CHEMICAL ENGINEERING TRANSACTIONS
VOL. 46, 2015
A publication of AIDIC
The Italian Association of Chemical Engineering
Online at www.aidic.it/cet
Guest Editors: Peiyu Ren, Yanchang Li, Huiping Song

ISBN 978-88-95608-37-2; **ISSN** 2283-9216
DOI: 10.3303/CET1546218

Data Fitting Based on Genetic Programming and Least Square Method

Pinchao Meng[a], Weishi Yin[a], Zhixia Jiang[a], Yanzhong Li*[b]

[a] Department of Applied Mathematics, Changchun University of Science and Technology, Changchun, China,
[b] College of mathematics and statistics, Beihua University, Jilin, China.
lyz@cust.edu.cn

Traditional data fitting techniques usually require estimating basis function and they are specific for different application areas. Based on dynamic characteristics of genetic programming, a two-phase data fitting algorithm is proposed. In this algorithm, genetic programming is used to optimize model structure and Least Square method is applied to estimate parameters. Proposed algorithm is tested for different types of data fitting. This high efficiency and accuracy algorithm can be applied in different areas.

1. Introduction

Professor Koza proposed Genetic Programming Algorithm in 1989. Subsequently, genetic programming has been widely studied. Algorithm studies mainly include the selection of fitness function, evolution strategies and genetic operator optimization, as well as corresponding algorithm designs. The studies also combine specific problems with practices. In theoretical research, with the help of fitness state diagram and pattern theorem, adaptive mechanism analysis of genetic programming has been carried out.
After that, genetic programming began to integrate with other computational intelligence algorithms, such as decision tree algorithms, artificial neural networks, fuzzy inference and rough set algorithms. The new improved variants have very good performance. Later on, some people have combined genetic programming with machine learning methods. Currently, applying genetic programming into machine learning has become a very popular research topic.
In this paper, in order to overcome the unknown solution structure of the least square method and errors occurred in parameter determination of genetic programming, genetic programming was combined with least square method to conduct data fitting.

2. Genetic programming

Genetic Programming is a new search optimization technology. It imitates the evolution of biological and genetics and follows the principles of "struggle for existence" and "survival of the fittest". Based on this, it can approach the optimal solution of corresponding problems from the initial solution step by step utilizing duplication, exchange and mutation operations. The hierarchical format of computer programs can be used to reflect problems, which is suitable for a variety of complex issues. Its mission is to implement computer programs which can reflect the essence of real problems. The problem solving process of genetic programming is to find out a computer program with best fitness in a search space which is composed of many possible computer programs. And genetic programming can exactly provide the method to find the computer program with the best fitness.

2.1 General steps of genetic programming

The general steps of genetic programming are listed as following:
(1) Generate an initial population randomly, i.e produce computer programs composed of random functions and variables;
(2) Run every computer program (individual) in the population and give each of them a fitness based on their problem-solving abilities (good or bad);

Please cite this article as: Meng P.C., Yin W.S., Jiang Z.X., Li Y.Z., 2015, Data fitting based on genetic programming and least square method, Chemical Engineering Transactions, 46, 1303-1308 DOI:10.3303/CET1546218

(3) Generate a new generation of computer programs based on the following two steps. The individuals to be duplicated are selected randomly according to their fitness;
(a) Generate a new generation of computer programs from the current generation by duplicating. The individuals to be duplicated are selected randomly according to their fitness;
(b) Generate new computer programs by exchanging the randomly selected parts of the two parent individuals. The parent individuals are also selected randomly according to their fitness;
(4) Execute steps (2) and (3) iteratively until the termination criterion is satisfied.
The best individuals produced by any generation are considered as potential results of genetic programming. These results may become the correct solutions of corresponding problem which was confirmed (M.Ebner et.al (2007)).

2.2 Generation of the initial population

The initial population consists of many initial individuals. And initial individuals, which are a variety of possible symbolic expressions of the problem to be solved, are produced randomly.
At the beginning, a function is selected from the function set F according to uniform distribution as the root node of the tree algorithm. The purpose of confining the node root in the function set F is to generate a hierarchical complex structure.
We assume that there n functions in the curve fitting function set F :

$$F = \{ f_1, f_2, \ldots, f_n \}. \cdot$$

The function f_i in the function set can either be arithmetic operators such as +, -, × and ÷ or standard math functions such as sin, cos, log, exp, etc, or be other subfuncions which are already defined. The terminator set T includes m terminators:

$$T = \{ t_1, t_2, \ldots, t_m \}.$$

The terminators in the terminator set can be variables, such as x, y and z, or constants, such as a and b.

