基于朴素基因表达式编程挖掘紧致函数

朱明放[1]，唐常杰[2]，陈安龙[3]，代术成[2]，于中华[2]

(1. 江苏技术师范学院计算机工程学院 江苏 常州 213003; 2. 四川大学计算机学院 成都 610065;

3. 电子科技大学计算机科学与工程学院 成都 610054)

【摘要】基因表达式编程(GEP)是一种基因型和表现型相分离的进化新模型，为了挖掘紧致的函数关系，分析了进化系统各因素对挖掘紧致函数的影响，提出了带紧致压力的适应度函数来进化函数紧致解。实验表明，带有紧致压力的适应度函数能自动进化计算机程序，适合挖掘的紧致关系，在挖掘紧致函数中，朴素基因表达式编程(NGEP)比GEP提高效率21.7%，与不带压力的系统相比，GEP的平均压缩了31.2%，NGEP系统平均压缩了42.5%；NGEP较GEP更容易发现紧致解，且函数表达形式更容易理解，丰富了NGEP理论.

关 键 词 紧致压力; 紧致解; 函数发现问题; 朴素基因表达式编程

中图分类号 TP311.6 **文献标识码** A doi:10.3969/j.issn.1001-0548.2010.02.028

Mining Compact Function Based on Naïve Gene Expression Programming

ZHU Ming-fang[1], TANG Chang-jie[2], CHEN An-long[3], DAI Shu-cheng[2], and YU Zhong-hua[2]

(1. School of Computer Engineering, Jiangsu Teachers University of Technology Changzhou Jiangsu 213001;

2. School of Computer, Sichuan University Chengdu 610065;

3. School of Computer Science and Engineering, University of Electronic Science and Technology of China Chengdu 610054)

Abstract Gene Expression Programming (GEP) is a new member of evolutionary algorithm family, and it is an artificial genotype/phenotype system. Aiming to discover compact mathematical functions for function finding, this study analyzes the factors that greatly affect the efficiency of GEP, proposes the fitness function with pressure parameter, and implements a naïve gene expression programming (NGEP) for compact function mining tasks. Extensive experiments show that the proposed fitness function with compact pressure can automatically mine the compact functions as well as an alternative strategy to find compact results, and NGEP boosts the convergence speed by 21.7% than GEP, in addition, the results are more understandable than that are found by GEP. Compared with the evolution system without compact pressure, the average compact rate are 31.2% in GEP and 42.5% in NGEP, respectively, which shows that NGEP is easier to find compact results than GEP and the results are more comprehensive than traditional GEP.

Key words compact pressure; compact solving; finding function problem; naïve gene expression programming

基因表达式编程(gene expression programming，GEP)是数据挖掘的新方法[1]，该方法属于遗传算法家族，其特点是将基因型和表现型分离。GEP技术在数据挖掘领域得到深入研究，被应用于符号回归[1]、函数发现[2-3]、分类[4-5]、聚类[6]、关联规则[7]、时间序列预测[8]等领域。

函数发现问题是GEP最典型和最成功的应用领域，但传统GEP挖掘到的函数有时很复杂，难以理解，影响使用。如逻辑合成问题可能有多个完美解，即正确求值的解，其中非紧致解的实现较复杂，需要较多的门电路和更多的运行时间。实践呼唤人们研究问题的完美紧致解的挖掘方法。获得紧致解有两种思路：(1) 对较简单问题，用传统GEP方法发现模型，人工简化或专用软件简化；(2) 直接在进化中得到紧致解。

1 GEP基因结构

GEP的解为计算机程序，即基因的表现型，编码为固定长度的多基因组成的染色体，遗传操作在染色体(基因的基因型)上进行[1]。

收稿日期：2008 – 07 – 25; 修回日期：2009 – 12 – 25

基金项目：国家自然科学基金(60773169)

作者简介：朱明放(1970 –)，男，博士，副教授，主要从事数据库与知识工程等方面的研究.

