

内嵌基因表达式编程及其在函数发现中的应用

向 勇¹, 唐常杰², 朱明放³, 陈 瑜², 代术成²

(1. 成都电子机械高等专科学校计算机工程系 成都 610031; 2. 四川大学计算机学院 成都 610065;

3. 江苏技术师范学院计算机工程学院 江苏 常州 213001)

【摘要】为了提高表达效率,提出了新的基因解码方案,形成了内嵌基因表达式编程算法EGEP;提出了极大表达树、嵌套表达树和拼接表达树等概念;分析了基因的表达空间和算法的复杂度。实验表明,该算法提高了函数发现的成功率;在小规模种群的函数中其能力明显优于GEP。在单基因情况下,目标为一元函数和二元函数时,EGEP平均成功辈数分别为GEP算法的25.5%和16.3%;各种规模下,在EGEP算法中二元函数的成功率平均比GEP提高43%以上。

关键词 函数发现; 遗传算法; 基因表达式编程; 基因内区

中图分类号 TP311.6

文献标识码 A

doi:10.3969/j.issn.1001-0548.2011.01.022

Embedded Gene Expression Programming and Its Application in Function Mining

XIANG Yong¹, TANG Chang-jie², ZHU Ming-fang³, CHEN Yu², and DAI Shu-cheng²

(1. Department of Computer Engineering, Chengdu Electromechanical College Chengdu 610031; 2. School of Computer, Sichuan University

Chengdu 610065; 3. School of Computer Engineering, Jiangsu Teachers University of Technology Changzhou Jiangsu 213001)

Abstract Gene Expression Programming is effective for function mining. In gene expression usually exist some un-expressed introns. To improve the expression efficiency, this paper makes following contributions: Proposed an evolutionary algorithm embedded gene expression programming (EGEP) based on a new decoding method of gene; Proposed some new concepts, i.e. the maximum expression tree, nested expression tree and spliced expression tree; Analyzed the expression space of gene and the complexity of algorithm. Extensive experiments show that the success rate is improved greatly and under the small size population, the ability of mining function surpasses GEP apparently. In single gene algorithms, when the objective functions are bivariate function and single-variable function, the ratios of the convergence generation of EGEP to that of GEP are 25.5% and 16.3% respectively; compared with GEP, the success rate of EGEP is averagly increased by 43% in bivariate function mining.

Key words function mining; genetic algorithm; gene expression programming; intron

基因表达式编程(gene expression programming, GEP)^[1]是遗传研究家族的新成员,其不同于传统遗传算法 (genetic algorithm, GA)^[2]的显著特点是将个体的基因型和表现型分离。与遗传编程(genetic programming, GP)^[3]相比,基因表达式编程的编码方式简单,计算效率较高,GEP与GA和GP相比快2~4个数量级^[4]。在电路进化^[5]、时序预测^[6]、分类^[7]、优化^[8]、关联规则^[9-10]等领域,基因表达式编程的研究已经取得了很多成果。

在解码基因时,传统GEP经常存在未表达的基因内区。已有实验表明,传统GEP基因表达效率不高,并且在小规模种群中发现解的性能欠佳。本文

采用了新的基因解码方法,从而提出了一个新的进化算法即内嵌基因表达式编程(embedded gene expression programming, EGEP),使基因遗传信息表达更充分。与GEP相比,EGEP中同一基因可蕴含多个表现型,GEP没有此特征;由于基因信息可以复用,EGEP个体的表达空间优于GEP;EGEP提高了小规模种群下的函数发现的成功率。

1 基因结构和基本概念

1.1 基因结构与表达树

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

收稿日期: 2009-06-05; 修回日期: 2010-01-22

基金项目: 国家自然科学基金(60773169); 国家“十一五”科技支撑计划(2006BAI05A01); 四川省教育厅科研资助(2006B067)

