

分类号: TP391

密 级: 公开

U D C: 004

单位代码: 10360

安徽工业大学  
硕士学位论文

论文题目: 基于遗传算法的自动组卷系统优化设计及应用



学 号: 1723190667

作 者: 张海洋

专 业 名 称: 计算机技术

2019 年 04 月 09 日

# 安徽工业大学硕士学位论文

论文题目: 基于遗传算法的自动组卷系统优化设计及应用

Optimization Design and Application of Automatic Test  
Paper System Based on Genetic Algorithm

作 者: 张海洋 学院: 计算机科学与技术学院

指 导 教 师: 刘宏申 单位: 安徽工业大学

协助指导教师: \_\_\_\_\_ 单位: \_\_\_\_\_

论文提交日期: 2019 年 4 月 9 日

学位授予单位: 安 徽 工 业 大 学

安徽马鞍山 243002

## 独 创 性 说 明

本人郑重声明：所呈交的论文是我个人在导师指导下进行的研究工作及取得研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写的研究成果，也不包含为获得安徽工业大学或其他教育机构的学位或证书所使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

签名 张海洋 日期： 2019.5.30

## 关于论文使用授权的说明

本人完全了解安徽工业大学有关保留、使用学位论文的规定，即：学校有权保留送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文，保密的论文在解密后应遵循此规定。

签名 张海洋 导师签名 刘宏申 日期： 2019.5.30

## 摘要

信息化高速发展的今天,各行各业都离不开计算机的帮助。以自动组卷为核心的在线考试系统广泛应用于各类考试之中。自动组卷系统的应用不仅能够减轻教师的工作压力,节省出卷时间,还能有效的减少教师在出卷时的个人主观因素,提高试卷质量,提高考试的标准化程度。

组卷问题的实质是求解多约束条件下的多目标优化问题。遗传算法作为一种高效的全局并行搜索算法,其简单易用、适于并行处理和强鲁棒性的特点使其非常适于用来解决此类多目标优化问题。本文以自动组卷作为研究内容,着重研究了自动组卷的优化问题模型以及如何利用遗传算法来实现。

本文首先在大量研究国内外文献的基础上,分析组卷系统中重要的试题属性以及试卷指标,明确组卷的基本步骤。然后结合试题的关键属性确定了组卷的关键约束条件,以此建立合适的数学模型和极小化目标函数。针对组卷问题引入遗传算法。最后实验运行结果表明采用本算法能够成功组卷并有效提高组卷质量。

在开发该系统的时采用了 C/S 架构, C#开发语言, 运用 VS 开发平台, 结合 ADO.NET 数据连接数据库。通过运行证明系统运行稳定,具有一定的使用价值。

**关键字:** 自动组卷;遗传算法;遗传算子;多目标优化

## Abstract

At present, along with the rapid development of information technology, all walks of life are inseparable from the help of computers. Online test systems based on automatic test paper generation system have been widely applied in various tests. The application of the automatic test paper generation system can not only reduce the teachers' working pressure and save the time spent on test paper design, but also effectively reduce teachers' individual subjective factors during test paper design, improve the quality of the test paper, and enhance the standardization of the test.

The essence of paper generation problem is to solve multi-objective optimization problems. As an efficient global parallel search algorithm, it is very suitable for solving such multi-objective optimization problems using the genetic algorithm because of its simplicity and robustness. With the automatic test paper generation system as the object of research, this paper focuses on the optimization problem model of automatic test paper generation system and how to realize it using the genetic algorithm.

Firstly, based on a large number of domestic and foreign literatures, this paper analyzes the important question attributes and test paper indexes in the test paper generation system, and clarifies the basic steps of test paper generation. Then combined with the key attributes of the test questions, the key constraints of test paper generation are determined, thus establishing a suitable mathematical model and minimizing the objective function. The genetic algorithm is introduced to solve the test paper generation problem. Finally, the experimental results show that this algorithm can be used to generate the test paper successfully and improve the quality of the test paper.

The genetic algorithm is then appropriately modified. Design coding methods and population initialization methods, as well as design genetic operators and algorithm termination conditions. Finally, experiments show that the system using the algorithm in this paper can successfully roll up the volume and effectively improve the quality of the test paper.

In the development of the system, C/S architecture, C # development language and VS development platform are used, and the database is connected combined with ADO.NET data. It is proved that the system runs stably and has certain application value.

**Keywords:** automatic test paper; Genetic Algorithm; Genetic operator; Multi-objective Optimization

## 目录

摘要 .....	I
Abstract .....	II
第 1 章绪论 .....	1
1.1 课题研究的背景和意义 .....	1
1.2 研究现状 .....	1
1.2.1 国外研究现状 .....	2
1.2.2 国内研究现状 .....	2
1.3 本文的主要工作 .....	2
第 2 章试卷评价指标及组卷过程分析 .....	4
2.1 试题的主要属性分析 .....	4
2.1.1 题型 .....	4
2.1.2 难度 .....	4
2.1.3 区分度 .....	5
2.1.4 章节 .....	5
2.1.5 曝光度 .....	6
2.2 试卷的主要评价指标 .....	6
2.2.1 试卷难易度指标 .....	6
2.2.2 试卷的覆盖度指标 .....	7
2.2.3 试卷的相似度指标 .....	7
2.2.4 试卷的区分度指标 .....	8
2.3 组卷的基本步骤 .....	8
2.4 本章小结 .....	11
第 3 章多目标优化与遗传算法 .....	12
3.1 多目标优化及组卷数学模型 .....	12
3.1.1 多目标优化简介 .....	12
3.1.2 组卷数学模型和约束条件 .....	13
3.1.3 试卷的目标函数 .....	15
3.2 遗传算法 .....	16
3.2.1 传统遗传算法简介 .....	16
3.2.2 种群的初始化以及染色体编码 .....	17
3.2.3 种群大小和适应度函数 .....	18
3.2.4 遗传算子和终止条件 .....	18
3.3 遗传算法的特点 .....	19
3.4 本章小结 .....	20
第 4 章系统实现 .....	21
4.1 组卷的基本要求和原则 .....	21
4.1.1 组卷要求 .....	21
4.1.2 组卷原则 .....	22
4.2 开发环境与相关技术 .....	23

4.2.1 开发环境 .....	23
4.2.2 系统架构 .....	23
4.2.3 ADO.NET 技术 .....	24
4.2.4 SQL SERVER 数据库技术 .....	24
4.3 系统设计 .....	24
4.3.1 可行性分析 .....	25
4.3.2 总体架构设计 .....	25
4.3.3 系统数据库设计 .....	26
4.4 遗传算法在系统中的实现 .....	31
4.4.1 染色体的编码 .....	31
4.4.2 初始种群的确定 .....	32
4.4.3 适应度函数确定 .....	32
4.4.4 精英保留策略 .....	33
4.4.5 选择操作 .....	33
4.4.6 交叉操作 .....	34
4.4.7 变异操作 .....	35
4.4.8 算法终止条件 .....	36
4.5 组卷实现的总体过程 .....	36
4.6 本章小结 .....	38
第5章系统运行分析 .....	39
5.1 测试过程及分析 .....	39
5.1.1 参数设置 .....	39
5.1.2 运行分析 .....	40
5.2 组卷算法分析 .....	41
5.2.1 遗传算法效率测试 .....	42
5.2.2 遗传算子测试 .....	42
5.2.3 精英保留策略对系统影响 .....	43
5.3 试卷的评价信息 .....	44
5.4 本章小结 .....	45
第6章总结与展望 .....	46
6.1 总结 .....	46
6.2 后续工作和展望 .....	46
参考文献 .....	47
在学研究成果 .....	50
致谢 .....	51

## 第 1 章绪论

### 1.1 课题研究的背景和意义

进入信息化时代的今天,传统教育考试不断地面对着信息技术发展带来的冲击,但同时也带来了新的发展机遇。在学校教育中,试卷的质量对于考试来说至关重要,考试成绩将对学校教学评价以及后续教学计划的制定产生直接影响。传统的试卷是由相关任课教师手工编写。教师在出卷过程中不仅要考虑教学大纲、考试大纲、教学进度,还要考虑试卷难度、题型、题量、试卷格式等多方面要求。以至于教师需要花费大量精力和时间来编写试卷,这无形中大大增加了教师的工作压力和强度。与此同时,试卷不可避免的会被教师的一些个人主观因素所影响。因此,传统的组卷方式在现代化教育中急需做出改变<sup>[1]</sup>。

通过组卷系统来生成试卷,一方面可以减轻教师工作量,节省出卷时间,使其可以将更多的时间和精力放在教学工作,帮助教学质量的提升;另一方面可以减少教师出卷时的主观因素,有利于提高试卷质量,保证考试的公正性、公平性、规范性和科学性。让考试结果可以客观的反应教师教学水平以及学生的学习情况。除此之外,组卷系统的应用还有利于对各个专业的试卷、考试试题以及学生成绩的管理和分析。这些都是宝贵的数据资源,对于以后的教学都有重要的参考和使用价值。

组卷系统对于提高教学效率和考核教学的科学性的重要作用不言而喻。而组卷的策略的选择和应用对于组卷系统来说是最为核心的部分。本课题主要研究的内容即为系统设计和组卷策略的选择和应用。

### 1.2 研究现状

智能组卷是计算机相关技术和教育考试两者作用的产物。随着计算机软硬件技术的快速发展,以及对教育更高工作效率和更高试卷质量的迫切追求。国内外诸多机构一直在此方向上做相关的努力和研究。



### 1.2.1 国外研究现状

计算机技术与教育教学相结合的系统首先是由西方国家开发出来的,由于国外起步较早,研究时间较长,所以已经出现较多成熟的技术和系统。美国国际商用机器公司于上世纪六十年代开发出世界上第一个试题和试卷管理系统CTSS系统<sup>[2]</sup>。虽然题库中试题数量不是很多,但是其依旧有效地提高了组卷效率。CTSS系统成为了一个成功的应用典范。著名的美国思而文学习系统公司(Sylvan Learning Systems)其考试涉及的语言种类丰富、考试类型众多、以及在全球六大洲共计140多个国家和地区均设有考试中心。为全球范围内的用户提供在线教育、考试测评等服务。与此同时,国外还有众多成熟的等级考试系统诸如美国TOEFL、GRE等考试系统<sup>[3]</sup>。虽然他们开发出许多的成熟系统,但是这些公司的开发技术、设计思想等通常是保密的,所以我们想要提升自己的现代教育管理水平和自身的不懈努力,并结合自身发展的实际需求去研究、设计,完成此类系统。

### 1.2.2 国内研究现状

国内相关理论技术虽然起步较晚,但随着现代化教育对教育的信息化要求越来越高,国内企业和高校对计算机在线教育、考试考核等教辅系统也越来越重视。在组卷系统和在线考试方面相关技术已经日趋成熟,已经从最开始的试题库简单,系统功能单一,发展到试题库丰富全面,功能日趋完善<sup>[4]</sup>。并且产生了大量成功的应用案例,如驾驶证考试系统,科目一和科目四的考试就是一个成熟的在线考试系统,其通过一定的组卷策略在题库中快速抽取试题,生成试卷以及答案,在考生答案确定后就对考生答案进行判断并显示出来,考试结束时考生的考试结果也显示出来。除此之外计算机等级考试、普通话考试等也是一个成熟的组卷考试系统。组卷系统相关的系统在生活中非常多且广泛,并且对其需求依旧非常大。

## 1.3 本文的主要工作

本文的主要工作是要设计并实现基于遗传算法的自动组卷系统。此系统首先要确定系统的整体框架,需求分析,关键技术等,接着设计合适的试题库系统,后根据试题、试卷相关的属性和约束条件,明确之间的关系,主要有试题的难度、区分度、曝光度,试卷的分值条件、难易度条件、覆盖度条件、相似度以及区分度等来完成符合要求的组卷工作,完成组卷。

本文共分为六章,每章节具体内容如下:

第一章绪论主要介绍自动组卷的研究背景和意义,以及该课题现阶段的国内外研究现状,并介绍本文的主要工作内容。

第二章重点介绍组卷的主要评价指标及组卷过程,详细介绍本文试题库中试题的主要属性,以及对试卷质量产生影响的主要评价指标,最后介绍本文中所选择的两种组卷流程。

第三章多目标优化与遗传算法。通过介绍多目标优化问题引出本文对组卷问题的数学建模,再介绍作为本文组卷策略的遗传算法,最后指明本文如何选用此算法。

第四章系统实现。主要对开发过程中所涉及的一些技术条件等加以说明,并将改进型的遗传算法应用到自动组卷中。

第五章系统运行结果分析。通过对运行结果的分析得出该算法在实际应用中的拥有较好的效果。

第六章主要对本文进行总结以及对后续工作的展望。概括本文的主要研究内容和研究成果。给整个系统给予客观的评价,指出不足之处,并对今后的研究工作与改进方向进行相关的展望。

## 第2章 试卷评价指标及组卷过程分析

组卷的目的是按照一定要求形成试卷。试卷的好坏需要有一定的指标来衡量，因此在组卷之前必须要确定好相应的试卷衡量指标。与此同时，试卷又是由一道道试题来组成，试题的属性对整张试卷的评价指标会产生直接影响，所以本文将先从试题的属性开始讲解。

### 2.1 试题的主要属性分析

为了能够生成好的试卷，试卷中试题必须要满足一定的条件，主要是通过试题的属性值来判断。所以我们在试题库建立的时候就需要充分考虑每道试题所必要的属性。以下是本文中试题的主要属性。

#### 2.1.1 题型

试卷中试题分主观题和客观题。主观题主要有简答题、计算题、设计题等题型，客观题主要有选择题、填空题、判断题等题型。试卷中具体题型由教师按照考试要求来选择确定，无需量化。试题在存入题库的同时就需要输入此属性。题型属性对于后期评价试卷的题型分值分布等具有重要作用。

#### 2.1.2 难度

难度是试题的核心属性之一，其可以反映学生解答试题的情况。本文中试题难度值由具有丰富教学经验的教师结合教学经验来给定。难度值需要在一个合理的范围之内，本文中的难易度取值范围为0.0~1.0之间，其值越大表示难度越低，能够答对的学生也就越多；难度值越小表示试题的难度越高，则能够答对的学生数就越少<sup>[7]</sup>。

在录入试题时我们要保障各个难度的试题都录入，且其数量要分配合理。这样有助于高质量试题库的建立，间接地有助于试卷质量的提升。

对于每次考试试卷结果出来后都要对试题的属性进行修正。试题新的难度公式为：

$$D = \bar{X}/A \quad (2-1)$$

其中 $D$ 表示试题 $i$ 的难度值,  $\bar{X}$ 表示考试中学生在该道试题的平均得分,  $A$ 表示该道试题的实际满分分值。从上式中可以明显看出平均得分越高则试题难度值 $D$ 越高, 即试题越简单; 平均得分越低则 $D$ 值越低, 即试题越难。

