

基于 B/S 模式下的优化遗传算法的自动组卷系统设计

王旭, 吴志雄, 周子勋

(长江大学 计算机科学学院, 湖北荆州, 434023)

摘要: 本文针对自动组卷通常采用传统遗传算法实现, 受题目难度、知识点覆盖以及答题时间等约束条件的限制, 组卷过程较慢且算法存在“早熟、局部最优”问题的困扰, 成卷质量偏低。文章将对传统遗传算法选择、交叉和变异算子进行改进, 实现了组卷策略的优化。并结合B/S模式的优势, 设计并实现了基于优化遗传算法的自动组卷系统, 取得了良好的实际应用效果。

关键词: B/S模式; 遗传算法优化; 自动组卷

DOI:10.16589/j.cnki.cn11-3571/tn.2023.04.027

0 引言

随着信息化时代的发展, 以及全国疫情的爆发, 各种云会议、网络授课等网络技术的出现, 同时教育行业的自动组卷模式也成为热点。

本文采用具有较高安全性、扩展性和兼容性等特点的B/S模式架构^[1], 在普通遗传算法的基础之上对选择算法的选择策略进行优化将更好的筛选试题及交叉概率和变异概率的调整, 将更好的防止“未熟就收敛, 局部最优, 生成的试题质量偏低”等问题, B/S和优化的遗传算法的相互融合很好的应用到自动组卷系统中^[2]。

1 技术概述

1.1 B/S 模式

Browser/Server 模式, 是指浏览器 / 服务器模式下的交互。B/S 相比于 C/S 模式不需要安装相应的软件、占用内存小、与服务器交互时响应速度快的特点, 而且可跨平台操作、业务逻辑简单方便、易于维护和较灵活结构等显著优势, 使得自动组卷系统交互下展现其特性。

1.2 JMV 体系

JMV 体系是指 Java+MySQL+Vue 技术的组合实现。采用具有多线程、安全性等特性的 Java 的 Spring Boot 架构编写前后端服务器功能达到接收前端发送的增删、修改、组卷操作请求命令和做出响应返回数据; 数据库服务器开发将使用 MySQL 数据库按照难度、知识点、区分度、时间等约束条件建立相应的表用以存储试题库; 前端界面将采用具有视图、数据、结构分离操作简单化、组件化、速度快等特性的 Vue 框架技术, 用于用户点击界面的某一操作实现交互功能, JMV 体系将很好适用于自动组卷系统中。

2 系统设计

2.1 总体设计

自动组卷系统采用三层架构的思想, 即表示层

(Demonstrate)、服务层 (Service)、数据层 (Database)^[3]。

基本设计流程: 表示层是用户登录 / 注册功能和用户点击增加或删除、修改、查看试题等操作时将采用 Ajax 异步请求服务, 向服务器发送用户的增删改查试题等操作命令; 服务层是对界面发送的增删改查试题等操作命令进行逻辑处理, 并且根据用户选择的难度系数、知识点、时间等约束条件通过优化的遗传算法向试题库进行抽选, 然后返回数据到前端; 数据层是每种试题按照难度系数、知识点、区分度、时间等约束条件进行存储, 接收服务器的查询命令, 返回数据^[4]。如图 1 所示。

