

基于 web 的英语在线学习系统^①

王战敏

(西安理工大学 计算机科学与工程学院, 西安 710048)

摘要: 介绍一个基于 web 技术的英语在线学习系统, 详述了系统的设计方案及关键技术问题。首先, 对系统中智能组卷算法与策略进行分析, 以算法的实用性和可操作性需求指标建立一个智能组卷的数学模型, 并设计一种基于改进遗传算法求解该模型的优化方法, 实现通过对试卷指标的组合来随机生成符合约束要求的试卷, 满足不同用户的个性化需求; 其次, 通过融合 Ajax 技术, 实现系统在线测试中对考试进行时间控制, 解决因 B/S 结构引起的“瓶颈”问题, 减缓了数据流量。

关键词: 智能组卷; Ajax; 遗传算法; 数学模型

English Learning and Test System Based on Web Technology

WANG Zhan-Min

(School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China)

Abstract: This paper introduces an English online learning system based on Web technology, and describes the system design schema in detail and the related key technologies. First of all, intelligent test paper generation algorithm and strategies of the system are analyzed, combining practicality and operability of proposed algorithm with practical technical indicators the mathematical model is established, and an improved genetic algorithm is designed for solving the model, a group of test papers are generated randomly by test paper indicators to meet the individual needs of different users. Secondly, by introducing Ajax technology, it realizes the time controll of examinations, resolve the bottlenecks problem which is caused by B/S structure, and ease the flow of data.

Key words: automatic test paper; Ajax; genetic algorithm; mathematical model

网络环境下的在线学习称为 E-Learning 或网络化学习, 主要是利用互联网进行学习、教学等相关活动, 是一种全新的学习方式^[1]。现今, 随着网络技术的飞速发展, 应用网络在线平台实施远程教育、教学已经成为未来发展的必然趋势。

在国外, 在线学习已经被广泛地应用, 美国 Campus Computing Project 的 2008 年调查表明超过 70% 的美国高等教育机构使用了校园范围内在线学习系统^[2], 2009 年调查发现在美国的 5 类大学中, 在线学习平均普及率从 2000 年的 15% 上升到了 2009 年的 55% 左右^[3]。在国内, 随着国内网络教育的兴起, 各高校纷纷开发了自己的网络在线教学平台, 如北京师范

大学的网络教学平台、上海交大的网络考试平台等^[4]。网络教学平台提供给学习者在线学习、在线测试等功能, 在线测试中所涉及的智能组卷技术也日臻成熟^[5]。本文针对目前较为流行的基于网络平台在线学习系统进行了深入的研究与分析。在组卷算法设计中, 建立了按实际需要生成试卷为目标要求的智能组卷问题数学模型, 并在该模型基础上设计了一种对智能组卷问题进行求解的优化方法; 利用融合 Ajax 技术实现对考试时间的控制; 最后进行了系统整体的设计和开发。

1 系统设计

1.1 系统设计目标

^① 基金项目: 陕西省自然科学基金计划(2015JM6355); 陕西省教育厅科学研究计划项目(2013JK1185)

收稿时间: 2015-10-09; 收到修改稿时间: 2016-01-07 [doi:10.15888/j.cnki.csa.005270]

本文研究目标是实现大学英语学习与阶段测试, 为大学生或普通学习者提供平台来衡量和提高英语水平, 学生在完成阶段性学习之后, 可以测试本阶段的学习成果. 在线测试形式实际上是传统考试形式的延伸, 充分利用网络的无限空间特性及数据库技术, 随时随地的对学习者进行测试, 大大简化了传统考试的过程.

1.2 系统设计环境

当前网络平台上的在线学习系统主要是基于 web 技术设计的^[6-8], Web 技术超越了传统的“客户机/服务器”两层结构, 采用了三层体系结构: 用户界面层、事务层、数据接口层.

