文章编号:1001 - 9081(2009)12 - 3280 - 03

# 基于堆栈解码的元胞基因表达式编程算法

杨 柳[1],何 锫[1,2],潘小海[1]
(1. 长沙理工大学 计算机与通信工程学院,长沙 410014; 2. 武汉大学 软件工程国家重点实验室,武汉 430072)
(yangliu198496@163.com)

**摘 要**:基因表达式编程(GEP)算法在评价个体适应度时需要将染色体转换为表达式树,并且在求解复杂问题过程中,由于多样性不足仍出现早熟收敛。针对以上问题,提出一种基于堆栈解码的元胞基因表达式编程算法(SD-CGEP)。利用堆栈直接对染色体进行解码和适应度评价,可以提高算法的运行速度;通过引入元胞自动机模型,从而提高算法跳出局部最优的能力。符号回归实验表明,SD-CGEP 算法在演化效率和预测精度上均超过传统 GP、GEP 算法。

**关键词**:基因表达式编程算法;元胞自动机;解码;符号回归
**中图分类号**: TP311 **文献标志码**:A

# Cellular gene expression programming algorithm based on stack decoding method

YANG Liu[1], HE Pei[1,2], PAN Xiao-hai[1]
(1. *School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha Hunan* 410014, *China*; 2. *State Key Laboratory of Software Engineering, Wuhan University, Wuhan Hubei* 430072, *China*)

**Abstract:** A novel Gene Expression Programming (GEP) algorithm named Stack Decoding Based Cellular Gene Expression Programming (SD-CGEP) was proposed. It did not need to transform the chromosome into expression tree, but directly used stack for decoding and evaluating chromosome to accelerate the evolution rate. Meanwhile, it imported cellular automata to avoid the problem of premature convergence. The result shows that SD-CGEP outperforms the conventional Genetic Programming (GP) and GEP in evolutionary efficiency and prediction accuracy.

**Key words:** Gene Expression Programming (GEP) algorithm; cellular automata; decoding; symbolic regression

## 0 引言

基因表达式编程算法(Gene Expression Programming, GEP)[1]是一种基于基因组和表现型的新型遗传算法[2],已被广泛地成功应用于符号回归、时间序列分析和分类等领域[3-5]。在 GEP 中,个体被编码为线性具有固定长度的字符串(基因组),这种结构在适应度评价过程中被转换成大小和形状不同的表达式树(表现型)。基因组和表现型分离的结构使得 GEP 比遗传程序设计算法[6](Genetic Programming, GP)具有更强的解空间搜索能力。元胞自动机(Cellular Automata, CA),又称点格自动机,最早由冯诺依曼提出,是一种时间、空间及状态均离散的局部网格动力学模型[7]。元胞自动机可视为由一个元胞空间和定义于该空间的变换函数组成,可以模拟复杂结构和过程的模型,已应用于许多领域[8-10]。

GEP 在计算个体适应度值时,需要反复构建和遍历表达式树,大大影响了演化效率。另外,GEP 在进化过程中种群多样性损失过快,仍存在早熟现象。本文在基本 GEP 算法的基础上,利用堆栈直接对染色体进行解码和适应度评价,提出了一种新的 GEP 解码方法(Stack Decoding, SD);同时以元胞自动机模型为框架,提出了基于堆栈解码的元胞基因表达式编程算法(Stack Decoding Based Cellular Gene Expression Programming, SD-CGEP)。

## 1 GEP 算法的基本原理

### 1.1 GEP 的编码和解码

GEP 的基因由头部和尾部组成。其中头部包含终结符号和函数符号,而尾部只由终结符号组成。假设基因头部长度为 $h$,那么其尾部长度 $t$ 由式(1)获得,其中 $n$ 是函数具有的最大参数个数。

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

假设有如下基因:

$$+ \ * \ - abcd \tag{2}$$

其头部长度为3,则尾部长度 $t = 3*(2-1)+1 = 4$。GEP 按照从左到右的顺序读取基因中的字符,根据语义规则,采用广度优先的顺序构成表达式树(图 1),然后通过中序遍历表达式树得到表达式 $(a*b)+(c-d)$。