### 2.1.3 区分度

试题的区分度主要是用来评价学生对相关知识内容的掌握情况。区分度的高低直接影响学生的在该道试题上的得分高低, 具有鉴别学生学习能力水平的作用, 所以区分度又称试题的鉴别力<sup>[8]</sup>。区分度是试题的重要属性, 也是评价试卷的指标之一。在实际选取试题时应当尽量选择区分度高的试题, 避免或者尽量少的选取区分度低的试题。本文中试题的区分度由教师在录入试题时给定, 初始值都是估计值。

对于每次考试试卷结果出来后都要对试题的属性进行更新。试题新的区分度通常采用高低分组的计算方法。具体公式如下:

$$T = (\bar{h} - \bar{s})/D \quad (2-2)$$

分组情况我们按成绩最高的20%为高分组, 成绩最低的20%为低分组。用 $h$ 表示高分组的在试题上的得分, 用 $s$ 表示低分组在试题上的得分,  $D$ 为试题的满分分值。 $\bar{h}$ 为高分组的平均得分,  $\bar{s}$ 为低分组的平均得分。 $T$ 为试题的区分度, 根据经验对区分度的评价如下表2-1

表2-1 试题区分度等级分类表

区分度值 $T$	评价
$\geq 0.4$	高(非常适合)
$0.3 \leq T < 0.4$	较高(适合)
$0.2 \leq T < 0.3$	较低(酌情使用)
$T < 0.2$	低(淘汰)

由上表可知, 理想的区分度值应该在0.3以上比较适宜, 在0.4之上表示此题非常合适。

### 2.1.4 章节

在本文中, 试题的章节信息主要用来确定试题的知识点分布, 以及后期生成试卷的覆盖度评价。所以试题的章节信息是试题必不可少的一项基本属性。在本文中试卷与章节属于包含关系, 试卷中只要出现某一章节的知识点, 则表示该章节在本张试卷中已考察, 即包含了该章节。

章节信息在本文中无需量化表示,该属性在录入试题时就用相应的数值标注好其所属章节信息。后期对试卷覆盖度进行检测时,直接通过此属性来判断试卷范围是否已经覆盖此章节。

### 2.1.5 曝光度

曝光度是记录试题被使用的信息。此信息对于组卷来说非常重要,为了保障考试的保密性、公平性必须要求试卷中的试题在过去出现的次数尽量少<sup>[9]</sup>。通过曝光度可以方便的来判断试题在过去被使用的情况。在本文中,试题初始的曝光度主要是记录该试题过去三年的使用情况,在成功组卷并在试卷被确认使用后,系统将会自动更新试题的曝光度。

在录入试题曝光度时需要注意,如果此试题在过去试卷中被使用则记录相应试卷的试卷号方便后期核对查找;如果试题从未被使用,则此值为空。后期评价试卷时,直接读取试题的曝光度值,通过统计此次试卷中与过去相应年份试卷重复的试题的总分值来判断此张试卷的曝光度。

## 2.2 试卷的主要评价指标

在实现本文组卷系统的过程中,需要考虑试卷的难易度、相似度、覆盖度等多方面的试卷指标,只有满足这些约束条件后,组出的试卷才是合格的。试卷合格是保证考试质量的关键和基础。本文中评价试卷质量的指标有试卷的难易度、覆盖度、相似度、分值以及区分度等。这其中有的指标是必须要满足的,只要不满足即判定组卷失败,如试卷的分值条件、覆盖度条件,试卷的总分值及覆盖度必须要满足。有的指标是可以优化趋近取值的,即尽可能的往目标值靠近,如试卷的难易度、区分度、相似度指标,这些指标往往无法同时全部满足,这时候就需要相互之间进行权衡,尽可能的往目标值靠近<sup>[22][23]</sup>。有的指标是排它性的,如试卷的章节指标,即只要有题目涉及到相应章节,则可以认为试卷内容已覆盖此章节。下面将具体讨论本文试卷的主要评价指标。

### 2.2.1 试卷难易度指标

如何来科学的设定试卷的难易度,对于本系统来说是一个非常重要的问题。在本文中我们将试卷的难度定义为以下形式:

$$D = \sum_{i=1}^N D_i S_i / \sum_{i=1}^N S_i \quad (2-3)$$

其中,  $D_i$  为试题  $i$  的难度,  $S_i$  为第  $i$  道试题的满分分值,  $N$  为试卷的试题数。由上式可知,  $D$  值越大表示试卷的难度越小;  $D$  值越小表示试卷的难度越大。为了更加清晰的表述试题的难度情况, 我们将试卷的难易度值分为五个等级<sup>[10]</sup>, 其等级分类如表。

表2-2试卷难度等级分类表

0.0~0.2	0.2~0.4	0.4~0.6	0.6~0.8	0.8~1.0
难	较难	适中	较易	易

通过大量的考试经验分析, 让试卷的难以度控制在0.3~0.7范围内为正常可以接受的程度。实践过程中如无特殊情况, 通常将难度控制0.4~0.6之间比较科学<sup>[11]</sup>。难度在此范围内有利于检测学生对课程知识的掌握情况以及老师的教学实际效。如果是已淘汰为主的选拔性的考试, 可以适当将难度增大。

### 2.2.2 试卷的覆盖度指标

试卷的覆盖度主要用来评价考试大纲要求的相关章节知识点在该试卷中的覆盖情况。很显然, 覆盖度越高, 那么整张试卷越合理, 覆盖度越低, 那么试卷需要改进的地方就越多。对于本课题的覆盖度我们主要由试卷中试题的章节信息分析得出。

对于覆盖度我们首先要计算它的整体覆盖度, 即试卷中是否包含了考试大纲要求的多有章节信息。公式如下:

$$C = S_i / S \quad (2-4)$$

其中  $S$  表示考试大纲要求考核的章节数,  $S_i$  为试卷中实际包含的章节数量。 $C$  值为试卷的章节覆盖度值。覆盖度是试卷必须要满足的指标。

### 2.2.3 试卷的相似度指标

对于一张试卷来说, 试卷中试题要尽可能的从未出现过或者在过往考试试卷中尽可能少出现。这样能够最大化的保证考试的公平性, 减少作弊等情况出现。这样有利于检测学生学习成果和教师的教学能力<sup>[12]</sup>。

本文中相似度主要是将本次试卷的试题与过往试卷进行对比, 对重复出现的试题以及它们的分值进行统计。以下为本文中相似度的计算公式:

$$R_{AB} = A \text{ 试卷与 } B \text{ 试卷中相同试题的总分值} / \sum_{i=1}^N A_i \quad (2-5)$$

上式中 $A_i$ 为试题  $i$  的分值。 $N$  表示试题的数量。 $R_{AB}$ 表示试卷 A 与之前考试用过的试卷 B 相比较, 试卷 A 的相似度。

本文中对于试卷的相似度主要由与此次考试最近的三次同级别同科目的试卷比较分析得来。因为时间上越接近本次考试的试卷, 其参考意义越大。分别用  $r_1$ 、 $r_2$ 、 $r_3$ 表示最近 1、2、3 次的试卷的相似度, 可以用集合 $[r_1, r_2, r_3]$ 来表示。时间越近的试卷的相似度对于本次试卷相似度指标评价指标权重越大, 分别按 5:3:2 的权重来计入试卷总得相似度。 $R = 0.5r_1 * 0.3r_2 * 0.2r_3$ 表示本次试卷的实际相似度。

#### 2.2.4 试卷的区分度指标

区分度指标对于试题以及试卷来说都是一个非常重要的指标。本文中录入时试题的区分度已经给出, 根据试题的区分度值, 试卷的区分度可以通过以下公式计算出来。

$$Q = \sum_{i=1}^N T_i S_i / \sum_{i=1}^N S_i \quad (2-6)$$

其中  $Q$  表示试卷的区分度值,  $T_i$ 为试题  $i$  的区分度,  $S_i$ 为试题  $i$  的分值。 $Q$  值越高说明考生在此试卷中的考核成绩区别越明显, 能够很好的反应学生的学习差距, 对于反馈其学习情况具有很好的指导作用。 $Q$  值越低说明学生通过此试卷考核的成绩差别不大, 说明不能很好的反应学生之间的学习差距, 无法有效地向教师反馈学习信息, 不利于教师后续的教学方向改进, 同时也不利于学生对于自身学习方法和成果的检测。

本文区分度值由授课教师结合经验给出如下评价标准:

表 2-3 试卷区分度等级分类表

差	一般	良好	优
$Q \leq 0.3$	$0.3 < Q < 0.4$	$0.4 \leq Q < 0.8$	$Q \geq 0.8$

#### 2.3 组卷的基本步骤

组卷系统就是让计算机执行系统中已经设定好的组卷程序, 从试题库中选取合适的试题组成试卷。完成的试卷需要满足教学大纲的要求, 知识点分布, 难度等条件。通过试卷能够科学地考察学生对知识实际的掌握情况, 同时反映教师的教学情况。

在本文中, 组卷策略有两种选择。第一种是从整体上对试卷整体提出一些要求, 主要有试卷总的分值、难易度、覆盖度、区分度、相似度、题数、题型数、

各题型分值。第二种是将试卷按题型分成相应的模块，分别对各题型模块提出相应的要求，主要有各题型模块的分值、题数、难易度、覆盖章节、区分度、相似度等。

第一种方式策略，系统主要执行步骤如下图：

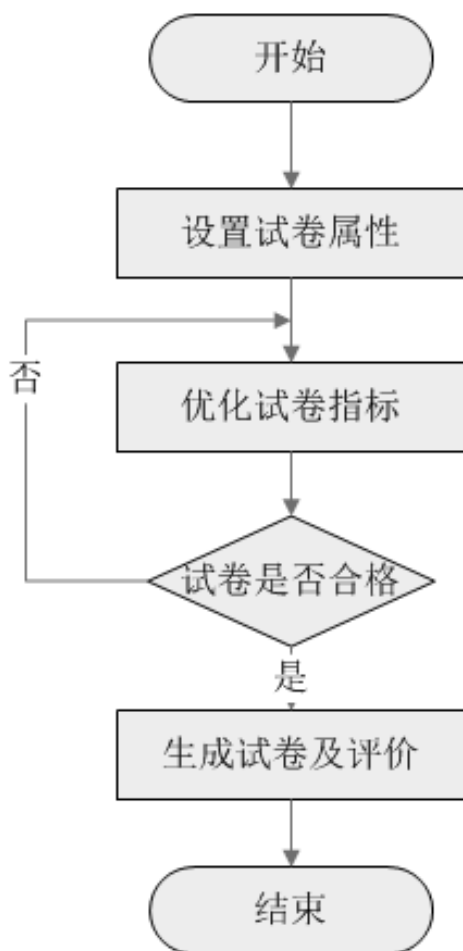


图 2.1 策略一组卷流程

- 1) 确定试卷课程、分值、章节、难易度、覆盖度等总体属性。
- 2) 通过系统已设定好的程序和算法对试卷各指标进行优化。
- 3) 判断试卷是否符合要求，如果符合要求则继续进行下一步；如果不符合要求则返回继续对试卷进行优化。
- 4) 输出试卷，同时产生该套试卷的相关评价信息。

第二种组卷策略的话，其主要过程如下图。



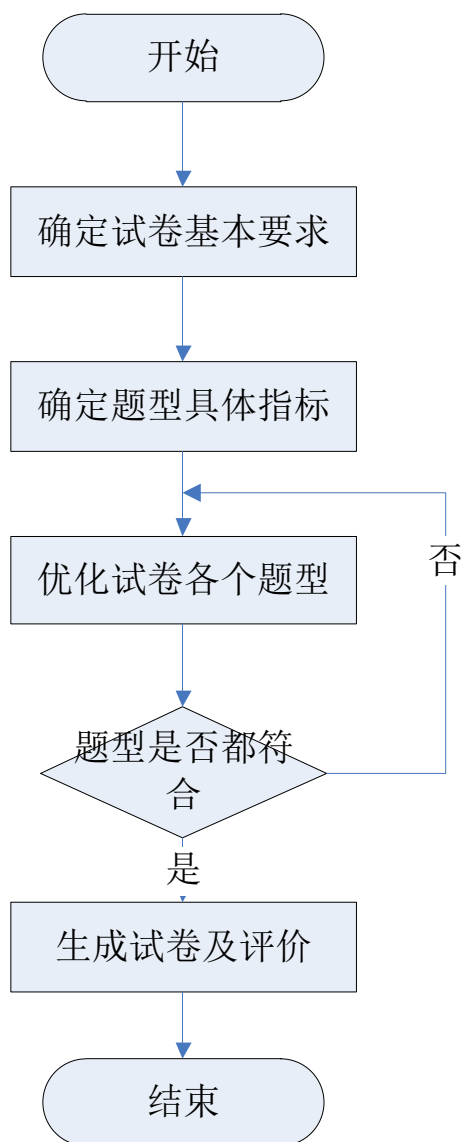


图2.2策略二组卷流程

- 1) 首先确定试卷的总分值、题型、题数等基本要求。
- 2) 确定各题型的难易度，区分度，覆盖度，相似度等指标。
- 3) 通过系统对各题型进行优化。
- 4) 对优化后的各题型进行判断，如果符合要求，则进行下一步，否则返回上一步。
- 5) 输出试卷同时生成试卷的评价指标。

从上述两种组卷策略我们可以清楚的看到，无论以何种方式进行组卷，都需要考虑到难易度、区分度、相似度等优化指标。而在现实操作过程中，这些指标间都是相互影响制约的，对系统来说是个多目标优化问题。

## 2.4 本章小结

本章主要对组卷过程的一些重要指标进行介绍,从试题的题型、难度、区分度、章节、重复度等重要属开,在此基础上对试卷的评价指标展开讨论,本文中主要讨论试卷的难易度、覆盖度、区分度、相似度等指标。最后介绍组卷的两种策略,一种是从整张试卷的指标要求入手来组卷;另一种是通过将各个满足要求的题型组合在一起完成组卷。

由于约束试卷质量的指标较多,属于多目标优化方面的问题,我们需要对其进行进一步的研究。接下来我们将具体介绍与本文有关的多目标优化问题,以及本文所要使用的组卷策略。

## 第3章多目标优化与遗传算法

本文中的组卷问题是一个需要考虑难易度、分值条件、覆盖度等多个约束条件的多目标优化问题。各个约束条件之间相互联系、相互约束，一个目标的实现往往是以牺牲其他目标为代价，很难存在使所有目标都实现的解。所以需要利用合适的理论基础对试卷的各项指标以及约束条件建立相应的数学模型。

