

基于自动组卷与判卷的在线考试系统设计

周志锋, 童 凌, 王浩茂, 李海燕

(南华大学 计算机科学与技术学院, 湖南 衡阳 421001)

摘 要:网上在线考试目前正在逐步取代传统的考试方式。针对程序设计类课程,实现了一种能够进行自动组卷和判卷的在线考试系统。首先利用遗传算法将试题模拟自然因子通过自然选择机制,自动组成符合要求的试卷。完成考试后,借鉴 ACM 在线测评机制自动对试卷进行评判,从而大大减轻了教师工作量。该系统具有实用性强、使用方便等优点,在实际应用中取得了良好效果。

关键词:程序设计类课程;在线考试系统;遗传算法;自动组卷;自动判卷

DOI:10.11907/rjdk.171014

中图分类号:TP319

文献标识码:A

文章编号:1672-7800(2017)006-0066-04

0 引言

随着计算机技术在教学中的广泛应用,考试方式由以前单一、固定试题和人工阅卷的传统方式,逐渐向多样化、灵活试题和计算机自动评分的无纸化考试过渡。在数字化考试中,使用在线网络考试是当今主流的考试方式。在欧美地区,较有影响力的机构如美国教育考试中心 ETS 举办的美国研究生入学考试 GRE 和计算机文化考试、高级就业计算机科学考试等,其考试形式即完全采用计算机在线进行。除此之外,目前的托福和雅思等标准化英语水平能力测试也采用了网络考试方式。很多职业资格认证机构、教育主管部门以及大型公司招聘应届生时都开发了针对各自目标考生的考试系统,如中国计算机学会计算机职业资格认证(CCF)即采用了网上出题、在线编程提交的方式。在各大高校,在线网络考试也得到了广泛认可,如清华大学、上海交通大学等高校都开发出了符合学校需求的在线考试系统^[1]。

在上述众多考试系统中,大部分考试系统都实现了对客观题的自动评分,部分考试系统还研究出了对于不同种类主观题的评分方法。如在 GRE 考试中,美国教育考试中心根据近年来对作文的自动评分研究机制,推出了对于考试写作部分的自动化评分系统,减轻了阅卷人过去批改大量作文的负担;中国计算机学会在其开展的计算机职业资格认证考试系统中,也对考生提交的代码实行自动评分。然而,实现主观题评分的系统适用性很窄,无法将其项目推广到程序设计类课程考试中。针对这些问题,本文提出一种借鉴 ACM 在线测评机制的自动判卷功能方式,

对程序设计类的主观编程试题进行客观评判。同时,自动组卷也是实现在线考试的重要环节之一,本设计采用遗传算法,按照教师对试卷构成的期望,随机、高效、快速地从试题库中抽取试题,组成一套高质量的试卷。通过自动组卷与判卷两者的结合,实现面向程序设计的高度自动化的考试系统,进一步摆脱了人工干预,提供了一个能更好地展示学生实践和思维能力的平台。

1 系统功能设计

本系统主要涉及 3 类用户:管理员、教师、学生。对于不同用户,则需要设置用户管理模块来维系这种角色分级,并设置不同权限。对于学生角色,系统设置在线答卷模块来管理整个考试过程;对于教师角色,需参与出卷、组卷、发布考试的考试准备流程。因此,设置试卷管理模块来实现对于试卷的常用操作,同时设置题库管理模块,由教师负责出题、管理题库;管理员角色具备所有权限,主要涉及对各种关键实体的管理,因此设置了后台管理模块。系统自动完成的模块是自动组卷模块和自动判卷模块,该模块是本系统的关键模块。本系统的功能架构如图 1 所示。

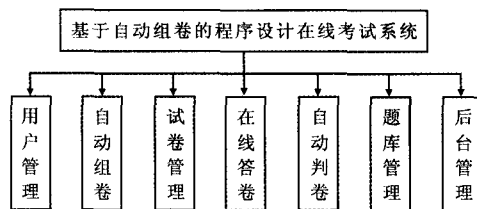


图 1 系统功能结构

基金项目:湖南省大学生研究性学习和创新性实验计划项目(湘教通[2016]283号—331)

