An Improved Gene Expression Programming Based on Niche Technology of Outbreeding Fusion

Chao-xue Wang, Jing-jing Zhang, Shu-ling Wu, Fan Zhang
School of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an, China
E-mail: 13991996237@163.com
http://www.xauat.edu.cn/en

Jolanda G. Tromp
Department of Computer Science, State University of New York Oswego, USA
E-mail: jolanda.tromp@oswego.edu

Keywords: gene expression programming, out breeding fusion, niche technology

Received: October 11, 2016

An improved Gene Expression Programming (GEP) based on niche technology of outbreeding fusion (OFN-GEP) is proposed to overcome the insufficiency of traditional GEP in this paper. The main improvements of OFN-GEP are as follows: (1) using the population initialization strategy of gene equilibrium to ensure that all genes are evenly distributed in the coding space as far as possible; (2) introducing the outbreeding fusion mechanism into the niche technology, to eliminate the kin individuals, fuse the distantly related individuals, and promote the gene exchange between the excellent individuals from niches. To validate the superiority of the OFN-GEP, several improved GEP proposed in the related literatures and OFN-GEP are compared about function finding problems. The experimental results show that OFN-GEP can effectively restrain the premature convergence phenomenon, and promises competitive performance not only in the convergence speed but also in the quality of solution.

Povzetek: V prispevku je predstavljena izboljšava genetskih algoritmov na osnovi niš in genetskega zapisa.

1 Introduction

Gene expression programming (GEP) was invented by Candida Ferreira in 2001[1, 2], which is a new achievement of evolutionary algorithm. It inherits the advantages of Genetic Algorithm (GA) and Genetic Programming (GP), and has the simplicity of coding and operation of GA and the strong space search ability of GP in solving complex problems [3]. GEP has more simplify on data set reduction than other intelligent computing technologies such as rough set, clustering, and abstraction. [4-6]. Currently, GEP becomes a powerful tool of function finding and has been widely used in the field of mechanical engineering, materials science and so on [7, 8], but the problem of low converging speed and readily being premature still exists like other evolutionary algorithms.

So far, the domestic and foreign scholars proposed different improvements about the traditional GEP in the field of function finding. Yi-shen Lin introduced the improved K-means clustering analysis into GEP, by adjusting the min clustering distance to control the number of niches, to improve the global searching ability [9]. Tai-yong Li designed adaptive crossover and mutation operators, and put forward the measure method of population diversity with weighted, to maintain the population diversity in the process of evolution [10]. Yong-qiang Zhang introduced the superior population producing strategy and various population strategy to improve the convergence speed of the algorithm and the diversity of population [11]. Shi-bin Xuan proposed the control of mixed diversity degree (MDC-GEP) to ensure the different degree in the process of evolution and avoid trapping into local optimal [12]. Hai-fang Mo adopted the clonal selection algorithm with GEP code for function modelling (CSA-GEP), to maintain the diversity of population and increase the convergence rate [13]. Yan-qiong Tu proposed an improved algorithm based on crowding niche, and the algorithm contributed to push out the premature individuals by penalty function and made the better individuals have greater probability to evolve [14].

To further improve the performance of the GEP, this paper proposes an improved gene expression programming based on niche technology of outbreeding fusion (OFN-GEP). The main ideas are as follows: (1) using the population initialization strategy of gene equilibrium to ensure that all genes are evenly distributed in the coding space as far as possible; (2) introducing the outbreeding fusion mechanism into the niche technology, to eliminate the kin individuals, fuse the distantly related individuals, and promote the gene exchange between the excellent individuals from different sub-populations. The experiments compared with other GEP algorithms

proposed in the related literatures about function finding problems are executed，and the results show that OFN-GEP can overcome the premature convergence phenomenon effectively during the evolutionary process, and has high solution quality, fast convergence rate.