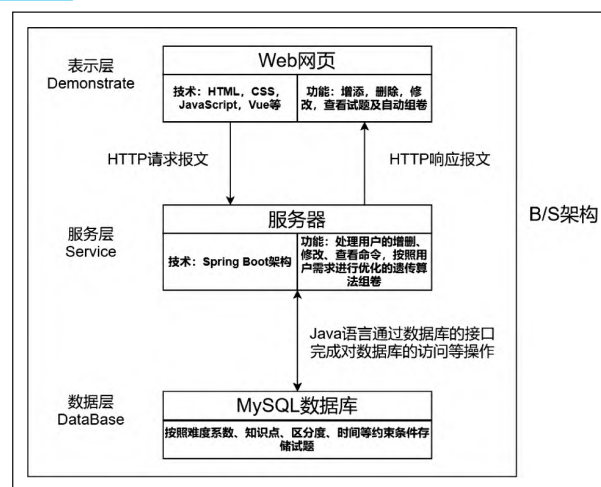


图 1 三层架构图

2.2 系统功能设计模块

本系统功能结构分为三大部分: 用户管理模块; 数据库模块; 考试管理模块。

(1) 用户管理模块

用户管理由管理员、教师、学生组成, 登录后的界面各不相同, 其中管理员权限高于教师权限和学生权限。

①管理员: 拥有创建或删除用户, 修改密码, 用户角色管理, 系统菜单管理, 考试管理以及试题库的管理, 维护系统数据等。

②教师: 拥有密码修改、试题管理、科目管理权限, 例

如增删试题，自动组卷等。

③学生：拥有密码修改、试题管理和试卷管理和自动组卷功能等。

(2) 数据库模块

MySQL 数据库是整个系统软件的根基，是存储数据和管理数据的库^[5]。

数据库表的设计，数据以二维表格呈现的方式，用来记录存储数据对象的关系。本系统包括用户表（user）、菜单表（menu）、角色表（role）、课程表（course）、试卷表（test）、试题表（paper）、成绩表（score）等表，其中在试题表中字段列举了每个题的难度（difficult）、区分度（distinction）、知识点（knowledge）、时间（time）等参数，有利于提高组卷系统的高效性。如表 1 所示。

表1 数据表

字段	描述	数据类型	长度
question_ID	编号	int	100
topic	题目	varchar	200
answer	答案	varchar	200
difficulty	难度	decimal	3
distinction	区分度	decimal	3
knowledge	知识点	set	21
answer_time	答题时间	int	256

(3) 考试管理模块

该模块主要有试题管理、科目管理、试卷管理。用户可对试题进行增加、删除、修改，还可按照难度系数、知识点、区分度、答题时间进行自动组卷操作。

3 组卷策略设计

通过遗传算法模拟自然界生物进化的思想，“物竞天择，适者生存”的生存法则^[6]，以种群为研究对象，对每个个体进行编码，依据预先设定的目标函数对个体进行评估给出相应的适应度值，通过逐代的选择、交叉、变异，逐渐形成新的、适应度高的种群，最后进行解码操作得到最大的适应度的个体并输出。其执行流程如图 2 所示。

本文将采用优化的遗传算法在选择策略上采用轮盘赌选择法，保留当前种群中 30% 最优个体并记为 oldmax，其不参与交叉变异，剩下的个体由轮盘赌法选择出来。本文将传统的只使用轮盘赌选择算法改进为结合最优个体保存策略，改进后既可以保证遗传算法能够收敛，又能避免提前收敛陷入局部最优。

算法整体执行流程如下：

(1) 种群初始化

系统将依据试题难度系数（difficult）、区分度（discrimination）、知识点（knowledge）、答题时间（time）

4 个参数作为约束条件，假设有 n 个试题则组成一个 n×4 的矩阵，如 (1) 式所示：

$$D = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & a_{n4} \end{bmatrix} \quad \text{式 (1)}$$

