基因表达式编程种群多样性自适应调控算法

李太勇[1,2]，唐常杰[2]，吴 江[1,2]，乔少杰[3]，姜 玥[2]，陈 瑜[2]

(1. 西南财经大学经济信息工程学院 成都 610074; 2. 四川大学计算机学院 成都 610065;
3. 西南交通大学信息科学与技术学院 成都 610031)

【摘要】为了解决基因表达式编程GEP种群多样性控制问题，提出了一种新的带权种群多样性的自适应调控方法。设计了带权的种群多样性测度方法，详细分析了选择、交叉及变异算子对种群多样性的影响。提出了初始种群的多样化算法DAIP，以保证初始种群多样性的最大化。设计了自适应的交叉和变异算子，提出了种群多样性自适应调控算法APDTA，使种群在进化过程中维持合适的种群多样性，进而提高进化效率。实验验证了APDTA的有效性。

关 键 词 自适应遗传算子; 进化计算; 遗传算法; 基因表达式编程; 多样性
中图分类号 TP311.13 **文献标识码** A doi:10.3969/j.issn.1001-0548.2010.02.027

Adaptive Population Diversity Tuning Algorithm for Gene Expression Programming

LI Tai-yong[1,2], TANG Chang-jie[2], WU Jiang[1,2], QIAO Shao-jie[3], JIANG Yue[2], and CHEN Yu[2]

(1. School of Economic Information Engineering, Southwestern University of Finance and Economics Chengdu 610074;
2. School of Computer Science, Sichuan University Chengdu 610065;
3. School of Information Science and Technology, Southwest Jiaotong University Chengdu 610031)

Abstract To cope with the problem of controlling population diversity in gene expression programming (GEP), an adaptive population diversity tuning algorithm is proposed. A weighted measurement for population diversity is designed. The impact in terms of selection, crossover, and mutation operators on population diversity is analyzed in detail. A diversity algorithm for initial population (DAIP) maximizing the initial population diversity is proposed as well. Aiming to appropriately maintain the population diversity and achieve high evolution efficiency, adaptive crossover and mutation operations are developed and an adaptive population diversity tuning algorithm (APDTA) is developed. Experiments show that APDTA is efficient and effective.

Key words adaptive genetic operator; evolutionary computation; genetic algorithm; gene expression programming; diversity

文献[1]基于遗传算法(genetic algorithm，GA)提出了基因表达式编程(gene expression programming，GEP)概念。GEP是一种有效的遗传进化方法，被应用于函数发现和分类、时间序列分析以及组合优化等领域[1-8]。

按照生物学的进化理论，多样性是生物种群进化的基础，也是GEP能够搜索到全局最优解的必要条件。因此，在GEP中保持种群的多样性具有重要意义。

文献[1]还对GEP中种群多样性的重要性进行了分析。文献[9]提出以基因空间均匀分布的策略实现初始种群的多样化，但未对多样性进行量化和评估。文献[10]提出以基因组多样性制导的分阶段进化挖掘算法，但未考虑不同的函数符和终结符具有不同的权值。

传统GEP中，各函数符和终结符在染色体中具有相同的(或随机的)出现概率。在实践中，各函数符和终结符出现的概率与具体的应用相关。例如，在符号回归问题中，如果知道问题答案的形式是多项式，可以让函数符“+”以较大的概率出现。

本文将带权的函数符和终结符引入到GEP中，为解决GEP种群多样性问题，提出了带权的种群多样性测度方法，分析了选择、交叉和变异算子对种群多样性的影响，并设计了自适应的交叉算子和变异算子，使种群在进化过程中维持较高效率。实验验证了算法的有效性。

收稿日期：2008 – 08 – 29; 修回日期：2009 – 12 – 18
基金项目：国家自然科学基金(60773169)；“十一五”国家科技支撑计划(2006BAI05A01)
作者简介：李太勇(1979 –)，男，博士生，主要从事数据库与知识工程等方面的研究.

