文章编号:1001 - 9081(2007)07 - 1760 - 03

改进的遗传进化算法及其在函数建模中的应用

贾丽媛[1,2],杜 欣[2,3], 习胜丰[1]
(1. 湖南城市学院 计算机科学系,湖南 益阳 413000; 2. 中国地质大学 计算机学院,武汉 430074;
3. 石家庄经济学院 信息工程系,石家庄 050031)
(jia_721008@ sohu. com)

摘 要:提出了一种改进的基因表达式程序设计的遗传进化算法(PGEP),新的算法引入三个算子:基于精英保存策略的精英子空间算子;基于全局收敛策略的变重组、变换概率和变变异概率算子及基于群体搜索技术的变维子空间算子。将改进的基因表达式程序设计应用于函数建模,获得满意的结果。

关键词:基因表达式程序设计;精英子空间算子;变维子空间算子
中图分类号: TP311;TP301.6 **文献标志码**:A

Application of an improved gene expression programming in functions modeling

JIA Li-yuan[1,2], DU Xin[2,3], XI Sheng-feng[1]
(1. *Department of Computer Science, Hunan City University, Yiyang Hunan* 413000, *China*;
2. *School of Computer, China University of Geosciences, Wuhan Hubei* 430074, *China*;
3. *Department of Information Engineering, College of Shijiazhuang Economy, Shijiazhuang Hebei* 050031, *China*)

Abstract: A new gene expression programming (GEP), PGEP, was proposed. It was based on three new operators: good subspace operator based on the best ones stored strategy; variable reorganization, conversion probability and variable mutation probability operator based on global convergence strategy; variable dimension subspace operator based on population search strategy. The result shows that PGEP behaves better than GEP in functions modeling.

Key words: gene expression programming; good subspace operator; variable dimension subspace operator

0 引言

基因表达式程序设计[1](Gene Expression Programming, GEP)的思想是 Candida Ferreira 于 2001 年首次提出的,它是一种知识发现的仿生计算新技术。GEP 融合了遗传算法[2](Genetic Algorithm, GA)和遗传编程[3](Genetic Programming, GP)的优点,Ferreira[4]指出 GEP 比 GA 和 GP 快 2 ~ 4 个数量级,通过简单紧凑的编码解决复杂应用问题。GEP 广泛应用于函数发现、时间序列预测、分类问题等领域。

GEP 在处理较复杂问题时,仍存在收敛速度慢、收敛后适应度不高和易陷入局部最优的情况。本文在 GEP 算法中引入新的算子:1)基于精英保存策略的精英子空间算子;2)基于全局优化策略的变重组、变换概率 P_c 和变变异概率 P_m 算子; 3) 基于群体搜索技术的变维子空间算子。

1 GEP 概述

基因表达式程序设计是在 GA 和 GP 的基础之上发展起来的基因组和表现组的新型遗传算法,它综合了 GA 和 GP 的优点,具有染色体简单、线性和紧凑、易于进行遗传操作等优点。GEP 不需要对结果公式做主观假定,而认为真理在训练数据中,以基本变量和算符为遗传本质的公式(例如由 +, -, *, /, (,), x, y, &, ~, true, false 组合表达的结果)。方法上更客观,结果上更接近事物的本质。

染色体(Chromosome)和表达式树(Expressing Tree, ET)是 GEP 技术中最重要的概念。染色体作为承载遗传信息的基因型实体,参与遗传操作;表达式树作为信息的表现型,表达遗传实体中的信息编码。染色体和表达式树结构简单清晰,通过简单的编码和解码规则可无歧义地互化。GEP 将这两者分别作为独立个体,对 GA 和 GP 的优点分别加以继承,使遗传操作易于实施,结果方便表达。

染色体由若干个基因(Gene)通过连接运算符连接组成。Gene 由头部(hail)和尾部(tail)组成。Hail 包含了函数(Function)和终结符(Terminals)。设头部长度为 h,尾部长度为 t,则两者满足关系式:

$$t = h(n - 1) + 1 \quad (1)$$

其中:n 表示在函数符号集中所需变量数最多的函数的参数个数。例如,在一般数学运算中,对于开方,$n = 1$;对于乘号或加号 $n = 2$。在逻辑运算中,对于 IF,$n = 3$;对于 AND,$n = 2$。

