种群的多样性, 可以提高遗传算法的局部搜索能力, 防止遗传算法过早地停滞。交叉操作和变异操作配合使用, 共同完成对可行空间的全局搜索和局部搜索, 使遗传算法保持良好的搜索性能。在使用改进的自适应遗传算法求解(3-35)式的过程中, 本文采用了如下形式的变异操作:

Step 1: 针对种群中的每一个个体 $G_p = \{g_1^{(p)}, g_2^{(p)}, \dots, g_{L'}^{(p)}\} (p=1, 2, \dots, N)$ 随机地生

成长度为 $L' = \sum_{i=1}^8 L_i$ 的代码串 $B = \{b_1, b_2, \dots, b_{L'}\}$, 其中 $b_i (i=1, 2, \dots, L')$ 为属于区间 $[0, 1]$

的、服从均匀分布的随机数;

Step 2: 根据第4章的(4-2)和(4-12)公式确定的变异概率 p_m , 如果 $b_i < p_m$

$(i=1, 2, \dots, L')$, 则即对 $G_p = \{g_1^{(p)}, g_2^{(p)}, \dots, g_{L'}^{(p)}\}$ 中的基因位 $g_i^{(p)} (i=1, 2, \dots, L')$ 执行如下

的变异操作: 判断 $g_i^{(p)} (p=1, 2, \dots, N; i=1, 2, \dots, L')$ 所对应的题型是什么以及是该题型

下的哪一道题; 之后, 随机生成一个在该题型范围中的试题编号, 如果该试题在某个体中已经存在, 则重新生成; 否则, 使用该试题编号的基因位由0变化为1, 而将基因位 $g_i^{(p)}$ 由1变为0。

5.2 改进的自适应遗传算法的性能验证

本文对改进算法的验证主要集中在下述的两个实验: 第一, 纯粹地比较三种遗传算法(标准遗传算法-SGA、自适应遗传算法-AGA、改进的自适应遗传算法-IAGA)

性能的优劣基于经典的测试函数；第二，应用性地比较三种遗传算法(SGA、AGA、IAGA)性能的优劣基于组卷的效能。

(1) 基于经典的测试函数比较 SGA、AGA 以及 IAGA 的性能

为了测试三种遗传算法的性能，本文挑选了六个经典的优化函数来验证这三种算法的求解效率。本文对这三种遗传算法的比较主要集中在迭代次数和最终求得的最优解的最优性。仿真实验的实现是以 Matlab 7.1 为实验平台，初始种群的规模为 500。这六个标准的测试函数的具体信息如下：

$$f_1(\vec{x}) = \sum_{i=1}^N x_i \sin(\sqrt{x_i}), \text{ 其中 } N=30, \vec{x}=(x_1, x_2, \dots, x_N), \text{ 对于 } \forall i \in \{1, 2, \dots, N\},$$

$x_i \in (-500, 500)$ ，此函数的最小值为 -12569.6；

$$f_2(\vec{x}) = \sum_{i=1}^N (x_i^2 - 10 \times \cos(2\pi x_i) + 10), \text{ 其中 } N=30, \vec{x}=(x_1, x_2, \dots, x_N), \text{ 对于}$$

$\forall i \in \{1, 2, \dots, N\}, x_i \in (-5.12, 5.12)$ ，此函数的最小值为 0；

$$f_3(\vec{x}) = -20 \exp\left(-0.2 \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}\right) - \exp\left(\frac{1}{N} \sum_{i=1}^N \cos(2\pi x_i)\right) + 20 + e, \text{ 其中 } N=30,$$

$\vec{x}=(x_1, x_2, \dots, x_N)$ ，对于 $\forall i \in \{1, 2, \dots, N\}, x_i \in (-32, 32)$ ，此函数的最小值为 0；

$$f_4(\vec{x}) = \frac{1}{4000} \sum_{i=1}^N x_i^2 - \prod_{i=1}^N \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \text{ 其中 } N=30, \vec{x}=(x_1, x_2, \dots, x_N), \text{ 对于}$$

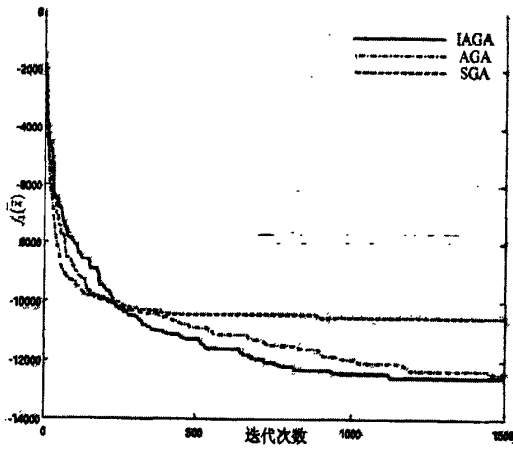
$\forall i \in \{1, 2, \dots, N\}, x_i \in (-600, 600)$ ，此函数的最小值为 0；

$$f_5(\vec{x}) = \frac{\pi}{N} \left(10 \times \sin^2(\pi y_i) + \frac{1}{N} \sum_{i=1}^{N-1} (y_i - 1) (1 + 10 \times \sin^2(\pi y_{i+1})) + (y_N - 1)^2 \right) + \frac{1}{N} \sum_{i=1}^N u(x_i, 10, 100, 4), y_i = 1 + 1/4(x_i + 1)$$

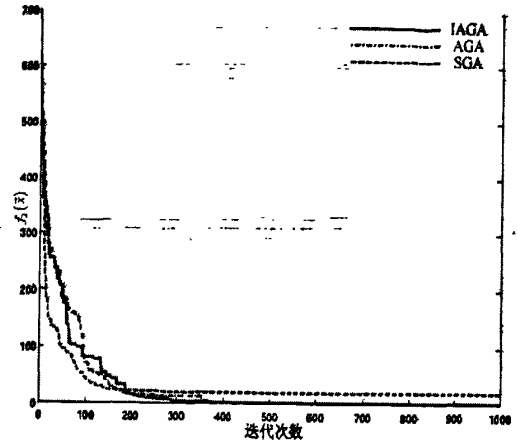
$N=30, \vec{x}=(x_1, x_2, \dots, x_N)$ ，对于 $\forall i \in \{1, 2, \dots, N\}, x_i \in (-50, 50)$ ，此函数的最小值为 0；

$$f_6(\vec{x}) = 0.1 \times \left(\frac{\sin^2(3\pi x_i) + \frac{1}{N} \sum_{i=1}^{N-1} (x_i - 1)^2 (1 + \sin^2(3\pi x_{i+1}))}{+(x_N - 1)^2 \times (1 + \sin^2(3\pi x_{N+1}))} \right) + \sum_{i=1}^N u(x_i, 10, 100, 4), \text{ 其中}$$