2.3 Fitness function

Fitness is the driving force of natural selection in genetic programming. Darwin's principle of evolution is "survival of the fittest", which means that good individuals will survive and bad individuals will be abandoned. Then optimization of the population can be achieved through generations of genetic recombination. A measurement is needed to judge whether an individual is good or bad. This measurement is fitness. It can provide a fitness measuring value for each individual utilizing valuation methods of specific issues. In general, there are four common fitness measurement parameters: original fitness, standard fitness, conciliatory fitness and normalized fitness. Fitness is the driving force of genetic programming.

2.4 Genetic operator

Like genetic algorithms, the main operators of genetic programming are duplication and crossover. The duplication operation (i.e, the better the fitness of an individual, the greater its possibility of being involved in the duplication) is a process in which best individuals are selected from the current generation based on fitness to generate a new generation through self-reproduction. Specifically, it consists of two parts: (1) select a parent individual based on fitness from the population using different selection methods; (2) the selected parent individual duplicates itself to the next generation from the current generation without any change.

2.4.1 Selection

When duplication takes place, a parent individual will generate an offspring individual and the duplication process consists of two parts. Firstly, a parent individual will be selected from the population according to the fitness; secondly, the parent individual duplicates the next generation without any change. Generally, selection methods can be divided into the following four types based on fitness: (1) Proportion selection; (2) Classification selection;(3) Competition selection; (4) Elite selection.
Generally, for a specific problem, the above methods can be combined or you can just customize more appropriate selection method based on the actual situation of the problem.

2.4.2 Crossover

Crossover refers to the operation in which parts of two parent individuals' structure exchange with each other to give birth to two new offspring individuals. Its method and procedures are listed as follows: select two individuals randomly from the parent population according to the selection method mentioned above, but both selection processes must be independent. In this way, both structure and size of the selected two parent individuals may be different with each other. The two selected parent individuals select exchange points using a random method of uniform distribution, generating a subtree whose root is the crossover point. The part of the subtree below the exchange point, including the crossover point, is called crossover section. Sometimes an exchange section is only a leaf. Then, the first individual deletes its exchange section, and then puts the

crossover section of the second individual on its own exchange point, and the second individual does the same thing. That is, two new offspring individuals are generated after crossover and exchange of the two parent individuals.

2.5 Mutation

Crossover operation adopts a hybridization method with partial match. That is, two bit string crossover points are generated based on uniform distribution and then the area between these two points will be selected as the match crossover area. And position swap operation will be carried out to exchange the bit strings in the matching crossover area. For the chromosome, gene bits will be selected randomly to execute mutation, and the mutation space is the set of all available actions of the corresponding layer, and idle actions are also involved in mutation.

In genetic programming, the mutation operation is a kind of auxiliary operation. The mutation probability is generally small and its usual range is 0-0.01. The selection method of each mutated individual is: select K_1 individuals from the population randomly and carry out mutation on the individual with the worst fitness K_1. When $p_\alpha = \tilde{p}$, random mutation is carried out.

2.6 Termination criterion

Genetic programming will execute the evolution iterations continuously. In this process, once a termination criterion is satisfied, the evolutionary process shall stop immediately. Generally, there are two termination criterions:

When the maximum allowable number of evolution generations G is satisfied, the evolutionary process shall stop immediately.

When the preset problem solving conditions are met, the evolutionary process shall stop immediately.

3. Data fitting algorithm based on genetic programming and least square method

The function structure has to be specified when the least square method is used to carry out data fitting, however, the structure of the fitting function is usually unknown in practical applications. Genetic programming method can determine the function structure based on the data, but the parameters of the function may not be optimal. So first of all, we use the genetic programming algorithm to determine the function structure, then utilize the least square method to optimize parameters. This process will be repeated to obtain excellent results.