在式(1)中,由于尾部与头部所具有的特殊关系,使得 GEP 的基因在任何遗传操作算子的作用下都不会产生不正确的个体。

GEP 先将个体编码为固定长度的染色体线性串,每个染色体通过连接符号连接形成基因组。其编码规则可以简单描述为:将基因组(染色体串)按从左到右的顺序逐个读取每个字符,并按照层次排放,形成表达式树。解码规则是:按照从上到下,从左到右的顺序遍历表达式树,最后形成基因组(染色体串)。GEP 染色体和表达式树结构简单清晰,通过简单的线性编码和解码规则可无歧义地互化。GEP 将这两者分别

收稿日期:2007 - 01 - 08;**修回日期**:2007 - 03 - 27。

作者简介:贾丽媛(1972 -),女,湖南益阳人,副教授,硕士,主要研究方向:算法和人工智能; 杜欣(1979 -),女,新疆石河子人,讲师,硕士,主要研究方向:演化算法、人工智能; 习胜丰(1970 -),男,湖南桃江人,副教授,硕士研究生,主要研究方向:计算机网络、图形学。

作为独立个体,对 GA 和 GP 的优点分别加以继承,使遗传操作易于实施,结果方便表达。

GEP 进化过程和传统的遗传算法很相似,每一代通过遗传算子[5,6](选择算子、变异算子、倒位算子、重组、变换算子)进化。文献[7]中描述了 GEP 算法的基本框架。

在 GEP 算法中,根据适应度函数选择出的双亲基因非常接近,那么所产生的后代相对双亲也必然比较接近,所期待的改善就比较小。基因模式的单一性不仅减慢进化历程,而且可能导致进化停滞,过早收敛于局部最优点,使算法搜索性能不高。

2　对基因表达式程序设计的改进

基于传统 GEP 解决复杂问题时收敛速度慢和易陷入局部最优等方面的欠缺,对基因表达式程序设计做三点改进:即在演化过程中使用精英子空间算子;变重组、变换概率 P_c 和交换概率 P_m 算子;变维子空间算子。

2.1　基于精英保存策略的精英子空间算子

本算子中,群体的 K 个最好精英个体直接进入 M 个个体参与下一代的产生。这样做的目的是使解的好的信息得到充分的利用,使算法更快地收敛到最优解。实验表明,采用这种精英保存策略,收敛速度明显加快。对 K 的选取也并不是越大越好,K 大固然可以更好地利用解的信息,但 K 越大杂交子空间的基的自由度就越小,容易使解的搜索空间在某个局部空间徘徊,经过试验证明 K 的取值范围应该是 $K \leqslant M/2$,这里我们定义 $K = (M)/2$。

2.2　基于全局收敛策略的变重组、交换概率和变异概率算子

鉴于传统的 GEP 遗传操作容易出现早熟收敛现象,因此引入变重组、交换概率 P_c 算子和变异概率算子 P_m。GEP 算法的参数中重组、变换概率 P_c 和变异概率 P_m 的选择是影响遗传算法行为和性能的关键所在,直接影响算法的收敛性。P_c 越大,新个体产生的速度就越快,然而,P_c 过大时遗传算法模式被破坏的可能性越大,使得具有高度适应度的个体结构很快被破坏;但是如果 P_c 过小,会使搜索过程缓慢,以至停滞不前。对于变异概率 P_m,如果 P_m 过小,就不易产生新的个体结构;如果 P_m 取值过大,那么遗传算法就变成了纯粹的随机搜索算法。

当适应度值低于平均适应度值时,说明该个体是性能不好的个体,对它就采用较大的重组、变换率和变异率;如果适应度值高于平均适应值,说明该个体性能优良,对它就根据其适应度值取相应的重组、变换率和变异率。可以看出,当适应度值越接近最大适应度值时,重组、变换率和变异率就越小;当等于最大适应度值时,重组、变换率和变异率的值为零。这种调整方法对于群体处于进化后期比较合适,但对于进化初期不利,因为进化初期群体中较优的个体几乎处于一种不发生变化的状态,而此时的优良个体不一定是优化的全局最优解,这容易使进化走向局部最优解的可能性增加。为此,可以做出进一步的改进,使群体中最大适应度值的个体的重组、变换率和变异率不为零,分别提高到 P_{c1} 和 P_{c2},这就相应地提高了群体中表现优良个体的重组、变换率和变异率,使得它们不会处于一种近似停滞不前的状态。此策略保证了群体多样性,克服了算法陷入局部最优,达到全局最优。

