基于基因表达式编程的计算机组卷算法研究

韩 啸[1],毕 波[1],唐锦萍[2]
(1. 东北石油大学 数学与统计学院,黑龙江 大庆 163000;
2. 黑龙江大学 数据科学与技术学院,黑龙江 哈尔滨 150080)

摘 要:随着计算机技术的发展,传统的手动组卷方法难以满足新时代下的各种领域的需求,为解决传统手动组卷在性能、速度、题型分配等方面的缺陷,基于计算机技术的智能组卷问题日渐变为人们关注的问题。然而目前的组卷算法存在成功率低、计算时间久、知识点覆盖不完整、难度系数难以把握、生成的试卷难以满足要求等问题,导致了生成的试卷无法达到理想的效果。为改善上述问题,引入了基因表达式编程算法,通过使用适当的遗传算子,采用线性定长的编码方式,构造了新的智能组卷方法,避免了传统组卷算法成功率低以及适应性差等问题,解决了多约束条件下试卷的分数分配、章节分配、难度等一系列问题。实验证明,该算法有着较高的效率,能够快速地生成满足要求的试卷。

关键词:基因表达式编程算法;智能组卷;数学模型;遗传算子;智能算法
中图分类号:O29 **文献标识码**:A **文章编号**:1673-629X(2020)05-0154-06
doi:10.3969/j.issn.1673-629X.2020.05.029

Research on Computer Test Paper Generation Algorithm Based on Gene Expression Programming

HAN Xiao[1], BI Bo[1], TANG Jin-ping[2]
(1. School of Mathematics and Statistics, Northeast Petroleum University, Daqing 163000, China;
2. School of Data Science and Technology, Heilongjiang University, Harbin 150080, China)

Abstract: With the development of computer technology, the traditional manual test paper generation method can't meet the needs of various fields in the new era. In order to solve the shortcomings of traditional manual test paper generation in performance, speed, assignment of questions and other aspects, the intelligent test paper generation problem has gradually attracted more interest. However, there are some problems in the current test paper generation algorithm, such as low success rate, long calculation time, incomplete coverage of knowledge points, difficulty coefficient control, and the generated test paper can hardly meet the requirements. As a result, the generated papers can't achieve the desired results. In order to solve above problems, a new intelligent test paper generation method is constructed by using gene expression programming algorithm. By introducing appropriate genetic operators and linear fixed length coding method, it avoids the shortcomings of premature and fast convergence of traditional test paper generation algorithm, and solves a series of problems such as score allocation, chapter allocation and difficulty of test paper under multi-constraints. Experiment shows that the proposed algorithm has high efficiency and can quickly generate test papers that meet the requirements.

Key words: gene expression programming; intelligent test paper generation; mathematical model; genetic operator; intelligent algorithm

0 引 言

随着当代教育技术及科学技术的迅速发展,传统的教学与考试方式已渐渐无法适应现代化的教学要求,因此网络化的教学考试已渐渐步入人们的视野。网络化的教学考试的关键在于能够快速智能地组成试卷,即在计算机上设置试题库,并通过适当的算法挑选出试题并构成试卷,这类算法相较于传统的人工组卷来说更为高效[1]。

通过网络化的考试模式,不仅可以减轻教师及学校的工作量,而且更能激发学生的学习热情、提高学生的学习效率。

目前的在线考试系统已经可以实现智能组卷、在

收稿日期:2019-06-27 **修回日期**:2019-10-29 **网络出版时间**:2020-01-10
基金项目:国家自然科学基金(11701159)
作者简介:韩 啸(1995-),女,硕士,CCF 会员(C4717G),研究方向为智能算法;毕 波,博士,副教授,通信作者,研究方向为智能算法、计算机视觉。
网络出版地址:http://kns.cnki.net/kcms/detail/61.1450.TP.20200110.1123.042.html