The steps of the data fitting algorithm based on genetic programming and least square method (LSGP) are listed as follows:

(1) Determine parameters, including function set and terminator set. Randomly generate variable parameters. For the function set:

$$P = \{+, -, *, \exp, \log, \sin\}.$$

For the terminator set:

$$T = \{p, rand\}.$$

(2) Generate of the initial population. The growth method is used to generate M initial individuals randomly. These M initial individuals will form the initial population.

(3) Fitness selection. Use original fitness to calculate the fitness individuals. If the termination criterion that the fitness is less than 0.01 is satisfied, then stop. Otherwise go to step 4.

(4) Use the genetic operators of genetic programming to carry out the following operations on M individuals:

Duplication: it is a selection method which adopts fitness proportion. The higher the individual fitness, the greater the probability of being selected. Doing this can ensure that the good individuals will enter the next population.

Exchange: select two individuals from M individuals randomly for exchange to generate two new individuals. The process will be repeated until the number of exchanges is satisfied.

Mutation: a single point mutation is used. Select two individuals from M individuals randomly, then carry out mutation randomly on the selected node. The process will be repeated until the number of mutations is satisfied.

(5) Select part of the individuals and use the least square method to optimize the corresponding function parameters of each individual, generating M new individuals.

(6) Go to step (3) and start looping until the termination condition is met.

4. Calculation examples

4.1 Example 1

A group of data (x_i, y_i), is shown in Table 1:

Table 1: First group of data

x_i	-1	-0.9	-0.8	-0.7	-0.6	-0.5	-0.4	-0.3	-0.2	-0.1	0
y_i	-1.9	-2.2	-2.4	-2.5	-2.5	-2.3	-2.0	-1.6	-1.1	-0.6	0
x_i	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	
y_i	0.6	1.1	1.6	2.0	2.3	2.5	2.5	2.4	2.2	1.9	

For the function set:

$$P = \{+,-,*,\exp,\log,\sin\}.$$

For the terminator set:

$$T = \{p, rand\}.$$

Parameters of genetic programming are: the number of individuals in the initial population M = 100, the number of iterations N = 100, the termination condition is that the fitness is less than 0.01. Run the genetic programming algorithm.

Parameters of the combination of genetic programming and least square method (LSGP) are: the number of individuals in the initial population M = 100, the number of iterations N = 100, the termination condition is that the fitness is less than 0.01. Take out the first 50 individuals with variable parameters after running the genetic programming algorithm every 5 times. Then use the least square method to optimize the parameters of these individuals. The functions after being optimized will rejoin the population to replace the original individuals. After that, the genetic programming algorithm will be conducted on the new population. This process is considered as one cycle and it will be repeated 20 times.

50 experiments have been carried out and the results are shown in Table 2:

Table 2: The experimental result of Example 1

Method	Iteration Number	Average Iteration Number
GP	26.36	21.22
LSGP	32.78	18.40

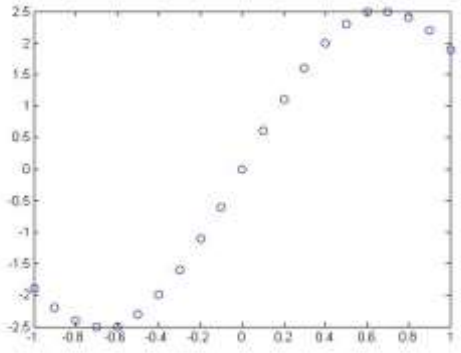

Figure 1: Scatter diagram of the experimental data of Example 1

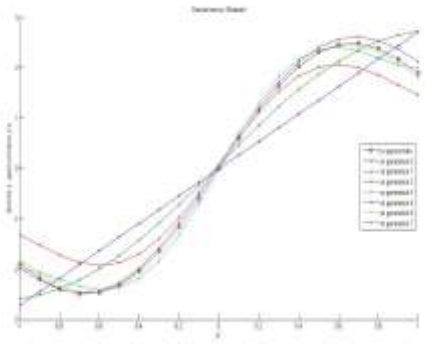

Figure 2: Graph of the fitting function of Example 1

It can be seen from Table 2 that: Compared with genetic programming, the combination between genetic programming and least square method has improved the number of convergences of the corresponding algorithm, and the average convergence algebra also improves. As can be seen from Figure 1 and Figure 2, the data points of the function obtained in this way distribute evenly next to the graph of the fitting function and the fitting accuracy is high. Considering these two aspects, the combination of genetic programming and least square method wins out.