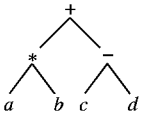


图 1 GEP 表达式树

### 1.2 适应度评价函数和算法流程

在 GEP 中,有两种适应度评价函数:

$$f_i = \sum_{j=1}^{C_t} (M - |C_{(i,j)} - T_{(j)}|) \tag{3}$$

**收稿日期**:2009 - 06 - 22;**修回日期**:2009 - 08 - 27。 **基金项目**:武汉大学软件工程国家重点实验室资助项目(SKLSE 20080701)。

**作者简介**:杨柳(1985 - ),女,湖南华容人,硕士研究生,主要研究方向:演化计算; 何锫(1963 - ),男,湖南长沙人,教授,博士,主要研究方向:软件工程; 潘小海(1982 - ),男,湖南邵阳人,硕士研究生,主要研究方向:智能软件系统。

$$f_i = \sum_{j=1}^{C_t}\left(M - \left|\frac{C_{(i,j)} - T_{(j)}}{T_{(j)}} \times 100\right|\right) \quad (4)$$

其中:$M$ 是一个常数,表示种群的选择范围;$C_{(i,j)}$ 表示第 $i$ 个染色体利用函数关系式在第 $j$ 个样本中的变量数据所求得的函数值;$T_{(j)}$ 表示第 $j$ 个样本包含的实际测量值;$C_t$ 表示样本的数目。

传统 GEP 算法的流程如下:1)初始化种群;2)按适应度函数对个体进行适应度评价;3)利用选择、变异、转录、重组遗传算子产生新种群;4)保存最优个体到下一代;5)若达到最大进化代数或计算精度,则进化结束,否则返回 2)[1]。

## 2　基于堆栈解码的元胞基因表达式编程算法

### 2.1　基于堆栈的 GEP 解码方法

传统 GEP 在计算个体染色体适应度值时,需先构建个体染色体对应的表达式树,然后中序遍历表达式树并结合测试数据集来计算染色体的适应度值。由于反复构建和遍历表达式树,对整个进化算法的效率有很大影响。本文利用堆栈直接对染色体进行解码和适应度评价提出了一种新的染色体解码方法(SD)。

文献[11]提出了一种基于前缀式描述的基因表达式编程算法(Prefix Gene Expression Programming, PGEP),该算法采用先序遍历树的方式进行染色体和表达式树的转换,具体步骤为:基因的首字符对应树的根节点,从左到右逐个读取基因串的字符,并按照从左到右,深度优先的顺序构造表达式树,直到树中所有叶节点都为终结符。其中,每个函数节点连接的子分支数目为该节点的参数个数,而终结符只作为叶节点。按照该方法将基因(2)转换为图 2 所示的表达式树,该表达式树对应的表达式为 $(a-b)*c+d$。

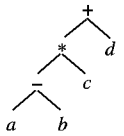


图 2　PGEP 表达式树

受到上述表达式树构造方法的启发,SD 方法采用堆栈直接对染色体进行解码和适应度评价,具体步骤为:从最后一位开始向前扫描基因,将所遇到的终结符依次放入堆栈中,一旦遇到运算符就从栈顶取出该运算符参数个数的终结符进行相应的运算,并把运算结果放入栈顶。扫描完一遍基因后,栈顶就保存着该基因对应的表达式值。以基因 $+^{*}-abcd$ 为例,SD 方法的解码和适应度评价过程如图 3 所示。

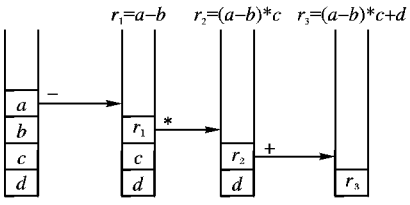


图 3　SD 方法解码和适应度评价过程

对比发现,采用 SD 方法解码得到的表达式与图 2 表达式树对应的表达式是一致的。SD 方法利用堆栈直接对染色体解码和适应度评价,无需反复构建和遍历表达式树,比传统 GEP 的解码方法具有更高的效率。