GEP的基因由头部和尾部组成，头部可以包含函数符号(取自函数集FS)和终结符号(取自终结符集TS)，尾部只能包含终结符号，目的是保证基因表达的是一个合法的计算机程序。基因的头部长度h和尾部长度t之间的关系为：

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

式中　n为函数集中函数的最大参数目数。

例1　设函数集FS={Q,×,/,–,+}和终结符集TS={a,b}，则n=2；若基因头部长度h=15，则尾部(用粗体标记)长度t=16，基因长度g=15+16=31。

考虑如下一个基因：

G_1: /aQ/b×ab/Qa×b×–***ababaababbabbbba***

基因G_1经Karva语言解码得到的表达树ET如图1所示。该表达树有8个节点，即该基因的1～8位为开放阅读区(open reading frames，ORFs)，9～31位为无码区。

GEP的染色体(基因组)由一个或几个等长的基因组成，每个基因解码为一个子表达树(sub-ET)，各子表达树相互作用，形成更复杂的多子单元的表达树(multi-subunit ET)。

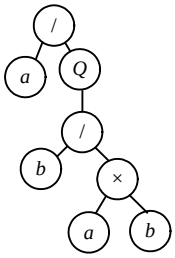


图1　G_1的表达树

2　朴素基因表达式编程

2.1　朴素GEP及性质

文献[3]提出了朴素基因表达式(NGEP)的初步概念，并研究了其在函数自动建模上的性能，本节将完整提出朴素基因表达式的概念。

GEP基因头长和尾长满足式(1)，当n=2时，对照完全二叉树的节点从上到下、从左到右的编号顺序，基因尾部基因位号恰好对应完全二叉树的叶子节点编号，头部基因位号对应完全二叉树非叶子节点编号。而二叉树具有良好的结构和性质，利用其性质特点，有望加速基因的解码过程。

定义 1　在经典GEP模型中，若函数集函数的最大目数为2，基因编码采用完全二叉树编码方式，则称该计算模型为朴素基因表达式编程(naïve GEP，NGEP)。

定理 1　NGEP的基因同构于完全二叉树。

证明　GEP的基因长度$g=h+t$，当函数集中函数的最大参数目数为2时，有$t=h+1$和$g=2h+1$。取二叉树的节点个数为g，构造映射。基因座子号i映射到完全二叉树节点的编号i，则映射是一一映射。根据完全二叉树的性质，具有h个非叶子节点，必有$h+1$个叶子节点，二叉树的总节点数为$2h+1$，分别是基因的头部、尾部和基因的长度。映射满足基因头部位号与二叉树非叶子节点编号，尾部基因位号与二叉树叶子编号一对一关系。证毕。

定理1表明，对于函数参数目数不超过2的GEP基因，对其表达可以通过一棵完全二叉树实现。将在GEP中使用的遗传操作应用到基因的二叉树表示中，只要维持完全二叉树前半部分符号既可是函数符号也可以是终结符号，后半部分只能是终结符号的结构特点，该二叉树编码的基因总能解码为合法的计算机程序。需要说明的是，对于操作目数为1的函数，其运算对象是其左子树根节点。

2.2　基因的二叉树解码

NGEP解码为：(1) 标记有效基因位，顺次扫描基因头部，标记第一个基因位，若是终结符号，则标记结束；若是函数符号，则根据它的运算目数标记其孩子节点序号。递归地，直到标记为终结符号时止，并记录标记的符号数目。(2) 解析为表达树，对标记过的基因位，若为基因的ORF，按编号生成的二叉树就是基因的表达树。