作者简介:李海燕(1972—),女,湖南衡阳人,硕士,南华大学计算机科学与技术学院副教授,研究方向为数据库应用、信息安全。

2 核心功能设计

2.1 基于遗传算法的自动组卷

遗传算法的机器自动组卷, 利用教师提供的参数, 灵活建立相应组卷模型, 然后对题目进行筛选, 以自动生成科学合理的试卷。

遗传算法 (Genetic Algorithm, 简称 GA) 是一种借鉴生物进化机制 (适者生存、优胜劣汰) 的随机搜索算法^[2-4]。其通过建立计算模型, 模拟达尔文的遗传选择和生物进化过程, 搜索出符合“环境”要求的最优解, 所以该算法常用来解决多约束下求最优解的问题。本系统在设计应用遗传算法之前, 首先要将试卷解码成“染色体”, 然后抽取一定数量的携带遗传物质的试卷集合初始化为种群, 再根据环境选择、交叉、变异等一些基本的演化操作, 模拟生物进化行为, 筛选出最适合“环境”生存的最优试卷集。

该自动组卷实现流程如图 2 所示。

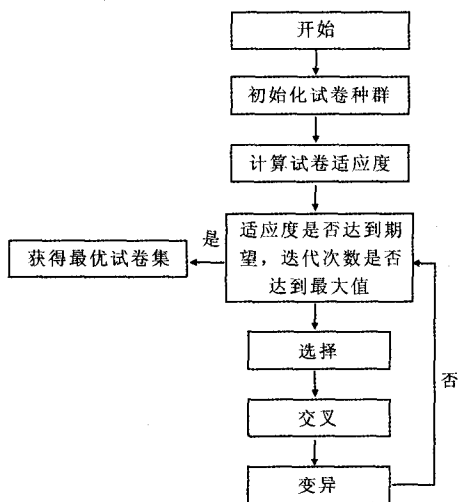


图2 自动组卷流程

2.2 量化试卷

自动组卷方案是根据出卷者给出的约束条件 (本文利用试题总数量 N 、总分 S 、知识点覆盖率 A 、难度系数 P 、题型比例 K 、曝光度权重 G 作为模型参数) 搜索题库中的题目, 以组成符合相关参数的试卷^[1]。从上述几点可以看出, 组卷问题正符合遗传算法的特性, 属于多约束下求解最优解问题。

首先量化试卷最基本的组成因子——试题。

$$S_i = [a_{i1}, a_{i2}, a_{i3}, a_{i4}, a_{i5}, a_{i6}], 0 \leq i \leq m \quad (1)$$

其中, S_i 为试题个体, m 为试题总数, a_{i1} 为题号, a_{i2} 为题型, a_{i3} 为题分, a_{i4} 为知识点, a_{i5} 为难度系数, a_{i6} 为曝光度 (出现在试卷中的次数)。

其中 a_{i5} 应根据实际考试情况进行动态变化, 所以对于客观题, 有:

$$a_{i5} = 1 - R_i / N_i \quad (2)$$

R_i 为该题目答对的人数, N_i 为回答该试题的人数。

而程序设计题:

$$D_i = (S_{iH} - S_{iL}) / N_i (W_{iH} - W_{iL}), a_{i5} = 1 - W_i / a_{i3} + D_i \quad (3)$$

D_i 为该题目的区分度, S_{iH} 为该试题前 27% 的高分段总分数, S_{iL} 为该试题后 27% 的低分段总分数, W_{iH} 为最高分数, W_{iL} 为最低分数, W_i 为中间段 46% 的平均分^[1]。

然后用上述 6 个指标来约束试卷的生成。

(1) 预先给出试题总数量 N 。

(2) 总分为:

$$S = \sum_{j=1}^N a_{j3} \quad (4)$$

(3) 知识点覆盖率:

$$A = M_A / N_A \quad (5)$$

其中 M_A 代表所有题目出现的知识点个数, N_A 则是试卷的知识点个数 ($M_A \leq N_A$)。

(4) 难度系数:

$$P = (\sum_{j=1}^N a_{j5} * a_{j3}) / \sum_{j=1}^N a_{j3} \quad (6)$$

(5) 预先给出题型比例 K 。

(6) 曝光度权重:

$$G = \sum_{t=1}^4 (\sum_{j=1}^{N_t} a_{j5} / \sum_{i=1}^N a_{i5}) \times (\sum_{j=1}^{N_t} a_{j3} / \sum_{i=1}^N a_{i3}) \quad (7)$$

