http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 10 (11): 2024-2033, 2011
ISSN 1812-5638 / DOI: 10.3923/itj.2011.2024.2033

A Study on H-GEP: Gene Expression Programming with Homeotic Genes

Liu Yijun, Zhu Mingfang, Tang Jiali, Zhu Guangping and Jiang Hongfen
School of Computer Engineering, Jiangsu Teachers University of Technology,
Changzhou 213001, China

Abstract: Homeotic Gene Expression Programming (H-GEP) is a kind of Gene Expression Programming (GEP) which infuses homeotic genes into the chromosome to achieve a new mapping from the biological mechanisms to the GEP coding. Present study makes a further study on H-GEP. At first we analyze the expression space of the H-GEP individual. Secondly, an algorithm named HNEC is presented which computes the value of the expression encoded by an H-GEP individual without building an expression tree and expression. Thirdly, the performance and characteristic of the H-GEP method for function discovery are studied. Providing a complex and flexible way to link sub functions to form the target function, H-GEP is superior in discovery of complex functions. Fourthly, we study the effect of normal gene number on H-GEP performance. Finally, one application of H-GEP in the area of environmental quality assessment is presented.

Key words: Gene expression programming, homeotic gene, function discovery, evolutionary computation

INTRODUCTION

Gene Expression Programming (GEP) proposed by Ferreira (2001), is a novel evolutionary algorithm developed from Genetic Algorithm (GA) and Genetic Programming (GP). They are all computing models simulating biological evolution but have the distinction of their own encoding method and representative form of results. With simple, linear and compact chromosomes and easy genetic operators, GEP is a powerful global search tool. It has become an active research area of evolutionary computation and has been applied in many fields such as regression, classification and time series prediction (Tang *et al*., 2006; Zhu *et al*., 2010; Cheng and Zhi-hua, 2007; Zuo *et al*., 2004).

In Ferreira's works (Ferreira, 2006), there are three typical GEP methods according to individual expression ways. We call them Single-gene GEP(S-GEP), Multi-genes GEP (M-GEP) and Homeotic-gene GEP(H-GEP). An S-GEP chromosome comprises a single gene coding for an expression. An M-GEP chromosome comprises multiple genes whose corresponding functions are linked by a preset link function to form an ultimate expression. In recent years numerous researchers have investigated GEP and propose a series of improved GEP methods which may process data in specific fields with more effectiveness and efficiency. Zuo *et al*. (2004) discuss two GEP-based methods for time series prediction: One is called GEP-SWPM (GEP-Sliding Window Prediction Method) combining traditional sliding window prediction method with GEP and the other is GEP-DEPM (GEP-Differentia Equation Prediction Method) which mines differential equations from training data and predict future trends based on specified initial conditions. Inspired by the biological nature known as "seek advantage, avoid disadvantage". Duan *et al*. (2004) present a Weak-adaptive Model (WAM) based on GEP and a Relative Error Fitness Algorithm (REFA) to mine functions from data with noises. Xiaodong *et al*. (2004) propose the method of UEM (Uniform Expression Mining) which deals with complex functions having n expressions (n>1) in different domains as well as those having only one uniform expression. Lin *et al*. (2008) present a hybrid GEP algorithm with niching. Combining a k-means clustering method and genetic mechanism, the algorithm adjusts the minimum clustering distance to decide the niching number to avoid premature convergence. In these improved GEP methods, the expression way of an individual or chromosome use that of S-GEP and M-GEP. Some researchers also propose GEP variations which have novel individual structures and expression ways. Peng *et al*. (2005) propose an evolution algorithm named M-GEP (note here M-GEP is not Multi-genes GEP mentioned above) based on the new concept of multi-layer chromosomes in GEP which builds a level-call model and storage structure between the different chromosomes. Shucheng *et al*. (2008) propose an algorithm called MEGP (Multi Expression Gene Programming) which builds a

Corresponding Author: Liu Yijun, School of Computer Engineering, Jiangsu Teachers University of Technology, Changzhou 213001, China

multi-level encoding and decoding model within one chromosome. Although, these GEP variations are more effective for some problems, their individuals are complex structured and difficult to decode. Besides there seems no biological explanation for their chromosome structure and working mechanism.

In Ferreira's work (Ferreira, 2006), a special kind of gene called homeotic gene is infused into GEP chromosome. Homeotic genes are in charge of controlling appearance of biological body. Normal genes, i.e., downstream regulatory genes, determine the characteristics and forms of a single organism, whereas homeotic genes determine the overall appearance of biological body by various permutations and combinations of normal genes. Mutations of either normal genes or homeotic genes will affect biological appearance. In the process of biological evolution with the multiple factors arising from gene expression systems and complexity of mechanisms and structures increasing, gene expression system is possible to have a rapid upheaval and big transformation, leading to leaps in biological evolution (Qichang, 2005). An infusion of homeotic genes into GEP achieves a novel mapping from biological mechanism to GEP encoding: Normal genes in chromosome encode sub functions called by the function encoded by a homeotic gene. Consequently the individual codes for a relatively complex function.

GEP with homeotic genes is named Homeotic Gene Expression Programming, H-GEP for brevity. H-GEP has been applied to the fields such as regression, classification, function discovery, etc. (Ferreira, 2006).

DEFINITIONS AND METHODOLOGY

Here, the definitions of several terms, such as gene, chromosome and population are given below.

Definition 1. (Normal gene): A normal gene G is a 3-tuple, denoted by G = (S, F, T), where S (String) is a string with fixed length, F (Function) is a set of computing functions, T (Terminal) is an alphabet, a finite set of labeling symbols.

Definition 2. (Homeotic gene): A homeotic gene G' is a 3-tuple, denoted by G' = (S', F', T'), where S' is a string with fixed length, F' is a set of computing functions and T' is a set of functions encoded by normal genes.

The S or S', also called gene for simplicity, is composed of two different domains of a head and a tail domain. The head domain contains symbols representing both functions and terminals whereas the tail is composed of only terminal.