### 2.2　基于 SD 的元胞 GEP 算法

实验发现采用轮盘选择方法的 GEP 在早期进化收敛比较快,后期却不能维持种群多样性,这主要是由于前期种群内个体适应度值差距比较大,而轮盘选择偏好于高适应值个体,造成大量好的基因片段没有被继承下来,导致算法早熟收敛。为了进一步提高 GEP 算法求解问题的精确度,本文在 SD 方法的基础上,通过引入元胞自动机模型提出了基于堆栈解码的元胞基因表达式编程算法(SD-CGEP)。

元胞自动机由元胞、状态、邻居和规则四部分组成。元胞自动机中的元胞状态依据一个已定义的局部规则,按照相邻元胞和元胞本身的前一时刻的值同步更新[7]。本文使用的二维元胞自动机由四方网格组成,每个网格代表一个元胞,所有的元胞构成元胞空间。理论上,元胞空间在各维上是无限延展的,但在实际应用中,无法在计算机上实现这一理想条件,因此,需定义边界条件,本文采用周期型,即对二维空间,上下相接,左右相接,形成一个拓扑的圆环面,形似车胎或甜点圈。本文采用 Moore 型邻居模型,一个元胞周围的 8 个元胞称为该元胞的邻居,如图 4 所示。在某一时刻一个元胞只能有一种状态,而且该状态只能从一个有限的离散状态集中取值。规则是根据元胞当前状态及其邻居状态确定下一时刻该元胞状态的动力学函数。

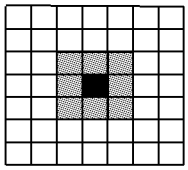

图 4　二维元胞自动机的 Moore 型邻居模型

SD-CGEP 将每个个体染色体定义为元胞,状态为染色体及其对应的适应度值,采用选择、重组和变异遗传操作来改变下一时刻元胞的状态。对于每个元胞,首先随机产生一个个体染色体并且评价其适应度值。在执行重组操作时,无需将种群的所有个体按适应度大小进行排序和采用轮盘选择方法,而是选择当前个体及其邻居中具有最好适应度值的个体作为父本。最后,下一时刻的元胞状态取进化得到具有较好适应度值的子代个体。

SD-CGEP 实现伪代码:

```
for every cell i do parallel
  generate_random_individual( C_i );        // 随机产生新个体 C_i
  evaluate( C_i );                          // 评价 C_i 的适应度值
end for
while not LastGeneration do
  for every cell i do parallel
    P = random( );                          //产生一个随机数 P
    if ( P < P_c ) then                                  //重组操作
      选择 C_i 邻居中具有最优适应度值的个体 C_j
      ( u , v ) = CrossOver( C_i , C_j );
    endif
    P = random( );
    if ( P < P_m ) then                                  //变异操作
      u = mutation( u );
      v = mutation( v );
    endif
    C_i = best_fitness( u , v );
                        //用较好适应度的子代个体替换当前个体
  endfor
endwhile
```

SD-CGEP 的个体只与周围的邻居相互作用,将好的基因片段慢慢地传播到整个种群,让具有不同适应度值的个体都有机会寻找更好的基因模式,维持了进化过程中种群多样性,从而避免算法早熟收敛。

## 3 实验与结果

实验1 目的是验证基于SD解码方法的GEP算法(简称SDGEP)的运行速度优于传统GEP算法。用函数$y = 5x^4 + 4x^3 + 3x^2 + 2x + 1(x \in [-5,5])$随机产生20个样本,通过SDGEP和GEP算法演化发现一个好的函数来拟合。算法参数设置如下:函数集为{+,-,*,/},终端集为{x},进化代数为1000,基因个数为4,联接函数为+,变异概率为0.03,重组概率为0.3,转录概率为0.1,适应度评价函数采用式(3)。对于不同的参数设置,每个算法运行20次。实验结果如表1所示。

从中可看出:在相同的种群大小和染色体长度下,SDGEP的成功率与传统GEP基本相同,说明其表示问题的能力不低于传统GEP算法。另外,SDGEP无需反复构建和遍历表达式树,因此在同样条件下其演化时间大大降低,平均运行速度比传统GEP算法提高了30%~50%。

