一种改进的多目标粒子群组卷算法*

马跃亮,靳志强,孙晨霞
(河北农业大学 信息科学与技术学院,河北 保定 071001)

摘 要:通过对粒子群优化算法(PSO)和遗传算法的研究,提出了一种改进的粒子群优化方法。该方法保留了粒子群算法成功率高、收敛速度快、不易陷入早熟的优点,并在此基础上加入了遗传算法的交叉及变异操作,改善了粒子群算法寻优初期关于粒子速度的两难问题,使得本算法在保持高成功率的基础上更加快速。通过在组卷系统中应用,验证了改进后粒子群早期寻优的优异性能。

关键词:粒子群;改进;遗传算法;组卷

中图分类号:TP30 **文献标识码**:A **文章编号**:1674-7720(2010)23-0011-03

A multi-object test paper generation through improved particle swarm optimization algorithm

MA Yue Liang, JIN Zhi Qiang, SUN Chen Xia
(College of Information Science & Technology, Agricultural University of Hebei, Baoding 071001, China)

Abstract: With the research of particle swarm optimization algorithm and genetic algorithm, modified particle swarm optimization (PSO) algorithm was given in the paper. It holds the advantages of high success rate, rapid velocity of convergence and not easy to fall into early-mature, includes the operations of crossover and mutation witch made the control of the particle's speed more easily, and keeps a speed more faster then ever. By using the improved PSO algorithm in the system of test paper generation system, the outstanding performance on early stage optimization was proved.

Key words: particle swarm; improve; genetic algorithm; test paper generation

利用计算机按照一定的组卷算法从试题库抽取试题组成符合要求的试卷,是实现考试规范化和科学化的手段。在各种计算机辅助考试系统、试题库系统软件中,自动组卷一直是该类系统的一个难点,其解决方法也一直是算法研究热点。可以说一个自动组卷系统的效率和组卷质量主要取决于组卷算法。以往的组卷算法多采用随机抽题法、回溯试探法及遗传算法。

随机抽题法根据状态空间的控制指标,随机抽选一道试题加入试卷中,不断重复直到组卷完毕或失败为止。该算法简单,但是当题库庞大时,时间复杂度大,而且常由于约束条件的局部满足而导致组卷失败。回朔试探法是将随机选取产生的每一状态记录下来,搜索失败时,释放上次记录的状态类型,然后再依据一定的规律变换一种新的状态进行试探,通过不断地回朔试探直到试题生成完毕或回到出发点为止。当试卷总题量较大时,时间和空间复杂度都很大。可以看出,随机抽题法和回溯试探法都具有较大的缺点,而且不具有智能性。

遗传算法[1](GA)模仿自然界的适者生存法则,根据遗传的变异交叉等操作进行重复迭代,直到满足某些收敛指标。GA最初用于函数优化及组合优化。现已广泛用于生产调度问题、图像处理、人工生命、遗传编码和机器学习等领域。其在计算机辅助教学方面的应用也是一大亮点,如在组卷应用中远优于以往算法,但也存在粒子速度控制等问题。

1 组卷问题数学模型

1.1 组卷问题描述

组卷问题即计算机根据要求从题库中自动抽取符合条件的分数、难度、章节、知识点等要求的试题组成试卷,因此,组卷问题包括了多个重要指标,如分数、难

*基金项目:河北农业大学综合试题库系统推广应用研究(项目编号:10-C14)

度、题型、章节、知识点等。每道题在录入时已经被赋予了相应的指标。在本库中,试题分为1、2、3、4四个难度等级;试卷难度从难到易分为A、B、C、D、E五个等级,每种级别的试卷中四种级别的试题所占比例不同。试题其他属性,如题型、章节、知识点等也都相应赋值。

1.2 数学模型

本文引用了王一萍[2]选用的关于粒子群算法设计的目标函数数学模型:

$$Z=p\sqrt{\sum_{1\leqslant k\leqslant m}|(\sum_{i=1}^{N}d_i x_{ik})/\sum_{i=1}^{N}x_{ik}-D|^p} \tag{1}$$