线答题、在线阅卷等一系列功能。但由于一些现实条件的约束,各类在线考试系统在智能组卷方面还不够成熟;大批量产生试卷时速度慢、质量差是现阶段组卷算法的主要缺点。

针对现阶段在线考试系统组卷速度慢、质量差、题型不均衡等不合理的问题[2],文中首先给出了组卷问题的数学模型,将组卷问题归结为一个约束非线性优化问题,并提出了基于基因表达式编程的自动组卷算法,将该算法用于求解组卷优化问题,实验证明该算法有着较高的计算效率和良好的实用性。

1　常见智能组卷算法

常见的智能组卷算法包括以下 4 种:

(1)基于随机算法的组卷算法。

随机组卷算法是依据已经确定的试卷标准进行随机抽取试题[3],在组卷时,利用二项分布函数建立数学模型,按照试题的类型、试题的难度、知识点等约束条件进行选择。

该算法结构简单、易于理解,常常应用于容量较小的试题库组卷系统中[4]。当系统中试题过多时,组卷的时间会变长且成功率低。

(2)基于回溯算法的组卷算法。

回溯算法实际上是一个类似于枚举的搜索尝试过程[5],是一种选优搜索的方法,按选优条件向前搜索以达到目标,在搜索尝试的过程中寻找问题的解,当不满足求解条件时就进行“回溯”[6],尝试别的路径,重新抽取试题。

在试题种类少、数量少的情况下,回溯算法优于随机算法[7]。但当试题种类增多、试题量增加后,使用回溯算法生成试卷的速度明显变慢[8],且选取的试题不能保证是最优的,所以在智能组卷中回溯算法很少被使用。

(3)基于遗传算法的组卷算法。

遗传算法(genetic algorithm,GA)是一类借鉴生物界的“适者生存,优胜劣汰”的遗传机制演化而来的随机化搜索方法[9]。该算法采用简单的编码技术对试题库中的试题进行编码,生成初始的种群。然后通过复制、交换、突变三种操作产生下一代新的种群。并且一代一代朝向适应度函数的方向发展,直至产生满足出卷要求的试卷。

遗传算法在智能组卷中是比较灵活的方法,适用于具有多约束条件的组卷问题[10]。但在组卷过程中由于试题量的增加,“早熟”的问题逐渐显现出来,所以如何提高算法执行效率,解决“早熟”问题成为了重要的研究问题。

(4)基于粒子群优化算法的组卷算法。

粒子群优化算法是模拟鸟群觅食行为而发展起来的一种基于群体协作的随机搜索算法[11]。粒子群算法首先从试题库中随机抽取若干套试卷,构成最初的粒子群(每个粒子代表一套试卷)。然后,运用适应度函数计算每个粒子的值,判断其是否符合要求。如果不符合继续迭代,直至选取到符合要求的试卷。

粒子优化种群算法在计算大规模试题库系统时编码简单,搜索速度更快,效率更高,但是不适宜处理离散的优化问题,且在选择遗传算子时比较麻烦。

2　组卷问题的数学模型

组卷是指从一个试题库中在满足给定规则的情况下,根据确定的参数及算法,从试题库中筛选组合出一套满足要求的、科学的、合理的试卷。

进行组卷时需要给定以下的约束条件:试题数量、总分、题型个数、题型分数等。首先用 x_n 来表示题型,用 y_n 来表示章节,$k_i=1$ 表示选择某题,$k_i=0$ 表示不选择某题,如表 1 所示。

表 1　试卷的约束条件

题型及章节	字母表示	分值(分)	个数(个)	备注
选择题	x_{i1}	3	10	
填空题	x_{i2}	2	10	
判断题	x_{i3}	1	10	
大题	x_{i4}	10	4	
第一章	y_{i1}	20		必有大 题
第二章	y_{i2}	20		必有选择题
第三章	y_{i3}	20		
第四章	y_{i4}	40		

根据表 1,可列出约束条件:

①试题数量约束。

$$\sum_{i=1}^{n} k_i = 34 \tag{1}$$

其中,$k_i=\begin{cases}1 & \text{选择第 } i \text{ 题}\\ 0 & \text{不选第 } i \text{ 题}\end{cases}$,式(1)表示试卷共有 34 道试题。