**表1 SDGEP和GEP运行速度的比较**

| 参数 | 种群大小 | 染色体长度 | GEP | | SDGEP | |
|---|---|---|---|---|---|---|
| | | | 平均时间/s | 成功率/% | 平均时间/s | 成功率/% |
| 比较1 | 50 | 71 | 3.96 | 80 | 2.75 | 80 |
| 比较2 | 100 | 71 | 7.02 | 90 | 4.82 | 90 |
| 比较3 | 50 | 95 | 5.23 | 85 | 3.27 | 85 |
| 比较4 | 100 | 95 | 11.67 | 95 | 6.29 | 100 |

实验2 目的是验证SD-CGEP算法通过引入元胞自动机模型,在求解符号回归问题时比传统GEP算法具有更高的精确度。实验数据来自于文献[12],这是一个1979年—1996年关于病虫害流行程度的统计数据,如表2所示(其中$X$表示感病种种植面积百分比,$Y$表示头年秋苗平均普遍率,$Z$表示冬季积雪天数,$T$表示病虫害流行等级)。要求根据1979年—1996年的数据记录建立一个病虫害流行等级的预测模型,并以1996年的数据进行检验。算法的参数设置如下:基因头长为12,基因个数为6,种群大小为400,最大进化代数为1500,SD-CGEP和GEP的交叉概率分别为0.3和0.8,变异概率为0.03,转录概率为0.1,网格的高度和宽度均为20,函数集为{+,-,*,/,sin,cos,ln,exp},终结符集为{x,y,z,r ∈ [0,20]},适应度评价函数采用式(4),实验结果如表2所示。

运行GEP算法10次,得到的最佳模型为:

$$(((\cos(\cos(x)) + \sin(\sin(\sin((x/9.628652530135 3))))) + (\sin(\sin(\ln(((\exp(\sin(\cos(z))) + z)/9.745060601000 78)))) + (y/(\ln(y) - z)))) + ((\sin(y) + (\cos(z)/x)) + (\exp((\sin((x/(11.847106890124 8 + \sin(10.026127786545 9)))) + \sin(10.562503889691 6))) + \sin(\cos(\sin(\cos(11.154780730916 2))))))))$$

该模型的拟合误差为1.3611。

运行SD-CGEP算法10次,得到最佳模型为:

$$(((\cos(\cos(x)) + (12.723498003153 1/x)) + (\sin((y \times \cos(\cos(\ln(\sin(y)))))) + \ln(z))) + ((\cos(14.743535565305 4) + (\cos(z)/12.945847903036 7)) + (\sin((x/(9.200712315170 56 + \cos(\sin((\sin((\sin(y) - x)) + \sin(\ln((x/z))))))))) + \sin(10.113822127753 9))))$$

该模型的拟合误差为0.3066。为了便于比较,表2中列出了文献[12]使用GP演化所得到的结果。从实验结果可以看出,通过引入元胞自动机模型的SD-CGEP得到的演化模型的解比传统GEP和GP更精确。