其中 t 代表题目类型, N_t 代表对应题目类型的所有题目个数。

2.2.1 染色体编码与初始群体设计

按照传统的遗传算法问题, 首先将要求解的问题化成 0、1 形式的编码染色体。染色体应该按题型分段, 因为本文针对的是程序设计类考试, 基本题型是单选、多选、判断、程序设计 4 种类型, 所以分段染色 (例如一个数据结构试题有单选 5、多选 5、判断 5、程序设计题 2), 如图 3 所示。

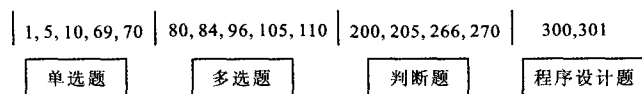


图3 染色体分段

初始化种群则不能完全按照随机产生, 而要根据题型比例、总分、总题数生成, 从而保证了每个初始化个体符合最基本的试卷要求。

2.2.2 适应函数设计

适应度函数是评定个体优良程度的一个指标, 适应函数对遗传算法的方向给予了指导, 不需要任何辅助信息, 从上文所示的几个相关点看, 适应函数 $f(x)$ 与知识点覆盖率、难度系数、曝光度权重有关:

$$f(x) = A \times f_1 - |EP - P| \times f_2 - G \times f_3 \quad (8)$$

其中, f_1 、 f_2 、 f_3 为输入的对应该权重, 可根据实际情况进行改变, EP 代表期望难度系数。其中, 知识覆盖率越高越好, 期望绝对值差越小越好, 曝光度越小越好。

2.2.3 遗传算子设计

(1) 选择算子。选择算子主要用于模拟自然选择的选择概率, 确定该个体是否能够“生存”。本文采用轮盘赌算法, 每个个体的选择概率为 $A_j / \sum_{j=1}^N A_j$ 。其中 A_j 代表第 j 个

个体的适应度,适应度高的个体将有更大机率“生存”下来。

(2)交叉算子。由于染色体是分段编码,所以采用多点交叉,交叉率设为 0.90 ~ 0.95 比较合适。在“交叉”时必须保证互换的题目分数一样,分数相加等于总分,而且不能在交叉点前出现题目,否则此次“交叉”不合法。

(3)变异算子。在遗传中,一般较少发生变异,变异率设置在 0.02 ~ 0.05 范围内比较适宜。

2.3 自动判卷

要实现较为客观地考核学生的编程能力,程序设计类考试的重点不是单纯地将选择、填空等客观题搬到计算机上,而是包含代码编写类试题的考核,这类试题可较为准确、客观地反映学生对程序设计思想和语言的掌握情况。本文根据程序设计试题代码的特点,设计沙盒来真实运行学生编写的代码,并根据得到的结果智能评判该类主观题的得分。

本文设计的自动判卷是面向程序设计类的客观题和主观题。对于客观题,系统可以直接通过与标准答案对比而得出对应分数^[5-6];对于主观题,它的输入和输出也具备唯一性,其中一个输入对应一个输出,利用预先设置的测试数据即输入集,执行输入程序,得出输出集。通过将输出集的每个点与标准答案输出集对比,即可将通过测试点作为该题的评分依据,得到该题的分数。通过该方式可大大减少对主观程序题的主观性判断,而是以更科学的客观依据进行判断。在此期间,自动判卷功能辅助子系统必须保持沉睡状态,以等待唤醒。自动判卷流程如图 4 所示。