### 3.1 多目标优化及组卷数学模型

#### 3.1.1 多目标优化简介

对于多目标优化问题，我们需要先了解多目标优化问题的数学描述。通常可以用与多个约束条件有关的等式以及不等式约束来表示多目标优化问题，其数学表示如下：

$$\begin{aligned}
 & \min \quad f_1(x_1, x_2, \dots, x_n) \\
 & \quad \dots \quad \dots \quad \dots \quad \dots \\
 & \min \quad f_r(x_1, x_2, \dots, x_n) \\
 & \max \quad f_{r+1}(x_1, x_2, \dots, x_n) \\
 & \quad \dots \quad \dots \quad \dots \quad \dots \\
 & \max \quad f_m(x_1, x_2, \dots, x_n) \\
 & \text{s.t.} \quad g_i(x) \geq 0, i = 1, 2, \dots, p \\
 & \quad \quad h_j(x) \geq 0, j = 1, 2, \dots, q
 \end{aligned} \tag{3-1}$$

其中函数  $f_i(x), \{i = 1, 2, 3, \dots, m\}$  为此多目标问题的目标函数<sup>[32]</sup>。 $g_i(x)$  以及  $h_j(x)$  为相应的约束条件。 $X = \{x | x \in R^n, g_i(x) \geq 0, h_j(x) \geq 0, i = 1, 2, 3, \dots, p, j = 1, 2, 3, \dots, q\}$  为此多目标问题的可行域。在上式中共有  $m(m \geq 2)$  个多目标函数， $r$  个极小化目标函数， $m - r$  个极大化目标函数，以及  $p + q$  个约束函数。

如果上式中所有的目标函数都为极小化目标函数，所有约束函数都为不等式形式，即为标准形式的多目标优化模型：

$$\begin{aligned}
 \min \quad & F(X) = [f_1(x), f_2(x), f_3(x), \dots, f_m(x)]^T \\
 \text{s.t.} \quad & g_i(x) \leq 0, i = 1, 2, \dots, p
 \end{aligned} \tag{3-2}$$

评价系统的性能指标主要由目标函数来判定,目标函数可以看作其数学表达式。用含有等式或者不等式的约束函数来表示设计变量时所需要满足的限制条件。满足所有约束条件(约束函数)的一组设计变量即为一个可行解,所有的可行解构成了整个多目标优化问题的可行域。目标函数主要作用是用来评价各指标的性能情况,在优化问题中要尽量使其达到最优。所有的目标函数  $f_1(x), f_2(x), f_3(x), \dots, f_m(x)$  构成多目标优化函数向量  $F(X)$ 。

### 3.1.2 组卷数学模型和约束条件

自动组卷系统需要解决的是一个满足多目标约束条件的复杂问题。其需要满足的约束条件大体上可以有:

- 1) 试卷的质量约束条件,主要有试卷的难度、覆盖度、相似度、区分度等。
- 2) 试卷的格式布局约束,主要有试卷的分值、题型、总体题数以及各个题型中的题数。
- 3) 其他约束条件,如某一章节最高分值约束,同一知识点考核次数限制等。

试卷中的约束条件相互之间并非毫无关系的,而是在试卷中相互影响、相互联系的。

本系统中一张试卷(S),其包含m个试题且每个试题(B)又含有n个属性(如分值,难易度,知识点,曝光度等等),那么试题可以表示为  $B = (b_1, b_2, b_3, \dots, b_n)$ , 试卷  $S = (B_1, B_2, B_3, \dots, B_m)$ 。这样试卷的数学模型就可以用下列矩阵来表示:

$$S = \begin{bmatrix} B_{11} & \cdots & B_{1n} \\ \vdots & \ddots & \vdots \\ B_{m1} & \cdots & B_{mn} \end{bmatrix} \quad (3-3)$$

其中n为试题的属性个数,m为纸卷中试题的总数。S中每一行代表一道试题的全部属性;每一列代表拥有该属性的全部试题<sup>[18]</sup>。具体的在本文中,试题属性有题型、分值、难度、章节、曝光度、区分度。具体的约束条件如下:

在上一章2.4节,介绍到本文有两种方式的组卷流程,下面我们将分别介绍这两种流程下所需要考虑的约束条件。

对于第一种组卷流程,由于是从整体上来完成组卷,对该流程有影响的因素为试卷总的分值、难易度、覆盖度、区分度、相似度、题数、题型数、各题型分值,但是总分值,题数,题型数,题型分值等是必须要满足的约束,不满足即组卷失败。所以真正对试卷选题起主要影响的是试卷的难易度、区分度,相似度约束。下面我们将详细介绍这三个约束。

#### 1) 难易度约束

$$H = \sum_{i=1}^m c_i B_i / F \quad (3-4)$$

此式表示试卷的实际难易度值。 $c_i$ 表示试题 $i$ 的难易度， $B_i$ 表示试题 $i$ 的分值。 $m$ 为试卷的试题数， $F$ 为试卷的总分值。试卷的实际难度值越接近目标难度值表示该指标完成的越好。难度指标通常很难准确的满足，所以在本文中该值尽量的往目标值趋近取值。其通常要将试卷的实际难易度与目标难易度作比较，要求差值最小，即 $\min |H_i - H_0|$ ， $H_i$ 为试卷的实际难易度， $H_0$ 为试卷的目标难易度。

## 2) 区分度约束

$$L = \sum_{i=1}^m c_i B_i / F \quad (3-5)$$

此式表示试卷的实际难易度值。 $c_i$ 表示试题 $i$ 的难易度， $B_i$ 表示试题 $i$ 的分值。 $m$ 为试卷的试题数， $F$ 为试卷的总分值。此约束条件通常很难准确满足，所以本文中该值的处理策略与难易度一样采用趋近取值。通常将试卷的实际区分度与目标区分度作比较，要求差值最小，即 $\min |L_i - L_0|$ ， $L_i$ 为试卷的实际区分度， $L_0$ 为试卷的目标区分度。

## 3) 相似度约束

由于重复的主要是与近三年用过的同类型同学科的试卷作对比。所以其表示如下：

$$Q_t = \sum_{i=1}^m B_i / Ft = (1, 2, 3, 4, 5) \quad (3-6)$$

此式表示试卷的实际难易度值。 $B_i$ 表示试题 $i$ 的分值。 $m$ 为本张试卷与过去试卷中相同试题的题数， $F$ 为试卷的总分值。 $t$ 表示用来做对比试卷的年份，本文中取近三年的考试试卷来作对比。上式表示与过去试卷中相同试题的总分值占试卷总分值的比例，用 $Q_t$ 表示其值。对于相似度我们通常其值越低越好。每次试卷的相似度值要求由出卷教师给出。通常将试卷的实际相似度与目标相似度作比较，要求差值最小，即 $\min |Q_i - Q_0|$ ， $Q_i$ 为试卷的实际相似度， $Q_0$ 为试卷的目标相似度。

对于第二种组卷流程，由于是通过对各个题型的优化，后将题型组合到一起完成组卷。各题型模块的优化指标主要有题型分值、题数、难易度、覆盖章节、区分度、相似度等。其中题型分值、题数、覆盖度等都是试卷整体的布局要求，属于必须要完成的指标，否则组卷失败。题型的难易度、区分度、相似度等位对题型选题产生主要影响的约束指标。其难易度约束、区分度约束、相似度约束与第一种组卷流程中相应的约束条件计算过程相同。

### 3.1.3 试卷的目标函数

对于本文而言,试卷的主要约束条件为试卷或题型的难易度、区分度、相似度。为了提高试卷的质量,需要在满足其它条件(试卷总分、题数、覆盖度等)的前提下,将上述三个约束条件的取值尽量向目标值趋近。由于多目标优化问题可知,对这三个指标取极小化目标函数。

由于本文考虑的是两种组卷流程,下面的介绍第一种组卷流程的目标函数。

- 1) 令  $h$  为试卷的实际难易度  $H_1$  与目标要求的难易度  $H_2$  偏差绝对值。 $S$  为试卷的总分值。

$$\text{即 } h = |H_1 - H_2| S$$

- 2) 同理令  $k$  为试卷的实际区分度  $K_1$  与目标要求的区分度  $K_2$  偏差绝对值。 $S$  为试卷的总分值。

$$\text{即 } k = |K_1 - K_2| S$$

- 3) 同理令  $q$  为试卷的实际相似度  $Q_1$  与目标要求的相似度  $Q_2$  偏差绝对值。 $S$  为试卷的总分值。

$$\text{即 } q = |Q_1 - Q_2| S;$$

本文取最近三次同级别科目考试试卷,且时间越近对本次试卷的相似度影响越大,权重按 5:3:2 分配。

那么本文中组卷的极小化目标函数

$$\min f = r_1 h + r_2 k + r_3 (0.5q_1 + 0.3q_2 + 0.2q_3) \quad (3-7)$$

其中每个偏差的权重为  $r_i$ , 且  $\sum r_i = 1$ 。

第二种组卷流程的目标函数介绍如下:

- 1) 设  $Q_i$  是题型  $i$  的实际难易度,  $Q_j$  为试卷要求难易度, 试卷中共有  $j$  类题型,  $S_i$  为题型分值。那么试卷题型难度的平均偏差分值为

$$\bar{Q} = \sum_{i=1}^j S_i |Q_i - Q_j| / j;$$

- 2) 令  $T_i$  是题型  $i$  的实际区分度,  $T_j$  为目标要求区分度, 试卷中共有  $j$  类题型,  $S_i$  为题型分值。那么题型区分度分值的平均偏差分值为

$$\bar{T} = \sum_{i=1}^j S_i |T_i - T_j| / j;$$

- 3) 令  $Y_i$  是题型  $i$  的实际相似度,  $Y_j$  是目标相似度, 共有  $j$  个题型,  $S_i$  为题型分值。(本文取最近三次同级别科目考试试卷), 那么题型的相似度偏差平均分值为  $\bar{Y} = \sum_{i=1}^j S_i |Y_i - Y_j| / j$ 。

综上采取第二种组卷流程的目标函数表达式如下：

$$\min f = r_1 \bar{Q} + r_2 \bar{T} + r_3 \bar{Y} \quad (3-8)$$

其中每个偏差的权重为 $r_i$ ，且 $\sum r_i = 1$ 。

## 3.2 遗传算法

### 3.2.1 传统遗传算法简介

遗传算法(Genetic Algorithm)<sup>[16][20]</sup>是一种基于种群的以遗传理论为基础的寻找问题最优解的随机搜索方法。该算法的基本理论是John Holland<sup>[43]</sup>于二十世纪70年代提出。该算法通过模拟达尔文生物自然进化过程(即优胜劣汰，适者生存)让种群的遗传基因通过多代的选择、交叉、变异等过程，最终保留下具有优良的基因的种群，再对种群进行全局并行搜索找寻种群中遗传基因最为优秀的个体，求得满足各约束条件的最优解<sup>[38]</sup>。

遗传算法从提出到发展至今已经非常成熟，关于该算法的应用领域非常多。例如可作为机器学习算法；也可与神经网络、模糊理论等智能计算方法相结合；除此之外经常与进化计算理论相结合。虽然算法本身已经成熟，但在遗传算法的应用以及延伸研究等方向上依旧非常热门和活跃<sup>[21]</sup>。

下面介绍下遗传算法的大致流程。

- 1) 初始化，构建初始种群 $P(0)$ ，设置好进化代数计数器 $i=0$  ( $i \leq t$ 其中 $t$ 为最大进化代数)。
- 2) 个体评价，对种群 $P(t)$ 中每个个体的适应度进行分析和统计。
- 3) 选择操作，通过选择算子对种群适应度进行排序，通过选择将符合条件优良的个体或是将配对交叉所产生的子一代转入到下一代。此操作是以种群中个体适应度评估为基础。
- 4) 交叉运算，通过交叉算子对种群进行交叉控制，交叉的概率由交叉算子控制，在遗传算法中交叉算子起核心作用，新种群将主要通过交叉操作产生。
- 5) 变异运算，通过变异算子对种群进行变异操作。即对交叉过后的子代个体基因进行小概率的改动。种群通过上述选择、交叉、变异过后形成下一代种群 $P(i+1)$ 。
- 6) 终止条件判断，即当进化代数 $i < t$ 时，回到第2步；当 $i = t$ 时，即达到最大进化次数时，此时输出种群中最大适应度值的个体作为最优解，遗传算法计算结束。

遗传算法具体流程图如下：

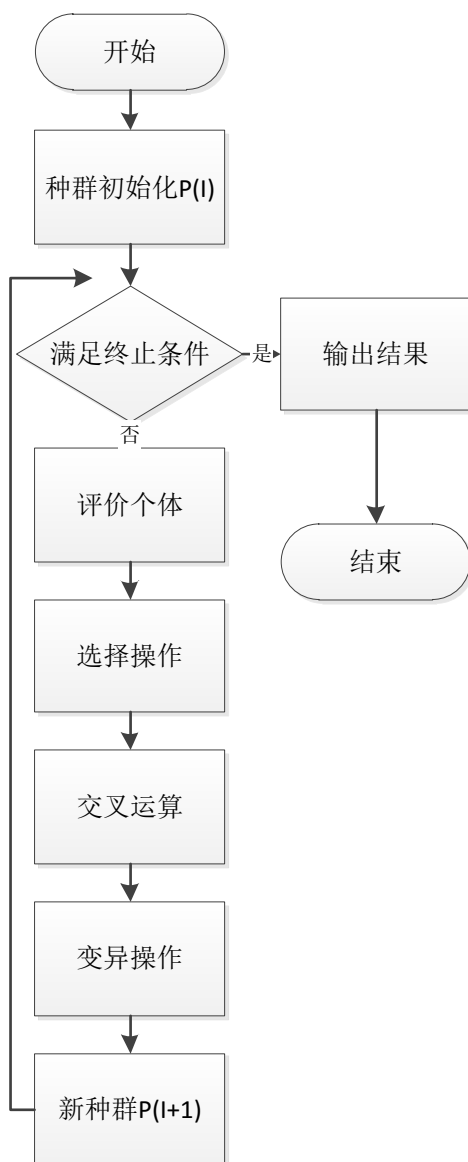


图 3.1 遗传算法流程图

### 3.2.2 种群的初始化以及染色体编码

遗传算法中种群的初始化主要是通过随机的方式产生<sup>[34]</sup>。种群中个体都是由染色体构成，染色体即为问题的可行解，构成可行解的元素即为染色体上的基因。在遗传算法中首先需要将染色体编码化。常用的编码方法主要有二进制编码<sup>[33]</sup>。