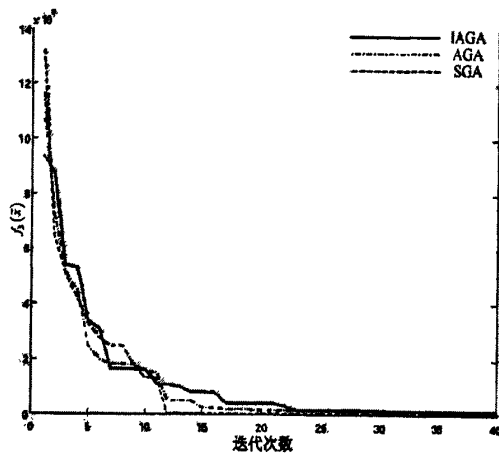
$N=30, \vec{x}=(x_1, x_2, \dots, x_N)$ ，对于 $\forall i \in \{1, 2, \dots, N\}, x_i \in (-50, 50)$ ，此函数的最小值为 0。



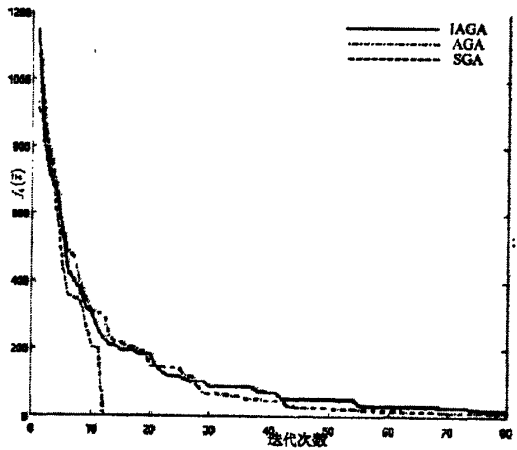
[A]. 函数 $f_1(\bar{x})$ 的收敛情况



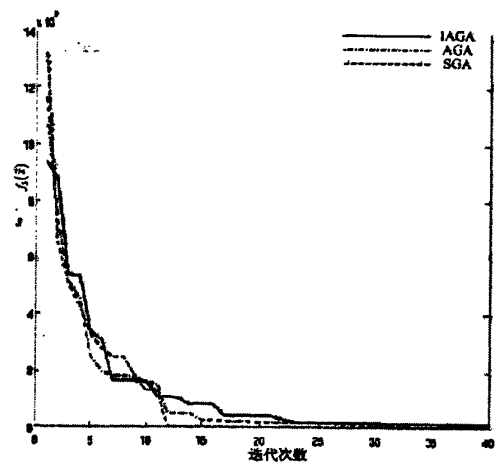
[B]. 函数 $f_2(\bar{x})$ 的收敛情况



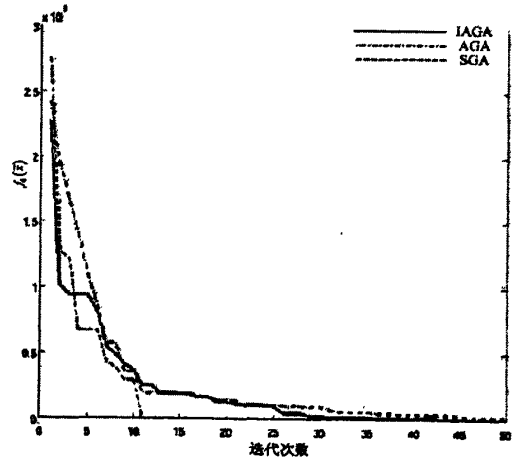
[C]. 函数 $f_3(\bar{x})$ 的收敛情况



[D]. 函数 $f_4(\bar{x})$ 的收敛情况



[E]. 函数 $f_5(\bar{x})$ 的收敛情况



[F]. 函数 $f_6(\bar{x})$ 的收敛情况

图 5-1 六个经典优化函数的收敛情况

Fig.5-1 The convergences of six classical optimization functions

本文的实验设置过程如下：处理种群的数量为 500，针对每个优化函数，分别采

用三种遗传算法进行求解 100 次, 所得的测试结果为 100 次的平均值, 并记录每次的求解结果, 根据求解过程中迭代次数与每次求得的函数值关系, 绘制了图 5-1, 图中曲线上的每个点均为 100 次求解的平均值。从图 5-1 中可以发现改进的自适应遗传算法 IAGA 较之 SGA、AGA 表现出了较强的收敛性能, 在经历相同的迭代次数的前提下, IAGA 一般总能够最先找到函数的最小值(与 SGA、AGA 相比), 并且在寻找到相同函数值的前提下, IAGA 的迭代次数为最少。这说明改进之后的自适应遗传算法在一定程度上能够克服遗传算法过早收敛或是停滞的缺陷。在表 5-3 中给出了这三种遗传算法对于这六个标准优化函数的最终求得的最小值的比较, 表中展示的结果同样为 100 次求解的平均值, 并且给出了 100 次求解的标准差。

表 5-3 使用三种遗传算法求解六个优化函数的结果

Tab.5-3 The results of six optimization functions using SGA、AGA and IAGA

优化函数 \ 算法	SGA	AGA	IAGA
$f_1(\bar{x})$	-7532.45±364.87	-8260.95±547.38	-10368.72±578.03
$f_2(\bar{x})$	175.85±14.52	76.59±32.16	13.89±6.98
$f_3(\bar{x})$	7.53±0.47	0.89±2.63	0.00±0.00
$f_4(\bar{x})$	6.78±0.79	0.95±0.87	0.01±0.05
$f_5(\bar{x})$	8.39±1.27	0.11±0.36	0.05±0.06
$f_6(\bar{x})$	6.91±0.93	3.45±1.67	0.09±0.63

由表 5-3 可知, IAGA 显著地改善了遗传算法的求解性能, 针对这六个优化函数, IAGA 所求得的最优值始终最接近函数的真实最优值, 说明 IAGA 在求解优化函数最优解方面的性能得到了改善, 这为使用 IAGA 求解改进模型(3-35)提供了坚实的实验基础。