When the head length h is chosen, the tail length t is evaluated by the Eq. 1:

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

where, n_{max} is the maximum number of arguments of the function in function set, also celled maximum arity. Thus, the gene encodes a legal program.

Although, S or S' is of fixed length, it is composed of two parts of variable length: The useful part and the useless part. The useful part is referred to as the Open Reading Frame (ORF) and the useless part is called the non-coding region.

Note that this study focuses on the GEP methods with a single output. Thus the chromosome is defined as below.

Definition 3. (Chromosome): An S-GEP chromosome C_S is a 1-tuple, denoted by $C_S = (G)$ where, G is a normal gene. An M-GEP chromosome C_M is a 2-tuple, denoted by $C_M = (U_g, f)$ where, U_g is a set of normal genes and f is a link function which is used to link genes to form an expression. An H-GEP chromosome C_H is a 2-tuple, denoted by $C_H = (U_g, HG)$ where, U_g is a set of normal genes and HG is a homeotic gene.

Definition 4. (Population): A population is a set of individuals which comprise a chromosome.

The expression encoded by a normal gene in an H-GEP chromosome is referred to as the Automatically Defined Function (ADF). A normal gene $gene_i$ is decoded into an ADF_i.

Example 1: To illustrate terms above, function set F = {+, -, ×, /} and terminal set T = {a, b} are used.

Assume n_{max} = 2 and h = 5 and hence, tail length t = 6.

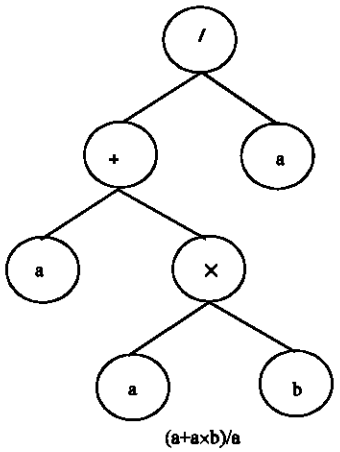

Fig. 1: ET of an S-GEP chromosome

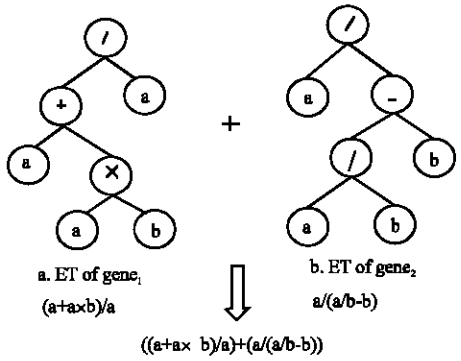

Fig. 2 (a-b): ETs of an M-GEP chromosome

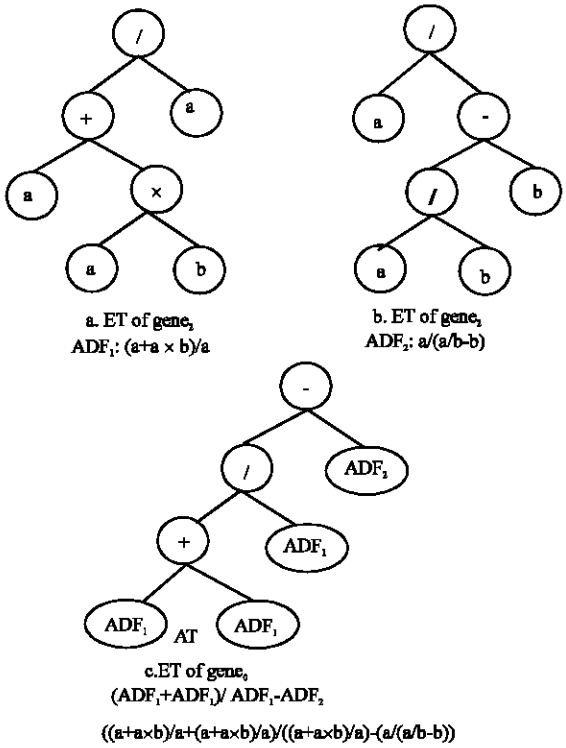

Fig. 3 (a-c): ETs of an H-GEP chromosome

An S-GEP chromosome, with the tail shown in bold, is given below:

/+aa×abbbaa

Figure 1 shows the Expression Tree (ET) encoded by the gene. The nodes of the ET correspond to the gene ORF, i.e., /+aa×ab.

An M-GEP chromosome comprising two genes is given below:

/+aa×abbbaa/a-/bababba

Let $gene_1$ be the first gene and $gene_2$ be the second one. Figure 2 shows the ETs and expressions encoded by two genes. The function "+" is used to link sub expressions to form an ultimate expression.

Example 2: Here, function set F = F' = {+, -, ×, /}, terminal set T = {a, b} and T' = {ADF_1, ADF_2} are used. Let h and h' be head length of the normal gene and the homeotic gene, respectively. Assume n_{max}= 2, h = 5 and h' =4 and hence, their corresponding tail length t = 6 and t' = 5.

An H-GEP chromosome comprising two normal genes and one homeotic gene is given below:

/+aa×abbbaa/a-/bababba-/2+11121

Let $gene_0$ be the homeotic gene, i.e., -/2+11121, $gene_1$ be the first normal gene, i.e., /+aa×abbbaa and $gene_2$ be the second one, i.e., /a-/bababba. The 1 and 2 in $gene_0$ represents ADF_1 and ADF_2 which are encoded in $gene_1$ and $gene_2$, respectively. Figure 3 shows the ETs and expressions encoded by genes.

ANALYSIS OF EXPRESSION SPACE

Definition 5. (Expression Space, ES): Let E_I be the expression encoded by the individual I. The expression space of I is denoted as ES (I) which is length of the expression E_I, i.e., the number of symbols of functions and terminals in E_I.

For the individual I in example 2, E_I = ((a+a×b)/a+(a+a×b)/a)/((a+a×b)/a)-(a/(a/b-b)). Length of E_I is 31 and hence, ES (I) = 31.