经过上述改进,P_c 和 P_m 计算表达式如下:

$$P_c = \begin{cases} p_{c1} - (p_{c1} - p_{c2})(f' - f_{avg})/(f_{max} - f_{avg}), & f' \geqslant f_{avg} \\ p_{c1}, & f' < f_{avg} \end{cases} \quad (2)$$

$$P_m = \begin{cases} p_{m1} - (p_{m1} - p_{m2})(f_{max} - f)/(f_{max} - f_{avg}), & f \geqslant f_{avg} \\ p_{m1}, & f < f_{avg} \end{cases} \quad (3)$$

上式中,$p_{c1} = 0.6$, $p_{c2} = 0.3$, $p_{m1} = 0.1$, $p_{m2} = 0.001$。

2.3　基于郭涛算法的群体搜索技术的变维子空间算子

郭涛在文献[8]中提出了一种基于搜索技术、多父体杂交的演化算法(郭涛算法 GT),该算法通用、简洁高效,已被广泛应用于各类复杂的优化问题的求解上,取得了良好的效果。该算法的高效性只是依赖于多亲杂交算子,该算子的描述如下:在每一代的演化过程中,随机从种群中挑出 M 个个体 $X_1, X_2, \cdots, X_M$,M 个个体杂交产生新个体 X', $X' = \sum_{i=1}^{M} a_i X_i$, 其中 $\sum_{i=1}^{M} a_i = 1$, $-0.5 \leqslant a_i \leqslant 1.5$。如果 X' 的适应度好于种群中最坏的个体,就取代它。在此对郭涛算法做两点改进。

2.3.1　变子空间的维数 M

子空间的维数 M 在郭涛算法中是保持不变的(即取 M 个父体进行重组、变换),不管当代群体的解的性质如何,它总是在维数相同的子空间内找一个替换点,这样在群体接近全局最优时,搜索范围依然较大,这显然会加大计算量而影响搜索的范围,即减少子空间的维数。因此我们在原算法中设置变维子空间,即在郭涛算法中的每一次循环结束前添加如下代码:

if abs $(f(X_{best}) - f(X_{worst})) \leqslant \eta$ and $M \geqslant 3$

then $M := M - 1$;

其中 η 是一个与要求的精度有关的量,且 $\eta > \mu$。例如计算要求精确到 $\mu = 10^{-14}$,则可将 η 设置为 10^{-2} 或 10^{-3}。

2.3.2　变被替换子空间维数 s

郭涛算法是在当代子空间 V 中随机找一个候选解,虽然他用子空间方式来描述,但实际上在算法中并没有真正采用子空间搜索法,而是一种多父体杂交算法。由于只是随机地从子空间中找一个个体,这样做有时会遗漏子空间中比较好的解,从而影响搜索的效果和效率。如果是随机地从子空间中选取多个(s)个体,然后用 X' 替代 s 群体中最差的个体,其搜索效果将会更好,因此我们将传统的遗传算法中的代码:“从当代子空间 V 中随机选取 1 个点 X'”替换成为:从当代子空间 V 中随机选取 S 个点 X_1', X_2', X_3', $\cdots$, X_s'($M/3 \leqslant s \leqslant M/2$), $X'' = \arg \min_{1 \leqslant i \leqslant s} f(X_i')$,如果 X' 适应度好于 X'',就取代 X''。

该算子采用了演化计算中的群体搜索策略,保证了搜索空间的全局性,有利于搜索问题的解;采用随机子空间中的随机搜索(多父体重组)策略,特别是子空间中随机搜索的非凸性:$X' = \sum_{i=1}^{M} a_i X_i'$, 其中 $\sum_{i=1}^{M} a_i = 1$, $-0.5 \leqslant a_i \leqslant 1.5$。

使算法搜索的子空间可覆盖多父体的凸组合空间,保证了随机搜索的遍历性,即解空间中不存在算法搜索不到的“死角”;采用了变维技术,当群体接近全局最优时,搜索范围明显减小,这样减小了计算量,加快了算法的收敛速度;采用了“劣汰策略”,每次将群体中适应性较差的个体淘汰出局,让适应度较差的一些个体保存下来,既保证了群体的多样性,也保证了适应性最好的个体可以“万寿无疆”。这种“群体爬山策略”,保证了整个群体最后集体登上最高峰(深谷)。