在二进制编码中，染色体的长度由固定字符长度的0或1来表示。其中通常1表示选中的，0表示位被选中。例在实际操作中表示一个具有4个基因的染色体可



以用 $G=(0101)$ 来表示。选用二进制编码方法主要因其可操作性较强,使用简便,同时具有良好的兼容性<sup>[36]</sup>。

### 3.2.3 种群大小和适应度函数

种群的大小主要对算法的运行效率和结果产生影响。过大则运行率降低,过小则结果的准确性将降低。所以种群选取时需要结合实际需求来确定。

自然选择的主要功能是将优良的遗传基因传递到下一代,同时淘汰掉不符合要求的差基因。在遗传算法中,适应度函数主要对个体的优劣进行评估,对染色体上优劣基因的筛选起重要作用。适应度函数的信息可以指导问题的搜索方向,其直接影响算法的运行效率和结果的准确性。所以适应度函数对该算法来说至关重要。

### 3.2.4 遗传算子和终止条件

遗传算法中遗传算子有选择算子、交叉算子、变异算子,通过选择、交叉、变异操作促使种群进化,产生适应度高的优良新个体。

#### ● 选择算子

选择的目的是为了选择合适的染色体来产生下一代,通过评价染色体的优劣来对其在下一代中是被复制或者淘汰进行评判。选择算子的常用方法有“轮盘赌策略”(Roulette Wheel Selection),它是遗传算法一直以来最为常用的选择算法。其通常表现为一种“比例选择”,具体操作中为个体被选中的概率和它的适应度值成正比。例种群共有个体总数为 $N$ ,其中个体 $X_i$ 的适应度为 $fX_i$ ,那么其被选中的概率为 $fX_i / (fX_1 + fX_2 + \dots + fX_n)$ 。

除此之外选择算法还有锦标赛选择方法、稳态选择方法等。

#### ● 交叉算子

交叉过程就是对群体中染色体进行两两配对,以此为基础产生下一代个体,即染色体的重组。常用的交叉方法有二进制交叉法。例两条染色体的部分等位基因进行相应的交叉,产生两条新的子代染色体。染色交叉概率记为 $P_c$ 。

例交叉前父辈个体 $F1, F2$ 染色体片段如下,

$F1: 0000\ 1010\ 0110$

$F2: 1110\ 0100\ 0001$

通过交叉产生子一代个体 $P1, P2$ 染色体片段如下。

子代 $P1: 1100\ 1110\ 0001$

子代P2: 0010 0000 0110

交叉操作对于保持种群的多样性以及保持种群中优秀个体数量有重要意义。

### ● 变异算子

在生物的自然进化过程中变异起到了非常重要的作用,在产生新染色体过程中基因会以一定机率出错,但是此概率非常小。在遗传算法中,基因发生变异的概率也是非常小的记为 $P_v$ 。例在二进制编码的染色体中基因发生变异如下:

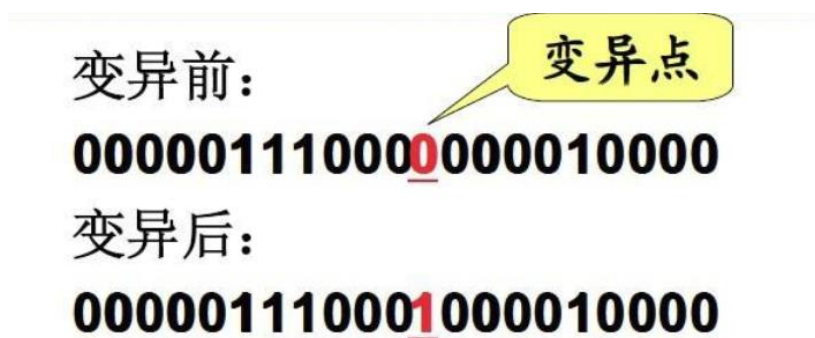


图3.2染色体变异图

其中在第12个基因位置发生了变异。常用的变异方法有基本位变异法和实值变异法等。

### ● 终止条件

在理论上进化的次数越多则产生的后代越优秀。但是在实际应用中不可能让算法无期限的运行下去,那么就需要结果在精确度和进化次数上建立合适的关系,其直接关系着算法的运行效率。

在什么时候算法终止。常用的有两种方法,第一种情况的终止条件是,设定最大进化次数 $N$ ,当运行次数达到了设定的最大终止代数 $N$ 时,无论有没有得到最优解算法都将终止;第二种是限定允许范围,事先给运行结果设定一个范围,当算法运行到一定代数其结果进入事先设定的范围,则终止算法。第一种算法可能在输入数据不同的时候其运行的代数差别非常大;第二种算法可能导致算法的执行效率不可控。具体选用那种要根据实际情况而定。

## 3.3. 遗传算法的特点

以生物进化为理论基础的遗传算法,具有很好的全局搜索能力,能够快速地从全体解中搜索出符合全部约束条件的最优解。在搜索的过程中可以对多个个体同时进行比较,存有潜在并行性。利用此特点能够进行相应的分布式计算,大大加快了求解速度。但是遗传算法也具有很明显的不足之处,当问题计算量大时,如果单纯的只使用遗传算法,则问题在局部搜索能力上效果较差,耗时较多以及

后期搜索的效率低。在实际应用中表现为，算法的早熟收敛，以及问题的搜索容易收敛到局部最优解。这也是遗传算法应用于本课题重点需要改进的地方。

### 3.4 本章小结

本章主要进行组卷系统的数学建模以及介绍本文的组卷策略。组卷问题的本质是一个多目标优化问题。试卷的约束条件（分值，难易度，题型，题量，覆盖度等）是多方面的，需要对其建立多约束条件下的多目标函数模型。介绍了两种组卷流程分别对应的目标函数。详细介绍了本文选用组卷算法的传统形式，重点介绍遗传算法中的一些重要参数（适应度值、遗传算子等），分析该算法的优缺点。

## 第 4 章系统实现

### 4.1 组卷的基本要求和原则

#### 4.1.1 组卷要求

此系统主要是用来协助教师来完成大学生的期末考试试卷的编写。需要充分考虑学校的考试要求以及本课程的考试大纲要求。所以此系统必须要满足以下要求：

- 1) 试题库内试题覆盖要足够丰富全面，高质量的试题库是组卷系统的重要基础，是完成高质量试卷的前提。其要求试题的难易度，覆盖度，以及题型的种类等都能满足课程考试大纲要求，这样的系统根据不同的要求时完成的试卷才更具有代表性、科学性。
- 2) 能够按照用户要求的各项指标完成试题抽取并组成试卷。教师按照教学大纲要求来指定考试范围内容、难度等信息，系统能够按照这些要求来完成相应的组卷。
- 3) 能够让教师手动组卷，即教师按照自己对教学大纲以及考试大纲的理解结合教学情况和学生整体学习情况来命题组卷。通过手动选择相应的题型和试题，同时系统也可以生成对此套试卷的难易度，覆盖度，重复度等相关的评价公交是作为参考。此功能可以保证在一些要求特殊情况下的组卷，使系统在实际应用中适应性更强。
- 4) 系统生成的试卷具有多样性。用户在分析好教学大纲以及考试大纲要求后在组卷系统中选择好试卷难易度、覆盖度、相似度等属性以后能够快速生成满足要求的试卷，并且每次运行过后所产生试卷的试题具有差异性。此组卷要求可以用完成多套备选试卷供考试使用，有利于防止作弊现象出现并有效保证考试的公平。
- 5) 能够调节试卷难度。试卷的难易度是评判试卷的一个重要指标，本系统能够将试题的难易度按照一定的方法策略换算成各大题型的难易度，或者直接将各试题的难易度转换成试卷的难易度。
- 6) 能够调节试卷的覆盖度。试题库的所包含的试题覆盖范围要全面，各属性的设置合理科学。即试题具有科学性，全面性，代表性。此组卷系统能够按照考试大纲的要求科学调整各章节试题的比重，完成符合一定要求的考试。

7) 试卷的相似度要尽量低。为了提高试卷的保密性,一方面要求试卷中与过去试卷中重复试题的总分值尽可能低;另一方面要求重复的试题距离本试卷的使用时间尽可能的远。

除此之外,对于系统的操作管理方面,有以下要求:

#### 1) 试题的管理

试题的管理是此系统的基础和前提,所以必须要能够安全、高效的对试题进行正常的增删改查功能,同时还要能够让试题在系统内部实现共享。要方便管理者的正常使用和维护。

#### 2) 组卷管理

常见的组卷由随机选题等方法来完成,在约束条件较多的情况下组出的试卷存在较多不足之处。本系统为基于改进型遗传算法的组卷系统,可以满足试卷区分度条件、难易度条件、覆盖度条件、相似度条件等多个约束条件来完成高质量的试卷。过程中主要考虑如何数学建模将试题和试卷结合起来,试卷的各评价指标量化,以及遗传算法的具体实现过程,以便整个组卷过程的顺利的执行。

#### 3) 用户管理

本系统将主要提供教师使用,同时有系统管理员管理功能,此功能操作简单、安全便捷,用户可以更改自己的密码等基本信息,防止账户被非法使用。管理员可以对用户进行管理,对用户进行增删等功能。

### 4.1.2 组卷原则

组卷只有在满足一定的原则的前提下,抽取满足要求的试题,才能够完成满足质量要求的试卷。这样的试卷在实际考试当中才能更加有效的反应学生对于课程知识的掌握情况。这样的考试结果更具科学性,更有利于教师了解自身教学过程中的不足之处。教师和学生都能够通过这份试卷的结果检测自己,对于后续的教学和学习具有重要积极的指导意义。组卷的基本准则主要有一下几方面:

1) 科学性原则: 试卷要能够科学的反应考试大纲要求,题型选择以及试题难度、数量等都要科学的选择排版。能够准确的反应实际情况。

2) 目的性原则: 好的试卷要在适合它的条件下才能发挥其作用。试卷是用来作为课程章节检测还是期中考试检测,还是学业水平检测亦或是期末考试检测,首先要明确。考试的目的不同,那么试卷必然要做出相应的变化调整。

3) 层次性原则: 一份试卷不可能全部都是简单或者困难的题目。试卷不仅要让学生检测出自己学习的不足之处,也要让教师从中得到教学的不足支持。在试卷编制过程中要充分考虑学生对知识认识掌握的差异性,以及课程中知识点的

难易度,让试题在试卷中呈现出一定的梯度。一方面是试题本身要有梯度,特别是一些主观题小问题的难度要递进排布;另一方面,整张试卷在试题排版的时候要遵循由浅入深,先易后难的排布。

4) 试卷所考核的知识要能够对学生的后期学习方法有一定的指导意义。考试对学生来说作用非常明显,通过成绩的形式让学生更加深刻的认识到自己与别人的差距,了解自己对知识的学习掌握程度,认识到不足之处促进其做出改变。

5) 试卷中语言要简洁准确,试卷中不要出现与考试无关的话语。这些有助于提高试卷科学性、专业性、权威性。

## 4.2 开发环境与相关技术

### 4.2.1 开发环境

本系统用C#高级语言进行开发。C#语言是从C和C++语言演化而来,面向对象的程序开发设计语言<sup>[31]</sup>。该语言是微软公司专为.NET平台而开发的。C#的发展比较新且继承了C语言的相关语法风格以及C++面向对象的特性。同时又有所区别,C#不再支持C语言的指针类型,不支持多重继承,使得C#语言更加健壮、安全。在C++中能完成的任务几乎都能用C#来完成。不同之处在于,C#使用的是.NET框架下的类库。

Microsoft Visual Studio(简称VS)是微软公司的开发工具包系列产品<sup>[30]</sup>。VS是一个基本完整的开发工具集,它包括整个软件生命周期中所需的大部分工具,如UML工具、代码管控工具、集成开发环境(IDE)等等。用该工具所编写的代码适用于微软支持的所有平台,具有跨语言优势<sup>[45]</sup>。

### 4.2.2 系统架构

C/S(Client/Server)体系结构由美国Borland公司开发。该结构使客户机及服务器的硬件优势得到充分利用,将任务合理的分配到客户机和服务器端来实现,能有效减少系统的通讯开销。

C/S模式的第一层结构(前端)在客户机的系统上结合了业务逻辑和形式逻辑;第二层结构(后端)主要是将数据库结合在计算机网络上。C/S模式将应用于网络的用户交互界面处理程序与对数据库的访问和处理分离开来,前端与后端的之间通过消息传递机制来进行对话。首先由前端发出请求给服务器端,服务器端接收到请求后做出相应的操作,将处理后的消息再传递给客户端。C/S模式的结构如下图所示:

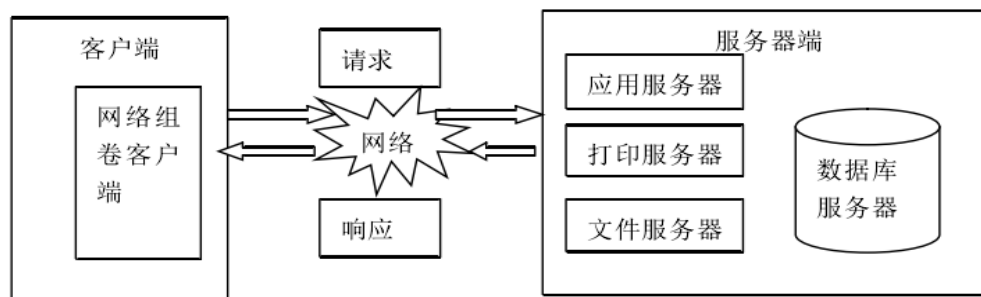


图 4.1C/S 模式结构图

C/S结构一般建立在局域网中，再通过特定的服务器提供与因特网的数据交换和连接功能，其面向固定的用户群体，系统有较强的控制能力，机密性较强的软件通常采用此结构。

对于本课题所设计的系统来说，主要是用来做验证性设计的。所以采用相对简单的C/S的模式开发即可。

#### 4.2.3 ADO.NET 技术

ADO.NET(ActiveX Data Objects)是一个COM组件库。它提供了对关系数据、XML和应用程序数据的访问，是.NET Framework中不可缺少的一部分。他是在.NET编程环境中使用的数据库访问接口<sup>[20]</sup>。其被集成到.NET Framework中，可以用于任何.NET语言，尤其是C#语言。ADO.NET数据库为用户提供了强大的数据操作管理功能，可以实现对数据信息的增加、删除、修改、查询等功能<sup>[41]</sup>。

#### 4.2.4 SQL SERVER 数据库技术

SQL Server 是一款功能强大的关系数据库管理系统。其是由 Microsoft 公司开发的，可以对数据库进行创建、设计、开发和管理<sup>[29]</sup>。SQL Server 具有图形化的用户处理界面，编程接口非常丰富，同时具有稳定的安全性、伸缩性。除此之外还具有操作简单、直观等优点。