例2　设G_2:/+a×/+***aaaaaaaaa***，标记有效基因位0～4和7～10，未被标记的基因位及位串为基因的中性码和中性区。

将标记的基因位号填写到完全二叉树的相应节点编号，如图2所示，其数学表达式为(a×a+a/a)/a。

G_2用GEP中的Karva语言解码的表达树如图3所示，其数学表达式为((a+a)×a+1)/a。

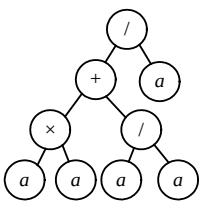


图2　G_2的二叉树方式解码

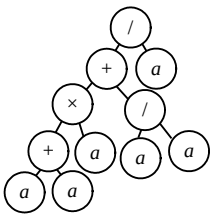


图3　G_2的Karva语言解码

3　影响GEP进化的因素

影响系统有效进化的因素包括种群的大小和初始种群质量，染色体的结构和特征，遗传算子的类型、概率和能力、适应度函数的设计、适应环境的质量等。其中最积极的因素是基因结构、染色体结

构和适应度函数。

3.1 GEP基因结构特征

GEP进化过程可以描述为：从一个种群开始，在一定的适应环境(问题的适应实例集)下，经过以个体适应度为依据的选择过程和带有基因突变的复制过程，形成新一代群体，如此循环，直到满足终止条件。

遗传算法进化的速度和质量，取决于基因表达方式和能遍历问题空间的遗传算子设计。GEP基因是长度固定的线性串，其尾部的作用是保证经过约束很少而能力强大的遗传算子的操作，产生的基因为有效的，即表达合法的计算机程序。

3.2 中性区和多基因技术

定义 2 设E为基因表达式中的基因代码：(1) 如果E在解码时其遗传信息不被表达，称E为基因型(显形)中性代码，其在染色体中占据的区域称作基因型(显形)中性区；(2) 基因码E解码时其信息被表达，但没有对个体的表现作出任何贡献，称E为表现型(隐形)中性代码，其在染色体中占据的区域称作表现型(隐形)中性区。

中性区及中性突变是GEP区别于其他遗传计算模型的重要特征。GEP基因中无码串存在，符合木村资生假设[9]，即中性突变的积累在进化中扮演至关重要的角色，而染色体的无码区域是中性突变的理想场所。

GEP中性区的控制有两种方式：(1) 在单基因染色体系统中，通过增加基因长度实现；(2) 在多基因染色体系统中，通过增加染色体的基因数量实现。实践表明，在染色体同样大小的冗余下，多基因染色体系统比单基因染色体系统的效率高。

GEP独有的多基因染色体结构和中性区是其成功的关键，其优点是进化系统解决实际问题时速度快、成功率高；缺点是挖掘到的知识庞大臃肿、不易理解。在GEP应用中，希望其既能有效进化，又能发现问题的紧致解，即发现问题的紧致解而不降低系统进化效率。

3.3 紧致性度量

GEP基因中存在中性区，是完备的基因型/表现型系统。GEP中性的表现有两种形式：(1) 在基因型上，基因存在中性无码区域，解码时在表现型上不表现出来，如例1的无码区代码；(2) 基因位于有效代码区，而解码的子表达树对整个表达树的结果没有作用，如表达式的加、减0；乘、除1等。

例3 设G_3://a×/+***aaaaaaaaa***，采用NGEP的基因解码规则，得到的表达树如图4所示，其ORFs长度为9，是不连续的几个片段长度之和，其余的代码是基因型中性代码，相应的数学表达为((*a*×*a*)/(*a*/*a*))/*a*。

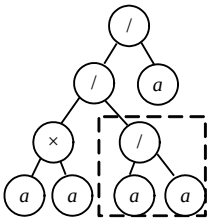


图4 G_3的二叉树方式解码

从图4可以看出，表达树的第2层的第一个节点的右子树(虚线矩形框内)的值为1，对表达式的值没有贡献，故4个节点(/,/,*a*,*a*)的代码属于表现型(隐形)中性代码。

G_3用Karva语言解码得到的表达树如图5所示，其数学表达式为(((*a*+*a*)×*a*)/(*a*/*a*))/*a*。基因G_3的表达树第2层第一个节点的右子树(虚线矩形框内)的值是1，对表达式无贡献，表达树有4个冗余结点(/,/,*a*,*a*)，属于表现型(隐形)中性代码。