(2) 应用性地比较 SGA、AGA 以及 IAGA 的性能

按照表 5-1 给出的试卷模式生成 30 份《微积分》考试试卷, 首先, 在邢台学院数学教研组挑选了 10 名至少具有 3 年微积分教学经验的优秀数学教师, 组成了试卷测评小组, 这十名教师的基本信息被列举在表 5-4 中; 之后, 确定了如表 5-5 中的试卷评价指标, 利用这些指标完成对试卷的评价, 教师可以根据这些指标为考试试卷的质量打分; 第三, 这十名优秀教师对试卷的测评维持 30 天, 每天仅对一份试卷测评, 测评的时间为每天上午的 8:30-10:30 之间, 下午不安排测评, 让测评小组休息; 第四, 教师的测评安排如下: 第一天测评使用标准遗传算法 SGA 生成的考试试卷, 对试卷的每个评价指标进行打分。第二天测评使用自适应遗传算法 AGA 生成的考试试卷, 对试卷的每个评价指标进行打分。第三天, 测评使用改进的自适应遗传算法 IAGA 生成的考试试卷, 对试卷的每个评价指标进行打分。此操作重复 10 次, 直至

表 5-4 教师基本情况表

Tab.5-4 The table of teacher's information

教师	性别	学历	专业	教学经验(年)
T1	男	硕士研究生	数学	5
T2	男	博士研究生	数学	10
T3	男	本科	数学	8
T4	男	本科	数学	9
T5	男	博士研究生	数学	3
T6	女	硕士研究生	数学	4
T7	女	硕士研究生	数学	6
T8	女	本科	数学	8
T9	女	硕士研究生	数学	5
T10	女	本科	数学	9

表 5-5 考试试卷的评价指标

Tab.5-5 The evaluation elements of the exam papers

#	评价指标	评价打分
E1	试卷题型安排	10
E2	试卷题型章节分布	10
E3	试卷试题符合大纲程度	10
E4	试卷试题难易程度	10
E5	试卷试题区分度	10
E6	试卷试题质量	10
E7	试卷试题效能	10
E8	试卷试题的基础性	10
E9	试卷试题的专业性	10
E10	试卷试题的智能型	10

对所有的试卷评价完毕；第五，收集依靠不同算法生成的试卷的评价信息，相同指标的评分值取平均，统计比较这三种策略生成试卷的质量。将这十名优秀教师对试卷的测评分值详细的列举在表 5-6、5-7 和 5-8 中，包括分指标分值以及对试卷测评的总分值。

表 5-6 使用标准遗传算法 SGA 所生成试卷的评价表

Tab.5-6 The evaluation table to the exam paper generated according to SGA

	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	总分
T1	7	5	5	5	5	6	7	6	5	7	58
T2	6	6	7	7	6	6	5	5	7	5	60
T3	5	6	7	5	5	5	5	5	5	5	53
T4	7	5	7	5	5	7	5	7	6	5	59
T5	6	7	5	6	6	5	6	6	7	7	61
T6	5	7	7	5	5	7	6	6	6	7	61
T7	7	6	5	6	7	6	7	5	7	7	63
T8	6	7	5	7	7	5	7	7	6	7	64
TS9	7	6	6	5	6	5	7	5	5	6	58
T10	7	6	5	7	7	7	5	7	6	6	63

表 5-7 使用自适应遗传算法 AGA 所生成试卷的评价表

Tab.5-7 The evaluation table to the exam paper generated according to AGA

	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	总分
T1	8	7	6	7	6	7	7	7	6	9	70
T2	7	8	9	7	9	8	8	7	9	6	78
T3	8	7	8	6	8	7	6	7	8	7	72
T4	8	6	8	9	7	8	9	9	7	7	78
T5	7	7	6	8	7	7	9	7	7	9	74
T6	7	8	7	6	7	9	7	8	9	7	75
T7	8	7	9	7	7	8	9	8	7	8	78
TS8	8	8	7	8	9	7	8	9	8	8	80
T9	9	8	7	7	7	6	7	9	7	8	75
T10	8	9	8	8	9	7	6	7	8	7	77

通过对比表 5-6、表 5-7 和表 5-8 中的结果，不难发现测评小组对 IAGA 考试试卷各项指标的评价较之 SGA、AGA^[67-70]有了明显的提升，在智能组卷系统采用了使用混合熵以及改进的交叉概率和变异概率的自适应遗传算法求解改进的数学模型 (3-35)之后，不仅使得 GA 的种群多样性得以保持并且组卷系统模型求解的稳定性以及可靠性得以增加，同时整个智能组卷系统的性能也得到了提升。对于这种现象的出现，给出如下的分析：

表 5-8 使用改进的自适应遗传算法 IAGA 所生成试卷的评价表

Tab.5-8 The evaluation table to the exam paper generated according to IAGA

	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	总分
T1	9	10	8	8	8	9	8	8	8	10	86
T2	8	9	10	8	10	10	9	9	10	9	92
T3	9	9	10	10	9	9	9	8	9	8	90
T4	9	10	10	10	9	9	10	10	10	8	95
T5	9	8	8	9	8	8	10	8	8	10	86
T6	9	9	8	8	8	10	8	10	10	8	88
T7	9	8	10	8	8	9	10	9	8	9	88
T8	9	9	9	9	10	8	9	10	9	9	91
T9	10	10	8	8	8	8	8	10	8	9	87
T10	9	10	10	9	10	8	10	8	9	8	91

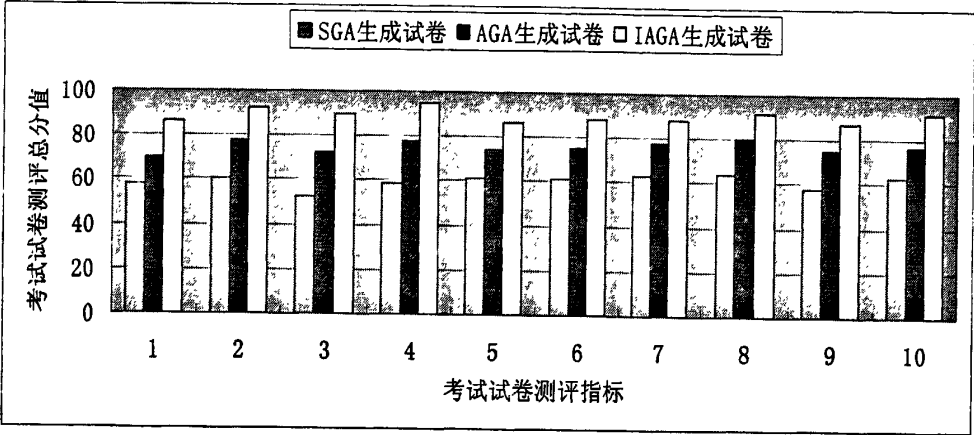


图 5-2 使用不同算法生成的试卷测评对比

Fig.5-2 The comparative results of exam papers generated according to different algorithms