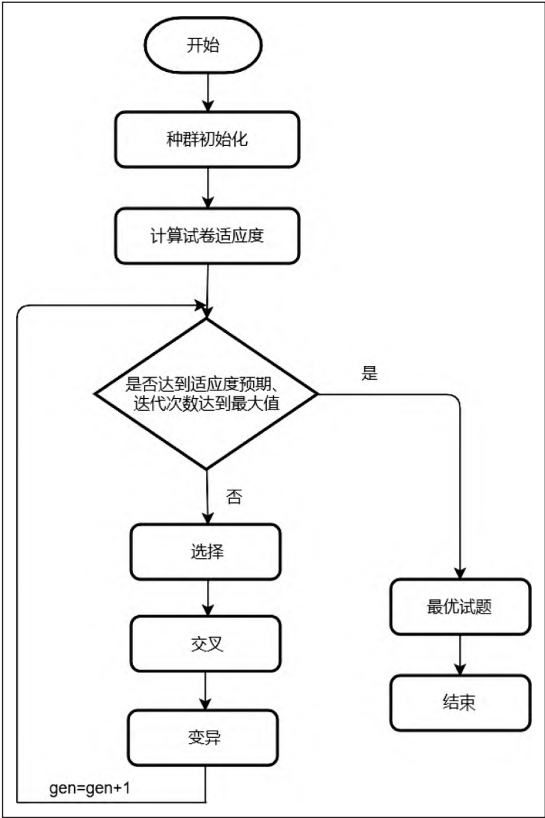


图 2 遗传算法执行流程图

D 是目标矩阵的解，某一题 $D_{i1} = [a_{i1}, a_{i2}, a_{i3}, a_{i4}]$ ，其中 $0 \leq i \leq n$ ， a_{i1} 代表难度系数， a_{i2} 代表区分度， D_{i3} 代表知识点， D_{i4} 代表答题时间，由上述约束条件生成试卷，构成“染色体”编码。

根据用户选定的考试类型和题量、难度系数、区分度、知识点、答题时间等条件，随机从试题库中抽取符合条件的题目，生成初代的 n 套卷。

(2) 适应度函数的建立

在遗传算法中，适应度的大小是区分种群优劣的主要指标，将直接影响着算法的性能^[7]，本系统将采用 4 个约束条件对适应度进行设计，模型如 (2) 式所示：

$$F = w_1 \times \left(1 - \left| \frac{D - D^*}{D^*} \right| \right) + w_2 \times \left(1 - \left| \frac{Q - Q^*}{Q^*} \right| \right) + w_3 \times K + w_4 \times \left(1 - \left| \frac{T - T^*}{T^*} \right| \right) \quad \text{式 (2)}$$

式中, $D = \frac{\sum_{i=0}^n d_i s_i}{S}$ 为难度系数, $Q = \frac{\sum_{i=0}^n q_i s_i}{S}$ 为区分度, $K = \frac{k_1}{k_2}$ $T = \sum_{i=0}^n t_i$ 为时间, S 为给定的总分值, T 为给定的总的时间。

(3) 遗传操作算子

① 选择算子

选择是指在种群中选择好的个体, 淘汰劣的个体, 是生物进化的“优胜劣汰”的应用, 建立在适应度的基础之上, 适应度越高, 被选择的概率就越大。通过轮盘赌和最优适应度保存策略对其进行选择操作。

假设种群由 n 个个体组成, 个体 n_i 的适应度用 F_i 表示, 则 n_i 个体被选择的概率如 (3) 式所示:

$$P_i = \frac{F_i}{\sum_{i=1}^n F_i} \quad \text{式 (3)}$$

式中, F_i 指第 i 个个体的适应度值。

经过轮盘赌选择后, 保存 30% 的最优个体记为 $oldmax$, 并且不参与个体的交叉变异操作, 最后再替换掉适应度低于 $oldmax$ 的个体, 较好的解决了“收敛快和局部收敛”的问题。

② 交叉算子

交叉是指模拟生物进化的过程, 从中获取优良个体, 产生新的下一代个体, 交叉过程是按照一定的交叉概率在相应库中两两随机交叉, 本系统采用两点交叉方式, 则交叉过程如图 3 所示。

随机交叉方式, 容易导致交叉后会出现重复的题目, 使得算法早熟早收敛。为了防止重复事件的发生, 通过控制交叉概率 P_c , 一般选择 0.4-0.99, 本系统使用 0.5 的交叉概率, 用以保证题目的不重复性。

③ 变异算子

变异是指生物模拟生物个体突变的过程, 通过变异概率

P_m 随机的反转某位二进制数值, 则交叉过程如图 4 所示。

变异过程是一种局部搜索, 为了防止因选择和交叉导致的信息丢失, 保持种群的多样性^[8], 变异概率一般选择 0.0001~0.1, 本系统使用 0.01 的变异概率, 防止出现未成熟就收敛的现象。