### 4.3 系统设计

一个好的组卷系统不仅需要实现对题型、试题、课程等基本信息的增删改查功能，还要对试卷整体的各项要求能够具体的细化到单独的试题中去。设计一个系统需要全面、合理的相关分析设计。下面将对本文所设计的组卷系统展开具体分析设计。

### 4.3.1 可行性分析

开发系统之前对此系统进行可行性分析是非常有必要的。虽然试题库等系统成熟产品较多，但是其使用范围和实用性等因素的约束使得其普及型不是太高。因而在对于特定的学校，针对其教学大纲，教学要求，考试要求，试卷要求等来设计完成一套符合其要求的系统是非常必要的<sup>[28]</sup>。本系统采用C#语言开发，基于SQL Server数据技术结合VS2013开发平台等一系成熟的开发技术来完成。其安全高效能够贯穿开发的整个过程，同时在后期维护上依旧能够保障其安全可靠，所以在开发技术可行性上，完全能够满足要求。

本系统的开发得到学院的支持，学校的校园网络资源、教师队伍资源都为本系统的开发提供了强大的软硬件环境和人力资源环境保障。

### 4.3.2 总体架构设计

通过对系统的分析可将本考试系统分为三大部分：用户管理，试题管理，组卷管理。

功能结构图如下所示：

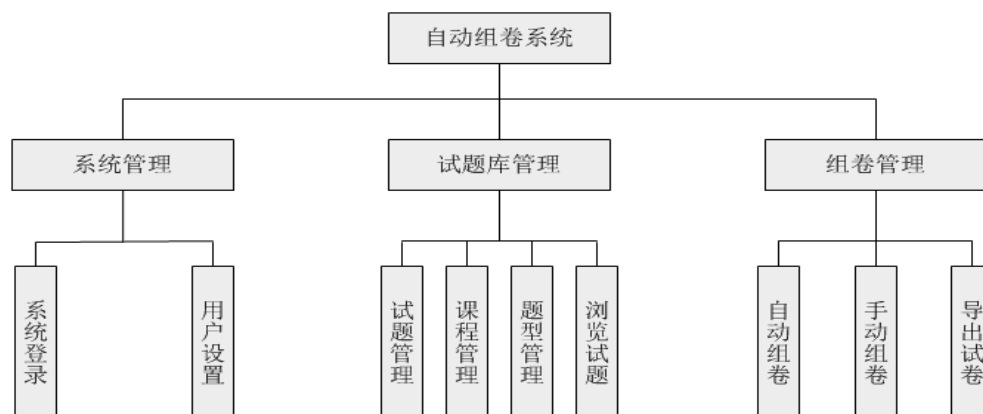


图 4.2 功能结构图

系统管理包括：

系统登录：包括注册，登录，找回密码功能。

用户设置：包括个人信息的管理，查看，修改，不同权限的用户功能有所区别。

试题库管理包括：



- 1) 试题管理：主要包括增加、修改、删除、查询试题的功能。
- 2) 课程管理：主要包括课程的增加、修改、删除、查询的功能。
- 3) 题型管理：主要管理题型的增加、修改、删除、查询。
- 4) 浏览试题：主要是浏览试题的功能。

组卷管理包括：

- 1) 手动组卷：通过人工手动选取试题，来确定满足要求的试题组成试卷。
- 2) 自动组卷：通过设定试卷难度、分值、题型、覆盖度、与过去试卷的相似度、各题型数目分值等条件，由组卷策略来完成组卷。
- 3) 导出试卷：将已经完成的试卷，以满足要求的排版输出Word文档。

#### 4.3.3 系统数据库设计

数据库设计原则：

在设计和实现该组卷系统的过程中，需要为其设计一个安全的、完善的数据库，来满足其正常的管理功能。数据库的设计需要保证数据的完整性，独立性，可靠性以及访问的高效性。同时要兼顾数据库后期发展，移植等方面的需要。本系统数据库系统应该具备以下功能。

- 1) 定义数据库的结构和逻辑结构和存储结构的功能。
- 2) 可以完成对数据库中数据的插入、修改、删除、查询以及数据库的备份与恢复功能。
- 3) 可以对数据库的安全性、完整性、以及在多用户情况下的状态的并发控制等方面进行控制。
- 4) 可以提供数据库的通信功能。

接下来将要介绍本文中数据库表的设计。

数据库表在组卷系统中数据库是一个非常关键的部分，是整个系统实现信息资源共享的前提基础。为了能够高效，安全的使用数据库来组卷，依据数据库相关概念设计要求，本文的实体E-R图如下。

- 1) 系统管理试题信息主要记录用户名称、密码、编号，其E-R如下：

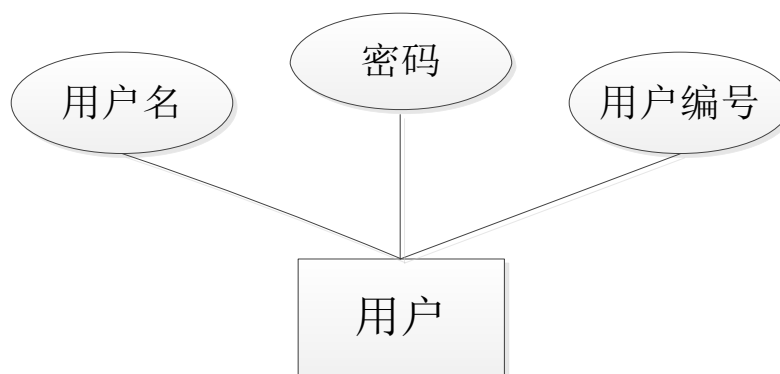


图4.3用户E-R图

- 2) 试题实体主要记录试题的编号，难易度，区分度，分值等信息，其E-R如下：

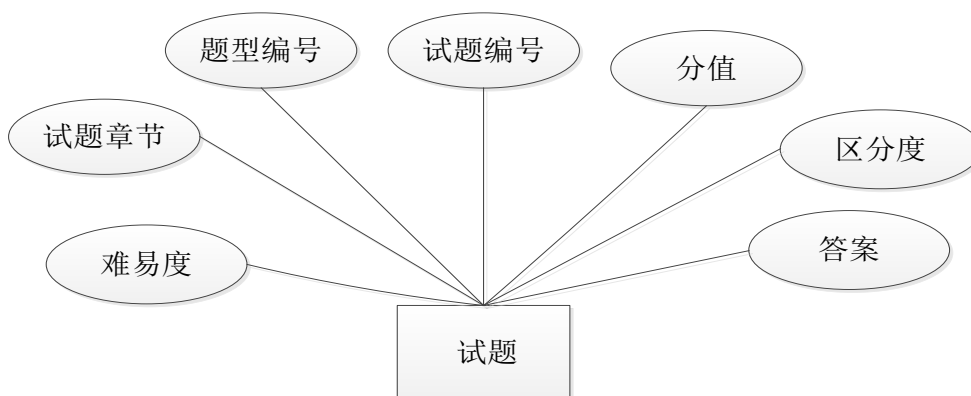


图4.4试题E-R图

- 3) 专业课程实体主要记录专业名称、课程号、课程名称其E-R如下：

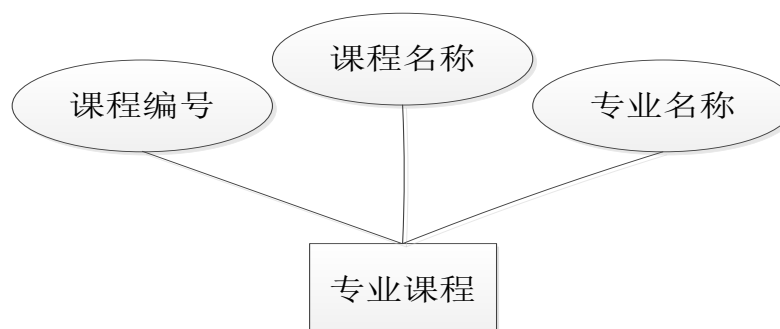


图4.5专业课程E-R图

- 4) 试卷实体信息主要记录试卷的编号、难度等信息其E-R如下：

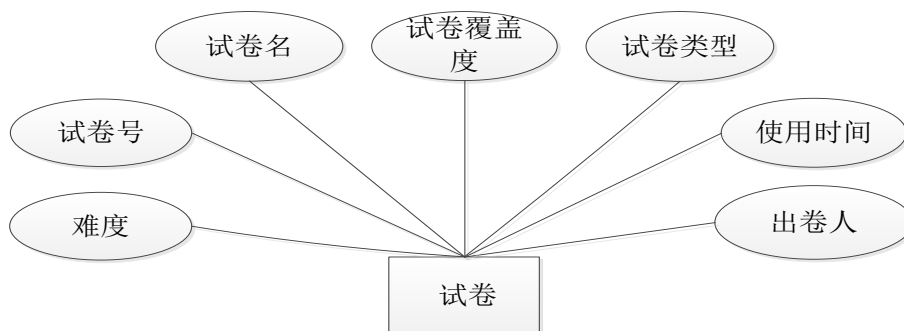


图4.6试卷E-R图

5) 题型实体主要记录题型的编号以及题型名称信息，其E-R如下：

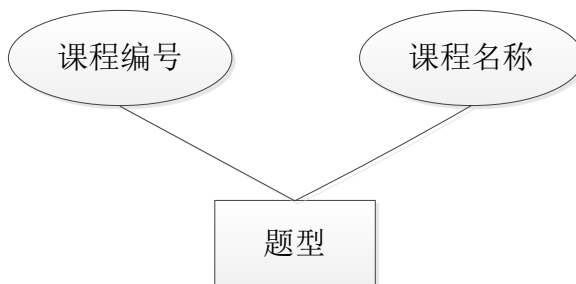


图4.7题型E-R图

6) 试卷的存储实体信息主要记录试卷的内容、试卷编号、课程编号，其E-R如下：

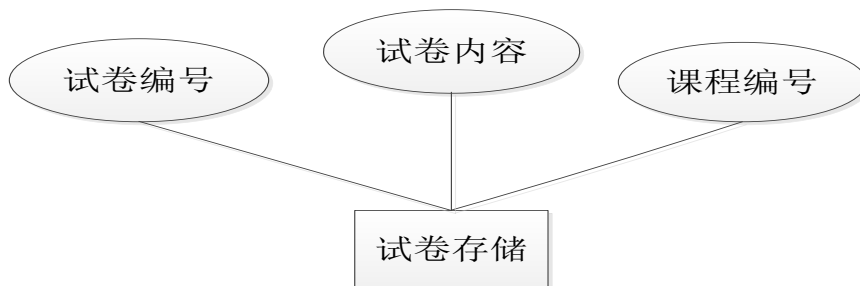


图4.8试卷存储E-R图

系统的数据表设计参照上述的数据库的E-R图设计。其具体设计如下。

#### (1) 用户信息表(User information table)

此表主要用于用户信息的存储以及系统对用户的管理，主要信息有用户编号、用户名、用户密码。

表4-1用户信息表

字段名称	类型	是否主键	备注
User_ID	Int	是	用户编号
User_Name	Nvarchar(20)	否	用户名
Password	Nvarchar(50)	否	账号密码

## (2) 试题表(Question)

试题表主要用于对试题信息的进行存储，主要信息有问题编号（每个试题都有唯一的试题号）、题型编号（用于确定所属题型）、问题所在章节（对试卷的覆盖度有重要作用）、问题难易度（是试题的重要属性）、试题内容、试题答案、试题分值。此表的试题质量直接关系整张试卷的质量。

表4-2试题表

字段名称	类型	是否主键	备注
QuestionNo	Varchar(50)	是	问题编号，主键
TypeID	Int	否	题型编号
Chapter	Nvarchar(50)	否	问题所在章节
Difficulty	Varchar(10)	否	问题难易度
Question	Nvarchar(MAX)	否	问题内容
Answer	Nvarchar(MAX)	否	问题答案
Score	Int	否	试题分值
Discrimination	Varchar(10)	否	试题区分度

## (3) 专业课程表(Course)

此表主要用于存储管理专业课程，主要信息有课程编号（每个课程在系统中都有唯一的课程编号）、课程名。

表4-3专业课程表

字段名称	类型	是否主键	备注
CourseID	Int	是	课程编号
CourseName	Nvarchar(50)	否	课程名称
Profession	Nvarchar(50)	否	专业名称

## (4) 题型表(QuestionType)

此表主要用于存储管理题型信息，题型表主要有题型号和题型名两字段。

表4-4题型表

字段名称	类型	是否主键	备注
TypeID	Int	是	题型编号
TypeName	Nvarchar(50)	否	题型名

#### (5) 试卷表(Paper)

试卷表主要是用于存储管理试卷信息，其包含试卷的各项指标，主要有试卷号（生成的每张试卷都有唯一的编号）、试卷名、试卷覆的章节信息（此信息可直观的查看试卷覆盖度情况）、试卷难度（试卷非常重要的信息）、试卷类型、试卷使用时间、出卷人。

表4-5试卷表

字段名称	类型	是否主键	备注
PaperNo	Int	是	试卷号
PaperName	Nvarchar(50)	否	试卷名
PaperChapter	Nvarchar(50)	否	试卷覆盖的章节
Difficulty	Varchar(10)	否	试卷难度
PaperType	Nvarchar(MAX)	否	试卷类型
UseTime	Nvarchar(50)	否	试卷使用时间
PaperCreator	Nvarchar(10)	否	出卷人

#### (6) 试卷存储表(PaperStorage)

存储试卷，主要含有试卷编号、试卷内容和课程编号三个字段。

表4-6试卷存储表

字段名称	类型	是否主键	备注
PaperID	Int	是	试卷编号
PaperContent	Nvarchar(MAX)	否	试卷内容
CourseID	Int	否	课程编号

## 4.4 遗传算法在系统中的实现

本文中通过将试题、试卷以及约束条件设计成遗传算法中基因、染色体等，模拟生物进化过程。在遗传算法的操作过程中通过种群初始化，染色体编码，交叉操作，变异操作等过程完成种群的多样化以及优化过程，最终实现优胜劣汰。本文在应用该算法的时候可能会涉及一些大家不太熟悉的生物学专业名词。下面是本文对这些专业名称的一些解释。