For S-GEP genes, M-GEP genes and H-GEP normal genes, let h be head length and t be tail length. For an H-GEP homeotic gene, let h' be head length and t' be tail length. Assume that an M-GEP individual has k genes and an H-GEP individual has k normal genes and 1 homeotic gene.

Lemma 1: For an S-GEP individual I_S, its maximum expression space max (ES (I_S)) = h+t.

Proof: In S-GEP method, an individual contains one normal gene. In best case, the gene is fully expressed. Hence, max (ES (I_S)) = h+t.

Lemma 2: For an M-GEP individual I_M, its maximum expression space max (ES (I_M)) = k (h+t) + k-1.

Proof: It follows from Lemma 1 that the maximum expression space of a normal gene is h+t when the gene is fully expressed in best case and as such, the maximum

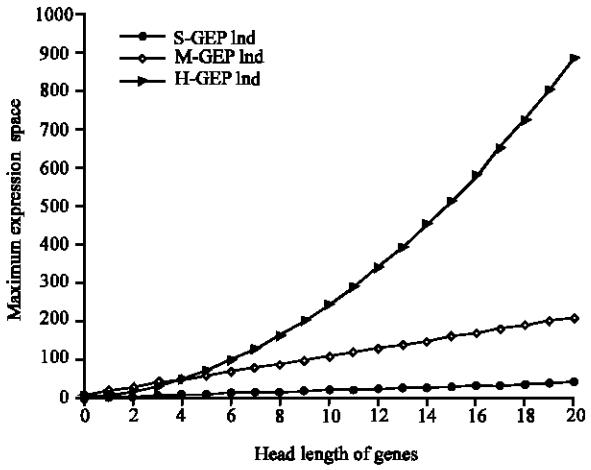

Fig. 4: Maximum expression space of individuals

expression space of k genes is k (h+t). The k-1 function connectors are needed to link k genes to form an ultimate expression. Hence, max (ES (I_M)) = k (h+t)+k-1.

Lemma 3: For an H-GEP individual I_H, its maximum expression space max (ES (I_H)) = (h+t) t'+h'.

Proof: Consider that in the best case the homeotic gene and normal genes of I_H are all fully expressed. The homeotic gene's head contains h' function symbols and the tail contains t' functions encoded by normal genes. It follows from Lemma 1 that the expression space of a normal gene is h+t. Hence, the length of the expression encoded in I_H is (h+t) t'+h'. As such, max (ES (I_H)) = (h+t) t'+h'.

Note that by Lemma 1-3, max (ES (I_S)) only relates to the gene length, max (ES (I_M)) relates to both the gene length and number of genes and max (ES (I_H)) relates to the length of normal genes and the homeotic gene but not the number of normal genes. To facilitate comparison, here we assume n_{max} in Eq. 1 is 2, k = 5, h' = h and hence, t' = t = h+1. Figure 4 shows maximum expression space for three kinds of individuals in the case of variable head length. With head length increasing, the H-GEP individual exceeds the S-GEP individual and the M-GEP individual fast in maximum expression space when head length is greater than 5, the value of k.

HNEC: NON-EXPRESSION COMPUTATION FOR H-GEP INDIVIDUALS

A key step to implement GEP algorithm is to translate a chromosome of linear symbol string to a nonlinear structured ET and math expression. Ferreira (2001) proposes the hierarchy method to build an ET. Datong and Qiaoyun (2008) propose two decoding methods for GEP to build an expression. One obtains the expression on genotype of GEP and the other obtains it by stack. Jiang *et al.* (2006) propose a Gene Read and Compute Machine (GRCM) which directly computes value of an expression without building an ET or expression but does not indicate how to compute ORF length of a gene. Mo and Kang (2008) present the algorithm of computing the ORF length. However, this algorithm is for genes having 1-place, 2-place and 3-place functions only. Here we present a universal algorithm to compute the ORF length, together with its computing principle. By extending the algorithms by Jiang *et al.* (2006) and Mo and Kang (2008) we present a algorithm named HNEC (H-GEP Non-Expression Computation) which computes the value of the expression encoded by an H-GEP individual without building an ET or expression.

Computation of ORF length: In an H-GEP chromosome C_H = (U_g, HG), Let U_g= {gene$_1$, gene$_2$ …, gene$_k$}, the homeotic gene HG be gene$_0$. Let T={t_1, t_2, …, t_k}, T' = {ADF_i| ADF_i is encoded by gene$_i$/\ i = 1, 2, …, n} and F = F' = {f_1, f_2, …, f_u}. The f_i is a d (f_i)-place function, e.g., d (+) = 2, d (log) = 1.

Let t (f_i) be the number of f_i in the gene ORF and such as:

$$Sum\_d = \sum_{i=1}^{u} d(f_i)\, t(f_i)$$

is the number of arguments needed by all functions in the gene ORF.

Theorem 1. The gene ORF length l and sum_d above satisfy the equation l = sum_d+1.

Proof. A gene ORF corresponds to an ET and hence, the number of nodes in the ET is l. Except the root in the ET, each node is an argument of its father node which a computing function and hence, the number of nodes as function arguments is l-1. Whereas sum_d is also the number of arguments needed in a gene ORF or its corresponding ET. Hence, l-1 = sum_d, i.e., l = sum_d+1.

The process to compute the ORF length of a gene is described in Algorithm 1.

The chromosome in Example 2, i.e., /+aa×abbbaa/a-/bababba-/2+11121, is used to illustrate Algorithm 1. Table 1 shows computation of the ORF length of gene$_1$, i.e., /+aa×abbbaa. Finally, length 7 is obtained and thus the ORF is /+aa×ab. In the same way, the ORF length of gene$_2$ is 7 and its ORF is /a-/bab and the ORF length of gene$_0$ is 7 and its ORF is -/2+111.

