# 基于基因表达式编程挖掘函数关系*

黄晓冬, 唐常杰+, 李 智, 普东航, 曾令明, 廖 勇

(四川大学 计算机学院,四川 成都 610065)

# A Mining Functions Relationship Based on Gene Expression Programming

HUANG Xiao-Dong, TANG Chang-Jie+, LI Zhi, PU Dong-Hang, ZENG Ling-Ming, LIAO Yong

(Department of Computer Science, Sichuan University, Chengdu 610065, China)

+ Corresponding author: Phn: +86-28-85466105, E-mail: tangchangjie@cs.scu.edu.cn, http://www.scu.edu.cn

Received 2003-05-10; Accepted 2004-07-16

**Huang XD, Tang CJ, Li Z, Pu DH, Zeng LM, Liao Y. A mining functions relationship based on gene expression programming. *Journal of Software*, 2004,15(Suppl.):96~105.**

**Abstract**: In many scientific researches, people try to discover the hidden relationships among factors of some certain phenomena. Most of these relationships can be expressed as functions. This paper analyzes the features of function expression and introduces a function discovery method based on Gene Expression Programming. The MEM method can deal with complex functions having n expressions ($n$>1) in different domains as well as those having only one uniform expression. The article also evaluates the complexity and performance of the method and shows that the MEM method has relatively low complexity due to the use of BDM algorithm and proves that the probability of successful mining is 20%~80%, the average time elapsed is less than 10s.

**Key words**: GEP; KDD; uniform expression mining (UEM); multi-expression mining (MEM)

**摘 要**: 在许多科学研究中,人们希望揭示隐藏在现象背后的规律,并用函数关系来表示.分析了函数关系表达式挖掘技术的特点,提出了一种基于基因表达式编程的函数关系发现方法.MEM 方法能处理具有一致表达式的关系和具有不同分域表达式的复杂函数关系.论文对该方法的复杂度和性能做了评价,论证了 MEM 方法具有对数数量级的复杂度.实验结果显示,基于 GEP 的函数关系发现方法在采用较高变异概率时有很好的性能,对于不同的目标函数,挖掘成功率可以达到 20%~80%,且运行时间较短,成功挖掘平均耗时在 10 秒以内.

**关键词**: GEP;KDD;一致表达式挖掘(UEM);分域表达式挖掘(MEM)

人类探索自然规律的精确描述的历史是一部充满艰辛的数据挖掘史.千百年来,这种挖掘发现工作通常由人来完成.人们根据大量的观测数据进行不断的计算、摸索和试探.例如第谷从事 30 年的天文观察,积累了大量的观察材料,但是并没有发现行星运动的三大定律,而开普勒在第谷的基础上坚持几十年做大量的数学计算工

* Supported by the National Natural Science Foundation of China under Grant No.60073046 (国家自然科学基金); the National Research Foundation for the Doctoral Program of Higher Education of China under Grant No.20020610007 (教育部博士点基金)

作者简介: 黄晓冬(1978－),男,重庆人,硕士,主要研究领域为数据库,知识发现;唐常杰(1946－),男,教授,博士生导师,研究领域为数据库,知识发现;李智(1978－),男,硕士,主要研究领域为数据库,知识发现;普东航(1973－),男,硕士生,主要研究领域为数据库,知识发现;曾令明(1975－),男,硕士生,主要研究领域为数据库,知识发现;廖勇(1968－),男,硕士生,主要研究领域为数据库,知识发现.

作,才最终做出重大发现.人们希望利用计算机技术从观测数据中去粗存精、去伪存真、发现规律.

本文在函数关系发现方法方面做了下列工作:(1) 对函数关系发现这一特殊知识发现形式的特点进行了分析;(2) 提出并实现了具有任意维度的定义域上的一致表达式和分域表达式的挖掘方法;(3) 提出并实现了基于基因表达式编程(GEP)[1]的从科学实验数据中发现函数关系的 UEM(一致函数表达式的挖掘)和 MEM(分域函数表达式挖掘)方法以及 BDM(二域式挖掘)算法;(4) 指出了 MEM 方法具有对数数量级的复杂度,证明了对于给定样本集当 UEM 得到最优解的概率大于等于 0.5 时,MEM 方法得到正确分域函数表达式的概率为 1;(5) 对 UEM 在不同参数设置下的性能进行了实验和讨论.

## 1 函数关系表达式发现的特点和难点

数据挖掘技术是人类分析和发现能力的延伸,随着计算机技术的发展,出现了用计算机在数据集中发现公式的方法,如曲线拟合等,在速度上有了很大提高,但是,传统方法在哲学层次上的先天不足,例如多项式拟合在函数形状上或多或少有主观臆断成分,通常得出的是真实函数关系在某种形式上的近似,对于一些较复杂的关系,或在要求较高的环境下,不能达到理想的精确度.