其他限制函数如下:

$$(\sum_{i=1}^{N}s_i x_{ik})/\sum_{i=1}^{N}x_{ik}\geqslant u_i \tag{2}$$

$$\sum_{i=1}^{N}x_{ik}c_{ik}/\sum_{i=1}^{N}c_{ik}\geqslant h_{ik},1\leqslant i\leqslant M \tag{3}$$

$$f(x)=(x_1,x_2,\cdots\cdots x_n),n\text{ 为整数} \tag{4}$$

其中 x_{ik} 表示试题,c_{ik} 表示知识点,h_{ik} 表示知识点覆盖率,d_{ik} 表示难度系数,D 表示目标难度系数,s_i 表示主观题,u_i 为主观题最小比例。

在上面各式中,式(1)为目标函数,式(2)为主观题比例函数,式(3)为知识点覆盖率函数,式(4)为题型分布函数。

2 改进的粒子群优化算法

2.1 简单粒子群优化算法

假设在一个 M 维的目标搜索空间[3]中有 n 个粒子,每个粒子的位置表示一个潜在的解。用 $x_i=(x_{i1},x_{i2}\cdots\cdots x_{iM})$ 表示第 i 个粒子的位置向量。用 $v_i=(v_{i1},v_{i2}\cdots\cdots v_{iM})$ 表示第 i 个粒子飞行的速度向量。将 x_i 带入适应度函数 $f(x)$ 就可以计算出其适应值 $f(x_i)$,根据适应值的大小即可衡量 x_i 的优劣。用 $p_i=(p_{i1},p_{i2}\cdots\cdots p_{iM})$ 表示第 i 个粒子迄今为止搜索到的最好位置,也称为个体极值 p_{best}。用 $p_g=(p_{g1},p_{g2}\cdots\cdots p_{gM})$ 表示整个粒子群迄今为止搜索到的最优位置,也称为全局极值 g_{best}。第 i 个粒子第 m 个分量根据下面公式来更新自己的速度和位置:

$$v_{im}(k+1)=\omega v_{im}(k)+c_1r_1(p_{im}^{(k)}-x_{im}^{(k)})+c_2r_2(p_{gm}^{(k)}-x_{gm}^{(k)}) \tag{5}$$

$$x_{im}(k+1)=x_{im}(k)+v_{im}(k+1)\quad i\in N,m\in M \tag{6}$$

其中,$v_{im}(k)$ 表示第 i 个粒子在第 k 次迭代中第 d 维的速度分量;$x_{im}(k)$ 表示第 i 个粒子在第 k 次迭代中第 m 维的位置分量;c_1 和 c_2 为加速度系数;r_1 和 r_2 是介于[0,1]之间的随机数;ω 称为惯性因子;$\omega v_{im}(k)$ 表示第 i 个粒子在第 k 次迭代中第 m 维的惯性速度;$p_{im}(k)-x_{im}(k)$ 表示第 i 个粒子当前位置与自己最好位置之间的距离,作为第 $k+1$ 次迭代源于第 i 个粒子本身的加速度;$p_{gm}(k)-x_{im}(k)$ 表示第 i 个粒子当前位置与群体最佳位置之间的距离,作为第 $k+1$ 次迭代源于整个群体的加速度。据上式知,第 i 个粒子在第 $k+1$ 次迭代中的位置和速度根据第 k 次迭代的位置 $x_i(k)$、第 k 次迭代的速度 $v_i(k)$、个体极值 p_{best}、全局极值 g_{best} 等生成。

由式(5)可见,粒子的速度进化方程由个体认知和社会信息共享两部分组成。

式(5)中的第一项 $v_{im}(k)$ 为粒子更新前的速度,第二项 $c_1r_1(p_{im}^{(k)}-x_{im}^{(k)})$ 为粒子本身的个体认知部分;第三项 $c_2r_2(p_{gm}^{(k)}-x_{gm}^{(k)})$ 为粒子间的社会信息共享部分。如果进化方程只有个体认知部分,则因粒子间缺少信息交流,得到最优解的概率非常小;如果只有社会信息共享部分,则粒子因无自身的认知能力,虽然收敛速度较快,但粒子速度难以得到更理想的控制。

