Researches on GEP Evolution Based on TMPDI_SHCRRI Method

Yonghong Yu [a], Zhou Zhou [b]

School of Management Science and Engineering, Anhui University of Finance & Economics, Bengbu 233030, China.

[a]ac120107@163.com, [b]1057578619@qq.com

Abstract

This paper mainly focuses on the resons for gene communication stuck in the whole population and evolutionary convergence to local optimal solution causing by gene expression programming(GEP). It introduced the concepts of GEP family Chromosomes and population fault and proposed a novel algorithm based on thread mechanism periodically population diversity improve(TMPDI) to optimize evolution process by differentiating the main thread population, and based on sorting hierarchical clone replenish randomized individual(SHCRRI) to achieve a better result of population diversity by setting a suitable hereditary generation to child thread. The proposed method can ensure population diversity, evolutionary efficiency and can avoid evolutionary local convergence. The experiment result shows that the TMPDI-SHCRRI accelerates evolution, increases the fitness of optimal chromosome compared to the result not using TMPDI_SHCRRI.

Keywords

Keywords

Gene expression programming, Local convergence, Thread mechanism periodically population diversity improve, Sorting hierarchical clone replenish randomized individual.

1. Introduction

In the process of searching the excellent solution by GEP algorithm[1,2,3,4], the individual genotype of the population approaches the direction of high fitness. The selection probability of dominant individuals and the new individuals generated by other individuals during generic process with higher fitness is high, and the excellent genes are inherited if they are successfully selected as the parent chromosome of the offspring. In the process of progeny evolution, it is possible that the individual or its former individual may produce individuals with higher fitness, so the number of excellent genes can be expanded in the population. On this basis, a better gene fragment may also be produced, which may arise from the mutation in the one or two gene sites of the excellent fragment of the parent chromosome, or by the recombination of two shorter excellent segments. But either way, they have a more favorable genetic probability and are similar to the dominant gene in the parent generation. After repeating the above procedure, a family chromosome can be formed after several generations, and the fitness values of individuals within the family are all close to each other.

Although the expression of standard GEP has created population diversity, it may also damage the genetic structure that has evolved. To realize the complex connection of the subexpression tree, Reference[4] proposed the homologous gene which is based on the individual structure to protect the better gene and the prior knowledge integrity of the evolution. The P-GEP algorithm[5,6,7] is proposed to overcome the deficiency of basic GEP by structural mapping. The subtree of the expression tree is directly connected with the linear gene fragment to achieve the acquisition and utilization of the fine gene fragment. A novel evolutionary algorithm named UGEP[8] is designed to solve the problem of premature convergence and slow evolution of the basic GEP algorithm. This method uses the mixed unified table to generate the dispersivity of the initial population and the multi parent crossover operator, and compares UGEP and GEP through a series of symbolic regression experiments. It shows that UGEP has a strong ability to search for global optimization and a faster convergence rate.

The population with high average fitness has strong assimilation, it is always in a dominant position in evolution, so its stability is very strong and continue to be strengthened. With the general improvement of the fitness in the family, the gap of fitness between family and the other non-family gradually increased until the obvious population fault occurs. Chromosomes with the highest fitness may have their own populations, this shows that the current solution space is convergent. If convergence and a serious population fault occurs before the number of genetic generations specified in GEP is far from over, the number of individuals in the same family will increase, which leads to the lower possibility of individuals with lower fitness being selected as the parent chromosomes, and the effective gene exchange of the offspring can be monopolized by the high fitness family chromosome. The individual genotypes produced by subsequent evolution are difficult to introduce excellent genes in parent chromosomes with normal fitness, and these individuals will have more structural characteristics of the parent chromosomes of the same family after evolution. When the potential genetic value of the elite gene is excavated, it is very difficult to produce a genotype that greatly improves the optimal fitness. If the optimal individual fitness value in the same family is still in the domain of a local optimal solution, then the subsequent genetic operation can only evolve towards the local optimal solution. If the GEP algorithm ends with a fitness value larger than the local optimal solution, the program is trapped in a dead loop.

Through the GEP algorithm, the initial population evolves from the random population diversity state to the dominant family chromosome state. The evolution process is a process from disorder to order outwardly, but if the dominance of the family chromosome is too obvious to cause population fault. It is not conducive to the communication of gene within the whole population, and the evolution result can only converge to the local optimal solution under the action of selection operator.