函数发现任务具有以下一些特点和难点:

(1) 函数类型的未知性.在仅有观察数据而不了解事物内在机制时,对函数形式的假设,例如设定其为多项式或幂函数,或多或少带有主观意志,可能导致挖掘结果的不准确.

(2) 分域性(domain split).许多事物在客观上具有分域性质,例如水在不同温度区间上有 3 种状态,各状态下物理性质截然不同,显然难用一个函数表达式去准确表达所有的情况,分域函数是很好的解决方案.

本文将讨论从实验样本中挖掘函数表达式以及判定确定目标函数是否应分域和确定分域边界的技术.

## 2 相关工作

美国密执安大学的 Holland 教授于 20 世纪 60 年代提出了遗传算法的思想,到了 20 世纪 80 年代,在一系列研究工作基础上,由 Goldberg 进行归纳总结,形成了遗传算法的基本框架[2].遗传算法(genetic algorithms,简称 GA)是一种借鉴生物界自然选择和进化机制发展起来的高度并行、随机、自适应搜索算法.由于其具有健壮性,特别适合于处理传统搜索算法解决不好的复杂的和非线性问题[7].此前有人研究过利用遗传算法进行函数拟合,这些方法通常是在假定了函数类型(例如多项式)的基础上对系数进行寻优,这种搜索通常只能得到真实函数关系的近似,而不能准确的反映事物的本质.

2001 年 12 月,葡萄牙的 Candida Ferreira 在遗传算法的基础上提出了基因表达式编程(gene expression programming,简称 GEP)的概念[1],GEP 同传统的遗传算法和遗传编程(genetic programming,简称 GP)在一些主要步骤上很相似,但在个体的编码方法及结果的表现形式等方面又有明显的区别.关于 GEP 的研究和应用参见文献[1,3~7]等.

GEP 具有许多优点,如染色体简单,线性和紧凑,易于进行遗传操作等.Candida 对利用 GEP 进行函数表达式挖掘进行了讨论,但仅局限于单表达式的处理,而且对许多方面(例如对系数的处理上)进行了较大程度的简化.因此本文在更加复杂的情况下,对一致表达式和分域表达式的发现进行了更深入的研究.

## 3 一致表达式的发现

设 $F$:Dom→Range 是从多维定义域 Dom 到 Range 的函数,$D_1,D_2,\ldots,D_n$ 是 Dom 的子域,$Dom=D_1\cup D_2\cup\ldots\cup D_n$,$Exp_i$ 是一个解析表达式,若对任意向量 $v$ in $D_i$,$Exp_i(v)=F(v)$在 $D_i$ 上成立,则本文中称 $Exp_i$ 是 $F$ 在域 $D_i$ 上的分域表达式(sub domain expression).当 $n=1$ 时,称 $Exp_1$ 是 $F$ 的一致表达式(uniform expression).当实验数据涉及的问题比较复杂时,常常不能发现一致表达式,需要用分域表达式表示,而一致表达式挖掘又是分域表达式挖掘的基础,因此我们先研究函数一致表达式的挖掘(uniform expression mining,简称 UEM).

### 3.1 编 码

把一个问题的可行解从其解空间转换到算法所能处理的搜索空间的转换方法称为编码.在遗传算法的运行过程中,它不对所求解问题的实际决策变量直接进行操作,而是对表示可行解的个体编码施加选择、交叉、变异等遗传运算,达到优化的目的.同样,利用 GEP 进行优化求解时,也要对操作对象进行编码,形成基因组.编码方法在很大程度上决定了如何进行群体的遗传进化等运算以及这些运算的效率,因此是我们的挖掘过程中一个很关键的步骤.

对于一个函数关系 $z=f(x,y)$,我们将表达式 $f(x,y)$进行编码,形成基因.其过程可简单描述为:将表达式根据其语义表示为表达式树(expression tree,简称 ET);然后从上到下,从左至右的按层遍历 ET,得到的符号序列即为基因编码的有效部分.例如,对于表达式:

$$\sin(x^2+xy) \tag{1}$$

其对应的 ET 如图 1(a)所示(其中"*S*"表示正弦运算).对该 ET 进行按层遍历,我们得到序列:

$$S+**xxxy \tag{2}$$

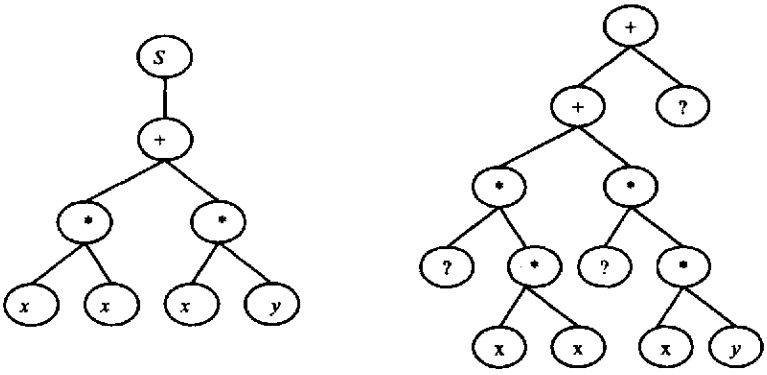


图 1 Expression tree

序列(2)被称为表达式(1)的 *K* 表达式(*K*-expression)[1],它构成了数学表达式的基因编码的主体.由于每一运算符的操作数个数是已知的(例如"S"是单目运算符而"+"是 2 目运算符)将 *K* 表达式按以上过程的逆过程进行便能解码得出对应的数学表达式.

GEP 中编码串的长度是固定的,它由位于前端的有效的 *K* 表达式和后面的填充部分组成.为了保证基因编码的有效性,编码被分为头部(head)和尾部(tail)两部分,头部可以出现运算符和变量,而尾部只能出现变量.设头部的长度为 $h$,文献[3]证明了,在运算符最多只有 2 个操作数的情况下(数学表达式中通常是这样),只要尾部长度 $t=h+1$ 就能保证编码的前 $k(1\leq k\leq$编码总长 $n)$个符号组成一个有效的 *K* 表达式,即不会出现运算符找不到操作数的情况,保证了遗传过程中不会产生无效个体及相应的程序不会因为空指针而出错.例如当 $h$ 为 5,则 $t=6$,于是式(1)的编码可能为

$$S+**xxxyxyyy \tag{3}$$

在进行解码时,利用前面 8 个符号(*K* 表达式)便得到完整的 ET,后面的填充部分将不再参加计算.采用统一长度编码的好处以及填充部分的作用将在遗传操作中体现出来.

实际应用问题往往比表达式(1)更复杂,可能在表达式中出现常量和系数.由于符号和数字在格式上的差异,在进行遗传操作的时候不便统一进行处理,作者在实现时采取了下列技巧:将表达式中的具体数值分离出来,在出现数字(常量或系数)的地方用一个统一的符号代替(例如"?"),而将具体这些数值依次存至一个向量中.相应的,其基因编码也分为两部分:表达式编码串和数值编码串.其中,数值编码串的长度($m$ 个数字)也是固定的,为保证足够的数字个数,我们令 $m=h$.

例如表达式:

$$5x^2+14xy+25 \tag{4}$$

其结构可用图 1(b)中的 ET 表示(其中数字已被"?"取代).在 $h$=9 的情况下,其可能的编码为

$$++?**?*?*xxxyxxyyxy\ 25\ 5\ 14\ 3\ 56\ 1\ 7\ 8\ 44 \tag{5}$$

解码时,在数值编码串中依次选取数值代替表达式中的"?",数值编码串中多余的数值在遗传操作中起作用.

### 3.2 适应度函数的设计

"适者生存"是生物在自然选择下进化的基本法则.适应度(fitness)是度量物种对环境的适应能力的指标.借鉴自然界的启示,遗传算法和 GEP 中使用适应度函数的返回值描述个体在优化计算中有可能达到或接近或有助于找到最优解的优良程度.因此,我们需要一个合适的适应度函数来计算各基因所对应个体的适应度,适应度函数的好坏直接影响着整个遗传操作的方向.

根据函数表达式的特点,本文采用基于误差的适应度函数,其构造要点如下:对于第 $i$ 个基因,其适应度函数的表达式为

$$f_i=\frac{1}{\dfrac{\sum_{j=1}^{n}\sqrt{\left|C_{i,j}-T_j\right|}}{n}+1} \tag{6}$$

其中,$C_{i,j}$ 表示根据第 $i$ 个基因对应的函数表达式利用第 $j$ 个样本中的变量数据求得的函数值;$T_j$ 表示第 $j$ 个样本中包含的实际测得的该目标函数值的真实值;$n$ 则为样本数量.例如:若某一代的群体中第 $i$ 个基因的表现型表示量 $x,y,z$ 之间可能的函数关系 $z=f(x,y)$,而实验测得第 $j$ 个样本为$(x_j,y_j,z_j)$,则 $C_{i,j}=f(x_j,y_j)$,而 $T_j=z_j$.