4.2 Example 2

A group of data (x_i, y_i), is shown in Table 3:

Table 3: Second group of data

x_i	22.00	27.00	32.00	37.00	40.00	45.00	48.00	49.00	52.00	59.00
y_i	3.08	4.25	4.01	2.97	4.43	4.65	3.10	2.94	4.94	4.71
x_i	65.00	66.00	70.00	77.00	80.00	82.00	83.00	85.00	88.00	90.00
y_i	5.00	4.16	5.02	5.34	3.39	4.72	5.39	4.27	4.51	5.39

For the function set:

$$P = \{+,-,*,\exp,\log,\sin\}.$$

For the terminator set:

$$T = \{p, rand\}.$$

Parameters of genetic programming are: the number of individuals in the initial population M = 100, the number of iterations N = 100, the termination condition is that the fitness is less than 0.01. Run the genetic programming algorithm.

Parameters of the combination of genetic programming and least square method(LSGP) are: the number of individuals in the initial population M = 100, the number of iterations N = 100, the termination condition is that the fitness is less than 0.01. Take out the first 50 individuals with variable parameters after running the genetic programming algorithm every 5 times. Then use the least square method to optimize the parameters of these individuals. The functions after being optimized will rejoin the population to replace the original individuals. After that, the genetic programming algorithm will be conducted on the new population. This process is considered as one cycle and it will be repeated 20 times.

50 experiments have been carried out and the results are shown in Table 4:

Table 4: The experimental result of Example 1

Method	Iteration Number	Average Iteration Number
GP	17.03	24.20
LSGP	25.59	20.00

It can be seen from Table 4 that: Compared with genetic programming, the combination between genetic programming and least square method has improved the number of convergences of the corresponding algorithm, and the average convergence algebra also improves. As can be seen from Figure 3 and Figure 4, the data points of the function obtained in this way distribute evenly next to the graph of the fitting function and the fitting accuracy is high. Considering these two aspects, the combination of genetic programming and least square method wins out.

It can be seen by comparing Figure 2 with Figure 4 that the experimental data of Example 2 look more disorderly than those of Example 1. It is difficult to carry out data fitting on such data using least square method. But the fitting results using the fitting function proposed in this paper are very good.

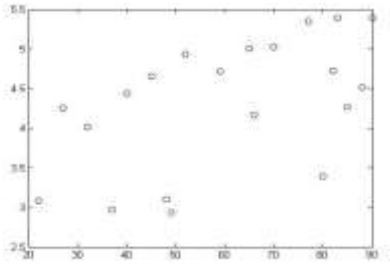

Figure 3: Scatter diagram of the experimental data of Example 2

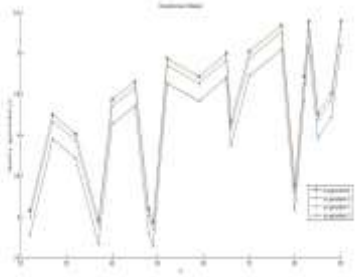

Figure 4: Graph of the fitting function of Example 2

5. Conclusions

The rapid development and wide application of genetic programming has been established in a history of about thirty years. As a branch of evolutionary computation, it has many common characteristics of evolutionary computation, such as global optimization, self-learning ability and independence of gradient. Traditional data fitting methods need to pre-determine the structure of the fitting function based on empirical knowledge, thus the designed programs can only solve a certain problem in a certain area, of which the portability is poor. While genetic programming has dynamic variable characteristics and does not require any prior knowledge, therefore, genetic programming has been introduced into the field of data fitting.

In order to improve the computing efficiency and avoid premature convergence and poor population diversity of traditional genetic programming algorithms, practical application experiences shall be utilized and changes with respect to fitness function design, exchange of individual selection and mutation operation shall be made to obtain new algorithms and good results.

In the data analysis, the least square fitting method is the most commonly used method to obtain approximate functions. However, on the premise of existence of function structure, this method has to determine all the parameters of the function and the function structure should not be complicated. While when genetic programming conducts data fitting, the obtained parameters may not be the optimal. Therefore, on the basis of this issue, a method which combines genetic programming combined with the least squares method is proposed in this paper.