作者简介: 向 勇(1975-),男,博士,副教授,主要从事数据挖掘、优化等方面的研究。

TS), 尾部只能包含终结符号。头部长度 h 和尾部长度 t 之间满足关系:

$$t = h(n_{\max} - 1) + 1 \quad (1)$$

式中, n_{\max} 为函数集FS中函数的最大参数数目。

GEP中借鉴了生物学的开放读码框架(open reading frame, ORF)概念。GEP中的基因是生物学中基因原理的简化, 其ORF起始码恒是基因的首位位置, 而结束码子却未必是基因的末位, 即在GEP基因中, 基因串的的最后一些位区有可能是非编码区(noncoding regions)或基因内区(intron)。

例1和例2说明数学表达式、基因的ORF以及表达树之间的转换过程。

例1 考虑代数表达式 $Sin((ab - c)(a+d))$, 对应的表达树如图1所示。其中 S 代表 Sin 。

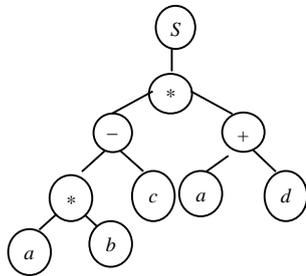


图1 表达树

在GEP中, 图1所示的是表现型(phenotype), 其对应的基因型(genotype)为:

$$S^* - + * cadab \quad (2)$$

式(2)的形成过程为: 对图1所示的表达树, 从上到下、从左到右地直接阅读, 即将其按层线性化, 得到的串就是该基因的ORF, 如式(2)所示。式(2)在GEP中称为基因的K-表达式(K-expressions)。文献[9]介绍了一种典型的基因解码算法, 但其解码后经常存在较多的非编码区域。为了使基因遗传信息表达更充分, EGEP采用了新的解码算法, 使基因头部全部得以表达。

例2 给定GEP基因 $++bcc+bbcbbbc$, 其中头部长度为6, 函数的最大参数数目为2(前6个符号为头部, 后7个符号为尾部)。分别用GEP和EGEP的方法解码为表现型。图2所示为GEP的解码结果, 图3所示为EGEP的解码结果。

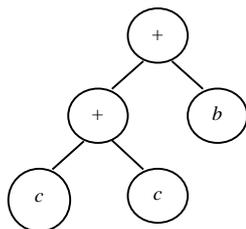


图2 GEP解码形成的表达树

由图2得知, 只有前5个符号被解码, 而其余码点*bbcbbbc未出现在表达树上, 形成了非编码区域。

与GEP不同, EGEP的解码过程是直到基因头部不存在非编码区域。观察图3可知, 该基因解码后可得到的表达树多达6个, 其中有的表达树是其他表达树的子树, 有的表达树是由其他表达树拼接而成。图3a所示为GEP解码得到的表达树; 以图3a的结束码子的下一个码子继续解码, 得到如图3e所示的表达树; 使用连接函数“+”将图3a和图3e所示表达树连接, 形成图3f的表达树; 图3b、图3c和图3d中的表达树是图3a的相异子树。由图3e本来可得到一个与图3d相同的子树, 但去掉重复只保留一个。