2.2 改进的粒子群算法

在整个粒子群寻优过程中,粒子飞行速度大小直接影响算法的全局收敛性。飞行速度大,则粒子到达全局最优解所需时间短,但容易飞越最优解;飞行速度小,则粒子到达全局最优解的时间长,故必需对粒子的飞行速度进行有效控制,使粒子在先期以较快速度飞行到全局最优区域,进而以较小的步长对最优区域进行局部搜索以获得高精度解。笔者将粒子群优化过程分为三个阶段。首先利用遗传算法的交叉操作对粒子群进行前期搅动,使得初始种群能够迅速地均匀遍布整个搜索空间;然后用变异操作进行局部寻优的微调搅动,使得种群更快地向目标移动靠拢;最后利用粒子群优化得出结果。

改进粒子群优化算法的实现步骤如下。

(1)随机生成初始群体

在粒子群算法中,每个粒子均表示一个可行解,以二进制编码形式表示。

$$A=\begin{pmatrix}a_{11}&\cdots&a_{1n}\\ \vdots&\ddots&\vdots\\ a_{41}&\cdots&a_{4n}\end{pmatrix}$$

在矩阵中,行表示试题,如:a_{2i} 为1表示试题 Q_i 在第2套试卷中被选择,每行表示一套试卷。每列依次表示题分、题型、章节、难度系数。

(2)计算并评价每个粒子的适应值

模型最初目标函数值用来衡量满足式(2)~式(4)候选解的质量。然而,基于粒子群优化算法的粒子可能不满足其中一个或多个限制条件。为解决此问题,评估粒子质量时,如果不满足某限制条件则要考虑惩罚函数,惩罚函数对应各限制条件。

①b 罚值是关于不满足知识点限制范围:

$$b=\sum_{k=1}^{s}\sum_{j=1}^{M}(h_j-\sum_{i=1}^{N}c_{ij}x_{ik})$$

这个值是弥补所选择试题知识点覆盖率的不足。

②d 罚值是违反多试卷同试题数量的限制:

$$d=\sum_{j\neq k}(\sum_{i=1}^{N}x_{ij}x_{ik}-S)$$

当两套试卷具有相同的试题数量超阈值 S 时使用。

③函数 $J(x)$ 是计算粒子 x 适应值

$J(x)=Z$ 当满足所有限制条件

$J(x)=Z+w_1b+w_2c+w_3d$

其中 w_1、w_2 和 w_3 是相关的权重，同样粒子 x 的适应值要考虑目标值和罚值。根据组卷的实际问题，适应值越小，粒子越好。

(3)个体极值 p_{best} 和全局极值 g_{best} 计算

使用限制标准确定 p_{best} 和 g_{best} 的值在原来的粒子群优化算法中，粒子的 p_{best} 和 g_{best} 最费时间，因此提出限制标准来加速确定粒子 p_{best} 和 g_{best} 的进行。根据分析得知，一个粒子的适应值是为了确定粒子 p_{best} 和 g_{best}，但不能直接用于粒子速度的更新。因为 Z 和 $J(x)$ 都能单独提高作用，可用当前 p_{best} 的适应值作为适应值的界限，当中间适应值超过此界限时则终止对第 i 个粒子适应值的计算。将第 i 个粒子的当前适应值与 p_{best} 值比较，如果优于它原来的 p_{best} 则用这个适应值更新其 p_{best}，这种限制能节省计算时间。

(4)若个体极值 p_{best} 和全局极值 g_{best} 满足条件或其他终止条件，则退出。

(5)将个体按照适应度排序，取适应度较差的 50%进行交叉或变异操作，遗传操作计数器加 1。

为加速粒子群初期的寻优速度，在前期加入遗传算法的交叉和变异操作。将排序后的 50%粒子复制一遍，并与原型随机进行交叉或变异操作，然后再按适应度排序，取适应度值前 50%放入原种群，保持种群规模不变。在总循环次数的前 25%次，交叉率取 0.8 进行交叉操作，使群体迅速均匀地进行全局搜索；接下来的 15%次以变异率为 0.2 进行变异操作，使得群体能更加快速地进行局部搜索。

