第32卷 第21期 计 算 机 工 程 2006年11月
Vol.32 No.21 Computer Engineering November 2006
·人工智能及识别技术· 文章编号：1000—3428(2006)21—0188—03 文献标识码：A 中图分类号：TP183

基于改进的基因表达式编程的复杂函数建模

方旺盛[1]，张克俊[2]，邵利平[3]
(1. 江西理工大学信息工程学院，赣州 341000；2. 江西理工大学机电工程学院，赣州 341000；
3. 西安交通大学电子与信息工程学院，西安 710049)

摘 要：介绍了基因表达式程序设计方法的基本原理，针对求解复杂函数模型反问题中经典 GEP 算法多样性表现不足，甚至出现早熟的问题，提出了一种基于动态变异算子的改进的 GEP 算法——IGEP 算法，从理论上对该改进算法进行了复杂度分析和收敛性分析。通过求解复杂函数模型反问题的多个实验将改进算法与传统方法、神经网络方法、经典 GEP 算法进行了对比，结果表明：该方法建立的复杂函数反问题拟合模型比经典 GEP 方法、传统方法、神经网络方法得到的模型更加优秀。

关键词：基因表达式程序设计；自动建模；复杂度；收敛性；参数模型

Complex Function Modeling Based on Improved Gene Expression Programming

FANG Wangsheng[1], ZHANG Kejun[2], SHAO Liping [3]
(1. School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou 341000;
2. School of Machinery and Power-generating Equipment Engineering, Jiangxi University of Science and Technology, Ganzhou 341000;
3. School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049)

【Abstract】 The basic principle of gene expression programming (GEP) is introduced. An improved GEP algorithm called IGEP based on dynamic mutation operator which deals with the problem of complex function auto modeling of complex function is presented, the algorithm complexity of the IGEP is given. Many simulation results show that the models set up by the paper are better than the models set up by classic GEP, traditional algorithm and nerve network.

【Key words】 Gene expression programming(GEP); Auto modeling; Complexity; Convergence; Parameters model

基因表达式程序设计[1]（Gene Expression Programming，GEP）是葡萄牙科学家Candida Ferreira发明的一种基于基因组（Genome）和表现型(Phoneme)的新的遗传算法。它与遗传算法(Genetic Algorithms,GA)和遗传规划(Genetic Programming,GP)的根本区别在于它们所采用个体的本性不同：即在GA中个体是固定长度的线形串（染色体）；在GP中个体是长度和形状不同的非线形性实体（分列树），而在基因表达式程序设计中个体首先被编码成固定长度的线形串（基因组或者染色体），然后被表达成不同长度和形状的非线形实体（简单图表示，或者表达式树）。由于它集GA与GP的优点于一身，因此目前被广泛应用于函数关系发现、分类器设计及时间序列预测等研究中。

1 改进的基因表达式程序设计算法

经典基因表达式程序设计的实现技术主要包括编码方式、遗传算子、插串操作、重组算子、适应度函数选择等几个部分[2]。经典的GEP算法中的变异算子较简单，使得算法在保持基因多样性上表现不足，甚至出现早熟现象。通常解决的方法是对种群重新初始化，使得算法效率大大降低。本文就变异算子的变异方法给出了一种改进的方案，可以避免经典的GEP算法中的变异算子带来的缺陷，并对适应值函数进行了新的构造，从而大大提高算法的效率。

1.1 IGEP 算法描述

改进的 GEP 算法（IGEP）如下：

(1) 初始化种群，随机产生 50 组（过大会增大算法运行时间，过小则很难收敛）初始染色体，每个染色体由 5 个基因（采用 5 个以上基因的，效果较好，但需要运行更长时间）构成，每个基因头长度为 8（或者更多）。

(2) 按照适应值函数求解初始种群的各染色体的适应值，（本文采用 $f_{fitness}(i)=1000\times\frac{1}{Ei+1}$ 作适应值函数，其中 $E_i=\frac{1}{m}\sum_{j=1}^{m}(Y_{(ij)}-Y_j)^2$ 为均方误差和计算公式，$Y_{(ij)}$ 是个体 i 在第 j 个训练集时用 $F(x,y)$ 计算出的值，Y_j 表示第 j 个训练集的实际值）并排序，保存适应值最高的个体。

(3) 执行变异，并按照染色体所含基因的多少决定变异的基因位个数，本文选择每个基因变异一个基因位的方法。设种群大小为 p_g，基因变异率为 p_m，基因长度为 g_l，染色体含有的基因个数为 c_p，则变异步骤如下：

1) 随机产生一个 0 到 1 之间的数 randompm，判断此数是否大于变异率 p_m，p_m 的取值随进化代数的增加而减小，并保证适应值高的染色体对应更小的变异率，适应值低的染