In order to prevent premature convergence of GEP in the process of evolution, we propose an improved idea of evolutionary process: Through randomly generating non-family chromosomal individuals with higher fitness periodically, and adding them to the population, to make GEP search break through the premature convergence of the solution space. On the one hand, adding the dominant non-family chromosomal individuals to the population increased the diversity of the dominant individuals, on the other hand, it fills the population fault and breaks the limitation of communications in the family chromosomes. The optimal fitness of the family chromosomes can converge to a higher optimal upper limit through the introduction of an excellent gene fragment with more abundant external chromosomes.

2. Preliminary

Terminals and functions constitute the primitive elements of GEP. The terminals represent to the constant, input or no parameter functions in the mathematical expression, and correspond to the leaf nodes of the expression tree. The functions represent operators of espical application, components of programm and middle symbols of system, they correspond to the non leaf node of expression tree.

The expression tree(ET) is used as the phenotype of GEP individuals to represent an expression visually. In order to facilitate computer genetic manipulation, the nonlinear structure of the expression tree is translated into a linear structure which Ferreira called K-expression as chromosome of GEP, corresponding to the genotype of GEP individuals.

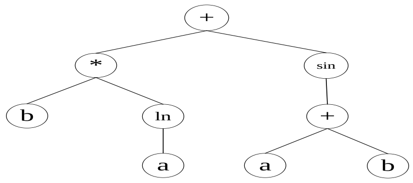
GEP genes are composed of a head and a tail[2,4]. The head contains symbols that represent both functions and terminals, whereas the tail contains only terminals. Therefore two different alphabets occur at different regions within a gene. For each problem, the length of the head *h* is chosen, whereas the length of the tail *t* is a function of h and the number of arguments of the function with the most arguments *n*, and is evaluated by the equation:

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

Consider a gene composed of {+,*,*ln*,*sin*, *a*, *b*}. In this case n = 2. For instance, for h = 7 and t = 8, the length of the gene is 7+8=15. One such gene is shown below：

$$\begin{array}{ccccccccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 0 & 1 & 2 & 3 & 4 \\ + & * & \sin & b & \ln & + & a & a & b & b & a & a & b & a & b \end{array} \tag{2}$$

and it codes for the following ET:



The corresponding meaning of each chromosome member in the initial population created by the GEP algorithm is different. The genetic operator acts on the genetic manipulation of the population, and the higher the degree of fitness of the resulting offspring individuals, the greater the probability of being selected. GEP adopts linear equal-length coding, so long as the length of the gene is constant and the tail consists only of terminals, a legitimate offspring chromosome can be obtained. Therefore, the GEP genetic operator can be designed more simply and flexibly. The basic genetic operators of GEP mainly include selection, mutation, transposition of insertion sequence elements, root transposition, gene transposition, one-point recombination, two-point recombination, and gene recombination.

Fitness is an indicator of biological ability to measure the ability of a species to adapt to the environment, it can also apply to the GEP algorithm. The fitness function is used to calculate the individual fitness to evaluate the pros and cons. By evaluating the chromosomes, expression trees can be obtained, then the objective function values are calculated from the phenotypes, and finally, according to the conversion rules, the phenotypic (individual) fitness values can be obtained using the objective function values. Ferreira proposed two fitness evaluation models: one is that the fitness f_j of an individual chromosome j is calculated based on the absolute error, and the other is calculated based on the relative error. Let T be a dataset containing n samples, T_i is the i-th input dataset, v_i is the observed value of T_i, and $\hat{v}_i$ is the estimated value. the fitness equations are:

$$f_j = \sum_{i=1}^{n}(K - | v_i - \hat{v}_i |) \tag{3}$$

$$f_j = \sum_{i=1}^{n}(K - | \frac{v_i - \hat{v}_i}{v_i} |) \tag{4}$$

Through the constraint of the fitness function, the population retains the high-adaptation chromosomes during the evolution process. When the individual chromosomes do not have relative or absolute errors in the sample estimates, the accuracy is maximized and the fitness is maximized.