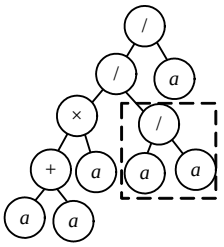


图5 G_1的Karva语言解码

可见，在保证进化效率前提下，需要消除染色体中的显性中性代码和隐形中性代码。

定义 3 进化系统得到的完备解集中，解码的表达树中节点数目最少的解称为紧致解。

4 具有紧致压力的适应度函数

进化系统对问题的成功解决依赖适应度函数的设计和环境选择的质量两个方面。

4.1 适应度函数设计

函数发现问题的适应度函数分为基于绝对误差、相对误差和统计指标3类。常见的适应度函数为带有选择带宽的绝对和相对误差的适应度函数[10]：

$$\mathrm{rFit}_i=\sum_{j=1}^{n}(R-|P_{ij}-T_j|) \tag{2}$$

$$\mathrm{rFit}_i=\sum_{j=1}^{n}\left(R-\left|\frac{P_{ij}-T_j}{T_j}100\right|\right) \tag{3}$$

式中 R为选择带宽；P_{ij}为第i个染色体对第j个适应实例的预测值；T_j为第j个适应实例的真实值；rFit_i

是第i个染色体对环境的适应度值，前缀r表示原始，以区分本文将设计的适应度函数。

4.2 紧致压力的适应度函数设计

以往的适应度函数设计过多地注重问题的解决，轻视问题解的简洁明了性，从而致使挖掘到的结果难于理解。

为使有效进化和获取问题的解紧致二者并举，设计具有紧致压力的适应度函数。设计准则：(1) 保证原有的进化效率不下降；(2) 保证获得问题的解的准确度不下降；(3) 能自动发现紧致解。文献[10]中有关于节俭压力适应度的讨论，主要用于决策树构建过程的树剪枝算法，本文采用表达树节点数目来度量解的复杂性。

根据以上设计原则，将紧致压力融入适应度函数，设计的适应度描述解的准确程度和描述解的紧致程度两部分构成，即设计的适应度函数为：

$$\mathrm{Fit}=\mathrm{rFit}+\mathrm{cFit} \tag{4}$$

式中　rFit是原始适应度函数；cFit是个体紧致性的适应度函数，前缀c表示紧致。

依据准则(2)，设计cFit时其值不能混入rFit内影响解的准确性。考虑Fit的值为非负实数，设计以小数点为界，整数部分为rFit值，小数部分为cFit值，保证适应度函数的这种结构，可满足设计要求。

如果rFit不是整数，可以适当放大或缩小或截断，cFit采用解的节点数来描述。

设染色体采用G个基因组成，每个基因的长度为g，为表达简便，设采用固定的连接函数(连接函数不参与进化)，在表达树节点数目计算中，不考虑连接函数带来的节点数。最复杂的表达是整条染色体的符号都参与信息表达，表达树的节点个数最大为$S_{\max}=G\times g$(连接函数的结点不计)；表达树节点个数最小为$S_{\min}=G$。

设计的cFit具有表达树节点数越少其值越大的特征，即要求cFit函数值为正值，为表达树节点数目的单调减函数。构造函数：

$$\mathrm{cFit}(S_i)=K\times(S_{\max}-S_i)/(S_{\max}-S_{\min}) \tag{5}$$

式中　S_i是第i个染色体对应表达树的节点数目；K是比例系数。可见cFit的值在0～K之间，其值越大，则S_i越小，解的紧致程度越高。

好的适应度函数应使群体中个体表现有差别，但不太悬殊，否则会造成进化迟钝或早熟。在设计K值时，应考虑对rFit函数值进行相应变换，如将适应度函数值进行缩放。

5 实验和性能分析

5.1 数据和进化环境

测试函数为$y=a^3+a^2+a+1$，在[−10,10]区间随机生成10个样本，选择式(2)为适应度函数，带宽为R=100，绝对精度p=0.01，则$\mathrm{rFit}_{\max}$=1 000。对带有紧致压力的适应度函数，其原始适应度函数取绝对误差适应度函数，紧致压力函数选用式(5)，其中K=9/10，进化代数为50，种群大小为30。函数符号集FS={+,−,×,/}，基因头部长度h=6，基因个数为4，连接函数取+，变异率为0.038 5，逆串率为0.1，单点重组和2点重组率为0.3，插串率和根插串率为0.1。