改进后的遗传算法很大程度的保证了染色体中的优良个体, 很大程度的防止早熟, 局部最优等问题, 最终通过不断的选择、交叉、变异达到预定的适应度或者迭代次数, 满足最高的适应度值的个体即为最优解。

4 系统实现

■ 4.1 用户管理实现

系统初始登录界面, 新用户可选择教师或学生来注册系统, 然后可登录系统, 已注册用户可直接登录, 成功后进入主界面, 如图 5 所示。管理员拥有最高权限, 可点击用户管理实现对教师和学生信息的查找和修改, 另外还可以实现角色菜单管理功能。教师或学生点击系统管理的密码修改按钮, 可在输入框中输入旧密码, 再输入新密码, 确认密码, 则可实现密码修改。如图 6 所示。

■ 4.2 考试管理实现

考试管理包括了 3 大部分, 即试题管理、科目管理、试卷管理, 教师和学生可在试题界面操作向服务器发送试题增删修改操作命令, 服务器响应并传输相应数据^[9], 本系统通过使用优化的遗传算法进行组卷, 教师或学生在试卷管理界面输入难度系数 (0-1, 数字越大, 难度越高), 区分度 (0-1, 数字越大, 难度越高), 选择知识点, 再输入答题时间, 如图 7 所示, 即可组出一套试卷, 并且可打印试卷, 如图 8 所示。