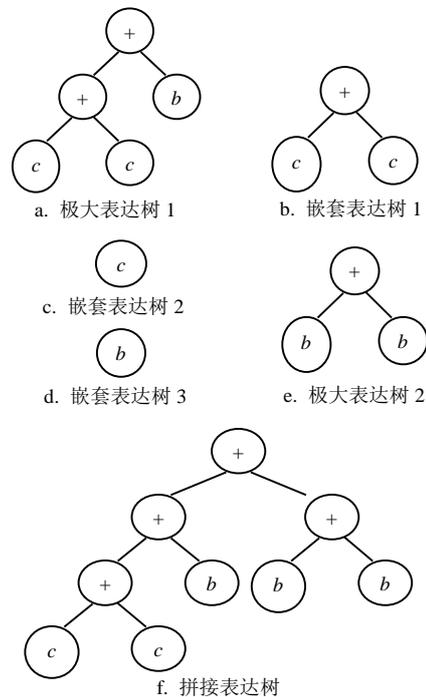


图3 EGEP解码形成的表达树

本文将图3a和图3e所示的表达树称为极大表达树, 将图3f所示的表达树称为拼接表达树, 将图3b、图3c和3d所示的表达树称为嵌套表达树。其中图3b、图3c和3d对应的K-表达式分别为 $+cc$ 、 c 和 b , 称为嵌套基因; 图3a和图3e对应的K-表达式分别为 $++bcc$ 和 $++bb$, 称为极大基因; 图3f对应的K-表达式为 $++++bbbcc$, 称为拼接基因。嵌套基因、极大基因和拼接基因统称为内嵌基因。

因此, EGEP与GEP的编码结构相同, 均为固定长度的线性符号串。不同的是, EGEP的基因表达更为丰富, 其基因头部不存在非编码区, 因而有限的遗传码点得以充分利用, 可表达更多的信息。

1.2 基本概念

本文给出EGEP中相关概念的形式化定义。

定义 1 设 G 为一个基因,使用传统GEP解码算法对 G 解码,有:

1) 若 t 是解码后得到的表达树,则称 t 为 G 的一棵极大表达树, t 对应的K-表达式称为 G 的极大基因。

2) 若基因头部存在未表达的符号串,则称该符号串为头部内区。

与GEP不同,EGEP基因解码时,若存在头部内区,则对头部重复使用GEP解码算法直到不存在头部内区,最后得到一个极大表达树的集合,本文称该集合为极大表达树森林。

定义 2 (内嵌表达树森林) 设 F_{\max} 是一个极大表达树森林, f 是一个参数数目为2的函数,一个集合 F_N 满足:

- 1) $t \in F_{\max}$, 则 $t \in F_N$;
- 2) $t \in F_{\max}$, e 为 t 的子树, 则 $e \in F_N$;
- 3) t 是 F_{\max} 中所有元素使用 f 连接形成的树, 则 $t \in F_N$ 。

则称集合 F_N 为内嵌表达树森林。

定义 3 (嵌套基因和拼接基因) 设 G 是一个基因, F_{\max} 为 G 的极大表达树森林, F_N 为 G 的内嵌表达树森林, $t \in F_N$, 有:

1) 若存在 $e \in F_{\max}$ 且 t 为 e 的子树, 则称 t 对应的K-表达式为 G 的一个嵌套基因;

2) 若 $\forall e \in F_{\max}$ 且 e 为 t 的子树, 则称 t 对应的K-表达式为 G 的拼接基因。

定义1和定义3中定义的极大基因、嵌套基因、拼接基因, 本文统称为内嵌基因。

定义 4 (个体与种群) EGEP环境中的个体 C 是一个4元组, 记为 $C=(G_s, g, u, s)$ 。其中 G_s (Genes)为基因的集合, g 为个体包含的基因数目, u 为用于形成拼接表达树的连接函数, s 为个体适应度。个体 C 的集合称为种群, 记为 $P=(C)$ 。

如 $C_1=(\{G_1, G_2, G_3\}, 3, '+', 0.35)$, 表示个体 C_1 由3个基因组成, 形成拼接表达树的连接函数为'+', 适应度得分为0.35。

根据定义1和定义4, 单基因EGEP算法具有如下特征: 种群中任意个体 $C=(G_s, g, u, s)$ 满足 $g=1$, 即只有一个基因。而多基因EGEP算法的特征是: 种群中任意个体 $C=(G, g, u, s)$ 满足 $g>1$, 即一个个体内包含多个基因。