3 改进的 GEP 的算法描述

算法 1 GEP 算法

输入 训练集及 GEP 配置

输出 最佳适应度个体

```
Init();                    //加载配置和种群初始化,其中种群大小为 M
DefaultJudge();                                   //调用评价函数
//循环遗传进化,直到达到最大辈数或成功
While( FCurrentGenaration < FMaxGenaration) Do Begin
  //采用精英保存策略将 M 中的排名前 K 个好的个体继承下
  // 来,然后随机产生 M - K 个个体,依次进行各项算子操作,
  //如式(2)、(3),做变异、重组、变换操作等
  For x: =1 to M do
    OperationList[ x];
  End For
  DefaultJudge();                                 //调用评价函数
  Sort();                                 //对种群按适应度进行排序
  Inc();                                          //进化辈数加 1
End;                          //如果最好进化结果满足要求,则退出
Result: = X best                                  //返回最好的个体
```

算法 2 改进的郭涛算法

输入 用 GEP 算法得到的较好的函数模型及其相应的 C_i 和常数;郭涛算法参数

输出 最优的函数模型

```
//用改进的郭涛算法对以上演化出的较好的函数模型进行优化
Init();        //加载配置和种群初始化为 X_i,其中种群大小为 N
DefaultJudge();                                   //调用评价函数
X_best = arg max_{1≤i≤N} f(X_i), X_worst = arg min_{1≤i≤N} f(X_i);
//循环遗传进化,直到成功
While( f(X_best) > f(X_worst) Do Begin
  //用郭涛算法产生 V 子空间有 M 个个体,产生 X' = Σ_{i=1}^{M} a_i X_i,
  //其中 Σ_{i=1}^{M} a_i = 1, -0.5 ≤ a_i ≤ 1.5
  Generate( X', M);
  //从 N 中随机选出 s 个个体中选取最差个体 X''_worst
  //如果 f(X') > f(X''_worst), X''_worst = X';
  Replace( X', s);
  DefaultJudge();                                 //调用评价函数
  Inc();                                          //进化辈数加 1
  If( abs(( f(X_best)) - f(X_worst)) ) ≤ η and M ≥3)
  Then M: = M -1;
  X_best = arg max_{1≤i≤N} f(X_i), X_worst = arg min_{1≤i≤N} f(X_i);
End
Result: = X best
```

4 用改进的 GEP 方法进行函数建模

实验平台:VC ++6.0

表 1 预测模型的样本集用 GEP 和 PGEP 预测结果比较

样本	煤层深度/m	煤层厚度/m	煤层瓦斯含量/(m^3/t)	煤层间距/s	日进度/(m/d)	日产量/(t/d)	绝对瓦斯涌出量(m^3/min)			相对误差(%)	
							实测值	GEP 预测值[8]	PGEP 预测值	GEP	PGEP
1	408	2.0	1.92	20	4.42	1825	3.34	3.27	3.30	2.0	1.2
2	411	2.0	2.15	22	4.16	1527	2.79	3.29	2.90	10.8	3.9
3	420	1.8	2.14	19	4.13	1751	3.56	3.47	3.49	2.4	2.0
4	432	2.3	2.58	17	4.67	2078	3.62	3.71	3.65	2.6	0.8
5	456	2.2	2.40	20	4.51	2104	4.17	4.10	4.12	1.8	1.2
6	516	2.8	3.22	12	3.45	2242	4.60	4.68	4.65	1.7	1.1
7	527	2.5	2.80	11	3.28	1979	4.92	5.03	4.95	2.2	0.6
8	531	2.9	3.61	13	3.68	2288	4.78	4.84	4.88	1.2	2.0
9	550	2.9	3.61	14	4.02	2325	5.23	5.1	5.19	0.8	0.6
10	563	3.0	3.68	12	3.53	2410	5.56	5.42	5.51	2.4	0.9
11	590	5.9	4.21	18	2.85	3139	7.24	7.35	7.29	1.5	0.7
12	604	6.2	4.03	16	2.64	3354	7.80	7.49	7.71	4.0	1.2
13	607	6.1	4.34	17	2.77	3087	7.68	7.58	7.61	1.3	0.9
14	634	6.5	4.80	15	2.92	3620	8.51	8.30	8.40	2.4	1.2
15	640	6.3	4.67	15	2.75	3412	7.95	7.94	7.93	0.1	0.2