**表2 预测模型的样本集及预测结果比较**

| 年份 | 实际数据 | | | | 预测值 | | | 相对误差/% | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $X$/% | $Y$/% | $Z$ | $T$ | GP[12] | GEP | SD-CGEP | GP[12] | GEP | SD-CGEP |
| 1979 | 61.0 | 0.405 | 12 | 3 | 2.8045 | 2.7711 | 2.8146 | 6.52 | 7.62 | 6.18 |
| 1980 | 64.8 | 0.397 | 16 | 3 | 3.1759 | 3.2361 | 3.0103 | 5.86 | 7.87 | 0.34 |
| 1981 | 50.8 | 0.002 | 10 | 1 | 0.8361 | 1.0155 | 0.9869 | 16.39 | 1.55 | 1.31 |
| 1982 | 52.9 | 0.317 | 18 | 2 | 2.3570 | 1.9333 | 2.0190 | 17.85 | 3.33 | 0.95 |
| 1983 | 61.2 | 0.111 | 18 | 3 | 3.0339 | 2.8182 | 3.0005 | 1.13 | 6.05 | 0.01 |
| 1984 | 76.0 | 0.521 | 22 | 4 | 3.5130 | 4.0205 | 4.0003 | 12.18 | 0.51 | 0.01 |
| 1985 | 80.4 | 0.887 | 39 | 5 | 5.0256 | 4.9333 | 4.9996 | 0.51 | 1.33 | 0.01 |
| 1986 | 50.9 | 0.391 | 6 | 1 | 1.1936 | 1.1822 | 0.9880 | 19.36 | 18.22 | 1.20 |
| 1987 | 50.4 | 0.082 | 10 | 1 | 0.9467 | 0.9757 | 0.9622 | 5.33 | 2.42 | 3.78 |
| 1988 | 41.2 | 0.081 | 9 | 1 | 1.2078 | 0.9876 | 1.0004 | 20.78 | 1.23 | 0.04 |
| 1989 | 60.0 | 0.900 | 25 | 3 | 3.0076 | 3.0602 | 3.0028 | 0.25 | 2.00 | 0.09 |
| 1990 | 80.0 | 1.910 | 27 | 5 | 4.9322 | 4.9933 | 5.0210 | 1.36 | 0.13 | 0.42 |
| 1991 | 70.0 | 0.750 | 9 | 3 | 2.9028 | 3.2462 | 2.9970 | 3.24 | 8.20 | 0.10 |
| 1996 | 80.0 | 1.020 | 36 | 5 | 5.0478 | 5.0127 | 4.9995 | 0.96 | 0.25 | 0.01 |

## 4 结语

GEP结合了GA和GP的优点,具有很强的函数挖掘能力。本文分析了传统GEP算法的不足,提出了SD-CGEP算法,该算法通过堆栈直接对染色体进行解码和适应度评价来提高算法的演化效率。同时引入元胞自动机模型维持进化过程中种群多样性,避免了算法早熟收敛。进一步研究工作在于,将SD-CGEP算法应用到分类等更为复杂的实际问题中。

(下转第3286页)

的平均收益接近15.6%,而收益的峰值可达到29.73%(表2的最右一列)。与此相反,建立一个个性化的背景模型也存在性能降低的情况,但如图4所示,平均损失不超过2.5%。

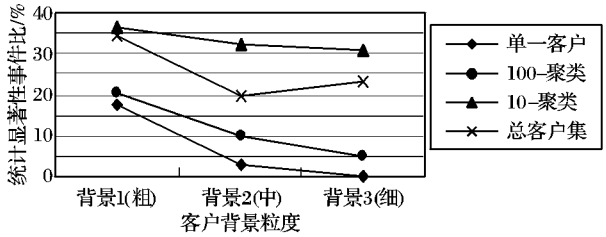


图6 性能损失时各客户分组粒度的统计显著性事件

3)客户背景粒度影响了对客户行为的预测准确程度。对客户背景知道得越细,对客户行为的预测就越准确。分析表2的最右边一列,唯一的负值仅发生在最粗的背景信息粒度上,背景粒度越细,性能收益也就越高。当分析单元是其他情况(表2向左移)时,虽然负值个数有所增加(特别是对于"10-聚类"的情况),但图3所示的平均收益率和平均损失率表明:背景粒度的性能收益在较宽的中带部分,从7.33%(最粗粒度)到13.94%(最细粒度)不等;而背景粒度导致的性能损失在非常窄的低带部分,绝对值从2.68%到4.31%不等,其最高点仅在中等背景粒度时出现,并且不超过4.5%。图5进一步说明了对于背景模型优于非背景模型的情况,除"10-聚类"模型的中粒度背景到细粒度背景外,统计显著性事件的数目随背景粒度的不断变细而增长。而图6进一步说明了对于相反的情况,除"总客户集"模型的中粒度背景到细粒度背景外,统计显著性事件的数目随着背景信息粒度的不断变细而衰减,特别是在"单一客户"模型的细背景上,其统计显著性事件的数目为0。这一事实说明:在"单一客户"模型的细背景上,没有非背景模型优于背景模型且差异是统计显著的事件。

## 4 结语

客户资源是商业企业的生命力,定量地分析和研究针对客户背景的预测模型(特别是个性化预测模型)的性能,能更好地预测客户下一步的购买意向。本文就单类型的背景数据集进行了研究,并证实:不同粒度的客户背景都不同程度地增强了对客户购买行为预测的准确度。

**参考文献:**

[1] BERRY M J A, LINOFF G S. Data mining techniques: For marketing, sales, and customer relationship management [M]. 2nd ed. Indianapolis: Wiley Publishing, 2003.