1.3 表达空间分析

定义 5(表达空间) 设 G 为一个基因, A 为该基因的一种解码算法, 若 S 是按照 A 对基因 G 解码形成

的表达树的集合, 则称 S 为基因 G 的表达空间, S 中元素的个数称为 G 的表达空间的大小, 记为 $D(G)$ 。

根据定义5, 在EGEP中基因的表达空间即为基因的内嵌表达树森林。为了叙述简便, 本文算法的最大表达空间即是该算法中所有可能基因的最大表达空间。算法 A 的表达空间的大小记为 $D(A)$ 。

定理 1 GEP算法的最大表达空间大小为1。

证明 设 G 为传统单基因GEP算法中的任意一个基因, 该基因解码后只能得到一棵表达树, 故其最大表达空间大小为1。在多基因GEP算法中, 使用连接函数将各个基因解码后得到的表达树连接形成一个表达树, 其最大表达空间大小仍为1, 故传统GEP算法的表达空间大小始终为1。

定理 2 单基因EGEP算法 A_s 的表达空间最大为 $\max(D(A_s))=h+\min(h(x-1)+1, m)$, 其中 h 为基因头部长, x 为所有函数的最大参数的数目, m 为终结符集中的元素个数。

证明 已知单基因EGEP算法中个体由一个基因组成, 头部长 h , 则尾部长 $h(x-1)+1$ 。由式(1)可知, 在最好的情况下, 基因头部只包含具有最大参数数目的函数, 此时基因头部和尾部被全部表达, 则形成具有 h 个非叶子节点的极大表达树。分解该极大树可以得到除本身以外的以非叶子节点为根的 $h-1$ 个子树。下面讨论分解该极大表达树可以得到的以叶子节点为根的子树的数目:

1) 当 $h(x-1)+1 < m$, 即基因尾部长度小于终结符的个数 m , 则以叶子节点为根的相异的子树个数最多为 $h(x-1)+1$;

2) 当 $h(x-1)+1 \geq m$, 以叶子节点为根的相异的子树最多为终结符集中的元素个数即 m ;

由1)和2)可知, 以叶子节点为根的子树数目最多为 $\min(h(x-1)+1, m)$ 个。故基因的表达空间最大为 $h+\min(h(x-1)+1, m)$, 即 $\max(D(A_s))=h+\min(h(x-1)+1, m)$ 。

定理 3 多基因EGEP算法 A_m 的表达空间最大为 $\max(D(A_m))=kp+1$, 其中 $p=h+\min(h(x-1)+1, m)$, h 为基因头部长, m 为终结符集中的元素个数, k 为基因个数, x 为函数最大参数数目。

证明 由定理2可知, 单基因EGEP算法中基因的表达空间最大为 $p=h+\min(h(x-1)+1, m)$, k 个基因的表达树通过个体连接函数最多可得到一个个体拼接表达树, 故有 k 个基因的表达空间最大为 $kp+1$, 则 $\max(D(A_m))=kp+1$ 。

在传统单基因GEP和多基因GEP中, 基因解码以后最终得到的表达树只有1个, 则表达空间大小均为1。

例3 根据定理1和定理2, 在基因数为1、函数最大参数数目为2、终结符个数为4、个体头部长度变化情况下, EGEP算法与GEP算法的最大表达空间大小比较如表1所示。

表1 表达空间示例

头部长度	1	2	3	4	5	6	7	12	19
EGEP	3	5	7	8	9	10	11	16	23
GEP	1	1	1	1	1	1	1	1	1

观察表1可知, 随着头部变长, 单基因算法中, GEP算法的表达空间大小始终为1, EGEP的表达空间大小随头部变长逐渐接近线性增长。

2 内嵌基因表达式编程算法

EGEP算法模拟自然界的生物进化, 按照“物竞天择, 适者生存”的原则, 对由个体构成的种群实施选择、重组、变异等遗传操作, 使种群一代代地进化, 寻找出最优或较优的个体, 最终获得问题的解。

2.1 内嵌基因表达式编程算法

算法1 EGEP算法框架

输入: 算法参数、训练集。

输出: 最后一代群体中最佳个体的表达式。

该算法的步骤如下:

BEGIN

- 1) 种群初始化;
- 2) $j=0$;
- 3) WHILE ($j < \text{Max_Generation}$) DO;
- 4) 个体解码为表现型;
- 5) 执行每个表现型;
- 6) 适应度评价;
- 7) 采用精英策略保留最优个体;
- 8) 对种群执行以下操作获得新种群:
 - ① 选择(Selection)
 - ② 复制(Replication)
 - ③ 变异(Mutation)
 - ④ 逆转(Inversion)
 - ⑤ 插串(插串IS, 根插串RIS, 基因插串)
 - ⑥ 重组(单点重组, 两点重组, 基因重组)
- 9) $j = j + 1$;
- 10) ENDWHILE;
- 11) 输出最优个体的表达式;

End.

在步骤4)中, 将个体解码时, 首先需要形成极大表达树。与常见的GEP非递归解码算法不同, 本

文采用递归方法实现EGEP中的极大表达树形成算法; 解码时, 若基因存在头部内区, 则重复调用极大表达树形成算法, 直到不存在头部内区。

2.2 适应度评价

在EGEP算法中, 基因中可能蕴含了多个内嵌基因、多个极大基因和一个拼接基因, 故基因的适应度值需要按照某种规则从该多个内嵌基因的适应度中产生, 经常使用的有最大值法、平均值法或加权平均法等方法。本文主要采用最大值法。

针对函数发现, 一般基于绝对误差、相对误差和均方误差等标准度量构造适应度函数。文献[4]通过大量实验发现, 采用基于绝对误差的具有选择带宽 R 的适应度函数(absolute error with selection range)适用于大多数函数发现问题。

个体绝对误差的计算为:

$$E_{ij} = |P_{(ij)} - T_j| \quad (3)$$

式中, $P_{(ij)}$ 为第 i 个个体的程序对第 j 个实例的预测值; T_j 为第 j 个实例的目标值; E_{ij} 分别为第 i 个个体的程序对第 j 个适应实例的绝对误差。 E_{ij} 越小, 说明第 i 个个体的程序越接近所要发现的目标函数。

基于绝对误差的具有选择带宽 R 的适应度函数为:

$$f_i = \sum_{j=1}^n (R - E_{ij}) \quad (4)$$

式中, R 为一常量, 称为选择带宽, 控制适应度 f_i 的取值范围; E_{ij} 为式(3)计算出的绝对误差; f_i 为第 i 个个体的适应度函数值。 f_i 越大, 说明第 i 个个体的程序越接近目标函数。 R 为设定的值, 是为了有效地区分各个个体的适应度函数值。

2.3 算法复杂度分析

定理4 设 n 为基因长度, p 为种群大小, m 为最大允许进化代数, k 为训练集的样本个数, 则算法1的复杂度是 $O(mknp)$ 。

证明 计算主要集中于求个体的适应度。在算法1中, 计算个体中每个内嵌基因针对 k 个训练样本的表达式值的复杂度为 $O(k)$ 。由定理3可知, 单基因的表达空间最大为 $h + \min(h(x-1)+1, m) \leq n$, 则计算单基因的个体适应度的复杂度为 $O(kn)$ 。该算法需计算种群中每个个体的适应度, 故种群适应度计算的复杂度为 $O(knp)$ 。因算法1最多进化 m 代, 故算法1的复杂度为 $O(mknp)$ 。

2.4 算子

本文提出的EGEP算法与传统GEP基本过程类似, 即变异—重组—新种群—选择。EGEP算法中用

到的GEP算子主要包括变异、根插串、插串、基因插串、单点重组、两点重组、基因重组和逆转等，详见文献[1]。

3 实验和性能分析

实验1和实验2中分别使用已知的具有常数系数的一元函数和二元函数作为挖掘对象，从合成数据集比较传统GEP和EGEP在函数发现的成功率和收敛辈数上的性能。实验3比较了3种群规模对算法性能的影响。