生物遗传学，在本文中直接解释为遗传算法本身。

基因：在试卷中表示为每道试题。在染色体中以试题的题号来表示。

等位基因：相同位置的基因不同。

染色体：一定数量的基因编码组合在一起形成一条染色体。染色体通过解码即可表示一套完整的试卷。

个体：表示优化问题的一个可行解，即一个满足一定条件的试卷。

种群：所有个体的集合，即所有试卷的集合。

选择：从当前种群中按一定要求选出适量个体去参与繁衍下一代的操作。

交叉：主要是产生新个体，通过染色体配对后，染色体间基因发生相应的交换，以此得到新的个体。

变异：对某个个体染色体上的基因以一定的概率用等位基因对其进行改变，产生新的个体。

适应度：评价个体优劣程度的标准。

下面将具体介绍算法在系统中的实现。

### 4.4.1 染色体的编码

遗传算法的应用中首先要解决染色体的编码。二进制编码作为常用的编码方式，用1表示选用试题，0表示不选用。该编码方式简单明了、易于之后的染色交叉及变异操作。但是具体在本系统中，由于试题库中试题量大，如果用二进制表示的话，那么编码串的长度非常大，相应的解码也会非常耗时，整体操作下来系统计算量大，耗时长，从而降低组卷效率<sup>[17]</sup>。所以本文将不选用此编码方式，使用另一种实数编码方式。

实数编码方式直接用试题的题号来表示染色体中选用的基因。出现的题号即是选用的试题。编码的长度就是所选试题题号的集合，这样编码可以大幅度减小编码串的长度。由此编码和解码耗时都会减少，组卷效率得到提高。本文中使用的

此编码方式时基因是按照题型的排布顺序来进行排列的，所以在连续的基因是由相同题型的试题来组成的。

例如完成一份试卷，要求选择题 14 题，填空题 10 题，计算题 4 题，设计题 2 题，其编码情况如下表所示。

表 4-7 试卷编码表

题型	试题编号													
选择题	4	13	6	22	26	34	54	46	64	55	78	63	17	39
填空题	101	109	126	135	145	127	156	142	150	153				
计算题	204	225	265	235										
设计题	285	296												

由上表，可以直观的看到试卷所选的试题在题库中的编号。实数形式的基因编码，简短明了。本文在染色体编码时将相同题型的试题连续的放在一起，这样编码有助于减少搜索空间和编码长度。

#### 4.4.2 初始种群的确定

在传统遗传算法中，种群是通过随机形式产生的。本文中为了提高系统的组卷效率，对于初始种群的设定一定的初始约束条件。本文中初始种群（即试卷）需要在满足总分值、总试题量、总题型以及各个题型中试题数等条件。这样可以有效减少后期染色体交叉，变异的迭代次数，有利于克服遗传算法的早熟收敛问题。

#### 4.4.3 适应度函数确定

适应度函数在遗传算法中，主要用来指导算法的搜索方向。通过适应度的值来区分种群中个体的优劣。本系统中，在种群初始化时已经将试卷的分值、题型、试题数量等约束条件考虑进去了。所以能够影响种群适应度的因素主要有难度约束、相似度约束以及区分度约束。这些也是试卷目标函数的主要约束条件。目标函数的值要求越低，说明试卷越满足要求。

适应度函数设计的好坏直接影响算法的迭代执行。当个体拥有合适的适应度值时即可认为个体达到了最优。通常情况下适应度函数直接和目标函数相关，适应度越高，表示个体越接近目标试卷。适应度的设定直接和目标函数相关。本文中遗传算法的适应度函数公式为 4-1

$$f = 1/(1 + \sum_{i=1}^3 r_i |e_i|) \quad (4-1)$$

式中 $|e_i|$  ( $0 \leq e_i \leq 1$ )为组卷目标函数中难易度、区分度、相似度的实际值与目标值之间的偏差绝对值。 $r_i$ 表示各项偏差值所占的权重。从上式可以看出,当试卷的约束条件误差越小时,目标函数值也就越小,试卷的适应度值也就越大,即个体越优秀,试卷完成的质量越高。

#### 4.4.4 精英保留策略

种群在进化的过程当中通过选择,交叉变异等操作会不断产生新的优良个体。但是也有可能破坏掉当前种群中的一些优良个体,很明显这是不利于种群进化的,会降低种群的整体适应度,不利于算法运行效率的提升和收敛。

所以为了避免优良个体在进化过程中被破坏,我们对当前种群中适应度最高的一些个体采取精英保留策略。即在当前种群中,适应度最高的一些个体不参与种群的选择、交叉、变异等操作过程。用这些个体来替换掉选择、交叉、变异过后种群中适应度值最低的一些个体。故能够保证产生的新的种群总体适应度值高于之前的种群。

#### 4.4.5 选择操作

选择的目的是为了增加种群中优秀个体的数量,使后代染色体具有较高的适应度值。本文结合组卷的实际需求情况,选用轮盘赌策略对种群进行选择操作<sup>[37]</sup>。其具体操作过程如下:

- 1) 计算出种群的总适应度值  $F$ 。计算公式如下:

$$F = \sum_{i=1}^m f_i \quad (4-2)$$

其中  $m$  为种群的个体数。

- 2) 计算个体被选中的概率, 公式如下:

$$p_i = f_i / F \quad (4-3)$$

- 3) 构造轮盘, 计算累积概率

$$K_i = \sum_{i=1}^m P_i \quad (4-4)$$

- 4) 通过轮盘选择个体, 随机生成一个 $[0,1]$ 之间的随机数  $q$ ,

当 $q \leq K_1$ 时, 则选择个体 $B_1$ 。当 $K_1 < q \leq K_2$ 时, 则选择 $U_2$ 个体。

重复以上步骤, 产生  $N$  个种群个体。

例如个体 1 被选中概率为 20%, 个体 2 被选中概率为 15%, 个体 3 被选中概率为 15%, 个体 4 被选中概率为 5%, 个体 5 被选中概率为 35%, 个体 6 被选中概率为 10%。则构成的转盘如下:



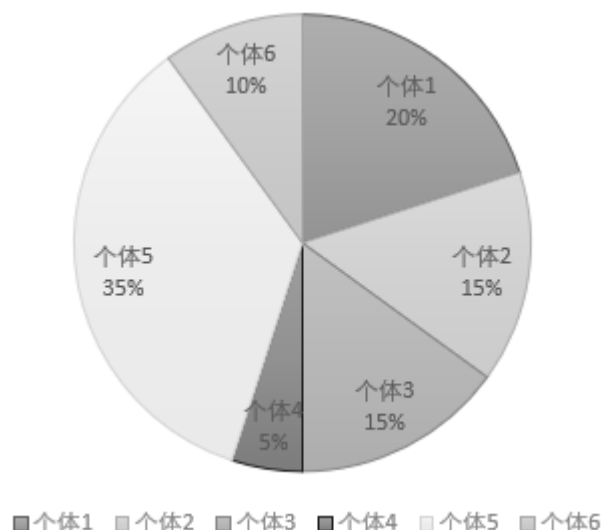


图 4.9 种群轮盘图

#### 4.4.6 交叉操作

交叉操作是种群产生新个体的主要方法，交叉的实质是染色体上基因的重组。其直接决定了算法的全局搜索能力。本文编码采用分段(按题型)实数编码方式，故交叉时采用题型内单点实数形式交叉形式。由于是分段的单点交叉，所以在整个染色体上表现为多点交叉。具体的交叉过程如下：

对种群内个体实行两两配对的方式组队。种群为  $M$  那么将会产生  $M/2$  对个体。设定好染色体内基因的交叉概率(本文中  $P_c$  表示)，同时指明交叉条件，这样就可以完成交叉，产生新的染色体。 $P_c$  的值越大新个体产生的速度也就越快，但同时会破坏适应度高的个体，使优势群体增加，不利于种群的多样性； $P_c$  过小会使搜索速度过慢，拖延种群进化速度，但会较好的保留适应度高的个体。因而在进化初期  $P_c$  选择的较大，则会加快新个体的产生速度，增强算法的搜索能力；在进化后期  $P_c$  选择的较小，则会避免破坏优良种群，加快算法的收敛速度。所以在本文中，染色体的交叉概率采用动态调整策略。具体交叉概率取值如下：

$$P_c = \begin{cases} L_1 & \dots \dots \dots f < f_{\text{avg}} \\ L_2 + (L_1 - L_2)(f_{\text{max}} - f)/(f_{\text{max}} - f_{\text{avg}}) & \dots \dots f \geq f_{\text{avg}} \end{cases} \quad (4-5)$$

其中  $f_{\text{max}}$  表示种群中最大适应度函数值， $f_{\text{avg}}$  表示该代种群平均适应度函数值， $f$  表示交叉操作中两个个体中大的适应度函数值。 $L_1$  表示两个个体间适应度函数大的一个， $L_1=0.9$ ； $L_2$  表示两个个体间适应度函数小的一个， $L_2=0.65$ 。

由于每一代的种群情况是不断变动的,所以用上式可动态描述表示每一代种群内个体间的交叉概率。

当  $f < f_{\text{avg}}$  时,即个体适应度较低时,这时需要进行较多的基因交叉操作来改良个体。所以染色体交叉概率为较大值  $L_1$ ;

当  $f \geq f_{\text{avg}}$  时,即个体适应度值较高时,这时需要更多的保留该个体基因,所以交叉概率较小。

当  $f_{\text{max}} - f$  接近 0 时,即个体适应度值较高,基因优良时,交叉概率较小,  $P_c$  值为  $L_2$ ; 当个体适应度值非常小,基因较差时,此时交叉概率就需要提高,  $P_c$  值为  $L_1$ 。通过此操作,有助于避免个体的早熟问题,通过动态调整交叉算子等形成稳定进化,加速种群的收敛。

#### 4.4.7 变异操作

变异是产生新个体的重要方式,主要是将染色体中某些位置的基因以按概率  $P_m$  用其它的等位基因代替,来产生新个体。常用的变异方法主要有基本位变异、插入变异、位移变异等。

进行变异操作时需要维持试题的题型、分值不变并将难易度等控制在合理的范围内。本文中变异主要发生在分段(题型)的染色体上,指定某个基因位置对其进行变异操作。因为是分段的进行单点变异操作,所以在整个染色体上表现为多点变异。下图为染色体多点变异,变异操作示意图。

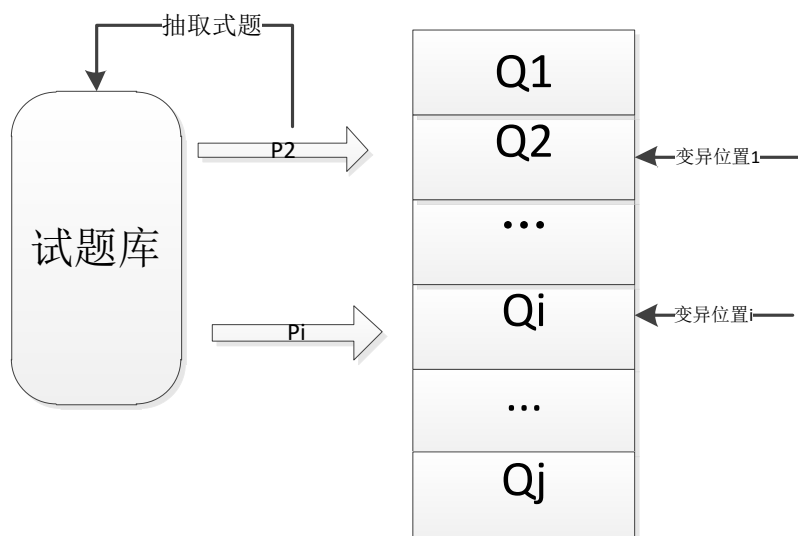


图 4.10 染色体变异图

上图中表示在变异位置 1 和 i 处,按照一定的条件在题库中抽取满足要求的试题 P2、Pi 代替染色体中 Q2 和 Qi。

$P_m$  的取值需要注意,过小则不利于种群中新个体的产生;过大将会使算法变为随机搜索算法<sup>[35]</sup>。通常为了保障种群的多样性,提高种群搜索效率,防止搜索陷入局部最优,变异概率的取值控制在较小的范围之内(通常在 0.001~0.1 间)。本文中变异概率按如下公式取值。

$$P_m = \begin{cases} K_1 & \dots \dots \dots f < f_{avg} \\ K_2 + (K_1 - K_2)(f_{max} - f)/(f_{max} - f_{avg}) & \dots \dots f \geq f_{avg} \end{cases} \quad (4-6)$$

$K_1$  取 0.1,  $K_2$  取 0.01;  $f$  为当前个体的自适应度;  $f_{max}$  当前种群中最有个体的适应度;  $f_{avg}$  为当前种群的平均适应度。

#### 4.4.8 算法终止条件

算法的运行过程中每一次迭代过后种群都将发生变化。可能存在满足约束条的个体早已出现,但是算法仍在执行;也可能出现多次迭代过后子代个体适应度值任然达不到目标适应度函数值。所以合理的算法终止条件对于提高组卷效率来说非常重要。如何去设定组卷终止条件,我们主要考虑以下几点:

- 1) 设定好最大适应度值,当适应度值达到此值时,算法终止。
- 2) 设定好最大迭代次数,当达到此值时,算法终止。
- 3) 当所有个体适应度值的方差小于某一值时,可考虑终止算法。

根据以上算法终止条件,本文的终止条件为:

- 1) 种群计划的迭代次数达到某一值。系统可设定范围在 100~200 代,本系统默认取值为 100 代。
- 2) 当最优个体的适应度值与目标要求的适应度值的比值达到 95%时,可终止算法。

#### 4.5 组卷实现的总体过程

综合以上内容,本文所用遗传算法实现组卷的总体过程如下:

- 1) 产生初始种群,计算个体适应度值:根据试卷的分值要求,题型要求,题数要求,每种题型题数要求,随机产生试卷,生成初始试卷。并根据适应度计算公式,计算出试卷的适应度值。

- 2) 判断试卷适应度值是否符合要求, 如果符合要求则输出试卷。如果不符合要求, 先对当前种群适应度值进行排序, 将适应度值高的一些个体保留到子代种群中, 不参加染色体选择、交叉、变异等操作。然后对剩下的个体进行步骤 3), 开始对染色体进行选择、交叉、变异等操作。
  - 3) 通过设定的选择概率选择染色体, 即从初始试卷中初步选择试卷。
  - 4) 通过交叉概率对染色体进行交叉操作。
  - 5) 通过变异概率对染色体进行变异操作。
  - 6) 通过设定的算法终止条件对算法的执行情况进行判断, 输出结果。
- 本文组卷核心流程图如下