实验 1 选择文献[5]中的 Schaffer 函数 *F*6 作为测试函数:

$$F6 = 0.5 + \frac{\sin^2\sqrt{(x_1^2 + x_2^2)} - 0.5}{[1.0 + 0.001(x_1^2 + x_2^2)]^2}$$

$$-100 \leqslant x_i \leqslant 100 (i = 1, 2)$$

该函数在定义域内只有一个极小点 $F6(0,0) = 1$。

实验中的 x_1 和 x_2 采用随机方法得到,y 由公式计算得到,总共生成 1000 组数。实验采用不同的参数分别对 GEP 和 PGEP 作了两组实验,每组实验中,两种方法各自进行 10 次实验。

适应度函数为:$fitness = 1 - SSE/SST$

$$SSE = \sum_{j=1}^{m}(y_j - \hat{y}_j)^2,\ SST = \sum_{j=1}^{m}(y_j - \bar{y})^2$$

其中,$\hat{y}$ 为变量 y 关于函数 g 的估计值。$\bar{y}$ 为变量 y 的平均值。称 *SSE* 为残差平方和,*SST* 为总离差平方和。显然,$-\infty \leqslant fitness \leqslant 1$。

图 1、图 2 给出了 PGEP 和 GEP 对 *F*6 的实验结果比较。

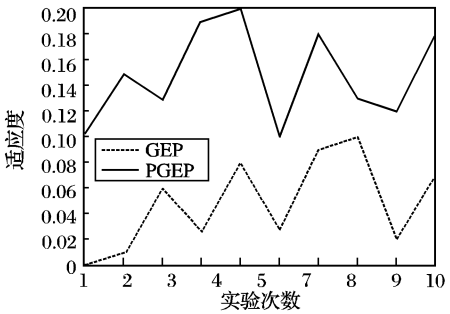

图 1 PGEP 和 GEP 的适应度比较

(下转第 1766 页)

0, 0, 0, 0)。

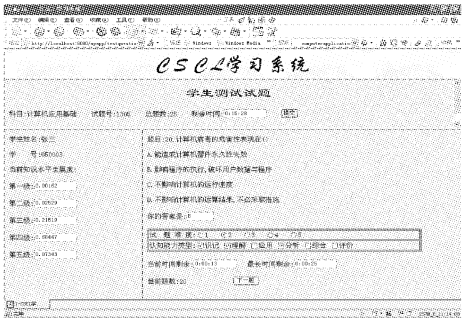

图2 学生测试界面

表1 学生知识水平隶属度的改变

题号	*H*	*N*	做题情况	隶属度
1	1	0.5	做对一题	(0.5, 0.5, 0, 0, 0)
2	1	0.5	做对一题	(0.25, 0.5, 0.25, 0, 0)
3	2	0.25	做对一题	(0.1875, 0.4375, 0.3125, 0.0625, 0)
4	5	0.5	做错一题	(0.40625, 0.375, 0.1875, 0.03125, 0)
5	2	0.25	做对一题	(0.3046875, 0.3828125, 0.234375, 0.0703125, 0.0078125)

从表1可看出,做5道题后,学生对学习内容的掌握最可能为第2等级。显然,随着做题的增加知识水平将不断发生变化。该学生经过5套题测试后,隶属度为:(0.00112, 0.02316, 0.0868, 0.63536, 0.25356),则其知识水平最可能为第4级(较高)。5套题的成绩分别为:95,82,85,90,88。其平均分为88,符合我们平常所认为的水平较高等级。因此,实验表明,随着做题的增加,知识水平的表示能比较接近学生的实际水平。

参考文献:

[1] 赵建华. 计算机支持的协作学习[M]. 上海: 上海教育出版社, 2006.

[2] GREER J, MCCALLA G, COOKE J, *et al.* The Intelligent Helpdesk: Supporting Peer-Help in a University Course[EB/OL]. [2006-12-12]. http://julita.usask.ca/Texte/Jim-html/I-Help.htm.

[3] SUN C-T, LIN S S J. Learning through collaborative design: a learning strategy on the internet[C/OL]// 31th ASEE/IEEE Frontiers in Education Conference. [2006-12-01]. http://citeseer.nj.nec.com/505392.html.

[4] INABA A, TAMURA T, OHKUBO R, *et al.* Design and analysis of learners' interaction based on collaborative learning ontology[C]// Proceedings of Euro-CSCL2001. Maastricht, Netherlands: [s.n.], 2001: 308-315.