①染色体的交叉。采用双点交叉运算[4]，随机选取两个双亲染色体确定两个随机交叉点进行交叉运算得出新个体，如：

Prt1： 1101 1001 1010 0101

Prt2： 0110 1110 0110 1110

设每两位为一个交叉单位，随机产生两个交叉点 4 和 7，则交叉之后变成：

son1： 1101 1010 1010 1101

son2： 0110 1101 0110 0110

②染色体的变异。随机产生若干个变异位置，然后在相应的位置随机选取一道试题。如变异前的染色体为：1010 1101 0001 1110，随机产生了 3 个变异位置，分别为：(2,5,8)，则变异后的染色体为：1011 1101 0101 1100。

(6)若个体极值 p_{best} 和全局极值 g_{best} 满足条件或其他终止条件，则退出。

(7)生成 $v_{im}(k+1)$ 和 $x_{im}(k+1)$ 构成下一代群体，循环次数加 1。

每个粒子的速度和位置的更新要遵循粒子群的离散方法，如通过转换函数 $S(\cdot)$ 和粒子中取值为 1 的位表示概率的速度变化，尺度为[0.0,1.0]。采用线性内插函数：$S(v_{ij})=v_{ij}/2v_{max}+0.5$，将速度转换成概率。

(8)检查终止条件

如果最优解不再变化或者达到最大迭代次数 GenMax 便终止迭代，否则返回(2)。

3 实验测试及结果分析

本实验在具有海量试题的题库内进行测试。本算法中每个题目含题分、题型、章节、难度系数四个属性(如表 1)，其他属性从略。设试卷总分为 100 分，各种题型比例为 1:1:1:2:5，章节比例 2:2:2:4，难度分布为 2:3:7:8，粒子群规模为 3，最大迭代次数为 600。

表 1 题库数据示例

试卷号	题分	题型	章节	难度
1	2	3	1	75
2	2	3	2	80
3	2	3	4	80
4	2	3	3	70
5	2	3	3	80
6	2	3	2	70
7	2	1	4	70
8	2	1	1	80
9	2	1	2	90
10	2	1	3	75
11	2	2	4	75
12	2	2	3	80

使用改进粒子群算法进行测试，运行 10 次的一组结果为(试题编号)：17,22,27,34,39,42,44,49,51,55,57,57,60,61,64,65,67,68,71,73,75,78,80,81,82,84,86,90,92,99。

分别使用遗传算法、粒子群算法和改进粒子群算法进行组卷实验。迭代记录如表 2。

表 2 三种算法运行 10 次的迭代记录比较

试卷编号	遗传算法	简单粒子群	改进粒子群
1	239	148	85
2	283	197	154
3	174	84	92
4	109	139	162
5	317	241	178
6	398	207	101
7	126	223	83
8	144	135	99
9	291	80	157
10	184	141	179

试验结果表明，当试题库中的试题具有一定规模时，此改进粒子群算法生成试卷的速度能达到较理想的效果，而且题库越大，效果越明显。此改进的粒子群算法

和遗传算法在同样的迭代次数条件下，相对于遗传算法和基本粒子群算法能更快地收敛于全局最优解。

本文利用遗传算法的优点，对粒子群优化算法(PSO)的组卷方法加以改进，从而设计出一种更加智能和高效的自动组卷算法，使智能组卷的速度得到了明显的提高，使组卷系统更加具有使用价值和现实意义。多数算法中提高速度都是以牺牲更多寻优目标实现的，增加寻优目标和提高速度是个两难问题，有待于进一步解决。

参考文献

[1] 王小平，曹立明.遗传算法——理论、应用与软件实现[M].西安：西安交通大学出版社，2002.

[2] 王一萍，曲伟建，潘海珠.一种基于粒子群优化的组卷算法[J].兵工自动化，2008(8).

[3] 田民格. 改进的粒子群优化算法实现多目标智能组卷[D].三明学院学报，2009(10)：404-405.

[4] 徐清振，肖成林.遗传算法成卷策略的编码实例[J].现代计算机，2006(6):40-41.

(收稿日期：2010-08-02)

作者简介：

马跃亮，男，1983年生，硕士研究生，主要研究方向：计算机网络与数据库。