显然,对于其表现型与真实的函数关系完全吻合的基因 $G_i$,有 $C_{i,j}=T_j(j=1,2,\ldots,n)$,在样本数据无误差时,其适应度 $f_i=f_{max}=1$.在实际应用中,由于实验精度等种种原因,采集到的样本一般具有一定的误差,即使是与真实情况相对应的基因,其适应度也只能在一定程度上接近 1.

### 3.3 遗传操作

我们通过对基因群体实施若干次的选择、交叉、变异等操作,使其一代代的进化,以从中寻求最优的个体,从而得到问题解答.要点如下:

(1) GEP 的遗传操作开始于初始群体的形成.首先需为初始群体中每个基因编码生成表达式编码串和数值编码串.

(2) 首先定义运算符集合 $F$(如{+,−,*,/,...})和变量集合 $P$(如{$x,y,\ldots$}),$P$ 中变量个数根据所求问题而定.

(3) 从 $F$ 和 $P$ 中随机选择元素构成指定长度 $h$ 的表达式编码串头部,从 $P$ 中随机选择元素构成长度为 $t=h+1$ 的尾部,头、尾部构成表达式编码串.再随机生成 $h$ 个整数构成数值编码串(每一整数用 8 位格雷码表示).由于从基因表达式编码到ET的变换是按层进行的,因此随机生成的基因都是有效的,即可以正确解码为合法的函数表达式(虽然基因中有些基因座并不起作用).初始群体中个体应达到一定的数量,比如 100 个.

(4) 个体选择.采用轮盘赌算法(roulette sampling).个体的适应度越高,被复制到下一代的份数就越多.各个体在下一代群体中的期望生存数目为

$$N_i=N\cdot f_i/\sum_{j=1}^{N}f_j\ ,\ i=1,2,\ldots,N \tag{7}$$

其中 $N$ 为种群中个体总数.这样便保证好的基因能够更多的保留到下一代配对库.同时我们将当前适应度最高的基因另外保存一份,以免其在交叉或变异中被破坏掉.若发现比当前保存的基因适应度更高的基因,则用其替换当前的保存.

(5) 我们需要不断产生新的基因以寻求更好的个体.交叉是产生新个体的手段之一,分 3 步进行:(a) 对配对库中的个体进行随机配对;(b) 在配对个体中设定交叉点,由于基因本身由两部分组成,因此我们在表达式编码串和数值编码串中各随机设定一个交叉点;注意,交叉点必须位于两个相邻基因座之间,而不能将一个符号或整数从中破开;(c) 对交叉点前或后的两个个体部分结构进行互换,生成两个新个体.交叉概率 $p_c$ 控制着交叉操作

被使用的频度.

(6) 变异操作.对个体编码串中以变异概率 $P_m$ 随机指定的某一个或几个基因座做变异操作.对于头部,可将运算符变异为变量或反之;而在尾部,不能将变量变异为运算符.这样,无论如何变异,所生成的新基因都是有效的.例如,将*x*+yyyxy 变异为*Q*+yyyxy(其中"Q"表示开平方运算.)虽然对应的 ET 在结构上发生了很大变化,但总是合法的.变异在 GEP 中起到维持解群体多样性的作用.

我们将在实验部分对交叉和变异概率取值对于挖掘成功率的影响进行讨论.

算法 1 描述了一致表达式(uniform expression)的挖掘过程:

**算法 1.** UEM(一致函数表达式的挖掘).

输入:Cases(样本数据集),Start,End.

输出:Best_Exp(最优表达式及其适应度).

```
Algorithm UEM (Cases,Start,End)
1. S=Initial Seed Population of N Genes;          //建立初始种群
2. Best_Exp.Gene=null;                            //初始最优解
3. Best_Exp.Fitness=0;
4. m=MaxGeneration;                               //确定遗传操作代数
5. While (m>0)
6.    {S=Selection(S);                            //选择
7.    S=CrossOver(S);                             //交叉
8.    S=Mutation(S);                              //变异
9.    If (Best_Exp.Fitness<biggest fitness value in S) Then
10.    {Best_Exp.Gene=the Gene with the biggest fitness value in S;   //保存最佳基因
11.    Best_Exp.Fitness=the biggest fitness value in S;}
12.    m=m-1;}
13. return(Best_Exp);
```

说明:算法 1 的目标函数有 $n$ 个自变量,参数 Start 和 End 为两个向量,表示为$(s_1,s_2,\ldots,s_n)$和$(e_1,e_2,\ldots,e_n)$,取满足第 $i$ 个自变量值位于区间$[s_i,e_i]$内($i$=1,2,…,$n$)的那些样本参与 GEP 运算.这为后面的分域函数挖掘中调用该算法提供了方便.返回值 Best_Exp 是一结构,它包含了经过若干代遗传操作所求得的最优基因表达式和该基因的适应度两项内容.