②总分约束。

$$\sum_{i=1}^{n} k_i(a_1x_{i1}+a_2x_{i2}+a_3x_{i3}+a_4x_{i4}) = 100 \tag{2}$$

其中,$k_i=\begin{cases}1 & \text{选择第 } i \text{ 题}\\ 0 & \text{不选第 } i \text{ 题}\end{cases}$,

$$x_{in}=\begin{cases}n=1 & \text{表示第 } i \text{ 为选择题}\\ n=2 & \text{表示第 } i \text{ 为填空题}\\ n=3 & \text{表示第 } i \text{ 为判断题}\\ n=4 & \text{表示第 } i \text{ 为大题}\end{cases},$$

$$a_t=\begin{cases}t=1\text{ 时} & a_t=3\text{ 表示选择题每题 3 分}\\ t=2\text{ 时} & a_t=2\text{ 表示选择题每题 2 分}\\ t=3\text{ 时} & a_t=1\text{ 表示选择题每题 1 分}\\ t=4\text{ 时} & a_t=10\text{ 表示大题每题 10 分}\end{cases},$$

式(2)表示试卷总分为 100 分。

③题型个数约束。

$$\sum_{i=1}^{n}k_ix_{i1}=10 \tag{3}$$

$$\sum_{i=1}^{n}k_ix_{i2}=10 \tag{4}$$

$$\sum_{i=1}^{n}k_ix_{i3}=10 \tag{5}$$

$$\sum_{i=1}^{n}k_ix_{i4}=4 \tag{6}$$

式(3)表示被选中的试题中选择题的个数为 10 个,式(4)表示被选中的试题中填空题的个数为 10 个,式(5)表示被选中的试题中判断题的个数为 10 个,式(6)表示被选中的试题中大题的个数为 4 个。

④每章所有题型的总分约束。

$$\sum_{i=1}^{n}k_iy_{i1}(a_1x_{i1}+a_2x_{i2}+a_3x_{i3}+a_4x_{i4})=20 \tag{7}$$

$$\sum_{i=1}^{n}k_iy_{i2}(a_1x_{i1}+a_2x_{i2}+a_3x_{i3}+a_4x_{i4})=20 \tag{8}$$

$$\sum_{i=1}^{n}k_iy_{i3}(a_1x_{i1}+a_2x_{i2}+a_3x_{i3}+a_4x_{i4})=20 \tag{9}$$

$$\sum_{i=1}^{n}k_iy_{i4}(a_1x_{i1}+a_2x_{i2}+a_3x_{i3}+a_4x_{i4})=40 \tag{10}$$

其中, y_{im} 表示第 i 题在第 m 章,

$$y_{im}=\begin{cases}m=1 & \text{表示第 } i \text{ 在第一章}\\ m=2 & \text{表示第 } i \text{ 在第二章}\\ m=3 & \text{表示第 } i \text{ 在第三章}\\ m=4 & \text{表示第 } i \text{ 在第四章}\end{cases}。$$

式(7)表示第一章被选中的所有试题的总分,第一章试题占 20 分。式(8)表示第二章被选中的所有试题的总分,第二章试题占 20 分。式(9)表示第三章被选中的所有试题的总分,第三章试题占 20 分。式(10)表示第四章被选中的所有试题的总分,第四章试题占 40 分。

⑤某章必有某种数量题型约束。

$$\sum_{i=1}^{n}k_ix_{i4}y_{i1}=1 \tag{11}$$

$$\sum_{i=1}^{n}k_ix_{i1}y_{i2}=2 \tag{12}$$

$$\sum_{i=1}^{n}k_ix_{i3}y_{i3}=1 \tag{13}$$

式(11)表示第一章被选中的题必须包含 1 个大题,式(12)表示第二章选中的题必须包含 2 个选择题,式(13)表示第三章选中的题必须包含 1 个判断题。

3 基因表达式编程算法在组卷算法中的应用

3.1 基因表达式编程算法的介绍

基因表达式编程算法(gene expression programming,GEP)是融合和遗传算法(GA)和遗传规划(genetic programming,GP)的特点演化而成的算法[12]。