1 带权的种群多样性测度

传统GEP中，在构建初始种群时，各函数符和终结符进入染色体的概率是随机的。而现实问题中，根据先验知识，可以人工设定种群内染色体中各符号的概率。如在进行泰勒展开式回归时，可以人为指定函数符“+”具有较大的出现概率。因为泰勒展开式是一个多项式，其中有较多的“+”。

基于以上考虑，可提出一种新的基因表达式编码方式，除了考虑函数符和终结符的形式外，还考虑它们出现的大致概率。在传统的GEP基础上增加两个集合P_h和P_t，P_h和P_t分别为各函数符和终结符在基因头部和尾部出现的概率。在用GEP求解时，准确预设各函数符和终结符在染色体中出现的概率是不现实的，但是可以通过各概率的相对大小表示各函数符和终结符在染色体中出现次数的多寡。

定义 1 概率基因。设F_s为函数符集，$f_s \in F_s$；T_s为终结符集，$t_s \in T_s$，$G=\{f_s,t_s\}^h\{t_s\}^t$，则：

(1) f_s和t_s为头部符号，t_s为尾部符号；

(2) h为基因头部长度，t为基因尾部长度；

(3) 如果g in G，且$t=h(n-1)+1$，则g为标准基因；

(4) 符号f_s和t_s在g中的位置为基因位；

(5) 如果基因头部符号出现的概率满足概率集P_h，尾部符号出现的概率满足概率集P_t，则基因g满足概率约束；

(6) 如果g是标准基因，且g满足概率约束，则g为概率基因。

显然，当种群中的所有基因都为概率基因时，则每一基因位上的符号也满足概率约束。为叙述方便，假设每个个体仅由一个基因构成。

定义 2 带权的多样性测度。设种群规模为n，种群包含的个体集合$P=\{a_1,a_2,\cdots,a_n\}$，其中$a_j=\{a_{1j},a_{2j},\cdots,a_{Lj}\}$，$j=1,2,\cdots,n$，则多样性测度为：

$$D(P)=1-\frac{1}{L}\left(\sum_{j=1}^{h}\sum_{i=1}^{|F_s|+|T_s|}|T_{ji}-p_{hi}|+\sum_{j=h+1}^{L}\sum_{i=1}^{|T_s|}\left|T_{ji}-p_{ti}\right|\right) \quad (1)$$

式中 $|F_s|$、$|T_s|$分别为函数符集和终结符集的元素个数，T_{ji}为在所有个体中第j位出现第i个函数符或终结符的概率。p_{hi}是P_h中的元素，为第i个头部符号在基因头部出现的期望值；p_{ti}是P_t中的元素，为第i个尾部符号在基因尾部出现的期望值。p_{hi}和p_{ti}通过人工设置。第t代种群的多样性测度记为$D_t(P)$。

性质 1 设$D(P)$为种群多样性测度，则：

(1) $D(P) \in [0,1]$

(2) 当种群中各个体都相同且各基因位上的符号与期望值不一致时，$D(P)=0$，即多样性完全消失；

(3) 当种群中各函数符和终结符在各基因位上出现的概率均与期望值相等时，$D(P)=1$，即多样性测度最大。

性质 2 当种群中各基因位上的函数符和终结符的种类和个数不变时，种群的多样性保持不变。

由于性质1和性质2相对简单，限于篇幅，证明从略。

对于良好的进化算法：(1) 在进化初期，种群的多样性较大；(2) 进化的中期，多样性逐渐降低；(3) 进化后期，多样性稳定在一个较低的水平，直到算法收敛到全局最优解。

2 遗传算子对种群多样性的影响

在GEP中，常见的算子包括选择算子、交叉(重组)算子和变异算子。本文分析该几种算子对种群多样性的影响。

2.1 选择算子的影响

在GEP中，选择算子通常采用精英保留策略、轮盘赌或联赛模式。最优个体直接进入下一代，其余个体按照适应度进行竞争，适应度越大的个体被选择进入下一代的机会越大。在单纯选择算子作用下，种群的最优个体得以保留，种群的平均适应度逐步提高，逐步收敛到初始种群的最优个体适应度。由于适应度大的个体被重复选择进入下一代的概率较大，在单纯选择算子的作用下，在进化后期，种群的个体趋同，由性质1知，种群的多样性降低。

2.2 交叉算子的影响

交叉算子改变了两个个体的结构，该两个个体的适应度可能提高。但是，由于交叉算子都是在不同个体的相同基因位上进行的，不改变种群中每一基因位上的函数符或终结符的种类和个数，由性质2知，单纯交叉算子不改变种群的多样性。

2.3 变异算子的影响

变异算子将基因中的一个符号变成另一个符号。如果变异位置在基因头部，则符号可以变异成函数符或终结符；如果变异位置在基因尾部，则符号只能变异成终结符。

因为变异算子改变基因位上符号的种类和个数，所以改变了种群的多样性。考虑极端情况，当种群中的个体都相同时，此时种群的多样性最小，改变一个个体中基因位上的符号，可增加种群的多样性；当种群中各函数符和终结符出现的概率都与期望值相等时，此时种群的多样性最大，改变一个