- 1) 改进的数学模型的使用可以显著地消除不同的指标分量偏差由于量纲的不同对目标函数效用的影响，使得最终得到的目标函数能够更加准确地反映考试组织对智能组卷系统的要求。计算机智能组卷的本质是一个基于对试卷的质量构成影响的指标求解多目标参数的优化问题，改进的数学模型简化了问题的求解并且更具针对性的对问题进行表述；
- 2) 改进的自适应遗传算法的采用较之原有的自适应遗传算法，改进的自适应遗传算法通过使用混合熵来度量种群的多样性，并在交叉概率以及变异概率的确定方面做出了相应的改进，使之更有针对性地求解智能组卷优化问题并且改进之后的自适应遗传算法收敛速率更快以及能够更加容易的搜索到问题的全局最优解。

为了形象地显示系统在改进之后性能的显著提升,通过比较对系统的整体评价来证实这一点,系统改进前后的整体评价被显示在图 5-2 中。通过图 5-2 显示,组卷系统在使用了改进的自适应遗传算法求解改进的数学模型之后整体性能得到了显著的提升,达到了系统设计的预期目的。

5.3 HEBITPGS 关键技术详述及代码展示

在本文中利用 JSP^[63, 64]编程语言和 SQL Server^[65, 66]数据库技术开发了邢台学院《微积分》课程考试系统 HEBITPGS,现在对 HEBITPGS 的界面以及功用进行简单的叙述。

HEBITPGS 的功能比较强大,和 CAT 比较起来在实际效能上基本一样,不同的是前者的规模不如后者那样大。一般,该系统用于测试中的部分工作,但其仍可以用于整体的工作过程。专业题库或者是通用题库,在构建题库系统时,对应的学科题库会有区别,但其功能基本上是一致的。题库系统的功能一般包括建库、查询和检索。试卷的组织、发送,试题的结果分析等功能,这是对所有的题库系统的基本要求。而对于各自的学科当然需要根据其特点进行适当的修正,去适应不同的学科特点。比如,像英语等语言学科,对于图形的元素没有要求,所以为了简化系统,就可以不包括图形处理。当然,微积分课程涉及到了图形处理,在设计题库系统时就需要考虑图形问题了。试题库中的图形采用了 RTF 格式进行存储,对于图文混合信息的存储,需要设计好字段的数据类型,在 SQL SERVER 2000 中可设计 Text 或 image 类型,然后利用 RichTextBox 控件的 TextRTF 属性,获得存放在 RichTextBox 控件中的全部 RTF 格式数据,该数据是既含有文本又含有图形的字符串。下面以微积分题库系统为例,如图 5-3 所示,为该系统的功能结构图,该图反映了题库系统的基本构成和该学科对题库系统的特殊功能要求。该系统包括以下几个子系统。

(1) 建库与维护子系统

一个数据库的建立与维护是指将确定好的题库内容按照其各种特点,如图形、文字特点、独特属性等,将内容分类分别送入各个不同的下级子数据库中,这样对库中的内容进行编辑会更加方便。

众所周知,微积分课程涉及到了很多的图形和特殊符号,因此该系统必须要有对这些内容进行编辑的功能,这应该在建立和维护时就设计完善,实现系统的正常运行。同时,应当将文字、图形和特殊符号分类进行储存,这样会方便管理与编辑。为此,建立了指标库、题文库、附图库、答案库和答图库等五个子库,将不同的元素进行分类管理。部分关键代码如下:

```
<HTML>
```

```
<HEAD>
```