Table 1: An example of computing the ORF length of a gene

	Step							
	1	2	3	4	5	6	7	8
l	0	1	2	3	4	5	6	7
e_l		/	+	a	a	×	a	b
d (e_l)		2	2			2		
sum_d	0	2	4	4	4	6	6	6

Table 2: An example of computing $\varphi_1(x)$

m	e_m	s_m	q	$s = (s_1, s_2, \ldots, s_7)$
7	a		7	(/, +, 10, 10, ×, 10, 20)
6	b		7	(/, +, 10, 10, ×, 10, 20)
5	×	200	5	(/, +, 10, 10, 200, 10, 20)
4	a		5	(/, +, 10, 10, 200, 10, 20)
3	a		5	(/, +, 10, 10, 200, 10, 20)
2	+	210	3	(/, 210, 10, 10, 200, 10, 20)
1	/	21	1	(21, 210, 10, 10, 200, 10, 20)

Algorithm 1 ORF_length

```
Input  : A gene gene
Output : ORF length l of gene
1      : l←0; sum_d←0
2      : Repeat
3      : l←l+1
4      : If e_l reprsents f_j, then sum_d←sum_d+d (f_j)
5      : // e_l is the l^th bit of the gene gene
6      : Until l = sum_d+1
7      : Return l
```

Computing value of expression for instances: Let $t = (t_1, t_2, \ldots, t_k)$ and the function expression encoded by individual I be $\gamma(t)$. For each normal gene $gene_i \in U_g (i = 1, 2, \ldots, n)$, it encodes the function expression $\varphi_i(t)$. Homeotic gene $gene_0$ encodes the function expression $\varphi_0(g)$, $g = (g_1, g_2, \ldots, g_n)$, $g_i = \varphi_i(t)$ $(i = 1, 2, \ldots, n)$. Substituting $\varphi_i(t)$ for g_i, $\gamma(t) = \varphi_0(\varphi_1(t), \varphi_2(t), \ldots, \varphi_n(t))$ is obtained. For an instance $x = (x_1, x_2, \ldots, x_n)$, $\gamma(x) = \varphi_0(\varphi_1(x), \varphi_2(x), \ldots, \varphi_n(x))$.

The global algorithm HNEC for an H-GEP Individual is outlined in Algorithm 3, in which $\gamma(x)$ is attained. Each $\varphi_i(x)$ $(i = 1, 2, \ldots, n)$ is computed by lines 1 to 3 firstly and then $\varphi_0(\varphi_1(x), \varphi_2(x), \ldots, \varphi_n(x))$, i.e., $\gamma(x)$, is computed by lines 4 to 5.

Algorithm 2 describes the computation of $\varphi_i(v)$. The variable v is x for $i = 1, 2, \ldots, n$ and $(\varphi_1(x), \varphi_2(x), \ldots, \varphi_n(x))$ for $i = 0$. The intermediate variable v unifies evaluation of normal genes and homeotic gene.

Similar to Algorithm 1, 2 and 3 are illustrated by the chromosome in Example 2.

Assume t = (a, b) and an instance x = (10, 20). Thus v = x = (10, 20) according to line 1 of Algorithm 3. For $gene_1$ the ORF length is 7 and as such the ORF $e_1e_2e_3e_4e_5e_6e_7$ = /+aa×ab. According to lines 3 to 6 of Algorithm 2, s = (/,+,10,10,×,10,20) is obtained by substituting v_1 and v_2 for a and b in ORF, respectively Table 2 shows the computation of $\varphi_1(x)$ on *s* according to lines 7 to 12 of Algorithm 2. Finally, $s_1 = 21$ and as such $\varphi_1(x) = 21$.

Algorithm 2 Gene_value

```
Input  : A chromosome C, i, v
Output : value val of φ_i (v)
1      : l←ORF_length (gene_i)
2      : // e_1e_2… e_l is ORF of gene gene_i
3      : s← e_1e_2… e_l // s = (s_1, s_2, …, s_l), s_j← e_j
4      : for m = 1 to l
5      : if e_m reprsents t_j then s_m← v_j
6      : if e_m reprsents ADF_j then s_m← v_j
7      : q← l
8      : for m = l downto 1
9      : if e_m reprsents f_j then
10     : s_m ← f_j(s_{q-d(fj)+1}, …, s_{q-1}, s_q)
11     : q←q-d (f_j)
12     : val←s_1
13     : return val
```

Algorithm 3 HNEC

```
Input  : A chromosome C, x
Output : Value val of γ (x)
1      : v←x
2      : For i = 1 to n
3      : G_i←Gene_Value (C, i, v)
4      : v←g // g = (g_1, g_2, …, g_n) = ( φ_1 (x), φ_2 (x), …, φ_n (x))
5      : Val←Gene_value (C, 0, v)
6      : Return val
```

Similarly $\varphi_2(x) = -0.513$ and hence, $g = (\varphi_1(x), \varphi_2(x)) = (21, -0.513)$, $v = g = (21, -0.513)$. For $gene_0$, the ORF length is 7 and the ORF -/2+111. Substituting v_1 and v_2 for 1 and 2 in ORF, respectively; s= (-,/,-0.513,+,21,21,21). By Algorithm 2, $\varphi_0(v) = 2.513$ and as such $\gamma(x) = 2.513$.

Complexity analysis of the algorithm HNEC: In Algorithm 3, each $\varphi_i(x)$ in $g = (\varphi_1(x), \varphi_2(x), \ldots, \varphi_n(x))$ is computed by calling Algorithm 2. In Algorithm 2 of computing $\varphi_i(x)$, Algorithm 1 is called to compute ORF length of $gene_i$. Scanning the $gene_i$, a string with length h+t, Algorithm 1 has computing time O (h+t). Thus, Algorithm 2 also has computing time O (h+t). Hence, Algorithm 3 HNEC has computing time O (n×(h+t)) to compute $\gamma(x)$.

A COMPARATIVE STUDY ON S-GEP, M-GEP AND H-GEP

Function discovery is one of the most successful applications of GEP. Here we study the performance and characteristic of H-GEP in function discovery. All experiments are conducted on computers with an INTEL Core 2DuoProcessorE2160 with 2G memory, running Windows XP. All algorithms are implemented in C and programs are executed on VC++ 6.0.