基金项目：国家自然科学基金资助项目“基于模糊神经网络的铜熔炼过程智能优化控制模型研究”(50364004)
作者简介：方旺盛(1963－)，男，副教授，主研方向：信息隐匿和演化计算；张克俊，硕士生；邵利平，博士生
收稿日期：2006-02-28 **E-mail：**fangwangsheng@163.com

色体对应更大的变异率，以实现种群的多样性。进化初期，取值较大，进化末期取值较小，定义为 p_m =P*(1-N/300)+P*(1- f_{Min} / f)（本文取 P 恒为 0.05，N 为当前进化的代数，300 为种群允许进化的最大代数，f_{Min} 为当前进化代中染色体适应值最小的值，f 为当前染色体的适应值）if randompm> p_m ，执行 2)，否则跳出(3)。

2) 随机产生一个 0 到 p_g 的数 randomnc，此数决定当前群体中将要执行变异操作的染色体号，继续执行 3)。

3) 根据randomnc号染色体所含基因的个数c_p（本文实验 1 中个数为 5，实验 2 中个数为 15），随机产生c_p个 0 到 g_i 的整数，这些 0 到 g_i 的整数值用来决定参加变异的基因位，按照语法生成规则，在基因头部，任何符号都可以变异成函数符号或者终点；在基因的尾部，终点只能变异成终点，本文中，每个基因仅变异一个基因位。

(4) 执行 IS、RIS、Gene 插串。

(5) 执行单点、两点、基因重组。

(6) 如果运行达到预先设定的最大代数或者获得了最大适应值，则停止运行，用图形输出结果并将结果保存到记录文件中，否则转到(2)继续运行。

1.2 IGEP 算法分析

1.2.1 复杂度分析

引理 算法的复杂度是 O（K*L*M），其中 K 为种群大小，L 为总进化代数，M 为训练集的长度。

证明 在算法中计算个体针对 M 个训练样本的复杂度为 O（M），算法需要计算种群中各个体的适应度值，故种群适应度计算的复杂度为 O（K*M），因算法最多进化 L 代，故算法复杂度为 O（K*L*M）。证毕。

1.2.2 收敛性分析

在基于 IGEP 的复杂函数参数识别反问题求解中，将适应度函数设计为

$$f_{fitness}(i)=1000\times\frac{1}{Ei+1}$$

其中，$E_i=\frac{1}{m}\sum_{j=1}^{m}(Y_{(ij)}-Y_j)^2$ 为均方误差和计算公式，简记为 SSE。设遗传变异率 $P_{mu}\le 0.5$，则可得如下两个结论[3]：

(1)IGEP 复杂函数反问题求解进化第 k 代的最小均方误差和的数学期望满足

$$E(SSE^k{}_{Min})\le E(SSE^0{}_{Min})-p_r\cdot p_t\cdot p_{all}\cdot e\cdot\sum_{i=1}^{k-1}E(b_i)$$

其中，p_r 为通过单点重组由上一代种群产生到下一代任一可能的染色体的概率且 $p_r=\frac{1}{L\cdot N\cdot(N-1)}$ ，p_r 为通过插串由上一代种群产生到下一代任一可能的染色体的概率，$p_t=\frac{2}{(h-2)\cdot(h+1)\cdot h}$ ，p_{all} 为一个染色体的所有位均发生变异的概率，且 $p_{all}=p_{mu}^L$ 。

其中，L 为染色体的长度，N 为种群大小，$e=Min(|SSE_i-SSE_j|),i,j=1,2,...,M$ （M 为 IGEP 搜索的群体空间中所有的 SSE 值不相同的染色体个数）。

(2)最小均方误差和以概率收敛，即对任意 $\varepsilon>0$，有

$$\lim_{k\to\infty}p(SSE^k{}_{Min}>\varepsilon)=0$$

其中 b_i 位第 i 代最小均方和小于第 $i-1$ 代最小均方和的染色体数量。

由以上分析可看出，基于 IGEP 的复杂函数参数识别反问题求解是依概率收敛到全局最优染色体的，但由于不是强收敛到全局最优点，因此不能排除收敛到局部最优点的可能。

2 算法对比分析

为了验证算法的有效性，进行了以下几个实验。实验中，使用最简单的函数集合 F={'+','-','*','/','sin','cos','sqrt','exp','ln','pow'}，其中 sin 表示 SIN()函数，cos 表示 COS()函数，sqrt 表示 sqrt()函数，exp 表示指数函数，ln 表示以 10 为底的对数函数，pow 表示幂函数。在此函数集合中，针对反问题的复杂性适当增加此函数集合项。实验表明，增加函数集合可以提高仿真精度。本文为了介绍方便，暂使用上面的简单函数集。终点集用 T={相应变量}。GEP 的相应参数参考文献[2]。