为了对比性能，EGEP采用了和GEP相同的适应度函数，即基于绝对误差的具有选择带宽 R 的适应度函数。实验中采用的两个性能指标为算法的函数发现成功率提高量和成功辈数比，计算方法分别如下：

$$\text{成功率提高量} = \text{EGEP成功率} - \text{GEP成功率} \quad (5)$$

$$\text{成功辈数比} = \frac{\text{EGEP平均成功辈数}}{\text{GEP平均成功辈数}} \times 100\% \quad (6)$$

实验1 具有常数系数的函数发现问题

测试函数 $f(a) = a^2/2 + 3a$ 。根据测试函数在 $[-10, 10]$ 之间随机生成20个样本数据作为训练数据。实验中选用的部分进化参数如表2所示。

表2 算法进化参数

参数	类型			
	多基因 EGEP	单基因 EGEP	多基因 GEP	单因 GEP
最大进化辈数	500	500	500	500
种群大小	20	20	20	20
训练集数	20	20	20	20
函数集合	+*/	+*/	+*/	+*/
基因连接符	+		+	
头部大小	5	16	5	16
基因数量	3	1	3	1
总长度	33	33	33	33
变异率	0.044	0.044	0.044	0.044
单点重组率	0.4	0.4	0.4	0.4
两点重组率	0.2	0.2	0.2	0.2
基因插串率	0.1	0.1	0.1	0.1
插串率	0.1	0.1	0.1	0.1
根插串率	0.1	0.1	0.1	0.1
精度	0.001	0.001	0.001	0.001

对每个算法运行100次，结果如表3所示。

从表3可知，EGEP算法性能明显好于传统GEP，在多基因情况下，EGEP平均成功辈数仅为GEP的53.7%；单基因情况下，EGEP的平均成功辈数仅为GEP的25.5%，但函数发现的成功率提高了68%。

表3 GEP和EGEP一元函数发现性能对比

参数	多基因	单基因
GEP成功率/(%)	100	32
EGEP成功率/(%)	100	100
成功率提高量/(%)	0	68
GEP平均成功辈数	95	263
EGEP平均成功辈数	51	67
成功辈数比/(%)	53.7	25.5

实验2 二元函数的发现问题

测试函数 $f(X, Y) = X^3 - X^2Y + Y$ 。根据测试函数随机产生20个样本，自变量取值范围均为 $[-10, 10]$ 。 $R = 1000$ ，种群大小50，最大进化辈数1000，其他参数同表2。对每个算法运行100次，结果如表4所示。

表4 GEP和EGEP二元函数发现性能对比

参数	多基因	单基因
GEP成功率/(%)	100	55
EGEP成功率/(%)	100	100
成功率提高量/(%)	0	45
GEP平均成功辈数	106	416
EGEP平均成功辈数	58	68
成功辈数比/(%)	54.7	16.3

由表4可知，对于该二元函数，EGEP算法性能明显好于传统GEP，在多基因情况下，EGEP平均成功辈数仅为GEP的54.7%；单基因情况下，EGEP平均成功辈数仅为GEP的16.3%，函数发现的成功率提高了45%。

实验3 种群规模变化对性能的影响

该实验测试种群规模变化对性能的影响。针对实验2中的二元函数和训练数据，种群规模从10~50变化。比较单基因算法EGEP和GEP在不同规模下的平均成功进化辈数和成功率，每个算法运行100次，结果如表5所示。

表5 种群规模对平均进化辈数和成功率的影响

参数	种群规模				
	10	20	30	40	50
GEP平均成功进化辈数	514	484	328	461	416
EGEP平均成功进化辈数	161	101	79	86	68
成功辈数比/(%)	31.3	20.9	24.1	18.7	16.3
GEP成功率/(%)	57	67	56	52	54
EGEP成功率/(%)	100	100	100	100	100
成功率提高量/(%)	43	33	44	48	46