首先,在基因型上(句法上),基因表达式编程算法继承了 GA 的定长线性编码的特点,以 K-表达式的形式表示[13]。GEP 的染色体可以由单个或多个 GEP 基因组成。GEP 的基因是由基因的头部 h 和尾部 t 组成,基因的头部既可以包含函数符又可以包含终结符,而基因的尾部只能包含终结符。其中头部的长度是根据具体的问题来决定的,基因尾部的数量由头部数量决定,公式为:

$$t=h\times(n-1)+1 \tag{14}$$

其中, n 表示所需变量数量最多的函数的参数个数(例如在常见的数学运算中,绝对值和三角函数运算 $n=1$,而大多数的算数运算如加减乘除 $n=2$)。GEP 采用这种将头部和尾部分开的基因形式,一方面整个基因的结构在设定了函数集合和头部长度之后就能够确定;另一方面,整个基因在该公式的前提下一定能够保证结构上的正确性,而不用担心会产生任何非法的个体。

其次,在表现型上(语义上),GEP 继承了 GP 的树形结构,称为表达式树。表达式树上的叶节点表示运算数,也就是终结符。分支节点表示运算符,也就是函数符。尽管 GEP 采用固定长度的基因,但其对应的表达式树形式却千差万别。但无论染色体怎样变异,这些表达式树都是有效的。GEP 的基本流程框架如图 1 所示[14],具体步骤如下:

①设置控制参数,选择函数集合并设置终结符;

②创建初始种群,包含若干个代表不同解答方案的个体;

③解析 GEP 的基因解码,计算每个个体的适应度值用以评价种群;

④若达到设定最大进化代数或计算精度,则进化结束,否则执行步骤⑤;

⑤根据“适者生存”原则,实施最优化保存策略;

⑥遗传操作产生下一代:分别利用选择、变异、倒串、插串(包括 IS 插串、RIS 插串、基因插串)、重组(包括单点重组、两点重组、基因重组)、随机变量变异对群体实施遗传操作;

⑦生成新种群,并计算各个个体的适应度函数值,

然后返回步骤④。

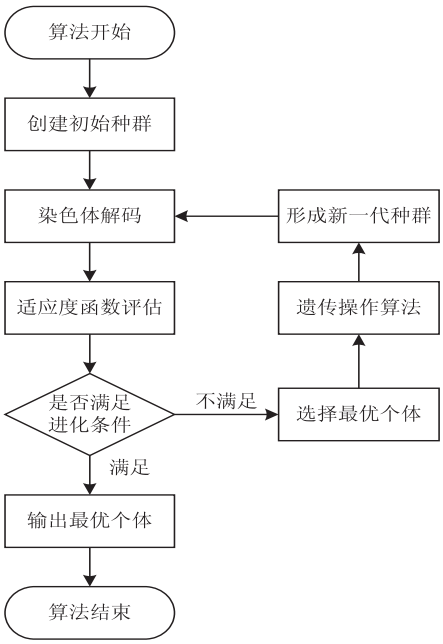

图 1　GEP 算法流程

3.2　基因表达式编程的适应度函数

基因表达式编程的适应度函数是进化算法与现实问题的接口,同时也是算法演化过程的主要驱动源泉。适当的适应度函数可以评价算法所求得的问题解的优劣。适应度函数的计算一般按照以下步骤进行:

①对个体的编码串解码,得到个体的表现型;

②由个体的表现型计算出相应个体的目标函数值;

③依据最优化问题的类型,由目标函数值按照一定的转换规则求出个体的适应度函数。

通过上述组卷问题的数学模型可以得到组卷问题关于基因表达式编程算法的适应度函数为:

$$W = 1\ 000 - (N + Q + M + P) \tag{15}$$

当 $W=1\ 000$ 时,达到最佳的适应度,即当前的最优解满足所有约束条件。其中 N 为计算求得的总分与给定 100 分之间的差距加权,权重为 $\frac{1}{2}$,有公式:

$$N = \frac{1}{2}\left|100 - \sum_{i=1}^{n} k_i(a_1x_{i1} + a_2x_{i2} + a_3x_{i3} + a_4x_{i4})\right| \tag{16}$$