## 4 分域表达式挖掘

对于复杂对象,须采用分域函数才能准确的表示各个量之间的函数关系,以揭示其内在规律.我们在一致表达式挖掘(UEM)的基础上进一步进行了分域函数中多个表达式挖掘(multi-expression mining,简称 MEM)的研究.为了叙述上的简便和直观,我们以单自变量函数,即每一样本只有两维的情况为例进行讨论.

### 4.1 分域边界的确定

对于一个给定的样本集,首先要判断样本各维之间存在的函数关系是否应该分域,在二维情况下表现为分段.本文采用宏观效果作为判别准则,即定义一个适应度阈值 $F=1-\varepsilon$,其中$\varepsilon$为一较小正常数.对于区间$[t_i,t_{i+1}]$上的函数关系,首先进行一致表达式挖掘(UEM),若得到的函数表达式 exp 的适应度 Fitness≥F,则称 exp 是令人满意的(如图 2(a)所示).如果经 UEM 挖掘得不到一个令人满意的表达式,则$[t_i,t_{i+1}]$内的函数关系应是分段的,本文把相邻分段间的交界点称为分段点(高维情况下为分域边界).

对于一个存在函数分段的区间 $R=[t_i,t_{i+1}]$,如果能够找到分段点 $t_d$,则可以通过分别在$[t_i,t_d]$和$[t_d,t_{i+1}]$上进行 UEM 挖掘找到各分段的表达式.本文设计了一种称为二域式挖掘(binary domain mining,简称 BDM)的方法来确定一区间上的分段点.其思想类似于二分法搜索:定义分段点可能存在的区间为 $R_d$(显然,一开始 $R_d$ 为整个待分

段区间 $R$),将 $R_d$ 的中点 $t_{d1}=(t_i+t_{i+1})/2$ 假定为分段点,分别在 $R_1=[t_i,t_{d1}]$和 $R_2=[t_{d1},t_{i+1}]$上进行 UEM 挖掘,得到表达式 exp1 和 exp2,这时,可能有几种情况:

(1) 若 exp1 和 exp2 的适应度均大于或等于 $F$(如图 2(b)所示),则我们认为 $t_{d1}$ 就是分段点,于是将 exp1 和 exp2 作为结果保存;

(2) 若 exp1 和 exp2 中其一(不妨假定为 exp1)的适应度小于 $F$,则认为分段点所在区间应为 $R_d\cap R_1=R_1$(如图 2(c)所示).于是令 $R_d=R_1$,重新假定 $R_d$ 的中点 $t_{d2}=(t_i+t_{d1})/2$ 为分段点在 $R$ 上重复进行二域式挖掘.以此类推逐步缩小分段点可能存在的范围 $R_d$,直至正确分段或遇到情况 3.

(3) 若 exp1 和 exp2 的适应度均小于 $F$(如图 2(d)所示),则认为 $R_1$ 和 $R_2$ 上均存在分段点,于是假定 $R_d\cap R_1$ 和 $R_d\cap R_2$ 各自的中点为分段点分别在 $R_1$ 和 $R_2$ 上进行二域式挖掘.

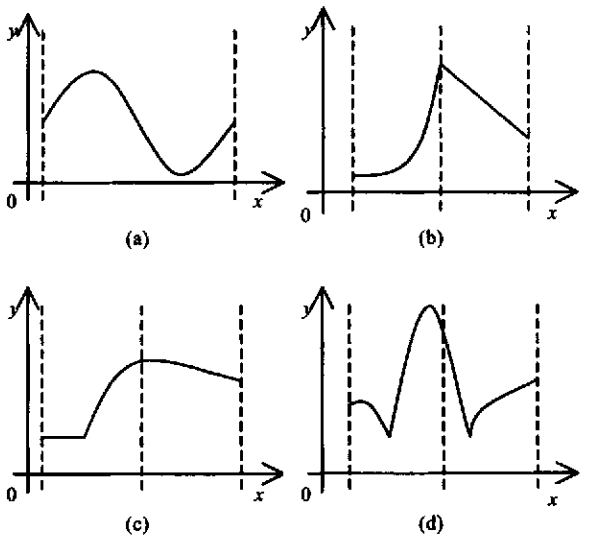


图 2 二域挖掘中各种情况示意

本文设计了二域式挖掘递归算法 BDM(算法 2).为便于之后的分段合并,采用数据库表 DOMAINS 来保存挖掘到的各表达式信息.DOMAINS 表结构如下:

| 序号 | 起始点 | 结束点 | 函数表达式 | 适应度 |
|---|---|---|---|---|

**算法 2.** BDM(二域式挖掘).

输入:Cases(样本数据集),Start,End,left,right.