从表5可以看出，无论是成功率还是平均成功进化辈数，EGEP性能大大优于GEP。各种规模下成功

辈数比变化范围为16.3%~31.3%,即EGEP平均成功进化辈数为GEP的16.3%~31.3%。随着种群规模的增大,EGEP的平均成功进化辈数呈总体下降趋势,但GEP的平均成功进化辈数始终比较高,且波动较大。

从表5可以看出,EGEP在种群规模较小(即使为10)时,成功率仍可达到100%,但GEP的成功率波动较大;在种群大于30时,其成功率呈总体下降趋势。总体上,在各种规模下比较单基因算法,EGEP成功率比GEP平均提高了约43%。实验表明,在种群规模变化时,EGEP的性能优于GEP。

4 结 论

为了提高基因表达效率,本文提出了一种内嵌基因表达式编程算法EGEP。EGEP表达空间比GEP大,表达能力优于EGEP。实验表明,EGEP提高了函数发现的成功率,在小规模种群中的函数发现能力明显优于传统GEP。

下一步的工作需要对内嵌基因表达式编程的种群进化过程进一步进行分析,尝试将其应用于其他更复杂函数的发现和其他应用领域。

参 考 文 献

- [1] FERREIRA C. Gene expression programming: Mathematical modeling by artificial intelligence[M]. 2nd ed. Berlin: Springer Verlag, 2006.
- [2] GOLDBERG D E. Genetic algorithms in search, optimization and machine learning[M]. Boston: Addison-Wesley, 1989.
- [3] KOZA J R. Genetic programming: on the programming of computers by means of natural selection[M]. Cambridge, MA: MIT Press, 1992.
- [4] FERREIRA C. Gene expression programming: A new adaptive algorithm for solving problems[J]. Complex Systems, 2001, 13(2): 87-129.
- [5] 谢方军,唐常杰,袁长安,等. 基于基因表达式的演化硬件进化和优化算法[J]. 计算机辅助设计与图形学报, 2005, 17(17): 1415-1420.
XIE Fang-jun, TANG Chang-jie, Yuan Chang-an, et al. An algorithm of evolution and optimization for evolvable hardware[J]. Journal of Computer-Aided Design & Graphics, 2005, 17(17): 1415-1420.
- [6] 陈宇,唐常杰,钟义萧,等. 一个基于基因表达式编程的时间序列预测新方法[J]. 计算机科学, 2005(增刊B): 269-271.
CHEN Yu, TANG Chang-jie, ZHONG Yi-xiao, et al. A novel approaches for time series prediction based on gene[J]. Computer Science, 2005, (Suppl B): 269-271.
- [7] DUAN Lei, TANG Chang-jie, ZHANG Tian-qing, et al. Distance guided classification with gene expression programming[C]//ADMA 2006, LNAI 4093. Berlin: Springer Verlag, 2006: 239-246.
- [8] XU Kai-kuo, LIU Yin-tian, TANG Rong, et al. A novel method for real parameter optimization based on Gene Expression Programming[J]. Journal of Applied Soft Computing, 2009, 9(2): 725-737.
- [9] ZUO Jie, TANG Chang-jie, ZHANG Tian-qing. Mining predicate association rule by gene expression programming[C]//Proceedings of the International Conference for Web Information Age 2002, LNCS 2419. Berlin: Springer Verlag, 2002: 92-103.
- [10] 曾涛,唐常杰,朱明放,等. 基于人工免疫和基因表达式编程的多维复杂关联规则挖掘方法[J]. 四川大学学报(工程科学版), 2006, 38(5): 136-142.
ZENG Tao, TANG Chang-jie, ZHU Ming-fang, et al. Mining multi-dimensional complex association rule based on artificial immune system and gene expression programming[J]. Journal of Sichuan University (Engineering Edition), 2006, 38(5): 136-142.

编辑 黄 莘