服务器端, 在 .Net 平台下开发, 连接 SQL Server 2000 数据库服务器, 以及 Web 服务器: IIS5.0. 其中, 若客户机要和数据库交互数据, 则由 Web 服务器通过 ADO.NET 技术访问数据库, 实现与数据库 SQL2000 的交互.

1.3 系统功能设计

基于 web 的英语在线学习系统是将英语学习、测试与评阅结合为一体, 同时涉及用户管理、试题资源管理、试卷管理、分数统计等.

系统包括六大功能模块: 在线学习、在线测试、试题库管理、试卷生成、统计分析、系统管理.

模块具体功能如下:

1) 在线学习模块

此模块为学生提供在线学习功能, 其学习内容是依据英语大纲要求按知识点来划分, 管理员可对知识点进行更新. 在线学习的知识点从后台数据库动态显示到用户界面(表示层). 单击知识点能够显示其必要属性、相关例题和难易指标.

2) 在线测试模块

主要包括: 考生登陆、试卷获取方式(由教师给定出题策略来决定)、考试控制(防止考生多次登陆、防止对已生成试卷进行刷新、控制考试时间等)、故障处理(防止考试过程中因软、硬件故障造成考试异常中断, 系统具备记忆储存考生试卷和答题情况的功能)、自动判卷、评分、存储成绩及统计报表与分析等.

3) 试题库管理模块

该模块是整个系统的基础, 试题库管理模块除包括试题添加、删除、修改、查询等功能外, 允许管理员(老师)将编辑好的试题批量导入题库中.

4) 试卷生成模块

为解决智能组卷问题, 对相关测试理论和教育统计进行了深入研究. 借助遗传算法实现了一种优化的自动组卷策略, 允许命题者按照不同的参数(如知识点、难度等级、题型、区分度、分值、考试时间等), 既按照一定的组卷指标, 生成符合特定要求的试卷. 本系统根据课程特点, 试卷的试题类型分为判断题、单选题、多选题、填空题四种.

5) 统计分析模块

对学生完成测试后的考试成绩进行常规分数统计和试题分析. 试题分析主要分析学生对不同知识点和题型的得、失分情况, 进而可对考生提出建设性参考意见, 便于学生在后续学习时需重点关注的知识点.

6) 系统管理模块

为保证整个系统的正常运行, 所设计系统具有备份功能及系统用户、考生基本信息管理等功能.



图 1 系统主页



图 2 在线测试页面

2 系统采用的关键技术

2.1 基于改进的遗传算法在组卷中的应用

本系统采用基于改进的遗传算法完成自动组卷,

避免了单纯人工组卷的试题重复使用、主观、随机性差等缺点,提高了组卷效率。

2.1.1 组卷设计

目前,智能方法在组卷中已得到广泛应用并取得较好效果。其中,遗传算法(Genetic Algorithms)以其具有自适应全局寻优和智能搜索技术等优点,更好地满足了自动组卷的要求^[9-12]。

设计本文组卷算法时,考虑到任意选取的两套试卷可能有相同的小题,若采用传统遗传算法中杂交算子操作,需对所生成的子个体进行重复小节的检查,算法效率下降,故设计时不考虑杂交算子作用。另外,考虑目标值偏差及约束冲突,本文专门设计的多点变异算子可以满足组卷时收敛速度和组卷质量的目标要求。

2.1.2 自动组卷数学模型

一份高质量试卷,必须考虑多方面因素,主要包括:题目难度、题目区分度、题目曝光度、题目的覆盖(试题章节分布和知识点分布)、题目考察层次、题目重复使用情况、解题时间等。

对一道试题*i*,需要*n*项指标,这里假设为*n*维向量(试题题型 a_{i1} 、总分 a_{i2} 、估时 a_{i3} 、知识点分类 a_{i4} 、难易度 a_{i5} 、区分度 a_{i6} 、能力层次 a_{i7} ...), a_{ij} 相当于第*i*题的第*j*项指标。这样,对1份含*m*题试卷,其试卷题属性可用一个 $m \times n$ 的如下矩阵表示。