输出:在表 DOMAINS 中记录各分段情况.

```
Algorithm BDM (Cases,Start,End,left,right).
Begin
1.  S_Point=(left+right)/2;
2.  Exp1=UEM (Cases,Start,S_Point);
3.  Exp2=UEM (Cases,S_Point,End);
4.  If ((Exp1.Fitness>=F) and (Exp1.Fitness>=F)) Then
5.     record the information in table DOMAINS;
6.     else if((Exp1.Fitness>=F) and (Exp1.Fitness<F)) Then
7.     BDM (Cases,Start,End,S_Point,right);
8.     else if ((Exp1.Fitness<F) and (Exp1.Fitness>=F)) Then
9.     BDM (Cases,Start,End,left,S_Point);
```

```
10.    else {BDM (Cases,Start,S_Point,left,S_Point);
11.          BDM (Cases,S_Point,End,S_Point,right);}
end;
```

说明:参数 Start,End 标明了需进行二域挖掘的区间;left,right 标明分段点所在区间.通常在第 1 次调用时有 Start=left,End=right.

### 4.2 分域的合并

由于 GEP 迭代运算产生最优解的概率不为 1,以及我们分段策略中为了降低复杂度进行的有意分段的原因,BDM 挖掘得到的结果往往存在多余分段,即将本来可用一个函数表达式表示的区间分为 $k$ 段,$k\geq 2$.这时我们需要对分段进行合并,降低冗余.

这种合并是两两进行的.在 BDM 挖掘得到的 DOMAINS 表中,两相邻区间 $R_i$ 和 $R_{i+1}$ 内的函数表达式 $\exp_i$ 和 $\exp_{i+1}$ 若完全相同,则可简单的将区间合并,使两项成为一项.但更复杂的情况是两表达式在形式上不同而本质上相同或极其近似,例如:$y=4x^3+2x^2$ 与 $y=2x(2x^2+x)$.这时便不能通过串比较来判定.这里我们采用将其中一段内的样本代入另一段对应的函数表达式进行适应度计算的方法来判定.通常,较大的区间内含有更多的样本,因此我们认为较大区间内挖掘到的函数表达式更可靠,假设 $R_i$ 较 $R_{i+1}$ 宽,则将自变量位于 $R_{i+1}$ 内的样本代入 $\exp_i$ 根据式(6)计算适应度 $f$,若 $f\geq F$,则认为 $\exp_i$ 对于 $R_{i+1}$ 同样适用,于是将 $R_i$ 和 $R_{i+1}$ 合并,将 $\exp_i$ 作为合并后区间 $R_i \cup R_{i+1}$ 的函数表达式.

完整的分域函数表达式挖掘过程可用算法 3 描述:

**算法 3.** MEM(分域函数表达式挖掘).

输入:Cases(样本数据集),Start,End.

输出:在表 DOMAINS 中记录各分段情况.

```
Algorithm MEM (Cases,Start,End)
1.  Exp=UEM (Cases,Start,End);
2.  If (Exp.Fitness>=F) Then
3.    record the information of the only sub domain;
4.    else {BDM (Cases,Start,End,Start,End);
5.      P points to the first row of DOMAINS;
6.      while (P⟨⟩null)
7.      {merge all the rows with expressions equal to P->Expression;
8.       P points to the next row;}
```

## 5 复杂度分析

在UEM中,我们所采用的编码方法使得初始种群以及遗传操作过程中的各代中不会产生无效个体,因此遗传优化的效率较高.

在确定分域边界(同时得到了各分域函数表达式)时采用的 BDM 以每次 1/2 的速度缩小分域边界可能存在的范围,快速的找到分域边界位置.同样以 2 维情况为例,设具有 1 个分段点的区间 $R$ 的宽度为 $W$,若要求分段点位置精确到 1,则采用 BDM 需要进行的查找次数最多为 $\log_2^W$ 次,平均约 $\log_2^W-1$ 次;而如果采用顺序查找,平均需进行 $W/2$ 次.例如,对于自变量在区间[0,1024]内的样本集进行分段点的确定,采用 BDM 最多仅需 $\log_2^{1024}=10$ 次查找便能精确到 1,进行不超过 14 次查找便能精确到 0.1(如果需要的话);而采用顺序查找要精确到 1 平均需要 1024/2=512 次查找,精确到 0.1 则需要 5 120 次.对分段点的每次查找需进行 2 次 UEM,因此利用 BDM 查找分段点需进行的 UEM 次数最多为 $2\log_2^W$ 次,这是完全可以接受的.