其中,Q 为计算求得的题型个数与所给个数之间的差距加权,权重为 $\frac{1}{4}$,公式为:

$$Q = \frac{1}{4}\sum_{j=1}^{4} P_j \tag{17}$$

当 $j = 1,2,3$,且 $\sum_{i=1}^{n} k_i x_{ij} \neq 10$ 时,

$$P_j = \left|10 - \sum_{i=1}^{n} k_i x_{ij}\right| \tag{18}$$

当 $j = 4$, $\sum_{i=1}^{n} k_i x_{ij} \neq 4$ 时,

$$P_j = \left|4 - \sum_{i=1}^{n} k_i x_{ij}\right| \tag{19}$$

其中,M 为计算求得的每章题型的总分与所给个数之间的差距加权,权重为 $\frac{1}{8}$,公式为:

$$M = \frac{1}{8}\sum_{j=1}^{4} P_j \tag{20}$$

当 $j = 1,2,3$,且 $\sum_{i=1}^{n} k_i y_{ij}(a_1x_{i1} + a_2x_{i2} + a_3x_{i3} + a_4x_{i4}) \neq 20$ 时,

$$P_j = \left|20 - \sum_{i=1}^{n} k_i y_{ij}(a_1x_{i1} + a_2x_{i2} + a_3x_{i3} + a_4x_{i4})\right| \tag{21}$$

当 $j = 4$, $\sum_{i=1}^{n} k_i y_{ij}(a_1x_{i1} + a_2x_{i2} + a_3x_{i3} + a_4x_{i4}) \neq 40$ 时,

$$P_j = \left|40 - \sum_{i=1}^{n} k_i y_{ij}(a_1x_{i1} + a_2x_{i2} + a_3x_{i3} + a_4x_{i4})\right| \tag{22}$$

其中,P 为计算求得的每章必有题型与所给个数之间的差距加权,权重为 $\frac{1}{8}$,公式为:

$$P = \frac{1}{8}\sum_{j=1}^{3} P_j \tag{23}$$

当 $j = 1$, $\sum_{i=1}^{n} k_i x_{i4} y_{i1} \neq 1$ 时,

$$P_j = \left|1 - \sum_{i=1}^{n} k_i x_{i4} y_{i1}\right| \tag{24}$$

当 $j = 2$, $\sum_{i=1}^{n} k_i x_{i1} y_{i2} \neq 2$ 时,

$$P_j = \left|2 - \sum_{i=1}^{n} k_i x_{i1} y_{i2}\right| \tag{25}$$

当 $j = 3$, $\sum_{i=1}^{n} k_i x_{i3} y_{i3} \neq 1$ 时,

$$P_j = \left|1 - \sum_{i=1}^{n} k_i x_{i3} y_{i3}\right| \tag{26}$$

3.3　基因表达式编程的遗传算子

GEP 采用的是线性等长的编码方式,所以基因表达式编程的遗传操作类似于遗传算法[15]。GEP 在进行各种遗传操作时满足“保持基因的长度不变,尾部只能出现终结符”的原则,因此,经过遗传操作后的子代染色体仍然是合法的。GEP 区别于遗传编程的一个特点是:GEP 的一个染色体可以多次进行不同的遗传操作,也可以不进行任何的遗传操作。

GEP 的遗传算子包括:选择、变异、倒串、插串(包括 IS 插串、RIS 插串、基因插串)、重组(包括单点重组、两点重组、基因重组)等算子[16]。对于组卷算法,

只可用到变异和重组(单点重组、两点重组)。

①变异。

在 GEP 中,变异是维持种群多样性的主要方法,变异可以发生在染色体上的任何位置,染色体的变异过程如图 2 所示,3-8 位为尾部,变异位置用下划线表示:

父代:

0	1	2	3	4	5	6	7	8
7	32	54	87	10	5	1	3	21

子代:

0	1	2	3	4	5	6	7	8
7	32	66	87	10	5	1	3	21

图 2 GEP 染色体变异过程

②重组。

重组包括了单点重组、两点重组和基因重组[17]。单点重组,顾名思义是在组卷染色体中寻找到两个父代染色体,然后在两个父代染色体中随机选取一个交换位置,然后互相交换交换位置后面的染色体部分,从而形成两个新的子代染色体。染色体的单点重组过程如图 3 所示,3-8 位为尾部,重组部分用下划线表示。

父代 1:

0	1	2	3	4	5	6	7	8
7	32	54	87	10	5	1	3	21

父代 2:

0	1	2	3	4	5	6	7	8
8	56	48	77	18	2	15	37	61

子代 1:

0	1	2	3	4	5	6	7	8
7	32	54	87	18	2	15	37	61

子代 2:

0	1	2	3	4	5	6	7	8
8	56	48	77	10	5	1	3	21

图 3 GEP 染色体单点重组过程

图 3 中选择父代 1 与父代 2 的 4 号基因位置为交换点,子代 1 是父代 1 交换父代 2 中的部分得到的,子代 2 是父代 2 交换父代 1 中的部分得到的。

两点重组就是首先寻找到两个组卷父代染色体,然后在两个染色体上随机选择两个交换位置,然后互相交换两个位置间的部分,以达到产生两个新的子代染色体的目的。染色体的两点重组过程如图 4 所示,3-8位为尾部,重组部分用下划线表示。

图 4 中子代 1、子代 2 是父代 1、父代 2 以 2 号位置和 4 号位置为重组点交换中间部分得到的。

4 仿真结果与分析

文中构造了一个题目总数为 1 000 道题的题库进行实验,其中 1 ~ 300 题为选择题,301 ~ 600 题为填空题,601 ~ 800 题为判断题,801 ~ 1 000 题为大题。

父代 1:

0	1	2	3	4	5	6	7	8
7	32	54	87	10	5	1	3	21

父代 2:

0	1	2	3	4	5	6	7	8
8	56	48	77	18	2	15	37	61

子代 1:

0	1	2	3	4	5	6	7	8
7	32	48	77	18	5	1	3	21

子代 2:

0	1	2	3	4	5	6	7	8
8	56	54	87	10	2	15	37	61

图 4 GEP 染色体两点重组过程

实验环境为:Windows7 操作系统,CPU 为 i5-4210 M,内存为 4 G,使用 python3.7 编程实现。

程序中变异、单点重组、两点重组的概率为 $P = (0.28, 0.41, 0.31)$,则实验使用参数如表 2 所示。适应度函数越接近 1 000 则得到的结果越好,当适应度函数为 1 000 时,达到最佳组卷效果。

表 2 进化中使用的参数

参数项	参数值
试题个数	1 000
种群大小	200
染色体长度	34
变异率	0.28
单点重组率	0.41
两点重组率	0.31
N 的权重	$\frac{1}{2}$
Q 的权重	$\frac{1}{4}$
M 的权重	$\frac{1}{8}$
P 的权重	$\frac{1}{8}$
适应度函数	$W = 1\,000 - (N + Q + M + P)$
曝光度	$S \leqslant 5$

迭代次数与适应度函数的变化如图 5 所示。通过图 5 可以看出,由于初始化种群时得到的种群比较理想,故开始时最佳适应度函数就已经在 980 ~ 1 000 的范围内波动,试卷的平均适应度则在 870 ~ 950 的范围内波动,经过 496 次迭代后,最佳适应度函数达到 1 000,即组卷达到最好效果。

通过试验输出的试卷中的试题在题库中的编号为

[538,61,748,683,275,330,50,979,510,774,705,168,945,759,471,589,27,77,835,798,679,454,53,682,663,177,321,158,273,68,691,219,215,806],也就是说这套试题符合给定的约束条件,该组卷算法成功地完成了组卷任务。