个体中基因位上的符号，则降低了种群的多样性。因此，变异算子既能增加也能降低种群的多样性。

3 自适应的多样性调控策略

多样性是生物种群进化的基础。在进化过程中使种群保持合适的多样性，可提高进化效率。

3.1 初始种群的多样性算法

初始种群应具有较大的多样性。本文提出一种初始种群多样性最大化算法。

算法 1 初始种群多样性算法(diversity algorithm for initial population，DAIP)

输入：F_s、T_s、P_h、P_t、h、t、种群大小P_s；

输出：初始种群

Begin

```
1  for(int i=0;i<h+t;i++){
2  foreach(s in F_s ∪ T_s){
3  计算s在第i位应出现的次数n;
4  生成n个s符号;}
5  将符号分配到P_s个个体的第i位;}
6  while(种群中有重复个体){
7  将重复个体与其他个体进行交叉;}
8  }
```

End

算法依次产生每一基因位上的符号，由于符号出现次数是按照期望值产生的，所以，产生的种群的多样性满足最大化要求。为了维持种群的多样性，对于种群中的重复个体，将其与其他个体进行交叉，产生新的个体，直到种群中不存在重复个体。算法1保证了初始种群多样性的最大化。

3.2 自适应的多样性调控策略

在进化的不同阶段，多样性不断变化，在进化初期，种群维持较大的多样性；在进化后期，种群收敛到全局最优解，种群维持较小的多样性。

选择算子采用精英保留策略和轮盘赌方式，将种群多样性与进化过程联系起来，提出自适应的交叉和变异算子。通过自适应算子将种群的多样性维持在理想状态，使种群以较高效率进化。

将进化分成α、β和γ等3个阶段，分别代表进化的初期、中期和后期。设交叉概率 $P_c \in [P_{c_{\min}}, P_{c_{\max}}]$，变异概率 $P_m \in [P_{m_{\min}}, P_{m_{\max}}]$。

在α阶段，种群应该维持多样性，而采用算法1得到的初始种群的多样性是最大化的。交叉算子可保持种群的多样性，所以，在α阶段，采用较高的交叉概率，较低的变异概率。α阶段的交叉和变异概率分别为：

$$\begin{cases} P_{c_\alpha} = P_{c_{\max}} - (P_{c_{\max}} - P_{c_{\min}})|t-t'|/t \\ P_{m_\alpha} = P_{m_{\min}} + (P_{m_{\max}} - P_{m_{\min}})|t-t'|/t \end{cases} \tag{2}$$

式中 t和t'分别为初始种群多样性和上一代种群多样性。

在β阶段，种群进行一般进化，交叉和变异概率维持不变，分别为：

$$\begin{cases} P_{c_\beta} = (P_{c_{\max}} + P_{c_{\min}})/2 \\ P_{m_\beta} = (P_{m_{\max}} + P_{m_{\min}})/2 \end{cases} \tag{3}$$

在γ阶段，种群将收敛到最优解，交叉和变异概率分别为：

$$\begin{cases} P_{c_\gamma} = P_{c_{\min}} + (P_{c_{\max}} - P_{c_{\min}})|t-t'|/t \\ P_{m_\gamma} = P_{m_{\max}} - (P_{m_{\max}} - P_{m_{\min}})|t-t'|/t \end{cases} \tag{4}$$

式中 t为进入γ阶段时种群的多样性，t'为上一代种群的多样性。

种群的α、β和γ等3个阶段可根据进化总代数人工设置，也可根据种群的最优个体适应度或种群的平均适应度确定。

算法 2 种群多样性自适应调控算法(adaptive population diversity tuning algorithm，APDTA)

输入：F_s、T_s、P_h、P_t、h、t、$P_{c_{\min}}$、$P_{c_{\max}}$、$P_{m_{\min}}$、$P_{m_{\max}}$、划分进化阶段的概率集合P_e、种群大小P_s、进化代数N；

输出：最优个体

Begin

```
1  根据DAIP算法生成初始种群P;
2  Best_Individual=null; //初始最优个体
3  n=0; //当前进化代数
4  while(n<N) {
5  switch(阶段) {
6  Case α阶段:
7  以式(2)更新交叉和变异算子;
8  Case β阶段:
9  以式(3)更新交叉和变异算子;
10 Case γ阶段:
11 以式(4)更新交叉和变异算子;}
12 对P进行选择、交叉和变异操作;
13 Keep(Best_Individual); //保存最优个体
14 n=n+1;
15 }
16 return Best_Individual;
```