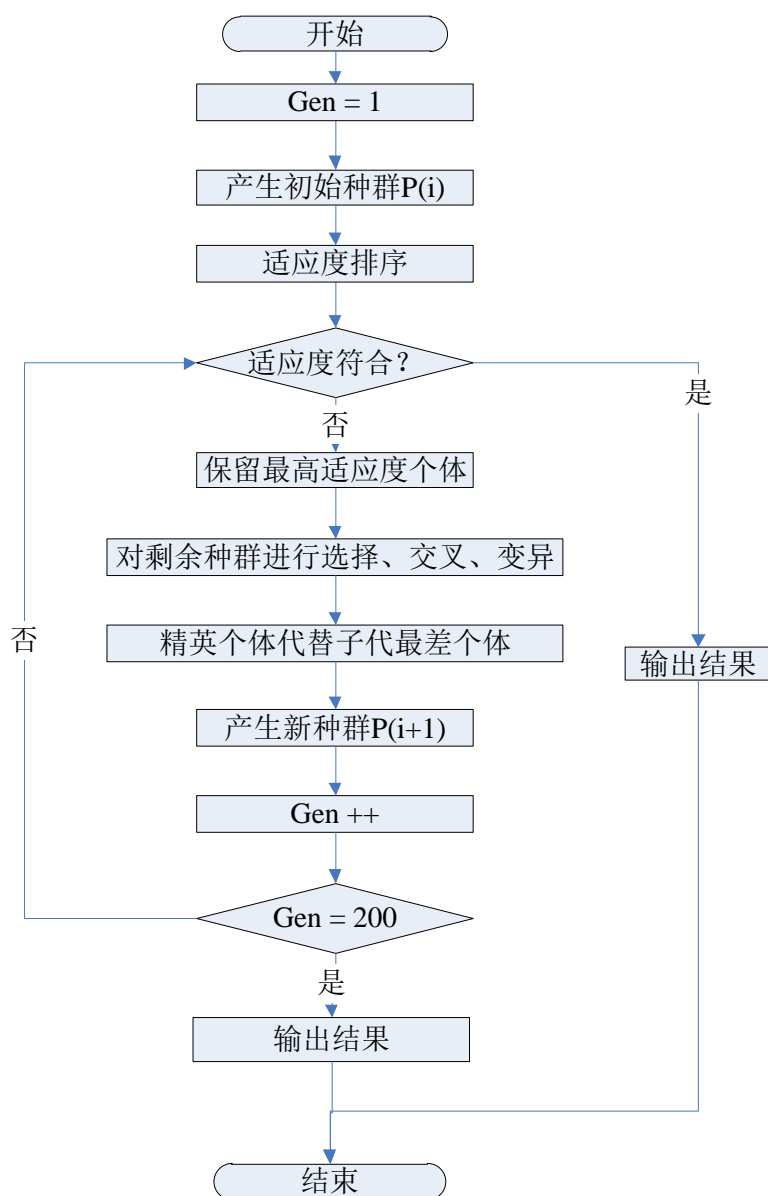


图 4.11 组卷核心流程图

## 4.6 本章小结

本章主要介绍系统的实现过程,首先明确组卷要求和原则,进行系统的可行性分析,架构设计,数据库相关设计,再具体介绍系统开发环境和开发技术等。最后是遗传算法的应用过程,通过染色体编码,初始种群确定,适应度函数分析,遗传算子设计等过程完成组卷系统。

## 第 5 章系统运行分析

为了验证本算法在系统中的实际组卷效果，特以《数据结构》课程为例，对该系统进行相关的实验测试。

### 5.1 测试过程及分析

系统试题库中含有《数据结构》各类试题共 600 道，共有五类题型（选择题 200 道、填空题 150 道、判断题 150、简答题 50、设计题 50），试题编号、难度、重复度、所属章节、区分度、分值等相关属性在录入试题库时已给定。并存入之前第一年、第二年、第三年的该门课程的试卷。本系统中，完成的试卷只有在确认使用后才能正式成为历史试卷，在之后的组卷过程其将会成为正式的参考试卷。

根据本文所设计的两种组卷策略，分别用这两种策略进行组卷。

#### 5.1.1 参数设置

针对本文组卷所用遗传算法，设置遗传算法参数：初始种群规模 50，每一代精英保留个体数为 5 个，初始种群交叉概率为 0.65，变异概率为 0.01，试卷各约束函数的权重值都为 4:3:3，终止条件为最大迭代次数 100 或者适应度函数值达到 0.48。

由于本文所考虑的组卷策略有两种，下面将分别按照这两种策略设定试卷相关的参数。

第一种组卷策略，即从整体上对试卷设定要求，试卷分值，覆盖度，题型，各题型题数和分值都是给定的确定值必须要满足的条件。

下面是本次运行测试时设定的试卷的主要指标要求难易度 0.5，目标区分度为 0.45，目标相似度为 0.10，这三个指标在实际操作过程中很难完全满足，在本文中采取趋近取值策略。

试卷具体要求如下表。

表 5-1 试卷题型-题量-分值表

题型	选择题	填空题	判断题	简答题	程序设计题
题量	14	10	10	4	2
分值	28	10	10	32	20

表 5-2 其它约束要求表

总分值	覆盖度	难易度	区分度	相似度
100	所有章节	0.5	0.45	0.10

以上是第一种组卷策略情况下试卷相关条件设定。

下面介绍使用第二种卷策略时需要设定的参数。即通过将满足一定约束条件的各个题型组合起来完成组卷。

由于试卷的总分值，覆盖度，题型的种类和分值，以及各题型的题数是指定的，必须要满足。所以表 5-1 依旧适合此组卷策略。下面是各个题型的一些要求，如下表。

表 5-3 题型-难易度-区分度-相似度要求表

	选择题	填空题	判断题	简答题	设计题
难易度	0.65	0.60	0.45	0.4	0.45
区分度	0.45	0.5	0.5	0.4	0.50
相似度	0.1	0.1	0.1	0.1	0.1

### 5.1.2 运行分析

根据第一种组卷策略，按照上述所设定的要求本系统进行了 8 次组卷最终形成 6 套试卷。下面表格是分析试卷各项指标的平均值。

表 5-4 策略一运行结果表

	试卷一	试卷二	试卷三	试卷四	试卷五	试卷六	平均值
难易度	0.53	0.55	0.47	0.46	0.56	0.51	0.513
区分度	0.41	0.47	0.39	0.50	0.46	0.42	0.4417
相似度	0.08	0.06	0.12	0.09	0.11	0.13	0.0983

由上表分析可知，第一种组卷策略下试卷的总体难易度、区分度以及相似度指标与目标差值较小。由第三章公式 3-7 可以算得第一种组卷策略下的试卷目标函数值。（其中难易度偏差、区分度偏差、相似度偏差占目标函数的权重值分别为 0.4、0.3、0.3）

$$\begin{aligned} \min g &= r_1 h + r_2 k + r_3 \\ &= 0.4 * 0.013 * 100 + 0.3 * 0.0083 * 100 + 0.3 * 0.001 = 0.808 \end{aligned}$$

再由本文适应度函数公式 4-1 可得，这六套试卷的平均适应度函数值为

$$f = 1/1 + 0.808 = 0.553$$

根据第二种组卷策略，系统同样进行了 8 次组卷，其中成功组卷 4 次。我们对成功的四次进行分析，得到其相关指标的均值。如下表

表 5-5 策略二运行结果表

	选择题	填空题	判断题	简答题	设计题
难易度	0.69	0.64	0.41	0.44	0.48
区分度	0.41	0.46	0.55	0.45	0.56
相似度	0.12	0.11	0.13	0.07	0.06

由上表，根据第三章公式 3-8，得用此方法完成的试卷的目标函数值

$$\begin{aligned}\min g &= r_1 \frac{\sum_{i=1}^j Q_i S_i}{j} + r_2 \frac{\sum_{i=1}^j S_i T_i}{j} + r_3 \frac{\sum_{i=1}^j S_i Y_i}{j} \\ &= 0.4 * \frac{3.8}{5} + 0.3 * \frac{4.82}{5} + 0.3 \frac{2.72}{5} = 0.7582\end{aligned}$$

可得这四套试卷的平均适应度函数值为

$$f = 1/1 + 0.7582 \approx 0.5688$$

对比分析可知，第一种组卷策略的组卷成功率较高，但是整体成功率仍有待提高。通过分析主要是由于试题库中各题型题量较少，可供选择的满足要求试题量较少。

综上可知，在题库内试题非常丰富时，选择第二种组卷策略较为合适，在题库较小时，选择第一种组卷策略。由于本系统试题题量较少，所以选用第一种组卷策略较为合适。

## 5.2 组卷算法分析

由于在本系统的题库下，第一种组卷策略较为合适，下面将以第一种组卷策略对系统的遗传算法进行测试。主要测试以下几方面：

- 1) 组卷算法的效率，通过在相同组卷条件和组卷要求下与传统遗传算法比较；
- 2) 研究交叉算子，变异算子取值对系统的影响；
- 3) 研究精英保留策略对系统的影响。

效率测试主要通过比较生成试卷的耗时；交叉算子、变异算子以及精英保留策略主要测试在不同取值情况下最优试卷的适应度值大小变化。



### 5.2.1 遗传算法效率测试

为了测试本文遗传算法在组卷效率上对系统的影响。我们在相同条件下分别用本文的遗传算法和传统未作变动的遗传算法按照 5.1 章节的试卷要求来组卷，分别统计两种算法生成满足目标要求的试卷所用时间。在忽略试卷质量差异的情况下，其结果如下图 5.1 所示。

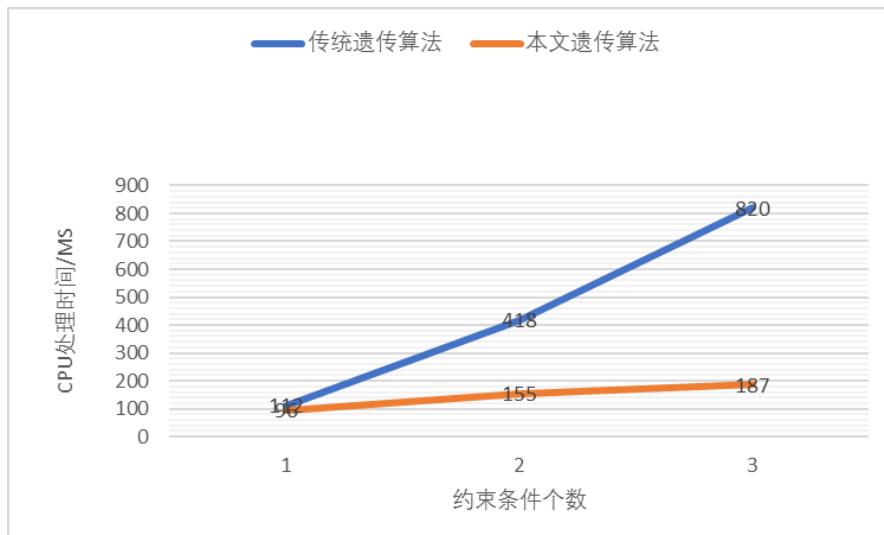


图 5.1 遗传算法运行效率图

由上图可知，在满足基本试卷要求无其他约束条件情况下，两者组卷耗时差别不大；当约束条件增加时（主要增加难易度、区分度、相似度约束），传统遗传算法组卷耗时将明显多于本文的遗传算法组卷。组卷问题的本质是多约束条件下的函数优化问题，所以在约束条件增多的情况下本文的遗传算法在组卷系统中具有显著的高效性。

### 5.2.2 遗传算子测试

最大迭代次数以及遗传算子都对最优试卷适应度函数有重要影响，本文与传统的遗传算法中遗传算子不同之处在于对交叉算子和变异算子采取了动态取值策略。为了检测他们之间的具体关系，本实验通过第一种组卷流程在不同迭代次数条件下记录未调整前的遗传算子和调整后的遗传算子所对应的适应度值。其结果如下表 5-11 所示。

表 5-6 迭代次数与遗传算子调整前后适应度值关系

迭代次数	10	30	50	100	150	200
未调整	0.483	0.554	0.566	0.568	0.573	0.574
调整后	0.267	0.375	0.514	0.628	0.673	0.724

为了更加直观地表示出改变他们之间的关系，我们给出图 5.2

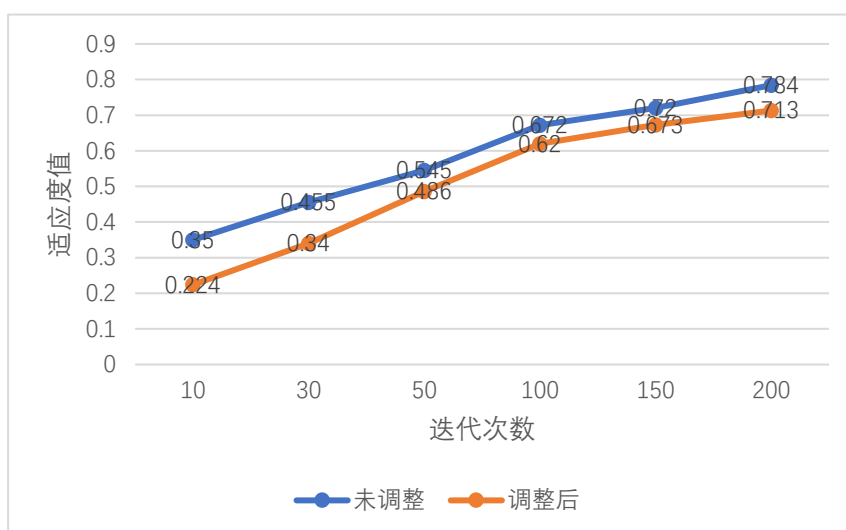


图 5.2 调整遗传算子在各迭代次数下的种群适应度图

由图可以清晰看出，进化代数在 70 代之前，随着进化代数的增加调整遗传算子后的最优适应度值在迅速向未调整遗传算子种群最优适应度值靠近。而在进化到 70 代之后，调整后遗传算子后的最优适应度值相较于未调整情况下迅速提高。主要是因为固定值遗传算子无法适应每一代的种群情况，不利于种群收敛到最优解。根据每一代种群情况调整遗传算子使适应度值高于平均适应度值的个体得以保留，加快淘汰差的个体，保证了种群收敛到全局最优解。

### 5.2.3 精英保留策略对系统影响

在对遗传算子采取自适应策略的同时对每一代种群合理选用精英保留策略。在本文中，选取种群中 10% 的个体作为精英直接保留到下一代种群中。本实验中种群数为 50，那么就需要设定 5 个相应的精英个体数量保留下来，采用第一种组卷策略组卷。下面将给出采取精英保留策略与未采用此策略所产生的种群最优适应度平均值的组卷运行结果。

表 5-7 迭代次数与精英保留关系表

最大迭代次数	10	30	50	100	150	200
保留精英	0.350	0.455	0.545	0.672	0.720	0.784
未保留	0.224	0.340	0.486	0.620	0.687	0.713