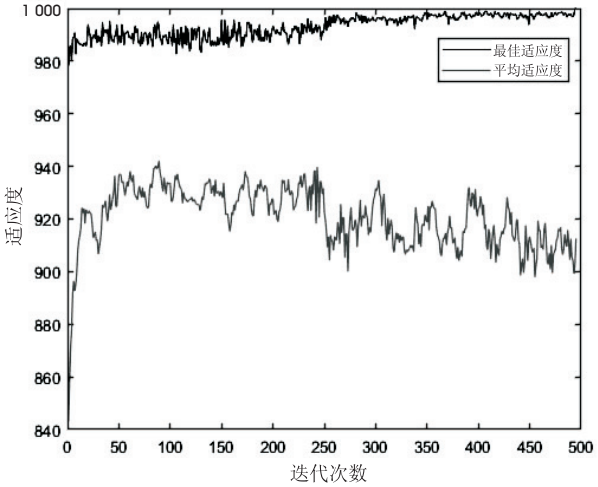

图 5　最佳适应度与平均适应度

5　结束语

通过试验验证,基于基因表达式编程算法的智能组卷算法在给定试卷约束条件的前提下,构建出组卷问题的数学模型及相应的适应度函数,便可以很快地计算得到理想的结果,即满足条件的试卷,能有效地防止“早熟”,且简单快捷,具有较高的实用价值。

参考文献:

[1] 杨晓吟. 基于蚁群优化遗传算法的智能自动组卷算法研究[J]. 现代电子技术,2018,41(21):121-123.

[2] 唐　静,舒小松. 基于回溯法的智能组卷算法的研究[J]. 信息与电脑,2018(19):69-71.

[3] 徐　波,胡玉涛,刘　昊. 医药考试系统 A1 及 A2 题型的随机组卷算法研究[J]. 计算机与网络,2016,42(21):63-65.

[4] 周文君. 一种高效的随机组卷算法的设计[J]. 电脑与电信,2016(7):49-50.

[5] 黄南泰,陶志穗,陈礼源,等. 一个多重约束目标的问题求解——物理题库智能组卷系统[J]. 华南理工大学学报:自然科学版,1990,18(1):11-17.

[6] 龚完全. 基于最小回溯代价的智能组卷算法[D]. 长沙:湖南大学,2005.

[7] WANG Zhigang,ZENG Yurong,WANG Sirui,et al. Optimizing echo state network with backtracking search optimization algorithm for time series forecasting[J]. Engineering Applications of Artificial Intelligence,2019,81:117-132.

[8] TSAI H C. Improving backtracking search algorithm with variable search strategies for continuous optimization[J]. Applied Soft Computing Journal,2019,80:567-578.

[9] 邓明学. 基于遗传算法的开放教育在线考试系统组卷实现[J]. 广西广播电视大学学报,2019,30(2):24-26.

[10] 贺建英,王光琼,唐青松. 一种基于遗传算法的智能组卷策略优化研究[J]. 计算机与数字工程,2019,47(1):130-135.

[11] 李建敏. 基于混合粒子群算法的智能组卷研究[J]. 电子技术与软件工程,2015(12):187-188.

[12] FERREIRE C. Gene expression programming:mathematical modeling by an artificial intelligence[M]. Berlin:Springer-Verlarg,2006.

[13] ZUO J,TANG C J,ZHANG T Q. Mining predicate association rule by gene expression programming[C]//Proceedings of the third international conference for web information age. Beijing:Springer,2002:92-103.

[14] 吴书玲. 基因表达式编程的改进及其在知识发现中的应用研究[D]. 西安:西安建筑科技大学,2016.

[15] 陈　超. 基因表达式编程优化算法及其在聚类分析中的应用[D]. 西安:西安电子科技大学,2013.

[16] 阮梦黎. 基因表达式编程研究及其在函数挖掘中的应用[D]. 济南:山东师范大学,2012.

[17] 吴　勇. 基因表达式编程算法及其应用研究[D]. 武汉:武汉理工大学,2008.