2 Standard gene expression programming

Standard gene expression programming (ST-GEP), which was firstly put forward by Candida Ferreira in 2001[1, 2], could be defined as a nine-meta group: $GEP = \{C, E, P_0, M, \varphi, \Gamma, \Phi, \Pi, T\}$, where C is the coding means; E is the fitness function; P_0 is the initial population; M is the size of population; φ is the selection operator; Γ is the crossover operator; Φ is the point mutation operator; Π is the string mutation operator; T is the termination condition. In GEP, individual is also called chromosome, which is formed by gene and linked by the link operator. The gene is a linear symbol string which is composed of head and tail. The head involves the functions from function set and the variables from the terminator set, but the tail merely contains the variables from the terminator set. Like GA and GP, GEP follows the Darwinian principle of the survival of the fittest and uses populations of candidate solutions to a given problem to evolve new ones, and the basic steps of ST-GEP are as follows [1, 2]:

(1) Inputting relevant parameters, creating the initial population;
(2) Computing the fitness of each individual;
(3) If the termination condition is not met, go on the next step, otherwise, terminate the algorithm;
(4) Retaining the best individual;
(5) Selecting operation;
(6) Point mutating operation;
(7) String mutating operation (IS transposition, RIS transposition, Gene transposition);
(8) Crossover operation (1-point recombination, 2-point recombination, Gene recombination);
(9) Go to (2).

3 Gene expression programming based on niche technology of outbreeding fusion

The flowchart of the OFN-GEP is schematically represented in Fig 1. Its main steps are as follows:

Step 1: Adopt the population initialization strategy of gene equilibrium to generate the population P, and set up the maximum *MAX* and the minimum *MIN* about the number of individuals in the niche. Then divide the initial population into several equal niches;

Step 2: Perform the genetic operators within each niche, which including point mutation, string mutation (IS, RIS, Gene transposition) and recombination (1-point, 2-point, gene recombination). Then use the pre-selection operator to protect the best individual in every niche;

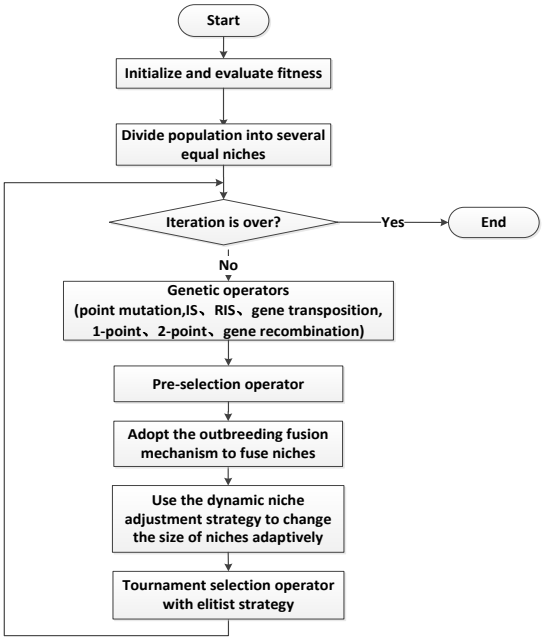

Figure 1: The flowchart of OFN-GEP.

Step 3: Use the outbreeding fusion mechanism to eliminate the kin individuals and fuse the distant relatives between two niches, and introduce some random individuals at the same time;

Step 4: Adopt the dynamic adjustment strategy to change the size of niches according to the maximum *MAX* and the minimum *MIN*;

Step 5: Perform the tournament selection operator with elitist strategy;

Step 6: Go to Step 2 until the iteration is over.

3.1 Population initialization

This algorithm adopts the population initialization strategy of gene equilibrium to increase the initial population diversity. The idea of this strategy is to let all genes are distributed uniformly in the coding space, so that the initial population diversity is rich. This strategy can reduce the time of search process and achieve the global optimal solution at a rapid speed [15].

3.2 Fitness function

In statistics, the method to assess the relevance degree between two groups of data usually uses the correlation coefficient. References [16], the fitness function is devised as: $fitness=R^2=1-SSE/SST$, where

$$SSE = \sum_{j=1}^{m} (y_j - \hat{y}_j)^2 \tag{1}$$

$$SST = \sum_{j=1}^{m} (y_j - \bar{y})^2 \tag{2}$$

where, y_j is the observation data; $\hat{y}_j$ is the forecast data which is computed with formula and observation data; $\bar{y}$ is the mean of y ; *SSE* is the residual sum of squares;

SST is the total sum of squares of deviations; *m* is the size of data.