前面提到,基于基因表达式编程(GEP)的 UEM 得到最优解的概率不为 1,那么,样本集经过 MEM 方法挖掘是否能够得到正确的分域函数表达式(分域数为 1 时为一致表达式)呢?事实上我们可以得到这样的结论:

**定理 1.** 对于给定的样本集,当 UEM 得到最优解的概率大于等于 0.5 时,MEM 方法得到正确分域函数表达式的概率为 1.

证明:考虑分域函数中存在分域表达式 $Exp_i$ 的子域 $D_i$,设一次 UEM 挖掘得到正确表达式的概率为 $p_s$,若未得到则将对其进行二域式挖掘(此间进行的分域将在二域挖掘结束后进行合并).因此,MEM 方法得到 $D_i$ 上正确表达式的概率为

$$p_m=p_s+(1-p_s)*(p_{m1}*p_{m2}) \tag{8}$$

其中 $p_{m1},p_{m2}$ 分别为二域式挖掘中两个子域 $D_{i1}$ 和 $D_{i2}$ 上得到正确表达式的概率($D_i=D_{i1}\cup D_{i2}$).由于对 $D_{i1}$ 和 $D_{i2}$ 的处理方法与 $D_i$ 完全相同(见递归算法 2),我们有理由认为 $p_{m1}=p_{m2}=p_m$,代入式(8),有 $p_m=p_s+(1-p_s)p_m^2$.于是可以得到

$$p_m=\begin{cases}1, & 0.5\le ps\le 1\\ ps/(1-ps), & 0\le ps<0.5\end{cases} \tag{9}$$

可见,要通过一次 MEM 成功的挖掘函数各段表达式只需要满足 $p_s\ge 0.5$ 即可,而这对于基于 GEP 的 UEM 而言是容易做到的.即便 $p_s$ 不能达到 0.5,我们也可通过重复使用平均 $1/p_m$ 次 MEM 方法以得到正确的结果.例如 $p_s=0.3$ 时,$p_m\approx 0.43$,得到正确结果平均需进行的 MEM 次数仅为约 2.3 次.

## 6 实验与性能分析

作者用 C++开发了一个交互式的实验系统,称为 GEPxP(GEP experiment platform).前面已经证明 MEM 的挖掘成功率取决于 UEM 的成功率,因此我们主要实验讨论了在不同的参数设置情况下对各种样本数据进行 UEM 挖掘的性能.

本文们选择了 3 个不同维数和形式的数学函数作为实验函数:

$F1$: $z=x^3+3x+1$

$F2$: $s=tv+0.5at^2$

$F3$: $z=\sin(2wt+5)$

实验函数在挖掘过程中又称为目标函数.实验用样本数据均根据 $F1$,$F2$ 和 $F3$ 产生.

实验中采用的部分 GEP 参数见表 1.

表 1 GEP 实验参数

| Population size | Gene length | Generation |
|---|---|---|
| 100 | 17 | 1 000 |

### 6.1 交叉和变异概率对挖掘的影响

交叉概率 $p_c$ 和变异概率 $p_m$ 控制着进化中开辟新的搜索空间的速度和方式.

GEPxP 在不同 $p_c$ 下对 3 个目标函数进行挖掘.从图 3 显示的实验结果中可以看出,对于实验函数 $F1$,$F2$ 和 $F3$,挖掘成功率随着交叉概率的增大呈波动的略微上升的趋势,但很不明显.

我们再在不同 $p_m$ 下对 3 个目标函数进行挖掘.从图 4 中可以看到,与前一实验相比,挖掘成功率均随着变异概率的增大呈明显上升趋势,在 0.6~1 间达到最大值.

传统遗传算法中普遍认为交叉应该是进化中产生新个体的主要手段,而变异只是作为一种辅助操作,因此 $p_c$ 应取较大值而 $p_m$ 取较小值.然而我们的实验结果显示,在 GEP 中使用较高的变异概率明显有利于提高挖掘成功率,而交叉概率的取值对性能影响并不大.我们认为这是由 GEP 的特殊性决定的.