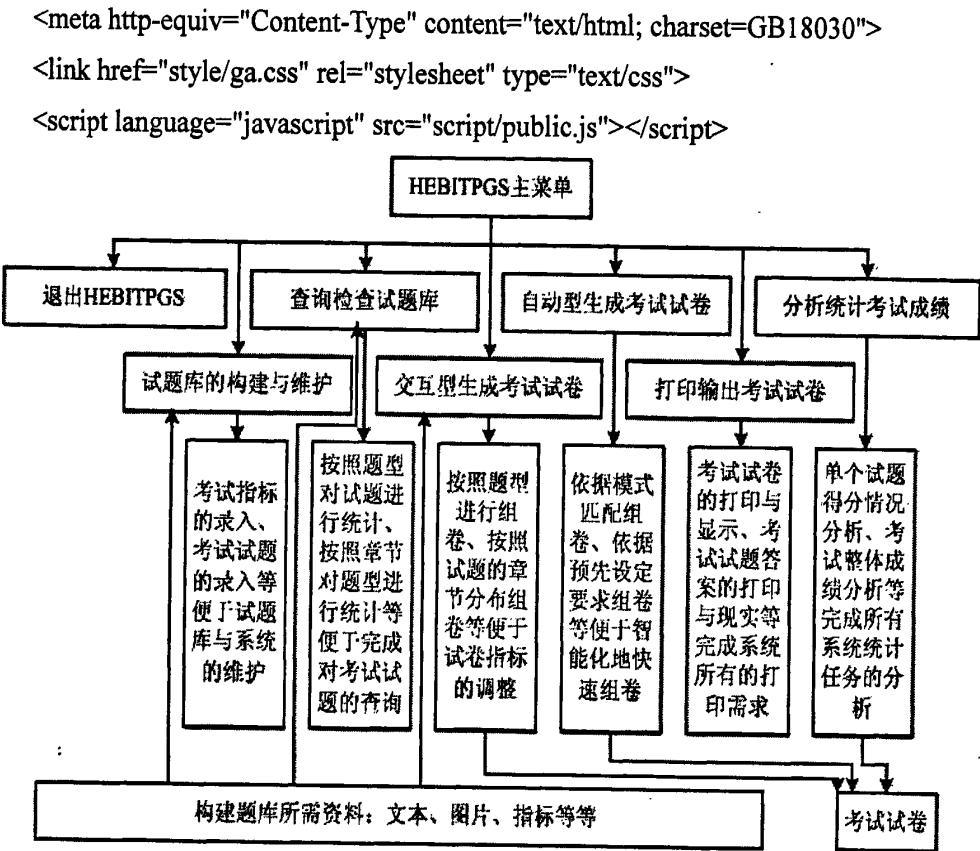


图 5-3 HEBITPGS 系统功能结构示意图

Fig.5-3 The flowchart of functions of HEBITPGS system

```
</HEAD>
<BODY bgColor=#dee7ff leftMargin=0 background="" topMargin=0 marginheight="0"
marginwidth="0">
<TABLE width="778" border=0 cellPadding=0 cellSpacing=0 borderColor=#ffffff bg
Color=#dee7ff style="FONT-SIZE: 10pt"> </TABLE>
<TABLE width="778" border=0 align=center cellPadding=0 cellSpacing=0 borderColor
=#ffffff style="FONT-SIZE: 10pt">
<TBODY>
<TR>
<TD width="82%" height=14 align=right vAlign=center no Wrap></TD>
<TD width="18%" align=right vAlign=center no Wrap></TD>
</TR>
<TR>
<TD height=14 align=right vAlign=center no Wrap></TD>
```



```

<link href="style/ga.css" rel="stylesheet" type="text/css">
<script language="javascript" src="script/public.js"></script>
<script language="javascript" src="script/findQuestion.js"></script>
<script type="text/javascript">
    function findTestQuestion(){
        var course = document.getElementById("courseId").value;
        var chapter = document.getElementById("chapterId").value;
        var section = document.getElementById("sectionId").value;
        var knowledgePoint=document.getElementById("knowledgePointId").value;
        document.getElementById("show").innerHTML="<A
href=testQuestion.do?courseId="+course+"&chapterId="+chapter+"&sectionId="+se
ction+"&knowledgePointId="+knowledgePoint+"> </A>";}
</script>
</HEAD>
<body bgColor=#dee7ff leftMargin=0 background="" topMargin=0 marginheight="0"
marginwidth="0" onload="findcs()">
<center>
    <TABLE width="778" border=0 cellPadding=0 cellSpacing=0 borderColor=#ffffff bg
Color=#dee7ff style="FONT-SIZE:10pt"></TABLE>
    <TABLE width="778" border=0 align=center cellPadding=0 cellSpacing=0 border
Color=#ffffff style="FONT-SIZE: 10pt"></TABLE>
    <TABLE width="778" border=0 align=center cellPadding=0 cellSpacing=0 border
Color=#ffffff style="FONT-SIZE: 10pt"></TABLE>
    .....
</HTML>

```

(3) 交互式组卷子系统

用户在使用时, 会根据自己的具体要求进行人机对话以满足不同的要求, 即根据具体的要求生成满足客户的具有独特属性的试卷并且可以调整。这对于特色化教学和测试有着重要的价值。部分关键代码为:

```

<HTML>
<HEAD>
    <meta http-equiv="Content-Type" content="text/html; charset=GB18030">
    <link href="style/ga.css" rel="stylesheet" type="text/css">
    <script language="javascript" src="script/public.js"></script>

```

```

</HEAD>
<BODY bgColor=#dee7ff leftMargin=0 background="" topMargin=0 marginheight="0"
marginwidth="0">
<center>
<TABLE width="778" border=0 cellPadding=0 cellSpacing=0 borderColor=#ffffff
bgColor=#dee7ff style="FONT-SIZE: 10pt">
<TBODY>
<TR height=35>
<TD align=middle width=20 background=images/title_left.gif bgColor=#
dee7ff></TD>
<TD align=middle width=11 background=images/title_middle.gif bgColor=#
dee7ff><FONT color=#f7f7f7>&nbsp;</FONT> </TD>
<TD align=middle background=images/title_right.gif bgColor=#dee7ff><FONT
color=#f7f7f7>&nbsp;</FONT> </TD>
</TR>
</TBODY>
</TABLE>
.....
</BODY>
</HTML>

```

HEBITPGS 考试系统添加考试试题界面如图 5-5 和 5-6 所示。

(4) 自动组卷子系统

组卷就是系统根据用户的要求, 基于一定的试题指标, 如试题数、试题难度等, 进行试题的设计。将该过程自动化, 只需要用户输入自己的要求, 系统就可以自动生成试卷, 关键代码为:

```

<HTML>
<HEAD>
<meta http-equiv="Content-Type" content="text/html; charset=GB18030">
</HEAD>
<BODY>
<pg:pager items="1001" export="currentPageNumber=pageNumber">
<pg:first><a href="{pageUrl}"></a></pg:first>
<pg:prev><a href="{pageUrl}"></a></pg:prev>
<pg:pages><c:choose>

```

基本信息	
总分数	100
及格分数	60
测试时间	60 (分钟)
考试次数	1
是否激活	<input checked="" type="radio"/> 是 <input type="radio"/> 否
查看答案	<input type="radio"/> 是 <input checked="" type="radio"/> 否
说明	微积分考试

基本设置	
字体	字号
段落格式	格式

图 5-5 HEBITPGS 的新增题库信息

Fig.5-5 The function of adding the new items of HEBITPGS

The image shows a computer exam interface. At the top, there is a navigation bar with buttons for 'Home', 'Exam', 'Subject', and 'Question Bank'. Below this, there are two buttons: 'Save' (保存) and 'Clear' (清空). The main area displays a question titled 'Question Information' (题目信息). The question text is: '当 $x \rightarrow 0$ 时 $1 - \cos x$ 与 x^n 为同阶无穷小, 则 $n =$ {FILL1}'. To the right of the question text, there is a dropdown menu for 'Type' (类型) set to 'Fill-in' (填空) and another dropdown for 'Difficulty' (难度) set to 'Level 1' (1级). Below these, there is a 'Score' (分数) field showing '25'. At the bottom of the question area, there is a 'Remarks' (备注) field. Below the question area, there is a toolbar with various icons for text formatting, including bold, italic, underline, and font color. The toolbar also includes dropdown menus for 'Font' (字体), 'Font Size' (字号), and 'Paragraph Format' (段落格式). The main content area below the toolbar is empty.

图 5-6 新增一道试题

Fig.5-6 Add a testing item

<c:when test="\\${currentPageNumber eq pageNumber}">

```

        <font color="red">${pageNumber}</font>
    </c:when>
<c:otherwise>
    <a href="${pageUrl}">${pageNumber}</a>
</c:otherwise>
</c:choose>
</pg:pages>
    <pg:next><a href="${pageUrl}"></a></pg:next>
    <pg:last><a href="${pageUrl}"></a></pg:last>
</pg:pager>
.....
</BODY>
</HTML>

```

(5) 打印与输出子系统

试卷在上述系统设计后, 需要根据要求输出具体格式的试卷。设定一定的指标, 就可以把任意设计好的试题从试题库中输出, 并可以进行各种格式的打印。部分关键代码为:

```

<HTML>
<HEAD>
    <meta http-equiv="Content-Type" content="text/html; charset=GB18030">
    <link href="style/ga.css" rel="stylesheet" type="text/css">
    <script language="javascript" src="script/public.js"></script>
</HEAD>
<BODY bgColor=#dee7ff leftMargin=0 background="" topMargin=0 marginheight="0"
marginwidth="0">
    <TABLE width="778" border=0 cellPadding=0 cellSpacing=0 borderColor=#ffffff
bgColor=#dee7ff style="FONT-SIZE: 10pt">
<TBODY>
    <TR height=35>
        <TD align=middle width=20 background=images/title_left.gif bgColor=
#dee7ff></TD>
        <TD align=middle width=120 background=images/title_left.gif bgColor =#
dee7ff><FONT color=#f7f7f7> <font color="#FFFFFF">&nbsp;</font></FONT>
</TD>