End

APDTA算法首先产生多样性最大化的初始种群，然后根据进化阶段自适应地调整交叉和变异算

子的概率。将每一代进化过程中的最优个体保留，进化完成后，返回最优个体。

传统GEP随机产生初始种群，在进化过程中采用固定的交叉和变异概率。当APDTA算法产生初始种群时，各符号在初始种群中出现的概率随机，并且各进化阶段采用固定的交叉和变异概率时，APDTA退化为传统GEP。所以，传统GEP是APDTA的特例。APDTA更具有通用性。

4 实验与性能分析

4.1 实验环境

实验平台为的硬件环境2.0 G CPU、512 M内存；软件环境为Windows XP、Eclipse+JDK 1.5。为了评估APDTA的性能，与传统GEP进行比较。

4.2 实验1

一元函数发现问题[1]：

$$F_1(x)=x^3+x^2+x+1 \tag{5}$$

实验中产生100个[−50,50]间的随机数作为训练集。实验重复10次，取平均值作为最后的实验结果。适应度函数为：

$$f_i=\sum_{j=1}^{n}\left(\left|\frac{P_{ij}-T_j}{T_j}\right|<0.001\right) \tag{6}$$

式中 n为训练样本的个数；P_{ij}为第i个个体相对于第j个样本的预测值；T_j为第j个样本的真实值。当两者的相对误差小于0.001时，表示第i个个体命中了第j个样本。当第i个个体命中了所有样本时，f_i取得最大值n。本文实验中，n的值为100。

实验参数如表1所示。在传统GEP中，交叉和变异操作的概率取最小概率和最大概率的中间值。

表1 F_1参数设置

参数名	参数值
种群大小P_s	40
函数符集F	{+,−,*,/}
终结符集T	{x}
头部长度H	8
进化代数N	400
基因个数	2
连接符	+
最小交叉概率 $P_{c_{\min}}$	0.1
最大交叉概率 $P_{c_{\max}}$	0.05
最小变异概率 $P_{m_{\min}}$	0.03
最大变异概率 $P_{m_{\max}}$	0.06
头部符号概率集P_h	{0.25,0.15,0.25,0.15,0.2}
尾部符号概率集P_t	{1}
进化阶段划分概率集P_e	{0.2,0.6,0.2}

表中，头部符号概率集表明+、−、*、/、x在头部出现的概率依次为0.25、0.15、0.25、0.15、0.2。进化阶段划分中的{0.2,0.6,0.2}表示按进化总代数从头到尾按照2:6:2将进化阶段分成α、β和γ等3个阶段。进化结果如图1所示。

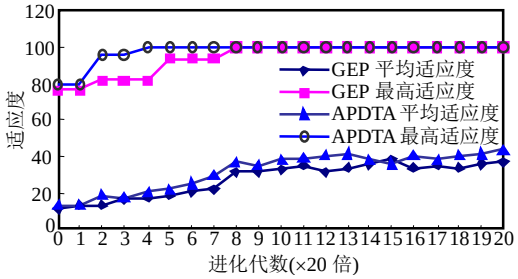

图1 F_1的最高适应度与平均适应度比较

实验1表明，相对于传统GEP，APDTA的平均适应度提高了约15%，而收敛到最优解的代数减少了约30%。可见，APDTA具有更高的进化效率。

4.3 实验2

五元函数发现问题[1]：

$$F_2(a,b,c,d,e)=\frac{\sin a\cos b}{\sqrt{\exp^c}}+\tan(d-e) \tag{7}$$

式中 exp是自然对数的底数，其值为2.718 281 83。实验中产生100组[0,1]间的随机数作为训练集。实验重复100次，取平均值作为实验的最后结果。适应度函数仍为 $f_i=\sum_{j=1}^{n}\left(\left|\frac{P_{ij}-T_j}{T_j}\right|<0.001\right)$。

实验参数如表2所示(其余参数与表1一致)。

表2 F_2参数设置

参数名	参数值
函数符集F	{+,−,*,/,S,C,E,T,Q}
终结符集T	{a,b,c,d,e}
进化代数N	2 000
头部符号概率集P_h	{−}
尾部符号概率集P_t	{−}

表中，函数符集中的S、C、E、T、Q分别为“正弦”、“余弦”、“exp的幂次方”、“正切”和“开方”运算符；表中的{−}表示各符号在染色体中按相等的概率出现。实验结果如表3所示。