3.3 Pre-selection

The pre-selection operator is: the offspring individual can instead of his father and access to the next generation only when the fitness of the new individual is bigger than his father. Due to the similarity of the offspring individual and his father, an individual can be replaced by his structure similar individual, which can maintain the diversity of population and protect the best individual in population.

3.4 A niche technology of outbreeding fusion

The niche technology of outbreeding fusion includes two aspects: one is to use the outbreeding fusion mechanism to eliminate the kin individuals, fuse the distant relatives between two niches and promote the gene exchange between the best individuals, which improving the diversity of population and the quality of solutions; the other is to use the dynamic niche adjustment strategy to change the size of niches adaptively [17], which maintaining the genetic diversity of niches.

Aiming at the judgment of the distant individuals in outbreeding fusion, this paper adopts the calculation methods of recessive hamming distance (individual fitness, the essence differences between individuals), and dominant hamming distance (the appearance differences between individuals), and the judge rules between kin and distant relatives as well in literature [18], to judge the kinship between individuals.

The niche technology of outbreeding fusion operators is as follows:

(1) Select two niches randomly, and merge all individuals of the two niches (which are supposed as N1 and N2, and before fusion, their sizes are S_1 and S_2 respectively) into niche N1; go to (2);

(2) Adopt the outbreeding fusion strategy (Algorithm 1) to eliminate the kin individuals, and then obtain the size S_1' of the modified N1; go to (3);

(3) If S_1' is bigger than the maximum *MAX*, corresponding *MAX* individuals will be selected out by tournament selection; then adjust S_1' and go to (5); else go to (4);

(4) If S_1' is smaller than minimum *MIN*, the new individuals will be introduced randomly until the smallest size *MIN* is satisfied; then adjust S_1' and go to (5);

(5) Construct N2 by the individuals generated randomly, and the size S_2' of N2 satisfies the equation $S_2' = S_1 + S_2 - S_1'$.

Algorithm 1: Outbreeding judgment

- Sort the fitness of the fusion individuals in ascend (or descend);
- Compute the dominant and recessive hamming distance between two adjacent individuals;
- If the dominant hamming distance between two adjacent individuals is less than the setting threshold M_1, and the recessive hamming distance is less than the setting threshold M_2, then the two individuals are kin relatives; otherwise, they are the distant relatives;
- Eliminate the lower fitness individual between the kin relatives, and retain the other one.

4 Experiments and results

In this section, two experiments are designed to justify the effectiveness and competitiveness of OFN-GEP for function finding problems, the general parameters setting of experiments are shown as Table 1. The source codes are developed by MATLAB 2009a, and run on a PC with i7-2600 3.4 GHz CPU, 4.0 GB memory and Windows 7 professional sp1.

4.1 Test for the effectiveness of OFN-GEP

To evaluate the improved effect of OFN-GEP, this paper adopts the F function, which was used in literature [1] as shown in equation (3), and the 10 groups of training data are produced by F. OFN-GEP is compared with the DS-GEP in literature [19]. The test results are shown in Table 2, the evolution curve is shown in Fig 1, Fig 2

Table 1: The parameter settings of experiments.

Option	Test A	Test B
Times of runs	50	50
Max evolution generation	200	200
Size of population	100	100
Function set	+,-,*,/	+, -, *, /, ln, exp, S, Q, sin, cos, tan, cot
Terminator set	a	
Link operator	+	+
The length of head	6	6
Number of gene	5	5
Point mutation rate	0.4	0.4
IS and RIS rate	0.3	0.3
Crossover rate	0.3	0.3
Recombination rate	0.3	0.3
Length of IS element	{1,2,3,4,5}	{1,2,3,4,5}
Length of RIS element	{1,2,3,4,5}	{1,2,3,4,5}
Size of tournament	3	3
The number of niches	5	5
The minimum threshold of the size of niche	10	10
The maximum threshold of the size of niche	60	60
The threshold of the recessive hamming distance	0.1	0.1
The threshold of the dominant hamming distance	0.5	0.5

Note: S is Square, Q is Sqrt, and Exp is e^x.

separately, where both the optimizing rate and the average generations of convergence of OFN-GEP are obviously superior to DS-GEP.

$$F:5a_n^4+4a_n^3+3a_n^2+2a_{\mathrm{n}}+1 \tag{3}$$

Table 2: The results of experiment A.