5.2 不带紧致压力的适应度进化性能

独立运行GEP系统100次，统计挖掘到的函数关系，染色体的表达树最大节点个数为53，最小为23，平均为39.36，平均进化代数为11.78。进化中获得的一个最大节点数目的解和最小节点数目的解对应的染色体分别如图6a和6b所示。

```
−/−−×+aaaaaaa××××//aaaaaaa
    +−××−aaaaaaa++a−+/aaaaaaa   [5,53]
a. GEP最大节点数目的解对应的染色体
  +aa+/aaaaaaaa+a−×+×aaaaaaa
   /aa+/+aaaaaaa×aaaaaaaaaaaa   [29,23]
b. GEP最小节点数目的解对应的染色体
−−+×−×aaaaaaa++//+−aaaaaaa
    /×/×//aaaaaaa+×××/−aaaaaaa   [4,55]
c. NGEP最大节点数目的解对应的染色体
   /aa+++aaaaaaa×+×/−×aaaaaaa
  ×a×+−/aaaaaaaa+++a+aaaaaaa   [7,27]
d. NGEP最小节点数目的解对应的染色体
```

图6　不带紧致压力下GEP和NGEP进化的染色体

图中，方括号中数字是进化代数和染色体解码得到的表达树的节点数，以下相同。同样地，运行NGEP系统，考察挖掘到的函数关系，染色体解码得到的表达树节点的最大数目为55，最小为27，平均为36.88；平均进化代数为11.12。进化中获得的一个最大节点数目的解和最小节点数目的解对应的染色体分别如图6c和6d所示。

分析实验结果，发现在选择不带紧致压力的适应度函数时，NGEP较GEP进化速度略快，函数关系也较为紧凑。

5.3 具有紧致压力的适应度进化性能

染色体的表达树最大节点个数$S_{\max}$=55，最小节点个数7，设计的适应度函数为：

$$\mathrm{Fit}_i=\sum_{j=1}^{n}(R-|P_{ij}-T_j|)+\frac{9}{10}\left(\frac{55-S_i}{55-7}\right) \tag{6}$$

可见，最大适应度函数值为$\mathrm{Fit}_{\max}$=1 000.9。

本例中，明显地S_i>7，$rFit_{max}$=1 000，故可增大K值，甚至超过1，以便分化个体，增强评价能力。

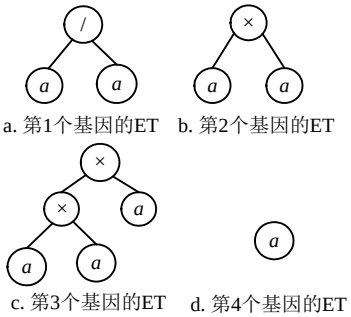
分别独立运行两个系统100次。在GEP系统中，得到的最大表达树节点数为43，最小表达树节点为数15，平均为27.080；最大适应度1 000.750，最小1 000.225，平均为1 000.524；平均进化代数30.500。其中，一个最大节点数目的解和最小节点数目的解对应的的染色体分别如图7a和图7b所示。

NGEP系统中，得到的表达树节点数最大为35，最小为15，平均为21.20；适应度最大值为1 000.75，最小为1 000.375，平均为1 000.634，平均进化代数为33.46，其中一个最大节点数目的解和紧致解对应的染色分别如图7c和图7d所示。