2.1 实验 1

在某化学反应里，测得生成物的浓度 y 与时间 t 的数据表如表 1 所示，试找出 t 与 y 的关系表达式。

表 1 生成物浓度与时间关系

t/in	1	2	3	5	6	7	8	9	10	11	12	13	14	15	16
Y	4	6.4	8	9.22	9.5	9.7	9.86	10	10.2	10.32	10.42	10.50	10.55	10.58	10.6

(1)传统方法[5]

文献[5]根据表 1 坐标分布，采用最小二乘法拟合得到两种模型，即：双曲线型和指数函数型。

1)双曲线型： $y=\frac{t}{at+b}$

采用最小二乘法拟合结果为：a=80.662 1，b=161.682 2，最小二乘误差均方差为 1.19×10^{-3}。（将数据代入文献[5]模型得到的实际均方差为 37.776 550。）

2)指数函数型： $y=ae^{\frac{b}{t}}$

采用最小二乘法拟合结果为：a=0.011 325，b=-1.056 7，最小二乘均方误差 0.34×10^{-3}（将数据代入文献[6]模型得到的实际均方差为：37.776 699。）

(2)神经网络[5]

在 Matlab6.5 下，采用 BP 神经网络计算该问题。设置一层隐含层含 5 个神经元，训练代数设为 2 000 代。运算 5 次得到最好模型，输入层和隐含层间权值向量为

$W1$ = (-1.076 7, - 0.186 8,-0.512 0, 0.421 6,-0.258 2)

隐含层和输出层间的权值向量为

$W2$ = (-1.151 9,-1.875 6,-2.048 8, 2.626 9,-0.552 8)

计算得到均方差为：0.242 439。

(3)经典GEP算法[7]

使用APS3.0[8]，经多次计算，得到较好的结果为平均均方误差 0.022 5，与实际结果进行比较如图 1（a）所示。其结果优于传统方法和神经网络方法。

(4)IGEP 算法

本文利用 IGEP 算法得到生成物浓度(y)与时间(x)之间数学模型为(参数 T 用 x 表示)：

$$Y=\cos((\frac{e^x}{x^x})^{e^{x^2}})+e^{x^{\frac{1}{4}}}+\cos(e^{(\sin(1))^x})+e^{\sin(\sqrt{\sin(\cos(1))\times x})}+\sin(\cos(1))$$

其中，e 为无理数 2.718 28，x 为仿真输入值，Y 为仿真输出值，与实际结果进行比较如图 1(b)所示，实验求出平均均方误差为 0.003 4。其结果优于传统方法、神经网络及经典 GEP。

2.2 实验 2

铜熔炼过程中，前 30 项熔炼数据如表 2 所示，试建立其

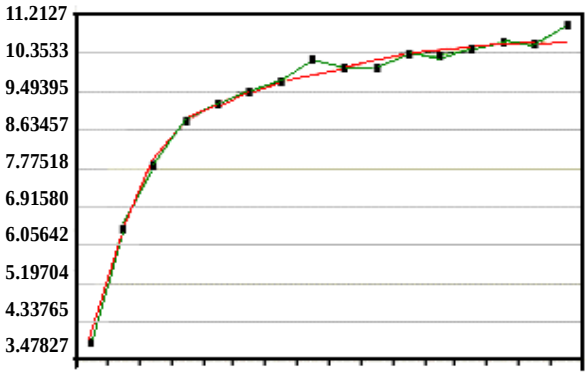
熔炼模型。

表 2 实验 2 的数据

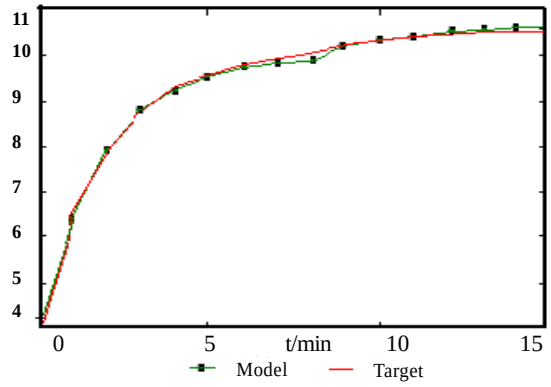
测试数据序号	数据变量 X1	数据变量 X2	输出值 Y	测试数据序号	数据变量 X1	数据变量 X2	输出值 Y
1	1.40	1.8	3.7	16	3.62	1.95	2.08
2	4.28	4.96	1.31	17	1.67	2.23	2.75
3	1.18	4.29	3.35	18	3.38	3.70	1.60
4	1.96	1.90	2.70	19	1.48	4.44	2.44
5	1.85	1.43	3.52	20	3.37	2.13	1.99
6	3.66	1.60	2.460	21	2.84	1.24	3.42
7	3.64	2.14	1.95	22	1.19	1.53	4.99
8	4.51	1.52	2.51	23	4.10	1.17	2.27
9	3.77	1.45	2.70	24	1.65	1.38	3.94
10	4.84	4.32	1.33	25	2.00	2.06	2.52
11	1.05	2.55	4.63	26	2.71	4.13	1.58
12	4.51	1.37	2.80	27	1.78	1.11	4.71
13	1.84	4.43	1.97	28	3.63	2.27	1.87
14	1.67	2.81	2.47	29	2.22	1.35	3.39
15	2.03	1.88	2.66	30	2.24	3.74	1.79