```

```
<TD align=middle width=11 background=images/title_middle.gif bgColor =#
dee7ff><FONT color=#f7f7f7>&nbsp;&nbsp;&nbsp;</FONT> </TD>

<TD align=middle background=images/title_right.gif bgColor=#dee7ff><FONT
color=#f7f7f7>&nbsp;&nbsp;&nbsp;</FONT> </TD>

</TR>

</TBODY>

.....

</BODY>

</HTML>
```

生成考试试卷界面如图 5-7 所示:

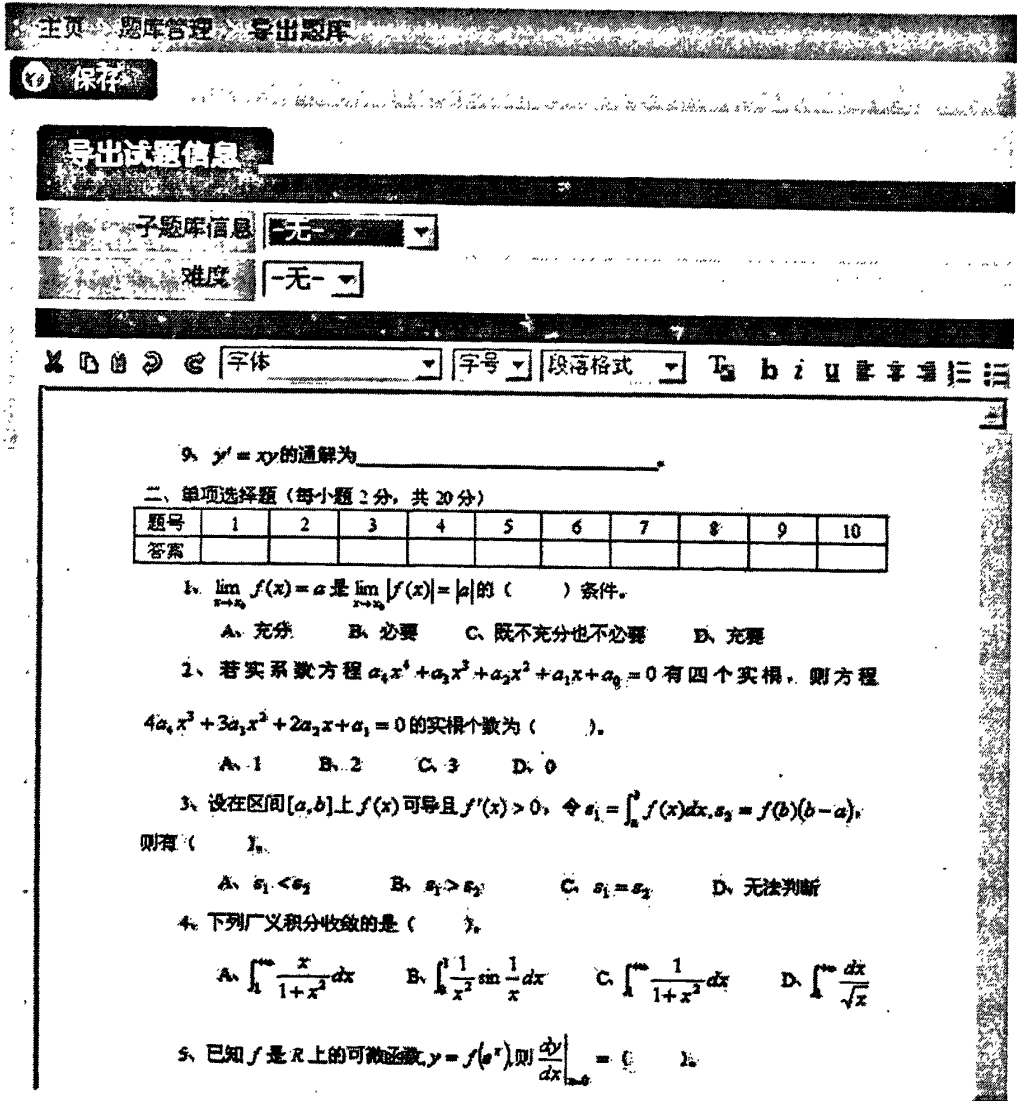


图 5-7 生成考试试题

Fig.5-7 Generate the testing paper

(6) 测试分析子系统

在测试完毕后, 需要将考生的答题情况输入, 并进行试卷的分析, 给出具体的指标, 并根据这些结果对试题库中内容进行修改完善。部分关键代码为:

```
<HTML>
<HEAD>
  <meta http-equiv="Content-Type" content="text/html; charset=GB18030">
  <script language="javascript" src="script/public.js"></script>
</HEAD>
<BODY bgColor=#dee7ff leftMargin=0 background="" topMargin=0 marginheight="0"
marginwidth="0">
<center>
  <TABLE width="778" border=0 cellPadding=0 cellSpacing=0 borderColor=#ffffff
bgColor=#dee7ff style="FONT-SIZE: 10pt">
    <TBODY>
      <TR height=35>
        <TD align=middle width=20 background=images/title_left.gif bgColor=#
dee7ff></TD>
        <TD align=middle width=11 background=images/title_middle.gif bgColor=#
dee7ff><FONT color=#f7f7f7>&nbsp;</FONT> </TD>
        <TD align=middle background=images/title_right.gif bgColor=#dee7ff><FONT
color=#f7f7f7>&nbsp;</FONT> </TD>
      </TR>
    </TBODY>
  </TABLE>
  .....
</BODY>
</HTML>
```

5.4 本章小结

本章对基于改进的自适应遗传算法求解改进数学模型的邢台学院《微积分》课程考试系统 HEBITPGS 的设计与实现过程进行了详细地介绍, 对 HEBITPGS 智能组卷模块的性能进行了实验性的验证, 最终的试验结果表明系统 HEBITPGS 的设计思路可行的、实际应用是有效的。

结 论

本文对现有智能组卷系统的相关内容进行了详细地分析与讨论,并在此基础上,针对当前智能组卷系统存在的不足,本文的主要研究内容为以下四个方面:

首先,针对已有智能组卷系统数学模型独立考虑各分指标的不足,提出了对分指标进行加权集成从而构建组卷系统数学模型目标函数的方法,对于各分指标权值的确定采用了拓扑扩展的方法,使得确定的权重可以显著地消除不同的指标分量偏差由于量纲的不同对目标函数效用的影响,从而使求得的权重可以有效地衡量指标分量偏差之间的差异程度。