In the process of genetic programming, this method uses the least square method to optimize individual parameters and then puts them back into the population. Doing so breaks the ice that traditional data fitting methods can only use fixed-function model. The number of convergences and average convergence algebra of the new fitting function are significantly better than the traditional methods. The fitting function also has higher precision.

The advantage of this method is that it can fit the function structure through genetic programming based on experimental data points when the function model is unknown. Then new initial population can be generated by using the least squares method to optimize parameters, and then genetic programming is applied. In this way, the obtained function has better structure and its number of convergence and average iteration algebra are very good.

Of course, this method needs improvement, for example, its iterative algebra and the number of individuals in the initial population have a great impact on the computation speed.

References

Abraham A., Ramos V., 2003. Web usage mining using artificial ant colony clustering and linear genetic programming. In: Proceedings of the 2003 IEEE Congress on Evolutionary Computation, 2: 1384-1391. http://dx.doi.org/10.1109/CEC.2003.1299832

Aslam M.W., Zhu Z.C., Nandi A.K., 2012. Automatic Modulation Classification Using Combination of Genetic and KNN, IEEE. Trans. Wireless Communication, 11(8): 2741-2750.

Angeline P.J., 1997. Comparing subtree crossover with macromutation. Proc. of the Sixth Annual Conference on Evolutionary Programming. Berlin: Springer- Verlag. 101-111. http://dx.doi.org/10.1007/BFb0014804

Chen Z.W., Wang W.L., 2003. Research actuality and development of Genetic Programming. Journal of Zhejiang University of Technology, 31(2): 153-159.

Ebner M., 2007, Fast Genetic programming On GPUs. In: Euro GP 2007, LNCS 4445, 90-101.

Freitas A.A., 2003. A survey of evolutionary algorithms for data mining and knowledge discovery. Advances in evolutionary computing. Springer Berlin Heidelberg: 819-845.

Hirasawa K., Okubo M., Katagiri H., 2001. Comparison between genetic network programming (GNP) and genetic programming (GP). In: Proceedings of the 2001 IEEE Congress on Evolutionary Computation, 1276-1282. http://dx.doi.org/10.1109/CEC.2001.934337

Jin R., Kou C.H., Liu R.J., 2014. Biclustering algorithm of differential co-expression for gene data, Review of Computer Engineering Studies, 1(1): 7-12. http://dx.doi.org/10.18280/rces.010102

Koza J.R, 1992. Genetic Programming: On the Programming of Computers by means of Natural Selection. Cambridge, MA: MIT Press.

Koza J.R, 2000. Automatic Creation of Human-Competitive Programs and Controllers by Means of Genetic Programming. Genetic Programming and Evolvable Machines, 1: 121-164.

Miller J.F., Thomson P., 2000. Cartesian genetic programming. Genetic Programming. Springer Berlin Heidelberg, 121-132. http://dx.doi.org/10.1007/978-3-540-46239-2_9

Niu A.G., Wang H.W., 2007. An Improved Algorithm of Genetic Programming. Operations Research and Management Science, 16(1): 67-70.

Nordin P., 1999. Book review: Genetic programming Ⅲ-Darwinian Invention and Problem Solving. Evolutionary Computation, 7(4): 451-453. http://dx.doi.org/10.1162/evco.1999.7.4.451

Shao G.F., Zhou Q.F., Chen G.Q., 2009. Data fitting based on improved genetic programming. Application Research of Computers, 26(2): 481-484.

Sun Y.G., Qiang H.Y., Yang K.R., Chen Q.L., Dai G.W., Dong M., 2014. Experimental Design and Development of Heave Compensation System for Marine Crane. Mathematical Modelling and Engineering Problems. 1(2): 15-21. http://dx.doi.org/10.18280/mmep.010204

Zhang M., Zhou Y.Q., Wang D.D., 2006. A data fitting method based on genetic programming. Journal of Harbin Engineering University, 27: 527-530.

Zhang Y, Zhang M, 2004. A multiple-output program tree structure in genetic programming. In: Proceedings of the 7th Asia-Pacific Conference on Complex Systems.

Zhou Q., 2015. Effective Kidney MPI Segmentation Method Based on Level Set with Prior Shape. Review of Computer Engineering Studies, 2(1): 43-46 http://dx.doi.org/10.18280/rces.020108