3. An Improved Evolution Flow Based on Thread Mechanism Periodically Population Diversity Improve

When a new chromosome individual is produced by the genetic manipulation of a GEP generation, the sequence of all chromosomes in the generation is descended according to the fitness value, and the population is traversed, it may appear such cases as : (1)chromosome individuals with higher fitness values are ranked before, although the ORF segments of these individuals are different, their genotype structure is very similar, especially the region of operator concentration in the head of the coding region. The expression tree of these chromosomes is similar, the corresponding function model is also very small, and the fitness decreases slowly. (2) The genotypic phase of some sequent chromosomes is very different from the previous individual, and the fitness decreases sudden, while the fitness of the subsequent individuals decreases slowly again. According to this phenomenon, two concepts of GEP family chromosomes and GEP population faults are proposed and defined.

Definition 1. GEP Family Chromosomes: In the GEP population, two or more individuals do not have duplicated or repeated chromosomes, but the head segments of the genotypic coding regions have similar functional structure, and the fitness values are very close, and they are called the same family chromosomes of GEP.

Definition 2. GEP Population fault: After the population is ordered by the fitness value, the GEP family chromosomes are clustered. The last individual of the same family is compared to the next individual outside the family, the gap of the fitness value is far greater than that in the same group, it was said that the population had a fault here.

This paper proposed an improved evolutionary process based on TMPDI, that is, a number of threads generated periodicity run GEP algorithms in parallel, and achieve the purpose of differentiating the main thread population ultimately. The main idea is: when the main thread is awakened and continued running, the probability selection method, instead of the championship, takes the first number of the previous bits which are sorted according the fitness in descend as the parent chromosome of the next generation, so as to avoid the situation that the excellent genes of diversity may be missed. After several (100) generations of evolution, the main thread population may be converged again, meanwhile, the next cycle of creating child threads for population differentiation is yet to come, then the tournament or other probability selection method is used, and the population diversity is temporarily maintained through a certain randomness. When the child thread creation cycle arrives, repeat the above steps and get the distinct population again. The evolution flow is shown in Fig. 1 :

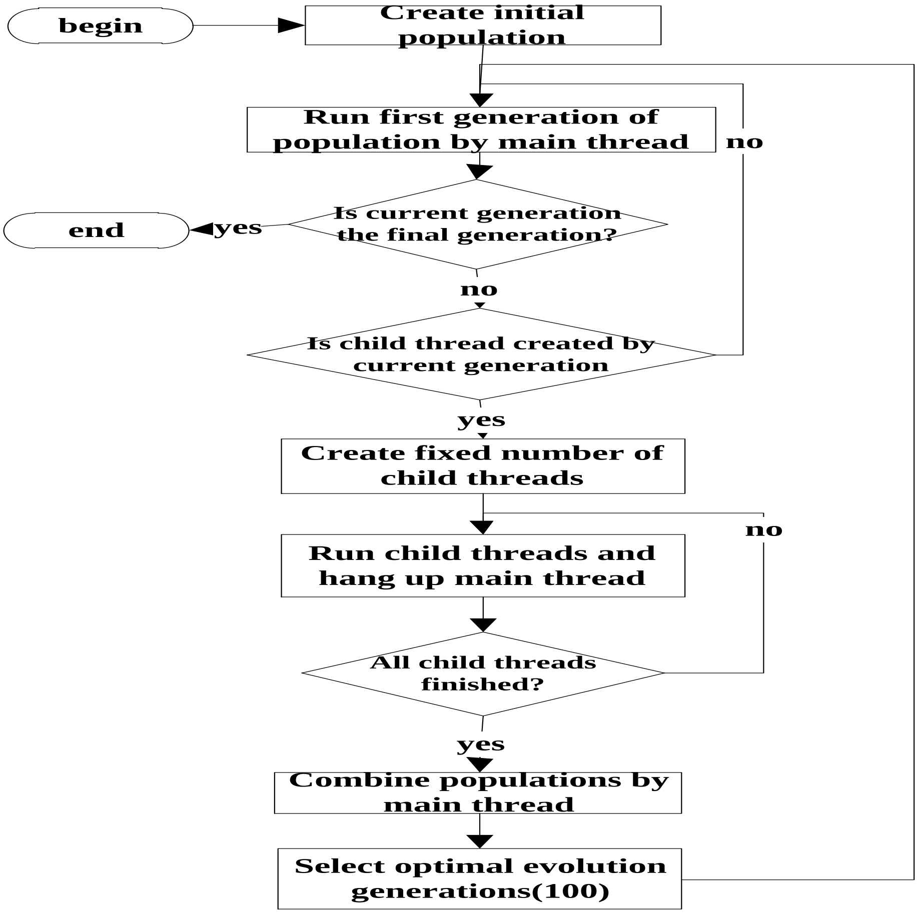

begin
Create initial population
Run first generation of population by main thread
no
end
yes
Is current generation the final generation?
no
Is child thread created by current generation
yes
Create fixed number of child threads
no
Run child threads and hang up main thread
All child threads finished?
yes
Combine populations by main thread
Select optimal evolution generations(100)

Fig.1 The improved evolution flows of TMPDI

It is necessary to create a population for the specified number of child threads in Fig.1, and to set a suitable hereditary generation to achieve a better result of population diversity.