接着,为了有效地求解智能组卷系统的改进数学模型,提出了一种改进的自适应遗传算法,在改进的自适应遗传算法中,对引起遗传算法过早成熟的原因进行了分析,从而提出了使用混合熵衡量种群多样性、并基于混合熵对传统自适应算法中的交叉概率和变异概率的计算公式进行了重新改进。

之后,在本文中利用 JSP 编程语言开发了邢台学院《微积分》课程考试系统 HEBITPGS,此系统具有良好简捷的操作界面,在此系统中实现了在建库维护、查询检查、交互组卷、自动组卷、试卷打印以及成绩分析等功能。

最后,对 HEBITPGS 的性能进行了实验性的验证,最终的试验结果表明 HEBITPGS 的设计思路可行的、实际应用是有效的,它有效地提高了生成试卷的质量。

参考文献

- [1] 张建国. 智能教学系统中的自动组卷算法研究. [河南大学硕士研究生学位论文], 2009
- [2] 叶颖泽, 章英. 改进的遗传算法在智能组卷中应用研究. 研究与开发, 2008, 285: 37-40
- [3] 周芬. 改进遗传算法在题库系统中的应用. 科技信息(学术研究), 2007, 36: 431-433
- [4] 段朋鹏, 郭庆平. 改进遗传算法在智能组卷系统中的应用. 软件导刊, 2008, 7(3): 118-120
- [5] 黄艳峰, 陈涛. 基于改进遗传算法的智能组卷系统的设计与实现. 煤炭技术, 2009, 28(10): 150-151
- [6] M.J. Huang, Y.B. Huang, Z.Y. Liu. Research on Auto-Composing Test Paper System Based on Improved Genetic Algorithm, Computer Applications and Software, 2007, 24(12): 191-194
- [7] K.L. Mao. The Design Of Online Test System Based On Genetic Algorithm, Journal of Xinxiang University(Natural Science Edition), 2010, 27(1): 59-61
- [8] 王一萍, 曲传建, 潘海珠. 基于遗传算法的自适应题库系统研究. 齐齐哈尔大学学报, 2008, 24(5): 1-4
- [9] 王洪峰, 吴聪. 基于遗传算法的智能组卷问题的研究. 福建电脑, 2007, 10: 28-29
- [10] 黄宝玲. 基于遗传算法的题库设计. 电脑知识与技术, 2009, 5(36): 10559-10561
- [11] 金涛. 基于遗传算法的组卷系统的研究. [燕山大学硕士研究生学位论文], 2001
- [12] 陈大平. 自动组卷算法的研究. [燕山大学硕士研究生学位论文], 2003
- [13] 刘艺. 智能教学系统中基于遗传算法的自动组卷的研究. [辽宁工程大学硕士研究生学位论文], 2004
- [14] 陈丽娜. 基于遗传算法的试题库组卷方法研究. [华东师范大学硕士研究生学位论文], 2005
- [15] 张月. 基于遗传算法的在线考试系统. [吉林大学硕士研究生学位论文], 2005
- [16] 陈蕾. 基于遗传算法的自动组卷系统研究与应用. [四川大学硕士研究生学位论文], 2006
- [17] 张彦. 遗传算法在计算机辅助考试系统自动组卷中的应用. [太原理工大学硕士研究生学位论文], 2007
- [18] 路景. 基于改进遗传算法的智能组卷研究. [中南大学硕士研究生学位论文], 2007
- [19] 林海涛. 基于遗传算法的 CAT 组卷问题研究. [大连理工大学大学硕士研究生学位论文], 2007
- [20] 徐江涛. 基于遗传算法的试题库智能组卷研究. [湖南师范大学硕士研究生学位论文], 2007
- [21] 宋正菊. 基于遗传算法的网络教学系统自适应组卷研究. [天津大学硕士研究生学位论文], 2007
- [22] 杨秀梅. 基于遗传算法的组卷系统的研究. [上海交通大学硕士研究生学位论文], 2007
- [23] 初文科. 基于遗传算法的试题库设计与组卷方法研究. [山东师范大学硕士研究生学位论文],

2008

- [24] 张义良. 基于遗传算法的网络智能考试系统的研究与实现. [湖南大学大学硕士研究生学位论文], 2008
- [25] 李军. 基于遗传算法的智能组卷系统研究. [天津大学硕士研究生学位论文], 2008
- [26] 吴树锦. 基于遗传算法智能组卷系统的研究与实现. [华东师范大学硕士研究生学位论文], 2008
- [27] 陈林琳. 试题库系统的设计与实现. [大连海事大学硕士研究生学位论文], 2008
- [28] 王珊. 无纸化考试系统组卷算法的研究与实现. [东北林业大学硕士研究生学位论文], 2008
- [29] 宋莉. 应用多方向进化遗传算法的自动组卷算法模型设计. [北京交通大学硕士研究生学位论文], 2008
- [30] 张强. 智能题库系统理论研究与实现. [辽宁工程大学硕士研究生学位论文], 2008
- [31] 马会. 基于遗传算法的智能组卷系统的研究与实现. [沈阳工业大学硕士研究生学位论文], 2009
- [32] 门立彦. 基于真实环境的计算机应用能力智能考试系统. [中北大学大学硕士研究生学位论文], 2009
- [33] 罗佳. 基于智能搜索算法的组卷系统研究与实现. [贵州师范大学硕士研究生学位论文], 2009
- [34] 金玉苹. 一种基于改进遗传算法的智能组卷方法研究. [哈尔滨工程大学硕士研究生学位论文], 2009
- [35] 卞灿. 基于人工鱼群算法的智能组卷研究. [湖南师范大学硕士研究生学位论文], 2009
- [36] 吴树锦. 基于遗传算法智能组卷系统的研究与实现. 天津职业院校联合学报, 2010, 5(2): 34-38
- [37] Z.Y. Meng. Intelligent Grouping Testpaper Based on Adaptive Immune Genetic Algorithm, Computer Engineering, 2008, 34(14): 203-206
- [38] Y. Liu, Y. Wang, Y. Du, J.Y. Zhang. Multi-object intellectual test paper assembling based on adaptive operator genetic algorithm, Journal of Computer Applications, 2008, 28: 22-24
- [39] 许璨. 基于自适应遗传算法的智能组卷策略. 光盘技术, 2008, 7: 42-43
- [40] 李国伟, 周雪燕, 李钜. 潜在倾向理论在计算机题库建库中的应用. 中原工学院学报, 2008, 19(2): 72-75
- [41] 胡中锋, 李方. 教育测量与评价. 广东高等教育出版社, 1999, 3
- [42] 王友仁, 陈则王, 洪春梅, 林华. 数字电子技术. 机械工业出版社, 2010
- [43] 河蓉, 李支尧, 刘翠英. 试题库智能组卷算法研究. 福建电脑, 2009, 11: 10-11
- [44] F. Liu, S.X. Gao. Research and Application for the Algorithmic of Examination Questions, Journal of Jixi University, 2008, 8(6): 141-142