Option	DS-GEP	OFN-GEP
Times of runs	50	50
Times of hit	41	48
Optimizing ratio	82%	96%
Average generations of convergence	87	41

As is shown in Table2, the average generations of convergence that achieves the optimal solution in OFN-GEP algorithm is less than the one in DS-GEP, so the OFN-GEP algorithm can improve convergence speed efficiently. From the evolution curves in Fig 2 and Fig 3 the volatility of average fitness in OFN-GEP is greater than the one in DS-GEP, and this says the differences between individuals are greater and the population diversity is better in OFN-GEP.

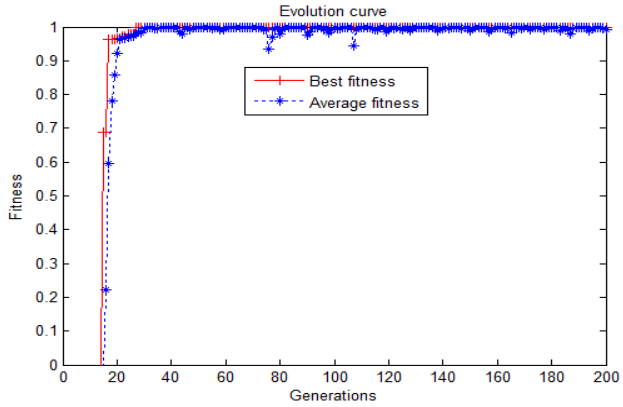

Figure 2: The evolution curve of DS-GEP.

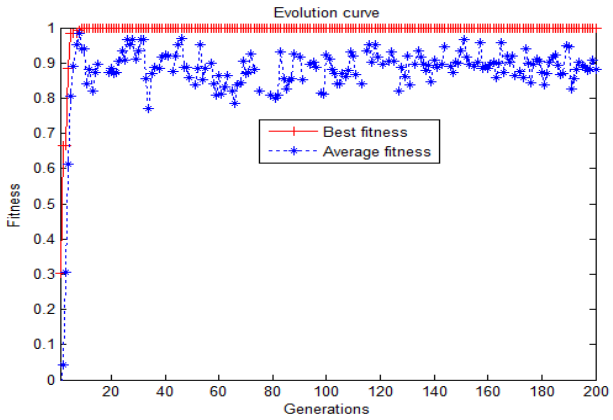

Figure 3: The evolution curve of OFN-GEP.

4.2 Test for the competitiveness of OFN-GEP

To test the competitiveness of OFN-GEP, MDC-GEP [12] and S-GEP [20] are chosen to compare with OFN-GEP. Test functions are partly the same with the ones in [12] and [20]. They are shown as (4) - (14). The test results are shown in Table 3.

$$F1:8+2e^{1-x^2}\cos(2\pi x),x\in[0,5] \tag{4}$$

$$F2:\cos^2(x^2),x\in[0,3] \tag{5}$$

$$F3:\frac{1}{2+\sin(x)},x\in[0,5\pi] \tag{6}$$

$$F4:5+\log(\cos^2(2\pi x)+x^2),x\in[0,5\pi] \tag{7}$$

$$F5:\frac{3x^3+2x^2+1}{5x^3+2},x\in[0,5] \tag{8}$$

$$F6:\cos(10^x),x\in[0,2] \tag{9}$$

$$F7:\frac{3x^2+5x+1}{5y^2+3},x,y\in[0,5] \tag{10}$$

$$F8:\frac{\sin(x^2+2)}{\cos(y^3+2.5)+3},x,y\in[-5,5] \tag{11}$$

$$F9:xye^{-x^2-y^2},x,y\in[-2,2] \tag{12}$$

$$F10:\frac{\sin x_1+\cos x_2}{\sqrt{e^{x_3}}}+\tan(x_4-x_5) \tag{13}$$
$$x_i\in[0,2\pi],i=1,2,....5$$

$$Fv:4.251a^2+\ln(a^2)+7.243e^a,a\in[-1,1] \tag{14}$$

Table 3: Test results of experiment B.