This paper presents an initial child threads population by SHCRRI. In order to amplify the genetic characteristics of individuals under different fitness levels of main thread population, the initialization algorithm first sorted them, and then cloned all the individuals to obtain a copy which was used as the initial population of the child thread 0, and appointed the hereditary generation of child thread 0 to *e*. For the remaining child threads 1 to *threadNum-1*, the sorted population of main thread is divided to *threadNum-1* segments, each child thread clones the chromosomes of their respective section numbers in turn. If the number is less than the number of the initial population, the individual is supplemented by randomization to the specified value. After sorting, the fitness of chromosomal of the latter number is low, and the algorithm increases the fitness by increasing the hereditary generation of child thread, and expects to obtain individuals with better fitness at the end of child thread, such as: child thread 1 inherits e generations, child thread 2 inherits 2*e generations,…, child thread threadNum-1 inherits (t-1)* e generations. The steps of SHCRRI algorithm are follows:

Algorithm 1. Sort Hierarchical Clone Replenish Randomized Individual

Input: the parameters of main thread population, the number of child thread [threadNum], the number of generic generation executed by child thread, symbol regression of child thread population, selection and choosing probability.

Output: array of child thread population threadPopulation[threadNum].

```
sorting the main thread populations;
creating array of child thread population threadPopulation[ threadNum];
for(t=0;t<=threadNUm,t++) {
    creating the instance of population(PopulationForThread) according to the parameters of heredity generations, size of population, symbol regression of child thread population, selection and choosing probability;
    threadPopulation[t]=populationForThread;
    if(t==0) {
        set the heredity generation of child thread 0 be e;
        empty the list of individuals of population of child thread 0;
        for(individual i=0;i<=Num;i++){
            add i to the population of child thread 0;  }
    }
    else{
        set the heredity generation of child thread t be e*t;
        empty the list of individuals of population of child thread t;
        for(i=(t-1)*(Num/(threadNum-1));i<=t*( Num/(threadNum-1));i++){
            add i to the population of child thread t;
        }
        if(the individual number of child thread t < the initial number){
            generate random chromosomes to supplement the individuals of child thread population.
        }
    }
}
return threadPopulation
```

4. Experiment and Analysis

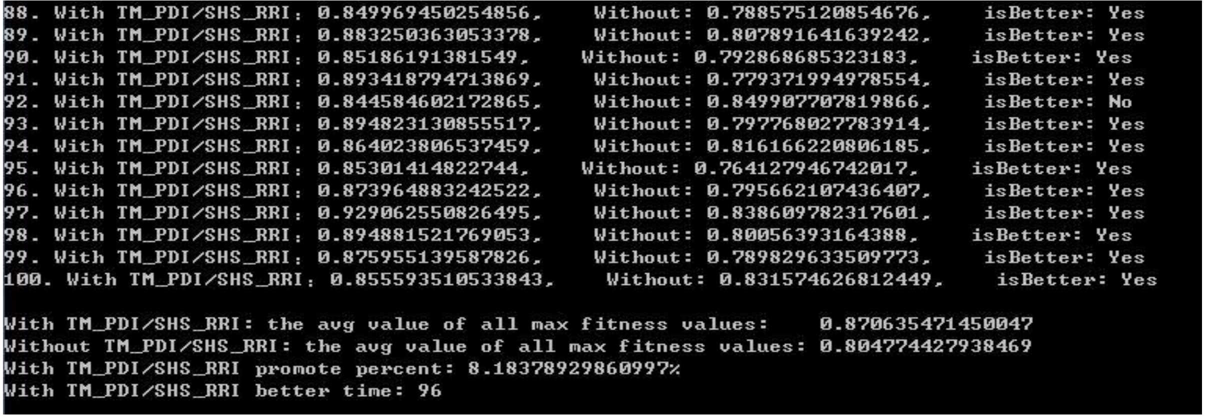
A part of the Iris data test of UCI data set is used to prove the validity of the improved GEP algorithm compared to the standard GEP. The data contain three flower samples, setosa, versicolor and virginica. Each sample contains 4 attributes, including sepal length, sepal width, petal length and petal width. The calyx length of the setosa sample was selected as the dependent variable y, and the other attributes were used as independent variables, which were expressed as a,b and c respectively. Setosa sample data, as shown in Table 1:

Table 1. Iris-Setosa Sample Dataset.