/−a/a−aaaaaaa×+−−×+aaaaaaa
+/a+aaaaaaaaa+××−+aaaaaaaa [10,43]
:1 000.225
a. GEP最大节点数目的解对应的染色体
×a×aaaaaaaaaaa/aa/aaaaaaa
×aa+/+aaaaaaa/aaa×aaaaaaaa [30,15]
:1 000.75
b. GEP最小节点数目的解对应的染色体
//aaa-aaaaaaa−//−+/aaaaaaa
+×+a××aaaaaaa/aa×//aaaaaaa [26,35]
:1 000.375
c. NGEP最大节点数目的解对应的染色体
/aa/a/aaaaaaa×aa×a×aaaaaaa
××aaa×aaaaaaaa×××++/aaaaaaa [18,15]
:1 000.75
d. NGEP紧致解对应的染色体

图7 紧致压力作用下GEP和NGEP进化的染色体

图8是NGEP发现的紧致解的染色体图7d的各基因的表达树，表明NGEP发现的解是紧致解，即含连接函数节点数，最少15个节点。



a. 第1个基因的ET b. 第2个基因的ET

c. 第3个基因的ET d. 第4个基因的ET

图8 NGEP的紧致解表达树

在紧致压力适应度的GEP和NGEP系统，统计最少节点的紧致解次数，分别是4次和18次。对该问题，NGEP的进化能力是GEP的4.5倍。

实验表明：(1) NGEP和GEP系统都能在紧致压力下发现紧致解。(2) NGEP更有发现紧致解的趋势，在相同的冗余下，基于NGEP技术的进化系统能较好发现问题的紧致解，既能保持一定的精度，又能简化描述的目的。

6 结 论

本文设计了基于完全二叉树的基因编码方案，提出了朴素基因表达式编程模型，证明了二叉树的结构与GEP的基因结构是同构关系。指出紧致解是生产实践和科学研究的追求目标，分析了影响挖掘紧致解的进化模型中的因素，提出了设计带有紧致压力的适应度函数设计方法。对不带紧致压力和带有紧致压力的适应度函数在GEP系统和NGEP系统中进行的实验表明，GEP系统平均压缩了31.2%，NGEP系统平均压缩了42.5%(NGEP比GEP提高效率21.7%)；在不带和带有紧致压力情况下，NGEP和GEP二者进化速度相当，但NGEP较GEP更容易发现紧致解。

本文研究工作得到江苏技术师范学院博士启动资金(KYY09001)的资助，在此表示感谢。

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] 黄晓冬, 唐常杰, 李 智, 等. 基于基因表达式编程挖掘函数关系[J]. 软件学报, 2004, 15(增刊): 96-105.
HUANG Xiao-dong, TANG Chang-jie, LI Zhi, et al. Mining functions relationship based on gene expression programming[J]. Journal of Software, 2004, 15(suppl): P96-105.

[3] 朱明放, 唐常杰, 陈 瑜, 等. 基于朴素基因表达式编程的函数自动建模[J]. 四川大学学报(工程科学版), 2008, 40(4): 126-131.
ZHU Ming-fang, TANG Chang-jie, CHEN Yu, et al. Function automatic modeling based on naive gene expression programming[J]. Journal of Sichuan University (Engineering Science Edition), 2008, 40(4):126-131.

[4] DUAN Lei, TANG Chang-jie, WEI Da-gang, et al. Distance guided classification with gene expression programming [C]//ADMA 2006. [S.l.]: LNAI 4093, 2006: 239-246.

[5] 张 赪, 蔡之华. 代价敏感的GEP分类算法实现[J]. 电子科技大学学报, 2007, 36(6): 1319-1321.
ZHANG Cheng, CAI Zhi-hua. Cost-sensitive classification by gene expression programming[J]. Journal of University of Electronic Science and Technology of China, 2007, 36(6): 1319-1321.

[6] 陈 瑜, 唐常杰, 叶尚玉, 等. 基于基因表达式编程的自动聚类方法[J]. 四川大学学报(工程科学版), 2007, 39(6): 107-112.

(下转第310页)