- [45] 孟宇桥. 自动组卷系统与遗传算法. 吉林工程技术师范学院学报(自然科学版), 2006, 22(9): 13-16
- [46] H.B. Li, C.Y. Zhou. Application of a Progressive genetic Algorithm in Generating Test Paper Intelligently, Science and Technology of West China, 2008, 7(8): 50-51
- [47] 吴迪, 龚启军. 遗传算法实现试题库智能组卷方法研究. 中国校外教育, 2009, 2: 94-95
- [48] 魏德志, 林丽娜, 吴旭. 遗传算法在智能组卷系统中的设计与实现. 长江大学学报(自然科学版), 2008, 5(1): 247-249
- [49] L.D. Liu. The Application of Genetic Algorithm in Maneuver of Composing Examination Papers, Journal of Guangdong Education Institute, 2010, 30(3): 75-79
- [50] H. Xu, R. Li, K. Zeng, J. Li Some applications of genetic algorithm to designing of the automatic test paper generation system, Journal of Shaanxi Normal University(Natural Science Edition), 2008, 36: 152-154
- [51] D.E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning, Kluwer Academic Publishers, 1989
- [52] A.E. Eiben, R. Hinterding, Z. Michalewicz. Parameter control in evolutionary algorithms, IEEE Transactions on Evolutionary Computation, 1999, 3(2): 124-141
- [53] Ho Shinn-Ying, Shu Li-Sun, Jian-Hung Chen. Intelligent evolutionary algorithms for large parameter optimization problems, IEEE Transactions on Evolutionary Computation, 2004, 8(6): 522-541
- [54] Leung Yee, Gao Yong, Xu Zong-Ben. IEEE Transactions on Neural Networks, 1997, 8(5): 1165-1176
- [55] F. Herrera, M. Lozano. Adaptive genetic operators based on coevolution with fuzzy behaviors, IEEE Transactions on Evolutionary Computation, 2001, 5(2): 149-165
- [56] M. Srinivas, L.M. Patnaik. Adaptive Probabilities of Crossover and Mutation Genetic in Algorithms, IEEE Transactions on Systems, Man and Cybernetics, 1994, 24(4): 656-667
- [57] J.J. Grefenstette. Optimization of control parameters for genetic algorithms, IEEE Transactions on Systems, Man, and Cybernetics, 1986, 16(1): 122-128
- [58] A.E. Eiben, Z. Ruttkay. Self-adaptivity for constraint satisfaction: Learning penalty functions, In Proceedings of the 3rd IEEE International Conference on Evolutionary Computation, 1996: 258-261
- [59] D Schlierkamp-Voosen, M uhlenbein. Adaption of population sizes by competing subpopulations, In Proceedings of the 1996 IEEE Conference on Evolutionary Computation, 1996: 330-335
- [60] J. Smith, T. Fogarty. Self-adaptation of mutation rates in a steady-state genetic algorithm, In Proceedings of the 3rd IEEE International Conference on Evolutionary Computation, 1996:

318-323

- [61] 岳盼想, 陈金梅, 李法朝. 带混合算子的自适应遗传算法的收敛性研究. 河北科技大学学报, 2006, 4: 272-276
- [62] 刘立民, 靳晨霞, 杨丽芸, 李法朝. 两阶段遗传算法的结构及性能分析. 河北科技大学学报, 2007, 1: 44-48
- [63] 刘砚秋, 郑鲲. PETS 网上报名系统的设计与实现. 河北工业科技, 2011, 1: 55-58+71
- [64] 王伍伶, 靳丽敏, 闫玉根. JSP 与 Web 数据库的连接技术. 河北工业科技, 2002, 3: 23-26
- [65] 陈显中, 张健. 基于 SQL Server 应用系统的安全性研究. 河北工业科技, 2003, 3: 10-13
- [66] 李颖, 张世民. 基于 Web 的高校户籍管理系统研究. 河北工业科技, 2009, 5: 319-320+334
- [67] R.J. Kuo, L.M. Lin. Application of a hybrid of genetic algorithm and particle swarm optimization algorithm for order clustering, Decision Support Systems, 2010, 49(4): 451-462
- [68] D.X. Chang, X.D. Zhang, C.W. Zheng, D.M. Zhang. A robust dynamic niching genetic algorithm with niche migration for automatic clustering problem, Pattern Recognition, 2010, 43(4): 1346-1360
- [69] C. Bielza, J.A. Fernández del Pozo, P. Larrañaga, E. Bengoetxea. Multidimensional statistical analysis of the parameterization of a genetic algorithm for the optimal ordering of tables, Expert Systems with Applications, 2010, 37(1): 804-815
- [70] J.W. Gu, M.Z. Gu, C.W. Cao, X.S. Gu. A novel competitive co-evolutionary quantum genetic algorithm for stochastic job shop scheduling problem, Computers & Operations Research, 2010, 37(5): 927-937

攻读硕士学位期间所发表的论文

- [1] 梁海丽. 浅谈《数据结构与算法》课程教学的改革. 邢台学院学报, 2009, (6): 104-105
- [2] 梁海丽. 计算机网络技术对物流信息化的影响. 中国商贸, 2011, (3) 138-139
- [3] 梁海丽. 通用智能组卷系统的设计与实现. 价值工程, 2011,(4)

致 谢

本课题的研究和论文的撰写得到了导师王晓红教授的精心指导和帮助，在此对王老师表示由衷的感谢和深深的敬意！王老师开阔的眼界、敏捷的思维以及严谨的治学态度和一丝不苟的工作作风，是我学习的榜样。

我还要感谢给研究生任课的各位老师，他们对我在研究生阶段的学习和生活都给予了极大的帮助。我要感谢院里所有进行研究生教学和管理的老师们，没有他们的辛勤工作也就没有的今天。

同时，我要感谢我的家人和同学们长久以来对我学业的理解与支持，由于他们的帮助我才克服了重重困难，最终完成了本文的全部工作。

感谢所有关心和帮助过我的人。

个人简历

姓名：梁海丽

性别：女

民族：汉

籍贯：河北邢台

出生年月：1972 年 8 月

个人简历：2000 年 6 月毕业于天津大学计算机及应用专业，获工学学士学位；
2000 年至今于邢台学院信息科学与技术系工作