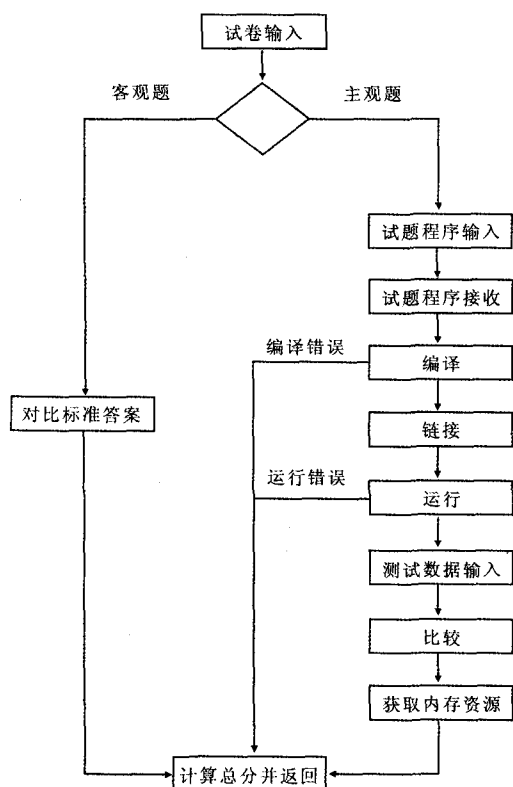


图 4 自动判卷系统体系结构

客观题的评判实现较为简单,本文着重描述主观题的

评判机制。

首先,定义一个信息结构,因为下列函数存在指针 ptr,信息块内容如表 1 所示。

表 1 信息块内容

字段	Id	Language	Text[]
解释	题号	使用的语言	提交的程序

其中每层的解释如下:

(1)试题程序传入。首先利用 Linux 消息队列方式,将试题中源程序的代码、题号、所用语言等信息封装在上述 msg 上,写入 Linux 系统下的消息队当中。通过系统 msgget 函数[`int msgget(key_t key, intoflag);`]创建消息队列,命名为 key,并获取消息队列的标识符,然后调用系统 msgsnd 函数[`int msgsnd(int msqid, const void * ptr, size_t length, int flag);`]将消息存入对应的消息队列中。

(2)试题程序接收。在接受消息时,通过系统 msgget 函数[`int msgget(key_t key, intoflag);`]获取对应 key 的消息队列的标识符,再通过系统 msgrcv 函数[`ssize_t msgrcv(int msqid, void * ptr, size_t length, long type, int flag);`]读取对应标识符消息队列中的一条消息 msg,从而获取队列中的消息内容。

(3)编译。对 GCC、JDK 等编译器进行必要的安全处理,利用 bash 命令对 C++/C、Pascal、Java 等语言程序的程序源码选择对应的编译器进行编译。编译成功则进行下一步,否则返回结果到父进程等待计算。

(4)链接。经过编译成功后,获取对应的 obj 二进制代码,通过调用 ld 链接命令,执行自动链接,产生可执行代码,提供给下一步使用。

(5)运行。通过系统调用子进程 child_process,child_process 运行编译链接后的程序,由父进程监控子进程,一旦子进程崩溃或正常退出,会发送相应的 SIGNAL 给父进程,此时父进程进行相应处理。

(6)测试数据输入。当前位置主要是为了数据的输入和输出位置的转换,通过调用 Linux 的系统函数,将数据输入从标准输入 Console 转换到标准测试文件输入,并将标准输出 Console 重定向到一个临时文件中,供下一步进行比较。

(7)比较。通过上述 4 步以后,转入对比结果阶段。本阶段主要是进行数据对比,通过调用 bash 命令 diff,将该程序通过测试输入产生的输出临时文件与标准输出答案文件进行对比,并得出正确率。

(8)获取内存资源等情况。通过父进程监控子进程,可以通过 wait4 函数,获取进程状态和资源占用情况等,判断是否符合题目要求,一旦检测出非法状况,也当作异常处理,转去计算结果。

(9)计算总分并返回。通过将原试卷主观题与客观题分类后,经过两个子进程得到的总分相加,得到本试卷的分数,并将其存入数据库对应试卷的学生 id 位置。消息队列的交互如图 5 所示。