表3 F_2实验结果

算 法	成功率	平均进化代数
GEP	59	1 316
APDTA	72	1 124

F_2函数相对复杂，传统GEP和APDTA均能正常发现其函数形式。但相对于传统GEP，APDTA发现最优解的成功率更高，而平均进化代数更低，仍然

表现了更好的性能。

以上实验表明，DAIP算法产生的初始种群对提高GEP进化效率是非常有效的；APDTA通过多样性自适应调控交叉和变异算子能，有效地提高进化效率。

5 结　论

种群多样性在GEP的进化过程中扮演了重要角色，传统GEP并未考虑多样性的影响。本文将多样性的概念引入GEP，提出了一种种群多样性的测度方法，分析了常见的选择算子、交叉算子和变异算子对种群多样性的影响。提出了一种初始种群的多样性算法DAIP，该算法产生多样性最大化的初始种群。在进化阶段，提出了通过自适应地调整交叉和变异算子的概率调控种群多样性的算法APDTA。该算法能有效提高GEP的进化效率。实验结果表明，相对于传统GEP，APDTA发现最优解的平均进化代数更低，而发现复杂函数的成功率更高。在未来的工作中，将研究进化阶段划分的其他方法，并研究APDTA在函数发现以外的应用。

本文研究工作得到西南财经大学科研基金(QN0805)的资助，在此表示感谢。

参　考　文　献

[1] FERREIRA C. Gene Expression programming: mathematical modeling by an artificial intelligence[M]. Second, revised and extended edition. Netherland, Berlin, Heidelberg: Springer-Verlag, 2006.

[2] FERREIRA C. Mutation, transposition, and recombination: an analysis of the evolutionary dynamics[C]//Proceedings of the 4th International Workshop on Frontiers in Evolutionary Algorithms, Research Triangle Park. North Carolina, USA: [s.n.] 2002: 614-617.

[3] FERREIRA C. Discovery of the boolean functions to the best density-classification rules using gene expression programming[C]//Proceedings of the 4th European Conference on Genetic Programming (Euro GP 2002). Berlin: Springer-Verlag, 2002: 51-60.

[4] ZUO J, TANG C J, ZHANG T Q. Mining predicate association rule by gene expression programming[C]// International Conference for Web-Age Information Management 2002(WAIM'02). Berling, Heidelberg: [s.n.], 2002: 92-103.

[5] ZHOU C, XIAO W M, TIRPAK T M, et al. Evolving accurate and compact classification rules with gene expression programming[J]. IEEE Transactions on Evolutionary Computation, 2003, 7(6): 519-531.

[6] 唐常杰, 彭　京, 张　欢, 等. 基于基因表达式编程发现知识的三项新技术[J]. 计算机应用, 2005, 25(9): 1978-1981.
TANG Chang-jie, PENG Jing, ZHANG Huan, et al. Three new techniques for knowledge discover by gene expression programming-transgenic, overlapped expression and back tracking evolution[J]. Computer Applications, 2005, 25(9): 1978-1981.

[7] 彭　京, 唐常杰, 李　川, 等. M-GEP:基于多层染色体基因表达式编程的遗传进化算法[J]. 计算机学报, 2005, 28(9): 1459-1466.
PENG Jing, TANG Chang-jie, LI Chuan, et al. M-GEP: A new evolution algorithm based on multi-layer chromosomes gene expression programming[J]. Chinese Journal of Computer, 2005, 28(9): 1459-1466.

[8] FANG Lei, ZHANG Huan-chun, JING Ya-zhi. A new fuzzy adaptive genetic algorithm[J]. Journal of Electronic Science and Technology of China, 2005, 3(1): 57-59, 71.

[9] 胡建军, 唐常杰, 段　磊. 基因表达式编程初始种群的多样化策略[J]. 计算机学报, 2007, 30(2): 305-309.
HU Jian-jun, TANG Chang-jie, DUAN Lei, et al. The strategy for diversifying initial population of gene expression program ming[J]. Chinese Journal of Computer, 2007, 30(2): 305-310.

[10] 刘齐宏, 唐常杰, 胡建军, 等. 多样性制导分段进化的基因表达式编程[J]. 四川大学学报(工程科学版), 2006, 38(6): 108-113.
LIU Qi-hong, TANG Chang-jie, HU Jian-jun, et al. Gene expression programming based on diversity guided grading evolution[J]. Journal of Sichuan University (Engineering Science Edition), 2006, 38(6):108-113.

编　辑　税　红