Experiments and results

Experiment 1: Discovery of function $f_1(x, y) = x^3y + x^2y^2 + xy^3 - x/y + 1$.

Let the 2-place function f_1 (x, y) be the test function. 30 fitness cases are randomly generated in the interval (-10, 10). The link function of "+" is used to link M-GEP genes.

Here the absolute error function with selection range is used as fitness function (Ferreira, 2006):

$$Fit_i = \sum_{j=1}^{n} \left(R - \left|P_{ij} - T_j\right|\right) \quad (2)$$

where, Fit_i is the fitness of the ith individual, R is the selection range, P_{ij} is the value predicted by the ith individual for fitness case j and T_j is the target value for fitness case j.

Individuals are selected according to fitness by roulette-wheel sampling. This kind of selection together with simple elitism and the fitness function Eq. 2 are used in all problems of this study.

In GEP, genetic operators except selection and replication are referred to as modification operators. Modification operators of normal genes comprise mutation, inversion, insertion sequence, root insertion sequence, two-point recombination, one-point recombination, gene recombination and gene transposition. Modification operators of the homeotic gene comprise mutation, inversion, insertion sequence and root insertion sequence. The parameter setting for all GEP methods are shown in Table 3. For information on how to set these parameters (Ferreira, 2006).

Table 4 presents the performance of three methods in 1000 executions, respectively. In a successful run, the generation, in which for the first time the target function is discovered, is referred to as success generation.

The results showed that H-GEP outperforms S-GEP significantly on the test function f_1. The success rate of H-GEP is 17.2 times that of S-GEP. Minimum success generation of H-GEP is 39.2% of that of S-GEP. Average and maximum success generations of H-GEP are close to that S-GEP. However, M-GEP outperforms H-GEP with higher success rate and the success rate of M-GEP is 8.26 times that of H-GEP.

Experiment 2: Discovery of function f_2 (x, y) = (x-y)/(2x+y) – x^2/y^2.

The parameter setting in Experiment 2 remains unchanged as Experiment 1.

Table 5 shows that both S-GEP and M-GEP fail to discover the target function f_2 and H-GEP discovers it with a low success rate of 0.2%.

Analysis and discussion: Note that M-GEP shows significantly different performance in Experiment 1 and Experiment 2. The reason is that the sub functions of f_1, e.g., x^3y, x^2y^2, etc. are linked in a simple way of addition which is consistent with the preset connector "+" of M-GEP. M-GEP in experiment 1 obtains a high success rate of 71%. However, the sub functions of f_2, e.g., x-y, 2x+y, x^2/y^2, etc. are linked by multiplication and division but not the preset link function. It's not amazing that M-GEP fails to discovery function f_2.

In addition, for both S-GEP and H-GEP methods, the results of Experiment 1 are more satisfactory than that of Experiment 2. In Experiment 2, S-GEP fails to discover the target function f_2 and H-GEP discovers it with a low success rate of 0.2%. Besides the randomness of algorithm execution we think the reason is that the function f_1 has more equivalents than f_2. By commutative law of addition, equivalents of f_1, such as $x^2y^2+ x^3y+xy^3-x/y +1$, $1+ x^3y +x^2y^2 +xy^3- x/y$, etc., can be attained. By distributive law of multiplication, equivalents of f_1, such as $xy (x^2 +xy +y^2)- x/y +1$, $x^2 (xy +y^2) +xy^3- x/y +1$, $xy (x^2+y^2) +x^2y^2 - x/y +1$, etc., can be attained. Hence, f_1 has a greater

Table 3: Parameter setting in experiment 1

Parameters	S-GEP	M-GEP	H-GEP
Number of Generations	1000	1000	1000
Population size	51	51	51
Function set of Normal genes	+, - , ×, /	+, - , ×, /	+, - , ×, /
Terminal set of Normal genes	x, y	x, y	x, y
Number of Normal genes	1	7	6
Function set of A homeotic gene	--	--	+, - , ×, /
Terminal set of A homeotic gene	--	--	ADF_1–ADF_6
Head length of Normal genes	38	5	5
Head length of A homeotic gene	--	--	5
Chromosome Length	77	77	77
Modification rate In normal genes	0.1	0.1	0.1
Modification rate In a homeotic gene	0.1	0.1	0.1
Selection range	100%	100%	100%

Table 4: Success rates and success generations in experiment 1

Gene	Success rate (%)	Average success generation	Minimum success generation	Maximum success generation
S-GEP	0.5	611	176	1000
M-GEP	71.0	436	46	999
H-GEP	8.6	667	69	995

Table 5: Success rates and success generations in experiment 2

Gene	Success rate (%)	Average success generation	Minimum success generation	Maximum success generation
S-GEP	0	--	--	--
M-GEP	0	--	--	--
H-GEP	0.2	636	278	994

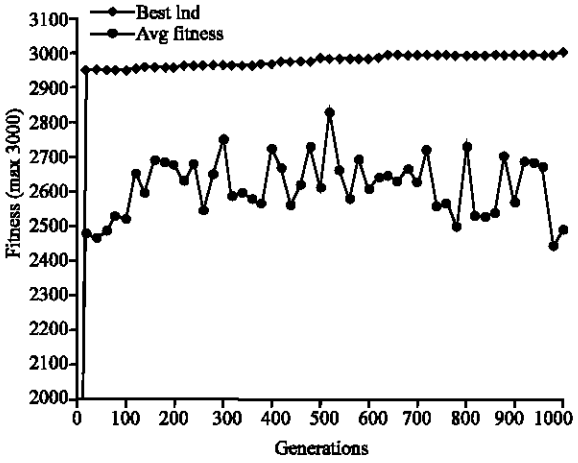

Fig. 5: Evolving curves of H-GEP on f2

probability to be discovered than f_2 for both S-GEP and H-GEP. In limited times of runs, it's acceptable that S-GEP fails to discover the function f_2.

Figure 5 shows the progression of average fitness of the population and the fitness of the best individual for a successful run of H-GEP on the test function f_2.