[5] INABA A, IKEDA M, MIZOGUCHI R. Learning goals and design rationales in collaborative learning: an ontological approach to support design of collaborative learning[EB/OL]. [2006-12-12]. http://www.ei.sanken.osaka-u.ac.jp/pub/ina/isir03.pdf.

[6] 黄荣怀, 林凉. 构建WebCL平台上的e-Tutor[EB/OL]. [2006-12-20]. http://www.etc.edu.cn/articledigest15/goujian.htm.

[7] 赵建华. 基于Web环境的智能协作学习系统的理论与方法[D]. 广州: 华南师范大学, 2002.

[8] 李芳, 乞建勋, 牛东晓. AHP法在虚拟企业中心伙伴选择中的应用[J]. 华北电力大学学报, 2004, 31(4): 82-85.

[9] 王万良. 人工智能及其应用[M]. 北京: 高等教育出版社, 2005.

[10] 陈自郁. 基于代理的远程教学系统及学生模型的研究[D]. 重庆: 重庆大学, 2002.

[11] KATZ S, LESGOLD A, EGGAN G, *et al.* Modeling the student in Sherlock II[J]. Journal of Artificial Intelligence in Education, 1993, 3(4): 495-518.

[12] GURER D, DESJARDINS M, SCHLAGER M. Representing a student's learning states and transitions[C]// The 1995 American Association of Artificial Intelligence Spring Symposium on Representing Mental States and Mechanisms. Stanford, CA: [s.n], 1995.

[13] 谢忠新, 王林泉, 葛元. 智能教学系统中认知型学生模型的建立[J]. 计算机工程与应用, 2005, 41(3): 229-232.

[14] 朱习军. ICAI中的认知型模型设计[J]. 泰安师专学报, 2001, 23(3): 36-39.

(上接第1762页)

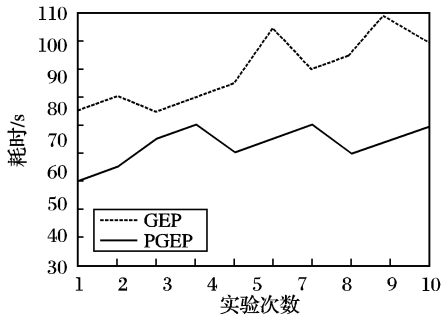

图2 PGEP和GEP耗时比较

实验结果表明:与传统GEP相比,PGEP性能良好。在10次实验中,PGEP每次的结果都优于GEP的结果,PGEP算法在速度上是GEP算法的1~2倍,在适应度上PGEP比GEP平均高0.11。

实验2 以下对由PGEP建模得到的结果与文献[6]中得到的GP建模得到的结果进行比较,以某矿3个工作面18个回采月份的采煤工作面瓦斯涌出的统计资料作为预测模型的样本集(表1),其中表1含有15个训练样本,用于建模。

由表1可以看出,PGEP建模得到的结果除了样本8和样本15相对误差比GEP得到的稍大以外,其余样本的相对误差比GEP所得到的模型计算结果的误差要小。

参考文献:

[1] FERREIRA C. Gene expression programming: a new adaptive algorithm for solving problems[J]. Complex Systems, 2001, 13(2): 87-129.

[2] MITCH M. An introduction to genetic algorithms[M]. Cambridge, Massachusetts, USA: MIT Press, 1992.

[3] KOZA J R. Genetic programming: on the programming of computers by means of natural selection[M]. Cambridge, Massachusetts, USA: MIT Press, 1994.

[4] FERREIRA C. Gene expression programming[M]. First Edition. Portugal: Angra do Heroismo, 2002.

[5] 周明, 孙树栋. 遗传算法原理及应用[M]. 北京: 国防工业出版社, 1999: 123-166.

[6] 李曲, 蔡之华, 朱莉, 等. 基因表达式程序设计方法在采煤工作面瓦斯涌出量预测中的应用[J]. 应用基础与工程科学学报, 2004, 12(1): 49-53.

[7] 段磊, 唐常杰. 基于基因表达式编程的抗噪声数据的函数挖掘方法[J]. 计算机研究与发展, 2004, 41(10): 1648-1689.

[8] 郭涛. 演化计算和优化[D]. 武汉: 武汉大学, 1999.

[9] 龚文引, 蔡之华. 基因表达式程序设计的原理与应用[J]. 微计算机信息, 2005, 32(11): 169-170.