(1)经典GEP算法[7]

使用APS3.0[8]，经多次计算，得到较好的结果为平均均方误差 0.037 57。与实际结果进行比较如图 1(a)所示。



(a) 经典 GEP 结果



(b) IGEP 结果

图 1 经典 GEP 与 IGEP 在实验 1 问题上结果对比图

(2)IGEP 算法

本文利用 IGEP 算法得到较好的数学模型如下：

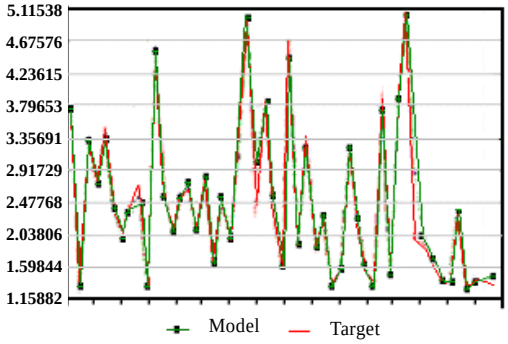
F(x,y)=

$$2\cos(x)+3\cos(\sqrt{x})+2\sqrt{x}+\sin(\cos(2x))+\sin(\sqrt{e^{e^{\frac{x}{\sin(e^{\cos(x)})}}}})$$

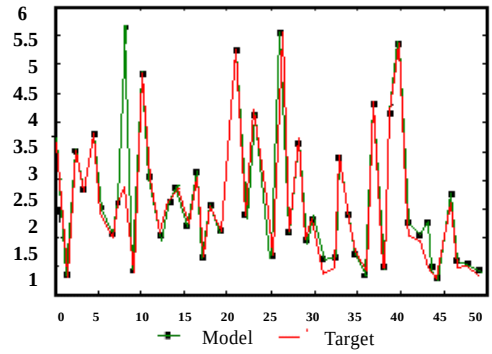
$$+\cos(\frac{y}{(\frac{y-x}{\sin x}+e^{y})})+\frac{\frac{\sin x}{y}+\sqrt{y}}{e^{y}}+\cos(\sin(x+y))$$

$$+\sin\sqrt{y^{\frac{1}{2}y^{2}x}+e^{\sin(y-x)e^{(y+e^{y})}}}+(\frac{e}{xy+y^{2}})^{y^{2x}}+\cos\sqrt{y}+1-e$$

与实际结果进行比较如图 2(b)所示。在图 2(b)中，实心线表示实际目标值，虚线加圆圈的为 IGEP 建立的模型值，实验求出平均均方误差为 0.024 702。



(a) 经典 GEP 结果



(b) IGEP 结果

图 2 经典 GEP 与 IGEP 在实验 2 问题上结果对比

3 结果分析

对上面两组实验结果分析表明，采用 IGEP 方法得到的模型比传统算法及经典 GEP 算法给出的模型更精确，MSE 值和 CC 值均优越于传统算法和经典 GEP 算法。在执行效率上 IGEP 也优越于经典 GEP，表 3 给出了 IGEP 与经典 GEP 在实验 1 和实验 2 问题上运行 100 次的结果。

表 3 IGEP 与 GEP 对比结果

实验名称	算法	得到较高适应值次数	平均所需代数	最短成功代数	最长所需代数
实验 1	IGEP	99 次	189	63	1 987
	GEP	97 次	273	117	2 658
实验 2	IGEP	96 次	269	102	2 865
	GEP	92 次	356	213	3 769

对比结果可以看出，IGEP 算法明显优于 GEP，平均所需代数仅为 GEP 的 69%和 76%，得到较高适应值次数也高于 GEP。

由于有丰富的基本函数可选，因此 IGEP 得到的模型结构较神经网络模型更丰富。通过上述两个实验实例可以看出，采用 IGEP 方法，可以对函数结构作灵活组合，因而采用这种方法所找到的模型比采用传统方法所找到的模型的拟合效

（下转第 205 页）