With a relatively high genetic diversity maintained in evolving, the H-GEP system is healthy and strong. The evolved solution appears in the 994th generation. The chromosome of the best individual is shown below:

- - + y + + y y x x x y - x y x x x y y x y x + - ×++xxxyxy+yxxxxxxxxy+yyyyxyyxyx×yyxyxxyxyy-//21366111

The expressions encoded in genes are presented in Table 6. Normal genes are $gene_1$-$gene_6$ and the homeotic gene $gene_0$.

The homeotic gene $gene_0$ is decoded into the function expression of $ADF_2/ADF_1 - ADF_3/ ADF_6$. With substituting the corresponding expression for each ADF, the target function $(x-y)/(2x+y) - x^2/y^2$ is obtained.

Table 6 shows that normal genes of $gene_1$, $gene_2$, $gene_3$ and $gene_6$ discover the sub functions of 2x+y, x-y, x^2 and y^2, respectively. Normal genes have powerful capability of searching simple problem spaces. The homeotic gene is capable of searching complex problem space. Compared with S-GEP and M-GEP methods, H-GEP is a more powerful global search tool.

Conclusions: The results of all experiments allow us to draw some characteristics of the three algorithms discussed in previous sections. An S-GEP chromosome that comprises a single gene coding for an expression is simple in structure. But for complex problems, such as complex function discovery, the S-GEP individual's expression space and search capability in problem spaces are limited by the characteristic that the single gene codes for one recursive function independently. An M-GEP chromosome comprises multiple genes whose corresponding functions are linked by a preset link function to form an ultimate function. The M-GEP method attains satisfactory results in case that the target function is formed by linking sub functions with the preset function of M-GEP. However, note that the link function is predetermined and the sub function encoded by a gene appears only once in the ultimate function. When the sub functions are linked in a sophisticated way to form the target function, the M-GEP method is outperformed by the H-GEP method. H-GEP provides a sophisticated and flexible way to link sub functions. In an H-GEP chromosome, normal genes search simple problem spaces and the homeotic gene is capable of obtaining the solution of a complex problem space.

Table 6: Genes and their corresponding expressions

Gene	String	Expression
$gene_1$	-+y++yyxxxy	ADF_1: 2x+y
$gene_2$	-xyxxxyyxyx	ADF_2: x-y
$gene_3$	+-×++xxxyxy	ADF_3: x^2
$gene_4$	+yxxxxxxxxy	ADF_4: x+y
$gene_5$	+yyyyxyyxyx	ADF_5: 2y
$gene_6$	×yyxyxxyxyy	ADF_6: y^2
$gene_0$	-//21366111	$ADF_2/ ADF_1 - ADF_3/ ADF_6$

EFFECT OF $|U_G|$ ON H-GEP PERFORMANCE

Experimental results: In an H-GEP individual, the chromosome comprises two kinds of genes. Here we study how the normal gene number n = $|U_g|$ affects the success rate of H-GEP.

Experiment 3: Effect of $|U_g|$ on the H-GEP success rate.

Here, the test function in Experiment 1 is used and the parameter setting remains unchanged. Figure 6 shows success rate for each n of 1-10 in 1000 runs.

When the normal gene number n is set to 1, H-GEP fails to discover the test function. When n is set to 2-6, H-GEP succeeds to discover the function and success rate increases with n. When n is set to 7, the best success rate of 9.9% is obtained. When n is set to 8-10, H-GEP also succeeds to discover the function but success rate decreases with n.

Analysis and discussions: Let f be the target function. Assume $f = g(\omega_1, \omega_2, \ldots, \omega_m)$, the set $S = \{\omega_i \mid i = 1, 2, \ldots,$

Fig. 6: Effect of |Ug| on the success rate

m} is a set of sub functions of f. Let φ_i be the function encoded in gene $gene_i$ of the individual I. As such, S' = {φ_i |i = 1, 2, …, n} is a set of functions encoded in normal genes of $gene_1$-$gene_n$ and the homeotic gene $gene_0$ codes for φ_0. When S⊆S', the set U_g of normal genes successfully discovers S. When $\varphi_0(\varphi'_1, \varphi'_2, \ldots, \varphi'_m) = g(\omega_1, \omega_2, \ldots, \omega_m)$, where $\varphi_0 = g$ and $\varphi'_i \in S' \wedge \varphi'_i = \omega_i$ (i = 1, 2, …, m), the homeotic gene successfully discovers g. In H-GEP, normal genes are used to discover S and the homeotic gene discover g and as such f is discovered by the individual I.

The intuitive reason is as follows. In evolution normal genes discover simple sub functions, such as x^3y, x/y etc., so the increase of normal gene number n help increase the probability to discover sub functions. Hence, at first the success rate is improving. However, with increase of n, it becomes difficult for a homeotic gene to search more normal genes to find out the ones coding for sub functions. Then the success rate decreases.

A simple mathematical illustration is given below. In time interval (t_1, t_2), let p_1 be probability of normal genes discovering S, p_2 be probability of a homeotic gene discovering g and p be probability of an individual I discovering f. Here $p=p_1p_2$. The normal gene number $n = |U_g| = |S'|$.

On the one hand, with larger n, i.e., larger size of normal gene set, comes the larger probability of normal genes discovering S by parallel search. As such, p_1 should be an increasing function of n and let $p_1 = \alpha(n)$. When $n \to 0$, $p_1 \to 0$ and when $n \to +8$, $p_1 \to 1$.

On the other hand, with larger n, |S|/|S'| decreasing, comes the smaller probability of the homeotic gene discovering g. As such, p_2 should be an decreasing function of n and let $p_2 = \beta(n)$. When $n \to 0$, $p_2 \to 1$ and when $n \to +\infty$, $p_2 \to 0$.