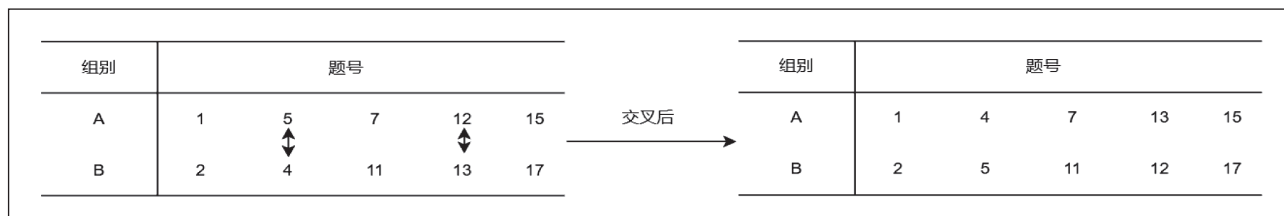


图 3 交叉过程图

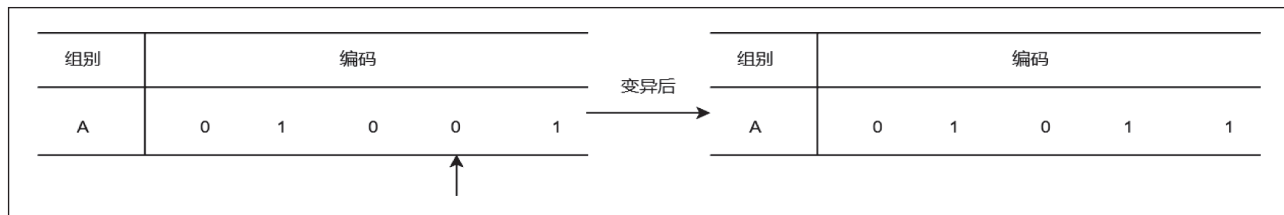


图 4 变异过程图

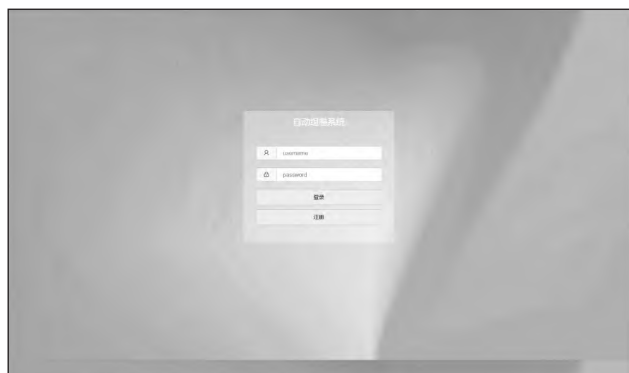


图5 登录/注册界面

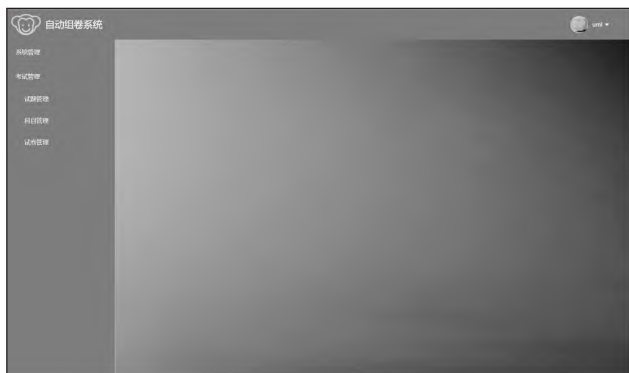


图6 教师界面

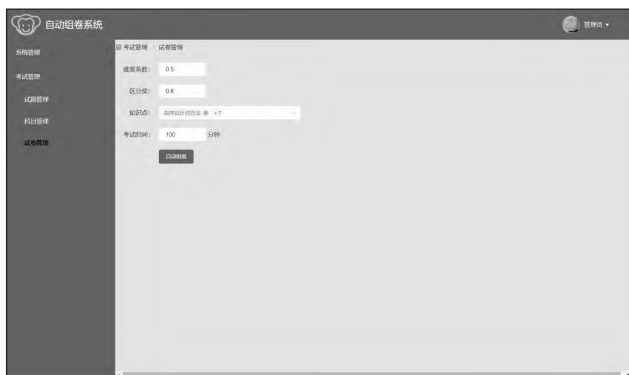


图7 自动组卷界面图



图8 试卷生成界面图

4.3 试题库管理实现

考试系统需要不断的填充题目以完善题库,保证测试的有效性,对试题进行必不可少的增添和修改功能。实现了试题按照题目、答案、难度、知识点、区分度、时间等参数添加相应试题,存储到 MySQL 数据库中并且定期检查和维护试题。

5 结束语

本文的自动组卷系统设计对传统遗传算法的选择、交叉和变异策略进行改进,很大程度上防止“未熟就收敛,局部最优,生成的试题质量偏低”等问题,较好的根据用户需求做到相应要求的试卷,实现组卷速度快且成卷质量高的优点,但还无法做到根据用户动态的调整难度系数,还需一定改进,总体上大大减少了教师在出卷时消耗的时间和精力,为教师和学生提供了方便。

参考文献

- * [1] 彭易波. 基于 B/S 在线考试系统的设计与实现 [J]. 科技传播, 2010, (12): 188+193.
- * [2] 叶欣, 陈磊, 杨小国, 刘柳, 朱接文. 基于 B/S 模式的在线考试管理系统的分析与设计 [J]. 电脑知识与技术, 2021, 17(35): 48-50.
- * [3] 陈尚书, 文斌. 移动端在线考试系统设计与实现 [J]. 软件导刊, 2019, 18(05): 117-121.
- * [4] 李冬, 王化喆. 基于 B/S 模式的通用在线考试系统的架构 [J]. 商丘职业技术学院学报, 2014, 13(05): 29-31.
- * [5] 王璐. 计算机软件数据库的设计原则与方式 [J]. 电子技术与软件工程, 2017, (04): 175.
- * [6] 苏小兵, 李天平. 基于遗传算法的在线考试系统的设计与实现 [J]. 中国教育技术装备, 2010, (36): 104-105.
- * [7] 王琦, 邓秀慧, 蔡玮, 潘磊. 基于遗传算法的在线考试系统设计与实现 [J]. 信息与电脑 (理论版), 2021, 33(21): 58-60.
- * [8] 吴英. 智能在线考试系统中遗传算法组卷的设计和实现 [J]. 制造业自动化, 2011, 33(02): 215-217.
- * [9] 宋巍. 基于 B/S 构架的在线考试系统设计与研究 [J]. 电大理工, 2021, (02): 18-20+25.

通信作者: 王旭。