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

其中,*m*为试题数,*n*为试题指标数。

试卷的主要指标约束分别描述如下:

1) 试题总分约束

$$\sum_{i=1}^m score_i = K \tag{1}$$

其中, $score_i$ 为第*i*道小题分数, *K* 为试卷总分。类似,根据每道题所属类型可得出每种题型的分数。

2) 知识点约束

知识点约束指试卷中包含的知识点个数以及各个知识点在总试卷中所占的分值比例。

假设题库所有试题按知识点有 *N* 类。针对一套试卷,不仅关注一套试卷中包含多少个知识点,还关注某类知识点的分值在试卷中所占的比重。

$$\sum_{i=1}^m f(kp_i) \geq k \tag{2}$$

其中 $f(kp_i)$ 根据第*i*小题的知识点是否在本试卷中已包含而取值为1或0。当第*i*小题所属知识点已包含则取值为0;不包含则取值为1,且将该知识点加入试卷知识点集合中。

若试卷要求对某几个知识点作为重点考核,可以规定这些知识点的分值在整套试卷中所占的比重。

假设有 N_1 类知识点是重点考核对象,那么需要先统计这 N_1 类知识点得分值,再计算它们在试卷中所占的比重。

对第*i*小题,若它属于第 $j(1 \leq j \leq N_1)$ 类重点考核知识点,那么:

$$sum[j] = sum[j] + score_i \tag{3}$$

计算出所有重点考核知识点后,可计算出每一类重点考核知识点在整套试卷中的比重 $\eta_j(1 \leq j \leq N_1)$ 。

$$\eta_j = sum[j] / K \tag{4}$$

对每一类重点考核知识点,它应满足:

$$\eta_j \geq g_j \tag{5}$$

其中, g_j 为给定的第*j*类重点考核知识点所占比重。

3) 试卷难度约束

难度约束是指试卷的总体难度系数,是将试卷中所有试题的难度加权平均得到。

$$\begin{aligned} |coeff_dif - coeff_dif_0| = \\ \left| \sum_{i=1}^m score_i * coeff_dif_i / (m * K) - coeff_dif_0 \right| \leq \epsilon \end{aligned} \tag{6}$$

其中, $coeff_dif_0$ 为给定试卷的难度系数, ϵ 为一个给定的较小数。上式说明,生成试卷的难度系数与给定难度系数的偏差应控制在一个既定的较小范围内。

4) 总时间约束

完成第*i*题所需时间为 t_i ,则全卷估计用时为:

$$T = \sum_{i=1}^m t_i \tag{7}$$

答题时间约束是指完成解答试卷中所有试题所用最长时间。一般情况下,答题时间满足在正常考试时间范围内即可。

2.1.3 目标函数

自动组卷问题实质上是一个多重约束目标优化问题,且满足约束条件的解不唯一。试卷中各考核知识点所占分数、试卷难度系数、总时间等约束条件可以

存在一定的误差. 在实际应用中, 各约束条件的重要性是不同的, 因此, 目标函数就取各项误差的加权和. 目标函数 F 可以表示为:

$$F = \sum_{i=1}^n f_i x_i \quad (8)$$

为了不使各项误差相互抵消, 实际值与要求值的误差均取绝对值. x_i 表示第 i 个约束条件的权值, x_i 通常由专家经验或试验给出, $0 \leq x_i \leq 1$.

$$\sum_{i=1}^n x_i = 1 \quad (9)$$

由(8)式可知, 目标函数 F 值越小, 即误差越小, 由算法得到问题的解就越优, 即所生成试卷越接近用户的需求.

2.1.4 算法描述

组卷算法步骤如下:

Step 1: 初始化种群个体;

Step 2: 计算个体的目标值(适应值);

Step 3: 若满足停止条件, 则停止计算; 否则继续执行下一步;

Step 4: 执行变异算子操作;

Step 5: 计算新产生个体的目标值(适应值);

Step 6: 产生下一代种群, 转 Step 2.