Num	Sepal length y	Sepal width a	Petal length b	Petal width c
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5.0	3.6	1.4	0.2
6	5.2	3.5	1.5	0.2
7	4.6	3.4	1.4	0.3
8	5.0	3.4	1.5	0.2
9	4.4	2.9	1.4	0.2
10	4.9	3.1	1.5	0.1

In order to prove the effect of TMPDI/SHCRRI algorithm on evolution, the experiment was divided into two groups, the experimental group and the control group, both groups use the same data set in Table 1. Both of them are executed based on reverse polish expression_sack decoding method to evaluate gene expression, and based on rough multiple linear regression initialization_adaptive correction constant to initialize constant, and based on creating population copy for the second choice-deleting same chromosome to remove duplicated chromosomes. In addition, the experimental group adopts the TM_PDI thread mechanism to differentiate the population diversity periodically and create the initialization populations during generic evolution process.

Set the size of population to 60, the head length of gene expression to 15, the probability of selection to 0.3, the probability of recombination to 0.75, and the mutation probability to 0.1. In addition, the experimental group creates 8 child threads every 600 generations(600,1200,1800,…), and each child thread executes 100 generations at least. That is, each child thread runs 100, 200, and... 800 generations. Comparing the optimal fitness of the two individuals obtained by the TM_PDI/ SHS_RRI and no TMPDI/SHCRRI algorithm after the 2000 generation, the individual with higher fitness wins. Fig.2 and Fig.3 show the results:



```
88. With TM_PDI/SHS_RRI: 0.849969450254856,    Without: 0.788575120854676,    isBetter: Yes
89. With TM_PDI/SHS_RRI: 0.883250363053378,    Without: 0.807891641639242,    isBetter: Yes
90. With TM_PDI/SHS_RRI: 0.85186191381549,    Without: 0.792868685323183,    isBetter: Yes
91. With TM_PDI/SHS_RRI: 0.893418794713869,    Without: 0.779371994978554,    isBetter: Yes
92. With TM_PDI/SHS_RRI: 0.844584602172865,    Without: 0.849907707819866,    isBetter: No
93. With TM_PDI/SHS_RRI: 0.894823130855517,    Without: 0.797768027783914,    isBetter: Yes
94. With TM_PDI/SHS_RRI: 0.864023806537459,    Without: 0.816166220806185,    isBetter: Yes
95. With TM_PDI/SHS_RRI: 0.85301414822744,    Without: 0.764127946742017,    isBetter: Yes
96. With TM_PDI/SHS_RRI: 0.873964883242522,    Without: 0.795662107436407,    isBetter: Yes
97. With TM_PDI/SHS_RRI: 0.929062550826495,    Without: 0.838609782317601,    isBetter: Yes
98. With TM_PDI/SHS_RRI: 0.894881521769053,    Without: 0.80056393164388,    isBetter: Yes
99. With TM_PDI/SHS_RRI: 0.875955139587826,    Without: 0.789829633509773,    isBetter: Yes
100. With TM_PDI/SHS_RRI: 0.855593510533843,    Without: 0.831574626812449,    isBetter: Yes

With TM_PDI/SHS_RRI: the avg value of all max fitness values:    0.870635471450047
Without TM_PDI/SHS_RRI: the avg value of all max fitness values: 0.804774427938469
With TM_PDI/SHS_RRI promote percent: 8.18378929860997%
With TM_PDI/SHS_RRI better time: 96
```

Fig.2 Results of TMPDI/SHCRRI and non TMPDI/SHCRRI

Fig.3 shows that in the comparison of the 100 experimental results, the optimal fitness of the chromosomes obtained by the TMPDI/SHCRRI algorithm after the 2000 generation is 96 times greater than that of the GEP without TMPDI/SHCRRI(both group using reverse polish expression_sack decoding method to evaluate gene expression, rough multiple linear regression initialization_adaptive correction constant to initialize constant, and creating population copy for the second choice-deleting same chromosome to remove duplicated chromosomes). The result shows that the TMPDI/SHCRRI algorithm can accelerate the evolution speed. One group uses TMPDI/SHCRRI algorithm and the other group does not use TMPDI/SHCRRI, each group is executed 100 times respectively, and after 2000 generations of evolution, the average fitness of optimal chromosome is 0.87064, compared to 0.80478 of not using TMPDI/SHCRRI, increased by 8.18379%.