Function	Algorithm	Max fitness	Min fitness	Average fitness
F1	MDC-GEP	0.9675	0.8231	0.9263
	OFN-GEP	0.96825	0.8782	0.9334
F2	MDC-GEP	0.9991	0.7865	0.9371
	OFN-GEP	1	0.8112	0.9446
F3	MDC-GEP	0.9916	0.8645	0.9843
	OFN-GEP	0.9959	0.8901	0.9505
F4	MDC-GEP	0.9812	0.8743	0.9476
	OFN-GEP	0.9898	0.9430	0.9696
F5	MDC-GEP	0.9587	0.6856	0.8735
	OFN-GEP	0.9413	0.7641	0.8408
F6	MDC-GEP	0.9954	0.8237	0.9465
	OFN-GEP	1	0.9603	0.9913
F7	MDC-GEP	0.9462	0.8133	0.8956
	OFN-GEP	0.9792	0.8387	0.9317
F8	MDC-GEP	0.9473	0.7012	0.8653
	OFN-GEP	0.9525	0.7464	0.8719
F9	MDC-GEP	0.9673	0.6782	0.8750
	OFN-GEP	0.9415	0.7099	0.8777
F10	MDC-GEP	0.9771	0.8954	0.9520
	OFN-GEP	0.9909	0.9109	0.9605
Fv	S-GEP	0.9991		
	OFN-GEP	0.9988	0.9796	0.9926

From Table 3, for most functions, the max fitness, min fitness and average fitness increase obviously in OFN-GEP compared with the MDC-GEP, S-GEP. This shows the effectiveness and competitiveness of OFN-GEP. For relatively simple function F2 and F6, the fitness of OFN-GEP can achieve 1, but for F5, F9, Fv, their fitness is less than (very close to) the results in MDC-GEP. The reasons are that GEP algorithm is a random algorithm, the algorithm parameters have great influence on the results of experiment, and the parameters of every function to obtain the best fitness are different. So, this situation exists which few functions can't obtain a better fitness value under the same parameters.

5 Conclusion

This paper puts forward an improved gene expression programming based on niche technology of outbreeding fusion (OFN-GEP), and verifies the effectiveness and competitiveness of the proposed algorithm about the function finding problems. The improvements in the paper are that: (1) using the population initialization strategy of gene equilibrium to ensure that all genes are evenly distributed in the coding space as far as possible; (2) introducing the outbreeding fusion mechanism into the niche technology, to eliminate the kin individuals, fuse the distantly related individuals, and promote the gene exchange be-tween the excellent individuals from different sub-populations. To validate the effectiveness and competitiveness of OFN-GEP, several improved GEP proposed in the related literatures and OFN-GEP are compared as regards function finding problems. The experimental results show that OFN-GEP can effectively restrain the premature convergence phenomenon, and promises competitive performance not only in the convergence speed but also in the quality of solution.

6 Acknowledgment

Support from the Natural Science Basic Research Plan in Shanxi Province of China (NO.2012JM8023), and the Scientific Research Program Funded by Shanxi Provincial Education Department (No.12JK0726) are gratefully acknowledged.

7 References

[1] Ferreira C (2001), Gene Expression Programming: A New Adaptive Algorithm for Solving Problems, *Complex System*, Complex Systems Publications, vol.13 no 2, pp 87–129.

[2] Ferreira C (2003), Function Finding and the Creation of Numerical Constants in Gene Expression Programming, In: Benítez J.M., Cordón O., Hoffmann F., Roy R. (Eds), *Advances in Soft Computing*, Springer-verlag, pp 257–266.

[3] Gerald Schaefer (2016), Gene Expression Analysis based on Ant Colony Optimisation Classification, *International Journal of Rough Sets and Data Analysis*, IGI Global, vol. 3, no.3, pp.51-59.