以下对编码、算子设计、选择策略等方面进行描述.

1) 编码方式及编码位数确定

编码采用位编码. 由试卷的总分值、题型及每类题型分值确定每类题型包含的小题数, 将所有小题数相加即得位编码长度.

2) 变异算子设计

选取的个体对随机的几个点按一定概率进行替换. 选取替换小题的原则为: a) 使目标函数偏差减小; b) 排除已选小题.

3) 选择策略

将经变异算子操作产生的小题与选取的父代个体的目标值进行对比, 优胜者进入下一代种群. 另外, 在每一代开始, 最优个体直接进入下一代种群.

2.2 考试时间控制设计

Ajax 由 HTML、JavaScript TM 技术、DHTML 和 DOM 组成. Ajax 是把 JavaScript 技术和 XMLHttpRequest 对象放在 Web 表单和服务器之间, 将传统的 Web 界面转化成交互性的 Ajax 应用程序.

本系统依靠客户浏览器与 Web 服务器相互配合共

同完成对考试时间的控制. 考试计时和试卷定时保存功能的实现思路类似, 均需使用 Ajax Timer 控件实现. 下一节介绍定时保存功能.

客户浏览器端解决剩余考试时间的计算、显示和考试时间结束时自动提交答卷内容. 服务器端解决从数据库中读取试卷开考时间, 以及当考生刷新页面或意外关闭浏览器时, 仍能使考试时间连续计算等问题.

以考试时间控制为例, 代码如下:

Aspx:

```
<form id="form1" runat="server">
  <asp:ScriptManager ID="ScriptManager1"
    runat="server" />
  <asp:TimerID="Timer1" runat="server"
    Interval="10000"
    OnTick="Timer1_Tick"></asp:Timer>
  <div>
    <asp:UpdatePanel ID="upFir" runat="server"
      UpdateMode="Conditional">
      <Triggers>
        <asp:AsyncPostBackTrigger ControlID="Timer1"
          EventName="Tick" />
      </Triggers>
      <ContentTemplate>
        <asp:Label ID="lbljs" runat="server" Text="upFir
          not refreshed"></asp:Label>
      </ContentTemplate>
    </asp:UpdatePanel>
  </div>
</form>
```

2.3 试卷定时保存功能设计

在数据库设计时, 考试信息表中应该包含考生标识(string, 考号)、开始答题时间(DateTime)、答题结束时间(DateTime)、已交卷(bool)四个字段(其他字段不做详述). 在客户端方面, 答题页面 onload 的时候调用一个 javascript 函数开始计时, 函数循环执行以便随时保存已用时间, 设置循环周期为 30 秒, 相关 js 代码如下:

```
<script language="JavaScript">
var myTimeOut=30; //可用时间, 单位为分钟
var myPassTime=0; //已用时间, 单位为分钟
window.attachEvent("onload",myTimer);
//绑定到 onload 事件
```

```
function myTimer(){
if(myPassTime<myTimeOut)
{
//已用时间是否小于可用时间
myPassTime+=1; //保存客户端已用时间
//显示友好界面的计时信息
}
else{ //执行强制交卷
//可以通过模拟点击一个 linkbutton 来实现,
//在服务器端把强制交卷的代码先在 linkbutton 的
Click 里 }
window.setTimeout("myTimer()",60000);
//一分钟循环一次;
</script>
```

考试页面放置一个 linkbutton, 在该控件的 Click 里编写强制交卷代码, 以供 js 模拟点击的时候执行。

2.4 出卷方式设计

试卷输出方式选择全部试题一次呈现的方式, 即将 html 中数据输出到 word 中, 然后对其进行调整, 可提高试题输出效果, 同时可以对试卷进行微调, 提高试卷质量, 采用此种方式显示试题过程如下:

1) 定义文档类型、字符编码

```
Response.Clear();
```

```
Response.Buffer=true;
```

```
Response.Charset="utf-8";//filename=FileFlow.xls
```

指定输出文件的名称, 注意其扩展名和指定文件类型相符, 可以为: .doc||.xls||.txt||.htm

```
Response.AppendHeader("Content-Disposition","on
line;filename=FileFlow.xls");
```

```
Response.ContentEncoding=System.Text.Encoding.
GetEncoding("utf-8");//Response.ContentType 指定文件
类型可以为 application/ms-excel||application/ms-
```

word||application/ms-txt ||application/ms-html||或其他浏览器可直接支持文档

```
Response.ContentType="application/ms-excel";
```

```
this.EnableViewState=false;
```

2) 定义一个输入流

```
System.IO.StringWriter oStringWriter=new
```

```
System.IO.StringWriter();
```

```
System.Web.UI.HtmlTextWriter
```

```
oHtmlTextWriter=new
```

```
System.Web.UI.HtmlTextWriter(oStringWriter);
```

3) 将目标数据绑定到输入流输出

```
this.RenderControl(oHtmlTextWriter);//this 表示输
出本页, 也可以绑定 datagrid 或其他支持
obj.RenderControl()属性的控件
```

```
Response.Write(oStringWriter.ToString());
```

```
Response.End();
```

3 结语

本文介绍了基于 Web 的英语在线学习系统的设计思路和实现方法。系统可方便学生自主学习、测试并及时得到反馈信息, 以便有计划、有步骤、系统地调整学习进度, 提高网络在线学习效率; 同时, 教师(或管理员)可通过系统对学生成绩进行分析统计, 有针对性地为学生提供指导。下一步工作重点是进一步完善对题库及试题方面的统计分析功能, 通过分析来指导更新题库, 使题库更具科学性和有效性, 以提高组卷质量及在线学习效果。

参考文献

- 1 常娜. 基于网络英语在线学习系统的研究与开发. 信息安全与技术, 2012, (6): 72-73.
- 2 Campus Computing. The 2008 Campus computing survey. Encino, CA: The campus computing project. <http://www.campuscomputing.net/survey-summary/2008>.
- 3 Campus Computing. The 2009 Campus computing survey. Encino, CA: The campus computing project. <http://www.educause.edu/E09+Hybrid/EDUCAUSE2009FacetoFaceConferen/The2009CampusComputingSurvey/176028>.
- 4 李娅. 基于 ASP.NET 技术的开放式在线学习及考试系统的研究与实现[学位论文]. 青岛: 中国海洋大学, 2012.
- 5 徐涛. 基于遗传算法的组卷研究与设计. 软件导刊, 2014, 13(12): 109-111.
- 6 吴琦. 基于 Web 在线学习系统的设计与实现. 长江工程职业技术学院学报, 2006, 23(3): 45-46.
- 7 陈明. 基于 B/S 结构的高校英语在线学习系统的设计开发[学位论文]. 昆明: 云南大学, 2013.
- 8 杨晋吉, 刘太根. 网络课程中基于 Web 的在线测试系统的研究与实现. 计算机应用研究, 2007, (2): 230-232.
- 9 孟祥娟, 王俊峰, 曹锦梅. 利用遗传算法实现试题库自动组卷问题. 计算机系统应用, 2010, 19(1): 180-184.
- 10 鲁萍, 王玉英. 多约束分级寻优结合预测计算的智能组卷策略. 计算机应用, 2013, 33(2): 342-345.
- 11 马德良, 陆昌辉, 王小乐. 基于改进遗传算法的智能组卷方法. 计算机应用, 2009, 29(7): 1884-1886.
- 12 邓新秀, 张敏, 葛斌. 混合遗传算法及其在智能组卷中的应用. 计算机系统应用, 2008, (7): 90-92.