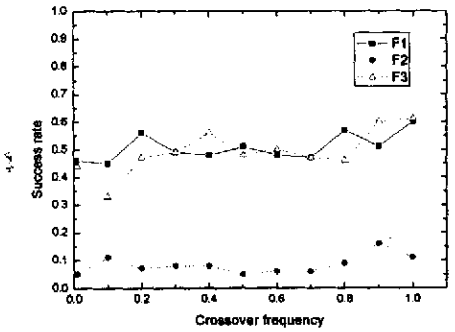


图 3 交叉概率对挖掘成功率的影响

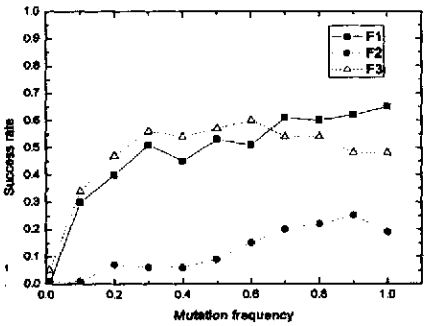


图 4 变异概率对挖掘成功率的影响

### 6.2 性能对比分析

传统遗传算法理论认为过高的变异概率将使得算法性能接近于随机搜索.而根据上节中得出的结论,我们应该在 GEP 中采用较高的变异概率(0.6~1).因此我们在保证二者搜索空间相同的情况下将 UEM 方法与随机搜索的成功率进行了对比.

从图 5 所示实验结果可以看到,对于 *F1* 和 *F3* 这样相对简单的函数关系,GEPFM 与随机搜索的成功率相差 1 个数量级以上;而对于 *F2* 这样较复杂的关系,在我们的实验中还未能用随机搜索法找到,而采用基于 GEP 的方法成功率可以超过 20%.

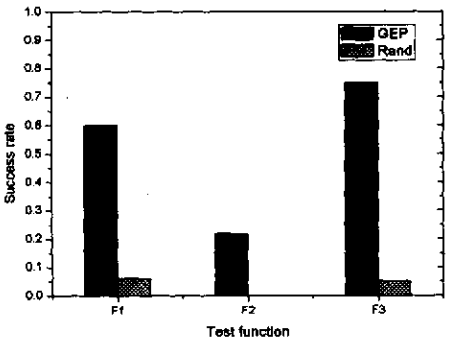


图 5 UEM 与随机搜索的成功率比较

一个有实用价值的方法除了较高的成功率外,还应该具有能够接受的较短的运行时间.我们对 GEPxP 的时间性能进行了观察.从表 3 中的实验结果可以看到,对于 *F1*,*F2* 和 *F3*,成功挖掘到函数关系的平均用时都在 10s 以下.

表 3 GEPxP 时间性能实验

| 目标函数 | 样本数 | 挖掘次数 | 成功次数 | 总用时(s) | 成功挖掘平均用时(s) |
|---|---|---|---|---|---|
| *F1* | 12 | 100 | 65 | 176 | 2.71 |
| *F2* | 17 | 100 | 26 | 249 | 9.58 |
| *F3* | 16 | 100 | 81 | 134 | 1.65 |

## 7 小 结

本文对根据实验样本数据发现对应的函数关系表达式进行了探讨,利用 GEP 较好的处理了函数表达式形式的多样性和未知性,以统一的观点解决具有一致表达式的函数和分域函数的挖掘问题,提出并实现了基于 GEP 的 UEM(一致函数表达式挖掘)、BDM(二域式挖掘算法)和 MEM(分域函数表达式挖掘)方法,并论证了 MEM 方法具有对数数量级的复杂度.从结果我们发现,基于 GEP 的函数关系发现方法在采用较高变异概率时

具有很好的性能,对于不同的目标函数,挖掘成功率可以达到 20%~80%,且运行时间较短,成功挖掘平均耗时在10s以内.

本文涉及的基于基因表达式编程的公式发现是一个新的研究分支.大量问题等待我们去研究.今后的主要工作包括建立更加完备的函数发现系统,进而可根据不同领域的特殊性进行更加有针对性和更准确的挖掘工作等.

**References:**

[1] Candida F. Gene expression programming: A new adaptive algorithm for solving problems. Complex Systems, 2001,13(2):87~129.

[2] Goldberg DE. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, 1989.

[3] Zuo J, Tang CJ, Zhang TQ. Mining predicate association rule by gene expression programming. In: Proc. of the Int'l Conf. for Web Information Age 2002. LNCS Vol. 2419, 2002. 92~103.

[4] Ferreira, C. Gene expression programming in problem solving. Invited tutorial of the 6th Online World Conference on Soft Computing in Industrial Applications, 2001. 10~24.

[5] Ferreira, C. Mutation, transposition, and recombination: An analysis of the evolutionary dynamics. In: Proc. of the 4th Int'l Workshop on Frontiers in Evolutionary Algorithms. 2002. 614~617.

[6] Ferreira, C. Discovery of the boolean functions to the best density-classification rules using gene expression programming. In: Proc. of the 4th European Conf. on Genetic Programming. LNCS Vol.2278, 2002. 51~60.

[7] Zuo J, Tang CJ, Li C, Yuan CA, Chen AL. Time series prediction based on gene expression programming. In: Proc. of the Int'l Conf. for Web Information Age 2004. LNCS, 2004. 55~64.