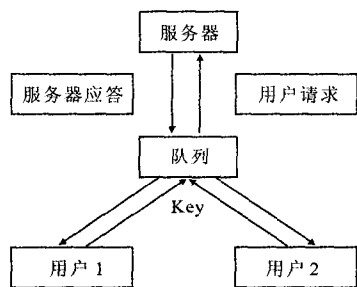


图5 消息队列交互

3 系统实现

本系统采用 B/S 结构模式,利用 Apache 服务器、MySQL 数据库和 SSH 开源框架进行搭建。Apache 服务器、MySQL 数据库是开源软件,所以具有更好的稳定性与安全性。SSH 即 Struts+Spring+Hibernate,利用该体系设计系统,能够为整个系统后期的维护、扩展等提供更多便利,程序的结构脉络更加清晰明了。实际应用结果表明,该系统运行性能良好。

4 结语

本项目旨在让目前高校计算机程序设计类课程考试脱离传统纸质考试的限制。平台采用了一种自动组卷算法,能够大大提高组卷效率以及卷面题目构成的合理性,以符合教师的组卷期望。同时,针对程序设计类课程的特点,引入并实现了特殊的自动裁判评分机制。该在线考试系统可大大减轻教师的工作量,在实践中更加有效地考核并提升学生的编码能力。

参考文献:

- [1] 余颖,李晓昀,左贵启,等. 基于自适应遗传算法的自动组卷策略研究[J]. 南华大学学报:自然科学版,2016(3):61-65.
- [2] 吕苗苗. 基于遗传算法的自动组卷系统的设计与实现[D]. 南京:南京理工大学,2012.
- [3] 张琨,杨会菊,宋继红,等. 基于遗传算法的自动组卷系统的设计与实现[J]. 计算机工程与科学,2012(5):178-183.
- [4] 袁锋. 遗传算法在自动组卷系统中的应用[J]. 山东师范大学学报:自然科学版,2006(1):53-56.
- [5] 杨志伟,曾艳嫒. 基于 Linux 的 ACM 在线评测系统研究[J]. 计算机与现代化,2010(6):166-169.
- [6] 郑传生. 基于 B/S 结构的程序设计竞赛自动测评系统的设计[J]. 计算机与现代化,2007(12):109-111.
- [7] 杨呈永. 基于 B/S 结构的大学英语在线考试系统设计与实现[D]. 成都:电子科技大学,2011.
- [8] 李佳. 《计算机基础》课程上机考试系统的设计与实现[D]. 成都:电子科技大学,2011.
- [9] 颜韵. 基于 B/S 模式的单独招生在线考试系统设计与实现[D]. 成都:电子科技大学,2013.
- [10] 楼玉萍. 基于 B/S 模式的计算机基础考试系统的设计与实现[D]. 长沙:国防科学技术大学,2005.
- [11] 李亦松. 在线考试系统分析与设计[D]. 昆明:云南大学,2012.
- [12] 李敏. 在线考试系统设计与实现[D]. 济南:山东大学,2013.
- [13] 刘楠,孙国道,田贤忠. ACM 在线评判系统设计与实现[J]. 计算机时代,2010(2):34-35,38.
- [14] 赵磊,吴玉培,廖彬. 基于 ACM/ICPC 在线测评系统的设计[J]. 电脑知识与技术,2013(8):5004-5006.
- [15] 张良银. 浅论 C/S 和 B/S 体系结构[J]. 工程地质计算机应用,2006(4):20-23,28.
- [16] 陈鹏. 在线评判系统的设计和实现[D]. 南京:东南大学,2015.
- [17] 余颖,李晓昀,欧阳纯萍. 一种 SSH 框架的在线程序自动评判系统的设计与实现[J]. 南华大学学报:自然科学版,2012(4):65-68.

(责任编辑:黄 健)

The Design of Online Examination System Based on Automated Questions Assemblage and Grading

Abstract: Online Testing System has gradually replaced the paper-based examination in actual institutions nowadays. With regard to the language programming course, We designed and realized an online examination system based on an automated questions assemblage technique. First, the system generated high-quality examination papers by means of adopting genetic algorithm which conducts natural selection according to question dataset. We borrowed the merit of online judge which deals with codes and marks programming-based test papers automatically. With high efficiency and convenience, the system could be applied to other test systems and achieve a great applicability.

Key Words: Language Programming Course; Genetic Algorithm; Automatic Question-assemblage System; Automatic Grading System