其结果用图表示如下：

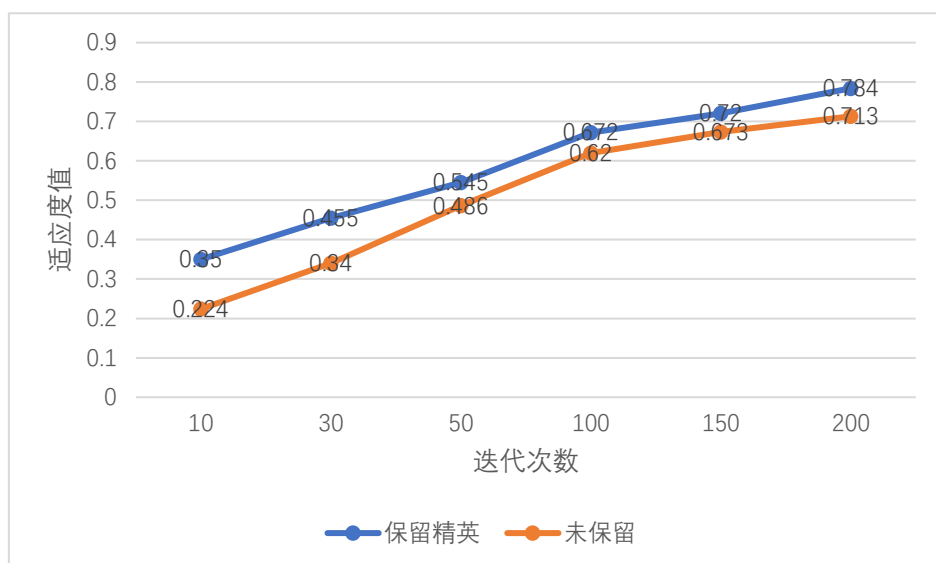


图 5.3 迭代次数与精英保留关系图

由图可以看出采用精英保留策略的遗传算法，其种群中个体的最优适应度明显改与未采用此策略的遗传算法。

### 5.3 试卷的评价信息

在系统完成组卷的同时会生成一份关于该试卷的初始评价报告。其主要包括试卷的实际难易度、实际区分度、实际相似度、分值、题型、题数、各题型分值等基本信息。

还包括以下信息。

- 1) 各章节分值比，记做

$$\sum_1 S_1 : \sum_2 S_2 : \dots : \sum_i S_i$$

$S$  表示章节试题总分值， $i$  表示试题所属章节。上式表示各个章节知识点总分值之间的比例关系。通过此章节分值比例信息我们可以直观了解此张试卷的知识点情况。

- 2) 相似度分值与重复试题

此信息主要包括该试卷与最近三次考试试卷中每张试卷重复的试题以及总分值。

- 3) 试卷的难度、区分度的分值分布

此信息通过统计试卷中的各个难易度、区分度级别分别对应的分值，并以表格形式显示出来。可以直观的看到试卷的难易度和区分度分布情况。

## 5.4 本章小结

本章主要介绍系统运行结果以及分析结果。首先介绍两种组卷策略下的组卷结果，并结合本系统的实际情况对组卷结果以及试题库进行分析。其次对本文所选用的组卷算法在实际运行过程中的情况进行分析，主要介绍遗传算子对于系统时间运行的影响、遗传算子和精英保留策略对种群最优适应度函数值的影响。最后介绍在完成组卷之后，系统生成的关于试卷的相关评价信息。

## 第 6 章总结与展望

### 6.1 总结

组卷问题是由多个约束条件组成的多目标优化问题。由于其约束条件的复杂性,在用传统的方法来解决此问题时往往不能够得到较好的结果。本文选用多目标优化问题中成熟的遗传算法来解决组卷问题。本文首先介绍了组卷过程及试卷的主要评价指标,重点介绍了本文所设计的两种组卷流程策略。随后对组卷问题进行数学建模,并介绍传统遗传算法以及遗传算子进行全面的叙述。紧接着,在此基础上开始实现组卷系统以及算法的应用。主要介绍了本系统的开发环境,数据库开发的相关技术等,并详细介绍算法的实现过程。最后进行系统的运行和分析,通过对比本文中两种组卷策略在实际运行过程的组卷情况,说明这两种的组卷策略的特点,再通过对分析组卷效率、调整的交叉算子变异算子以及精英保留策略对试卷质量的影响,进一步说明本文所用遗传算法的高效性和可行性。

### 6.2 后续工作和展望

虽然使用本文中遗传算法的系统能够实现组卷的基本功能,但是由于开发者能力以及开发时间等多方面因素的限制,系统在整体的界面,部分功能等需要进一步改进和完善。

首先,对试题库进行改进。需要继续丰富试题,在今后还需要添加更多专业,更多课程的试题。本文中第二种组卷流程的成功率不高最主要就是因为试题库内试题量不够,没有足够的试题供选择。

其次,对每次考试结束后的试卷进行分析,通过考生的实际答题情况计算出试题的相关难易度,区分度等属性。并对试题库中原来试题的属性进行更新。使该试题库中试题能够更准确的检测本校学生的实际学习掌握情况,增强本系统试题的科学性、可靠性,保证组卷的有效性。

最后,可以进一步的开发本系统,丰富系统的功能,如用户只需要给定一个目标试卷考试的期望考试平均分,系统可以自动根据过往考试情况等生成相关试卷供用户选择;又或者继续添加上在线练习测验的功能,帮助快速完善数据库。

所以,本人今后还需继续学习系统相关的开发设计技术以及优化算法等知识来提升和改进本系统。

## 参考文献

- [1] 肖豆. 智能组卷系统的设计与实现[D]. 电子科技大学, 2017.
- [2] 王欢, 陈莹. 高校试题库管理系统的开发研究[J]. 价值工程, 2010, 29(36):225.
- [3] 盖洋侨. 智能组卷系统设计与实现[D]. 大连理工大学, 2016.
- [4] 张琦, 郑河荣, 刘志. 基于优化遗传算法的智能组卷系统研究. 浙江工业大学学报, 2009, (06):11-12.
- [5] 李紫蔓. 基于改进遗传算法的在线考试智能组卷系统[D]. 郑州大学, 2013.
- [6] 杨晨霞, 涂风涛. 基于 B/S 模式的《计算机基础》试题库系统设计与实现[J]. 科技广场, 2010, (01):139-141.
- [7] 孙文清, 张旭东. 高校网络试题库平台开发设计研究[J]. 信息记录材料, 2017, 18(08):80-81.
- [8] 程秀丽. 高等学校经济数学试题库系统的建设与应用[D]. 青岛大学, 2017.
- [9] 陈加粮, 黄贤顺. 基于.NET 平台下试卷自动生成系统研究[J]. 科技视界, 2017 (16):70-71.
- [10] 张钰柯. 改进遗传算法在智能组卷中的应用[D]. 浙江理工大学, 2018.
- [11] 刘净. 基于智能组卷策略的考试系统设计与实现[D]. 河北工业大学, 2014.
- [12] 雷亮, 汪同庆, 杨波. 改进遗传算法在图像挖掘中的应用[J]. 计算机工程与应用, 2009, 25 (03):412-414.
- [13] 金萍. 基于遗传算法的智能组卷方法研究[D]. 北京理工大学, 2015.
- [14] 杨俊学. 遗传算法在智能组卷系统中的应用研究[D]. 天津工业大学, 2018.
- [15] 毛秉毅. 基于遗传算法的智能组卷系统数据库结构的研究[J]. 计算机工程与应用, 2003, (06):230-232.
- [16] 张京钊, 江涛. 改进的自适应遗传算法[J]. 计算机工程与应用, 2013, 6(04):23-24.
- [17] 刘学增, 周敏. 改进的自适应遗传算法及其工程应用[J]. 同济大学学报(自然科学版), 2009, 7(03):412-414.
- [18] Qiufang Ma. Researching of Intelligence Test Paper Generation Software based on Improved Genetic Algorithm in Java Program Design Teaching[A]. Wuhan Zhicheng Times Cultural Development Co., Ltd.Proceedings of the 2nd International Conference on Mechatronics Engineering and Information Technology(ICMEIT 2017)[C].Wuhan Zhicheng Times Cultural Development Co., Ltd:武汉志诚时代文化发展有限公司, 2017:4.
- [19] 王萌, 金汉均, 王晓荣. 集合随机抽选法在智能组卷中的研究[J]. 计算机工程与设计, 2014, 21(19):241-243.
- [20] 孙凤喜. 基于遗传算法的智能组卷系统的设计与实现[D]. 河北科技大学, 2013.
- [21] 王友仁, 张砦, 崔江, 姚睿, 储剑波. 智能组卷系统的建模与算法研究[J]. 系统工程理论与实践, 2004, (09):85-89+97.
- [22] 罗毅. 基于知识点-难度比例的组卷算法研究[J]. 武汉理工大学学报, 2009, 31(10):154-156.
- [23] 王雍钧, 黄毓瑜. 基于知识点题型分布和分值的智能组卷算法研究[J]. 计算

- 机应用与软件, 2004, (08):111-113.
- [24]Hêmin Golpîra,HassanBevrani. A Framework for Economic Load Frequency Control Design Using Modified Multi-objective Genetic Algorithm[J]. Electric Power Components and Systems, 2014, 42(8).
- [25]Shang Gao~(1,2) 1 School of electronics and information, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu 212003, China 2 State Key Laboratory of CAD & CG, Zhejiang University, Hangzhou, Zhejiang, 310027, China. Solving Traveling Salesman Problem by Genetic Ant Colony Optimization Algorithm[A]. 江南大学信息工程学院. Proceedings of 2008 International Symposium on Distributed Computing and Applications for Business Engineering and Science Volume[C]. 江南大学信息工程学院, 2008:6.
- [26]Li J M, Li J, Zhang J P, Ding N. Research on intelligent test paper generation based on multi-variable asymptotic optimization[A].The 2nd International Conference on e-Business and Information System Security [C]. Wuhan:IEEE Press,2010:177-181.
- [27]Xiaoqin Wu. Research and Design of Intelligent Generating Test Paper System[J]. Computational Intelligenceand Software Engineering, 2009, (11):1-4.
- [28]C Lu, GM Xu. Application Research of Web Examination System Based on College[J]. Energy Procedia, 2012, 17(Part A): 528-533.
- [29]Andrew J. Brust. Programming Microsoft SQL Server 2005[M]. America: Microsoft Press, 2006.
- [30]Patrick Desjardins. Introducing Visual Studio[M]. Apress:2014-06-15.
- [31]Ben Brosgol. An introduction to the C# language and .NET infrastructure[J]. ACM SIGAda Ada Letters, 2009, 29(3).
- [32]<https://blog.csdn.net/paulfeng20171114/article/details/82454310>.
- [33]严莉娜. 遗传算法在计算机自动组卷中的应用研究[J]. 电脑编程技巧与维护, 2018, (02):30-32.
- [34]陈国彬, 张广泉. 基于改进遗传算法的快速自动组卷算法研究[J]. 计算机应用研究, 2015, 32(10):2996-2998+3003.
- [35]张清富, 倪子伟. 基于动态自适应技术遗传算法的智能组卷策略[J]. 电脑与电信, 2009, (07):42-44.
- [36]<http://www.iitk.ac.in/kangal/codes.shtml>.
- [37]尚莹. 基于遗传算法的组卷系统应用研究[D]. 大连海事大学, 2010.
- [38]Edwin K. P. Chong, Stanislaw H. Zak. An Introduction to Optimization[M]. 北京:北京电子工业出版社,2015. 197-199.
- [39]Yong Ou-Yang , Hong-Fang Luo. Design of personalized test paper generating system of educational telenet based on genetic algorithm[C]. International Conference on Computer Science Education, Nanning, 2009:170-173.
- [40]Shi J, Miao Y. Research and Application of Auto Test Paper Generation Based on Genetic Algorithm[J]. Computer Knowledge & Technology, 2014.
- [41]Kouresh Ardestani, Kwvin Hoffman. Fast Track ADO.NET C# Edition[M]. 北京:清华大学出版社, 2003. 214-230.
- [42]Guang Cen, Yuxiao Dong, Wanlin Gao. A implementation of an automatic examination paper generation system[A]. Mathematical and Computer Modelling

- in Agriculture. 2010, 51(11-12): 1339-1342.
- [43]Holland J H. Adaptation in Nature and Artificial System [M]. The University of Michigan Press, 1975, MIT Press, 1992. 15-20.
- [44]C Lu, GM Xu. Application Research of Web Examination System Based on College[J]. Energy Procedia, 2012, 17(Part A): 529-532.
- [45]Krzysztof Cwalina. Framework Design Guidelines[M]. 北京:机械出版社, 2007. 15-40.



## 在学研究成果

软件著作权申请

自动组卷系统, 登记号: 2019SR0219168

## 致谢

研究生生涯两年的学习时光匆匆而过，回首过去的这两年，心中有太多的感慨。从刚来学校时，宿舍在哪，逸夫楼在哪，阶梯教室在哪，上课、吃饭、买东西去哪都要问一下同学，一切都是那么陌生和新奇。到后来每天宿舍、食堂、逸夫楼三点一线的校园节奏，学习生活变得有序和紧凑下来。内心在此过程中也慢慢的平静下来，积淀着对脚下这片土地的情感。

这两年研究生的学习经历给我不仅仅是学业上的收获，更多的是做人和学习上的教导。由于自身的原因，从一开始的专业课程学习和课程作业就让我非常的不适应，难以跟上大家的学习节奏，一度怀疑自己读研的选择是否正确。但是在亲人的不断鼓励下，导师的耐心指导下，同学们交流帮助下，让我慢慢的适应了这样的生活和环境。我的内心也在此过程中更加成熟和坚强。

首先，我要感谢我的导师刘宏申。感谢他在我刚进学院的时候能够接受我成为他的学生，并在整个研究生阶段根据我的实际情况给我细心的安排和指导，在论文阶段从选题、前期准备、写作指导、质量把关给予我的帮助。我由衷的感谢刘宏申老师。此外刘老师严谨治学、与人和善、事必躬亲的精神都值得我学习。在这里诚挚的祝刘老师工作顺利和生活美满。

同时，我还要感谢我的师兄王旭晨。虽然师兄这两年在学校的时间非常少，但是每次见面时都是非常关心我们的情况，并积极给与我学习上的指导，给我很多学习建议和学习资源。感谢我的同学姚浩男，在实验室里一同学习的两年研究生时光里，我的很多问题都是他在第一时间给我以帮助，并且他所特有的山西式幽默和生活态度深深的感染了我。感谢实验室的老师和同学们，是你们丰富了我的研究生时光，开阔了我的视野，并给予我无私的帮助。

最后，我要感谢我的家人和亲友，感谢你们一直以来的支持与付出！