For simplicity, assume $p_1+p_2=1$ and as such $\alpha(n)+\beta(n) = 1$. Hence:

$$p = p_1 \times p_2 = \alpha(n) \times \beta(n) = \alpha(n) \times (1-\alpha(n)) \text{ s.t. } \alpha(n) \in (0,1) \quad (3)$$

Since $\alpha(n)$ is an increasing function, $\exists n = n_0$, $p_{max} = \alpha(n_0) \times (1-\alpha(n_0))$. $(0, n_0)$ is an increase interval where p increases with n increasing and $(0, n_0)$ is a decrease interval where p decreases with n decreasing.

The performance of the H-GEP method correlates with the normal gene number more or less but there is no rigorous theory indicating how to set n. The above empirical result and analysis shows that for a concrete problem, there should be an optimal value range for n, with which the H-GEP algorithm has a high success rate. In conclusion, the algorithm has the best performance when the normal gene number matches the problem space.

APPLICATION CASE

Definition of the problem: Here, one application of environmental quality evaluation built on H-GEP are briefly presented which show that H-GEP and its implementation described in this study work well in real domain.

Let $P = \{p_1, p_2, \ldots, p_m\}$ be set of monitoring points, $X = \{x_1, x_2, \ldots, x_n\}$ be attribute set. For attribute value vector $(x_{1i}, x_{2i}, \ldots, x_{ni})$ of each monitoring point p_i, there is a corresponding evaluation score y_i of p_i. Hence there is a mapping f between them and as such $y_i = f(x_{1i}, x_{2i}, \ldots, x_{ni})$. Here H-GEP algorithm is used to discover f', an approximation of f. For a monitoring point to be assessed, the input of f' is its attribute value vector and output is the evaluation score.

Results and comparisons: In this study, attribute set X = {atmosphere (x_1), surface water (x_2), groundwater (x_3), soil (x_4)}. Environmental quality is divided into 4 classes of A, B, C and D where A represents no pollution, B light pollution, C moderate pollution and D signifies heavy pollution. Each evaluation score in the interval (0, 10) is mapped to one class. The interval (0, 1) is mapped to A. (1, 2.5), (2.5, 5) and (5, 10) are mapped to B, C and D, respectively.

Twenty samples are tabulated in Table 7 which are from 20 different monitoring points in a certain region (Yong *et al*., 2009). The samples of monitoring point 1-8, 14 and 15 are fitness cases and the rest are test cases. These attribute values have been normalized to a relatively small range (0.1, 1) by using a linear mapping below:

Table 7: Normalized data set in the environmental quality evaluation

Monitoring point	Atmosphere	Surface water	Ground water	Soil	Evaluation score	Environmental quality
1	0.1010	0.1000	0.2171	0.1130	0.9	A
2	0.1000	0.1024	0.2220	0.1130	0.9	A
3	0.2173	0.2156	0.2951	0.1000	1.8	B
4	0.2006	0.1094	0.2366	0.1097	1.6	B
5	0.2168	0.1496	0.1000	0.1173	1.7	B
6	0.2284	0.2864	0.9000	0.1336	2.2	B
7	0.6198	0.1755	0.4220	0.1227	4.4	C
8	0.3554	0.2864	0.5732	0.1195	2.9	C
9	0.4322	0.2628	0.2463	0.1217	3.3	C
10	0.2067	0.1590	0.2951	0.1152	1.7	B
11	0.9000	0.5224	0.3537	0.1314	6.6	D
12	0.5880	0.3336	0.2902	0.1758	4.5	C
13	0.2775	0.1496	0.2951	0.1173	2.1	B
14	0.6082	0.1330	0.5878	0.4735	5.2	D
15	0.5329	0.5224	0.3439	0.9000	5.9	D
16	0.5243	0.1850	0.7049	0.1693	4.1	C
17	0.2896	0.1519	0.3098	0.1227	2.2	B
18	0.5637	0.9000	0.7146	0.3349	5.4	D
19	0.3073	0.1684	0.8951	0.1520	2.7	C
20	0.5030	0.7442	0.5878	0.1671	4.5	C

$$a' = \frac{a - \bar{a}_{min}}{\bar{a}_{max} - \bar{a}_{min}} \times 0.9 + 0.1$$

where, a is the value to be mapped, the minimum value, the maximum value and a' the normalization of a.

The basic parameter setting for environmental quality evaluation is shown in Table 8.

After 1000 generations, the chromosome of the best individual in population is as follows:

- $+x_1x_1x_2\text{-}x_1x_1x_2x_3x_2x_1\times x_4++x_3x_2x_3x_2x_1x_3x_1\times x_3x_2x_3x_4x_2x_2x_2x_1x_2x_3 +x_1+\text{-}+x_1x_1x_1x_2x_4x_4++\times x_1x_4x_1x_3x_3x_3x_2x_0x_1+x_1x_2\text{-}x_1x_1x_1x_2x_2x_1+ /x_2\text{-}x_2x_2x_1x_1x_1x_4x_1+ +++2165477$

The chromosome codes for f': $6x_1+x_2+x_4+x_1x_3+x_2x_4+2x_3x_4$.

Table 9 tabulates the test results of H-GEP together with that of Support Vector Machine (abbr. SVM) and neural networks (abbr. NN). There are numerous types of neural networks and we choose the widely used BP neural networks with the fastest Levenberg-Marquardt (abbr. LM) learning algorithm. The prediction of SVM and LM-NN are from Yong *et al.* (2009) and Yijun *et al.* (2010) respectively. According to Table 9, average relative error of H-GEP, SVM and LM-NN prediction are attained 2.09, 2.19 and 6.13%, respectively. According to predictive values, all test cases are labeled with right quality classes in H-GEP and SVM. In LM-NN, the 18th monitoring point with heavy pollution is labeled with the wrong quality class of moderate pollution. These three methods are all effective in environmental quality evaluation because their predictive accuracy is all beyond 90%. Nevertheless, the results of H-GEP and SVM are significantly better than that of LM-NN with a lower average relative error while the result of H-GEP is slightly better than that of SVM. Superior in predictive accuracy to SVM and LM-NN, H-GEP also presents an easy understood explicit expression between independent variables and the dependent variable. Hence, this function discovery task is successfully accomplished by H-GEP.