[2] LILIEN G L, KOTLER P, MOORTHY S K. Marketing models [M]. Upper Saddle River: Prentice Hall, 1992.

[3] ADOMAVICIUS G, SANKARANARAYANAN R, SEN S, *et al*. Incorporating contextual information in recommender systems using a multidimensional approach [J]. ACM Transactions on Information Systems, 2005, 23(1): 103 - 145.

[4] BETTMAN J R, LUCE M F, PAYNE J W. Constructive consumer choice processes [J]. Journal of Consumer Research, 1998, 25(3): 187 - 217.

[5] DEY A K, ABOWD G D, SALBER D. A conceptual frame work and a toolkit for supporting the rapid prototyping of context-aware application [J]. Human Computer Interaction, 2001, 16(2): 97 - 166.

[6] FRIEDMAN N, GEIGER D, GOLDSZMIDT M. Bayesian network classifier [J]. Machine Learning, 1997, 29(1): 131 - 163.

[7] LU HONG-JUN, LIU HONG-YAN. Decision tables: Scalable classification exploring RDBMS capabilities [C]// Proceedings of the 26th International Conference on VLDB. Cairo: Morgan Kaufmann, 2000: 373 - 384.

[8] KOTLER P. Marketing management [M]. 11th ed. New Jersey: Prentice Hall, 2003.

[9] JIANG T, TUZHILIN A. Segmenting customers from population to individuals: Does 1-to-1 keep your customers forever? [J]. IEEE Transactions on Knowledge and Data Engineering, 2006, 18(10): 1297 - 1311.

[10] DUNHAM M. 数据挖掘教程[M]. 郭崇慧, 田凤占, 靳晓明, 等译. 北京, 清华大学出版社, 2005.

[11] HANLEY J A, MCNEIL B J. The meaning and use of the area under a receiver operating characteristic (ROC) curve [J]. Radiology, 1982, 143(1): 29 - 36.

(上接第3282页)

**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] 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.

[3] 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 - 54.

[4] ZUO J, TANG C J, LI C, *et al*. Time series prediction based on gene expression programming [C]//WAIM 2004: 5th International Conference. Berlin: Springer, 2004: 55 - 64.

[5] CHI Z, WEI M X, THOMAS T, *et al*. Evolving accurate and compact classification rules with gene expression programming[J]. IEEE Transactions on Evolutionary Computation, 2003, 7(6): 519 - 531.

[6] 刘大有, 卢奕南, 王飞, 等. 遗传程序设计方法综述[J]. 计算机研究与发展, 2001, 38(2): 412 - 423.

[7] BASTIEN C, MICHEL D. Cellular automata modeling of physical systems [M]. London: Cambridge University Press, 1998.

[8] ALBA E, DONRONSORO B. The exploration/exploitation tradeoff in dynamic cellular genetic algorithms [J]. IEEE Transactions on Evolutionary Computation, 2005, 9(2): 126 - 142.

[9] 田志友, 王浣尘, 吴端明. 区域市场连锁经营选址与布局的元胞自动机模拟[J]. 系统工程理论方法应用, 2005, 14(1): 50 - 54.

[10] 朱刚, 马良. TSP的元胞蚂蚁算法求解[J]. 计算机工程与应用, 2007, 43(10): 79 - 100.

[11] XIN L, CHI Z, WEI M X, *et al*. Prefix gene expression programming [C]// GECCO'2005. Washington, DC: [s.n], 2005.

[12] 唐丽钰, 李淼, 张建, 等. 基于自动定义函数GP的自适应建模研究[J]. 小型微型计算机系统, 2005, 26(6): 1000 - 1005.