[4] Debi Acharjya, A. Anitha (2017), A Comparative Study of Statistical and Rough Computing Models in Predictive Data Analysis, *International Journal of Ambient Computing and Intelligence*, IGI Global, vol.8, no.2, pp.32-51.

[5] Hans W. Guesgen, Stephen Marsland (2016), Using Contextual Information for Recognising Human Behaviour, *International Journal of Ambient Computing and Intelligence*, IGI Global, vol. 7, no.2, pp.27-44

[6] Ch. Swetha Swapna, V. Vijaya Kumar, J.V.R Murthy (2016), Improving Efficiency of K-Means Algorithm for Large Datasets, *International Journal of Rough Sets and Data Analysis*, IGI Global, vol.3, no.2, pp.1-9.

[7] A.H. Gandomi, A.H. Alavi (2013), Intelligent Modeling and Prediction of Elastic Modulus of Concrete Strength via Gene Expression Programming, *Lecture Notes in Computer Science*, Springer-verlag, vol. 7928I, pp 564–571.

[8] Y. Yang, X.Y. Li (2016), Modeling and impact factors analyzing of energy consumption in CNC face milling using GRASP gene expression programming, *International Journal of Advanced Manufacturing Technology*, Springer-verlag, Vol.87, no.5, pp.1247–1263.

[9] Y.S. Lin, H. Peng, J. Wei (2008), Function Finding in Niching Gene Expression Programming, *Journal of Chinese Computer Systems*, Chinese Computer Society, vol.29, pp.2111–2114.

[10] T.Y.LI, C.J. Tang (2010), Adaptive Population Diversity Tuning Algorithm for Gene Expression Programming, *Journal of University of Electronic Science and Technology of China*, UESTC Press, vol. 39, no.2, pp. 279–283.

[11] Y.Q. Zhang, J. Xiao (2010), A New Strategy for Gene Expression Programming and Its Applications in Function Mining, *Universal Journal of Computer Science and Engineering Technology*, Springer-verlag, vol.1, no.2, pp.122–126.

[12] S.B. Xuan, Y.G. Liu (2012), GEP Evolution Algorithm Based on Control of Mixed Diversity Degree, *Pattern Recognition and Artificial Intelligence*, Chinese Association Automation, vol. 25. no.2, pp.187–194.

[13] H.F. MO (2013), Clonal Selection Algorithm with GEP Code for Function Modeling, *Pattern Recognition & Artificial Intelligence*, Chinese Association Automation, vol.26, no9, pp.878–884, 2013.

[14] Y.Q. Tu, X. Wang (2013), Application of improved gene expression programming for evolutionary modeling, *Journal of Jiangxi University of Science and Technology*, JUST Press, vol. 34, no.5, pp.77–81.

[15] L. Yao, H. Li (2012), An Improved GEP-GA Algorithm and Its Application, Communications in Computer & Information Science, Springer, vol.316, pp 368-380.

[16] J. Zuo (2004), *Research on Core Technology of Gene Expression Programming*, Sichuan: Sichuan University.

[17] Y. L. Chen, F. Y. Li, J. Q. Fan (2015), Mining association rules in big data with NGEP, *Cluster Computing*, Kluwer Academic Publishers, vol.18, no2, pp.577-585.

[18] Y. Jiang, C. J. Tang (2007), Outbreeding Strategy with Dynamic Fitness in Gene Expression Programming, *Journal of Sichuan University (Engineering Science Edition)*, Sichuan University Press, vol.39. no2, pp.121-126.

[19] C. X. Wang, K. Zhang, H. Dong (2014), Double System Gene Expression Programming and its application in function finding, *the Proceedings of International Conference on Mechatronics, Control and Electronic Engineering(MCE2014)*, Atlantis Press, pp 357-361.

[20] Y. Z. Peng, C. A. Yuan, X. Qin, J. T. Huang (2014), An improved Gene Expression Programming approach for symbolic regression problems, *Neurocomputing*, Elsevier, vol.137, pp 293-301.