Table 8: Parameters for the environmental quality evaluation

Number of generations	1000
Population size	51
Function set of normal genes	+, -, ×, /
Terminal set of normal genes	x_1, x_2, x_3, x_4
Number of normal genes	7
Function set of a homeotic gene	+, -, ×, /
Terminal set of a homeotic gene	ADF_1–ADF_7
Head length of normal genes	5
Head length of a homeotic gene	5
Modification rate in normal genes	0.1
Modification rate in a homeotic gene	0.1
Selection range	100%

Table 9: Test results of H-GEP, SVM and LM-NN

		Predictive value		
Monitoring point	Evaluation score	H-GEP	SVM	LM-NN
9	3.3	3.1761	3.2730	3.0548
10	1.7	1.6617	1.7008	1.7213
11	6.6	6.5337	6.3412	6.0187
12	4.5	4.3687	4.4721	4.3065
13	2.1	2.1006	2.1530	2.1916
16	4.1	4.1397	4.0048	4.4872
17	2.2	2.1966	2.2524	2.2636
18	5.4	5.8000	5.2250	4.9802
19	2.7	2.7370	2.6127	2.9165
20	4.5	4.5458	4.3763	4.8181

CONCLUSIONS AND FUTURE WORK

Inspired by the phenomenon of biological evolution and modern genetics, GEP is a heuristic stochastic search method and has been proven to be a powerful global search tool. Furthermore, for H-GEP method, homeotic genes are infused into the chromosome. The artificial

evolutionary system of H-GEP is driven by two types of genes, normal genes and homeotic genes.

Analysis and experiments show that homeotic genes not only expand expression space of individuals but also make the evolution of linking functions possible and H-GEP method is more capable for function discovery. An algorithm called HNEC is presented to compute the value of the expression encoded by an H-GEP individual without building an ET and expression. Algorithm HNEC has computing time O (n×(h+t)) to compute the value predicted by an individual for an fitness case. Hence, the gene number and gene length affect the time performance of the algorithm. In addition, although we find out the normal gene number has a great effect on performance of H-GEP we can't present rigorous rule indicating how to set the number. To improve the performance of H-GEP algorithm, it's necessary to make a further study on how the gene length, gene number, genetic operators, etc., affect the evolving of H-GEP systems.

A weakness of our study on H-GEP is that, although, we apply H-GEP to environmental quality evaluation, the data set is still small. In most real world tasks, both the training set and the test set are very large, comprising tens of thousands or even millions of instances. Try to apply present results to more real-world applications, then, is a very important issue for future work.

ACKNOWLEDGMENTS

The authors are indebted to their colleagues of School of Computer Engineering, Jiangsu Teachers University of Technology, for their suggestive discussion and zealous help. This study is supported by the National Science Foundation of P.R.China under Grant No.60773169, the Youth Scientific Research Foundation of Jiangsu Teachers University of Technology (No.KYY10040) and Qinglan Project of the Education Department of Jiangsu Province (Su-teachers(2010)27).

REFERENCES

Cheng, Z. and C. Zhi-hua, 2007. Cost-sensitive classification by gene expression programm. J. Univ. Electron. Sci. Technol. China, 36: 1319-1321.

Datong, X. and C. Qiaoyun, 2008. Two decoding methods of gene expression programming. Comput. Eng., 34: 210-213.

Duan, L., C.J. Tang, J. Zuo, Y. Chen, Y.X. Zhong and C. Yuan, 2004. An anti-noise method for function mining based on GEP. J. Comput. Res. Dev., 41: 1684-1689.

Ferreira, C., 2001. Gene expression programming: A new adaptive algorithm for solving problems. Complex Syst., 13: 87-129.

Ferreira, C., 2006. Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence. 2nd Edn., Springer, USA., pp: 3-116.

Jiang, D., Z. Wu, Z. Jian, K. Lishan, T. Ming-Duan and L. Kang-Shun, 2006. New method used in gene expression programming: GRCM. J. Syst. Simul., 18: 1466-1468.

Lin, Y.S., L.P. Hong and W. Jia, 2008. Function finding in niching gene expression programming. J. Chin. Syst., 29: 2111-2114.

Mo, H.F. and L.S. Kang, 2008. Automatic modeling of complex functions based on gene expression programming. J. Syst. Simul., 20: 2828-2831.

Peng, J., C.J. Tang, C. Li and J.J. Hu, 2005. M-GEP: A new evolution algorithm based on multi-layer chromosomes gene expression programming. Chin. J. Comput., 28: 1459-1466.

Qichang, F., 2005. Analysis of Life. High Education Press, Beijing.

Shucheng, D., T. Changjie, Z. Mingfang and C. Yu, 2008. Automatic complex function discovery based on multi expression gene programming. J. Sichuan Univ. 40: 121-126.

Tang, C., L. Duan, J. Peng, H. Zhang and Y. Zhong, 2006. The strategies to improve performance of function mining. Gene Expression Programming Genetic Modifying, Overlapped Gene, Backtracking and Adaptive Mutation. Invited Talk and Paper, Okinawa, Japan.

Xiaodong, H., T. Changjie, L. Zhi, P. Donghang and L. Yong, 2004. Mining functions relationship based on gene expression programming. J. Software, 15: 96-105.

Yijun, L., G. Chunsheng, Z. Guangping, J. Hongfen and C. Dan, 2010. Levenberg-marquardt neural network for environmental quality assessment. J. Southern Yangtze Univ. (Natl. Sci. Edn.), 9: 213-216, (In Chinese).

Yong, C., L. Jian and W. Changqing, 2009. Environmental quality evaluation based on SVM. Comput. Eng. Appl., 45: 209-211.

Zhu, M.F., C.J. Tang, S.J. Qiao, S.C. Dai and Y. Chen, 2010. Naive gene expression programming based on genetic neutrality. J. Comput. Res. Dev., 47: 292-299.

Zuo, J., C.J. Tang, C. Li, C.A. Yuan and A.L. Chen, 2004. Time series prediction based on gene expression programming. Adv. Web-Age Inform. Manage., 3129: 55-64.