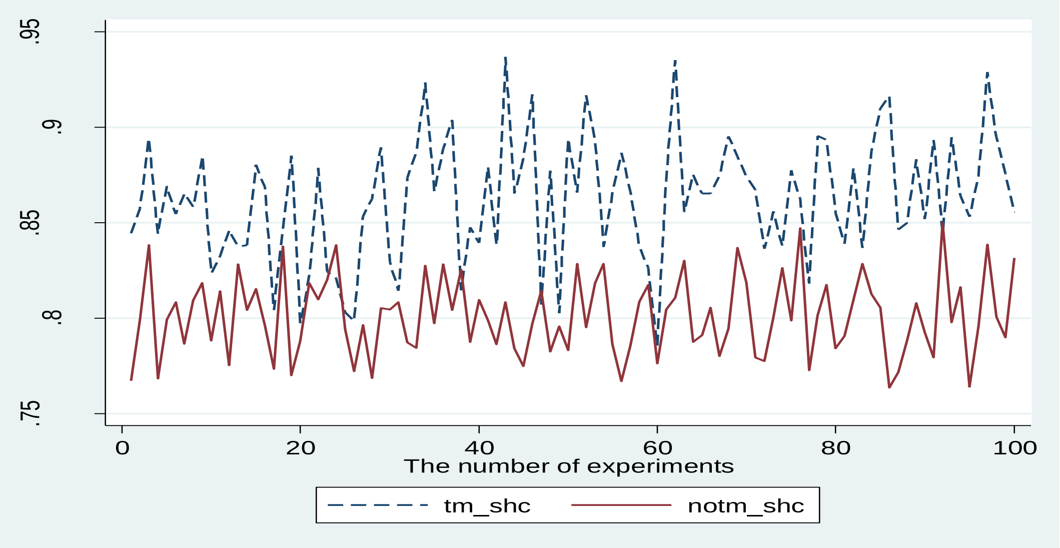


Fig.3 Fitness value of TMPDI/SHCRRI and non TMPDI/SHCRRI

5. Conclusion

In order to solve the problem of gene communication stuck in the whole population and evolutionary convergence to local optimal solution during the genetic process, This paper introduced two concepts GEP family chromosomes and GEP population fault, and proposed an improved evolutionary process based on TMPDI, by generating a number of threads periodically and running GEP algorithms in parallel, to maintain the population diversity through a certain randomness temporarily. It also presented an initial child threads population by SHCRRI to amplify the genetic characteristics of individuals under different fitness levels of main thread population. The experiment results show that the TMPDI/SHCRRI accelerates evolution, increases the fitness of optimal chromosome compared to the result not using TMPDI/SHCRRI.

References

[1] Information on *https://www.gene-expression-programming.com.*

[2] Ferreira Candida. Gene Expression Programming: A New Adaptive Algorithm for Solving Problems. Complex Systems, vol. 13(2001), p.87-129.

[3] Ferreira Candida. Genetic Representation and Genetic Neutrality in Gene Expression Programming. Advances in Complex Systems, vol. 4(2002), p.389-408.

[4] Ferreira Candida. *Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence* (Springer- Verlarg, Germany 2006).

[5] Yuan Rao: *Improved gene expression algorithm and its application in function optimization* (MS., Guangxi Normal University, China 2007).

[6] Li X, Zhou C, Xiao W M. Prefix Gene Expression Programming. *Proceedings of Genetic and Evolutionary Computation*(2005), p.25-29.

[7] Li X, Zhou C, Xiao W M. Direct Evolution of Hierarchial Solutions with Self-emergent Substructures. *Proceedings of The 4th International Conference on Machine Learning and Applications*(California, USA, 2005), p.337-342.

[8] Yunliang Chen, Dan Chen, Samee U.Khan. Solving Symbolic Regression Problems with Uniform Design-aided Gene Expression Programming. Journal of Supercomputing, vol.66 (2013), p. 1553-1575.