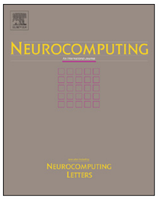
Neurocomputing 137 (2014) 293–301

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



An improved Gene Expression Programming approach for symbolic regression problems

YuZhong Peng, ChangAn Yuan*, Xiao Qin, JiangTao Huang, YaBing Shi

Key Lab of Scientific Computing & Intelligent Information Processing in Universities of Guangxi, Guangxi Teachers Education University, Nanning 530001, China

ARTICLE INFO

Article history:
Received 7 February 2013
Received in revised form
23 April 2013
Accepted 29 May 2013
Available online 18 February 2014

Keywords:
Genetic computing
Gene Expression Programming
Evolutionary algorithm
Symbolic regression
Data modeling

ABSTRACT

Gene Expression Programming (GEP) is a powerful evolutionary method for knowledge discovery and model learning. Based on the basic GEP algorithm, this paper proposes an improved algorithm named S_GEP, which is especially suitable for dealing with symbolic regression problems. The major advantages for this S_GEP method include: (1) A new method for evaluating individual without expression tree; (2) a corresponding expression tree construction schema for the new evaluating individual method if required by some special complex problems; and (3) a new approach for manipulating numeric constants so as to improve the convergence. A thorough comparative study between our proposed S_GEP method with the primitive GEP, as well as other methods are included in this paper. The comparative results show that the proposed S_GEP method can significantly improve the GEP performance. Several well-studied benchmark test cases and real-world test cases demonstrate the efficiency and capability of our proposed S_GEP for symbolic regression problems.

1. Introduction

Gene Expression Programming (GEP) is developed by a Portuguese scientist, named Ferreira, in 2001, which were derived and improved from Genetic Algorithm (GA) and Genetic Programming (GP) [1]. It is a new revolutionary member of the genetic computing family, benefiting from the genetic expression of the knowledge discovery technologies, owning to the merits of GP and GA, that evolves computer programs. In fact, they can take many forms such as mathematical expressions, neural networks, decision trees, polynomial constructs, logical expressions, and so on [1].

With simple, linear and compact chromosomes and easy genetic operators, GEP is a powerful global search tool. Since Ferreira released the first GEP research results, GEP has become an active research area of evolutionary computation and has been applied in many fields and well solved a large variety of complex problems, such as classification [2,3], symbolic regression and function mining [4–6], time-series analysis [7], optimization [8], and so on.

In recent years, numerous researchers have investigated GEP and proposed a series of improved GEP methods, processing data in specific fields with high effectiveness and efficiency. Li et al. [9], proposed a prefix K-expression structure to try to preserve good structures, which achieves a better convergence and efficiency in the classification; Duan et al. [10], posed a new dynamic adjustment of individual coding length through the ORF filter operator to reduce the situation of GEP lowering efficiency caused by an overly long string of individuals. With all the theories aforementioned, however, the primitive GEP should be applied to describe the genotype into the expression tree, and further to traverse the expression tree to calculate the fitness. As the tree construction and traverse were of very time-consuming operation, it greatly affected the efficiency of the algorithm; Elena et al. [11], presented an adaptive GEP algorithm, which automatically adapted the number of genes used by the chromosome. The adaptation process taken place at chromosome level, allowing chromosomes in the population to evolve with different number of genes to reduce the computational effort; Ryana and Hiblerb [12] presented a Robust Gene Expression Programming which used the simple grammar of prefix expressions and the simple encoding of bit vectors, reaping the benefits of encoding the expressive structures of trees and the power of breaking the “phenotype barrier”.

Symbolic regression, namely symbolic function identification, is a function discovery approach for analysis and modeling of numeric multivariate data sets for a purpose of getting insights about data-generating systems [14]. Symbolic regression has had both successful academic [15–18]and industrial applications [19,20].

Based on the basic GEP proposed by Ferreira, this paper describes the power S_GEP, which is specifically suitable for

* Corresponding author. Tel.: +86 077 1561 2556.
E-mail address: yca@gxtc.edu.cn (C. Yuan).

http://dx.doi.org/10.1016/j.neucom.2013.05.062
0925-2312 © 2014 Elsevier B.V. All rights reserved.

symbolic regression problems. The S_GEP has several other improvements containing: a new method for decoding and evaluating chromosome based on our previous research in literature [13], and a corresponding expression tree (ET) constructing and its traversing schema, a new approach for manipulating constants. The proposed new method for decoding and evaluating chromosome does not require constructing and traversing the ET but directly using stacks to decode chromosome and evaluate the fitness, to reduce the time–space complexity. The proposed new approach for manipulating numeric constants improves the convergence of population. Experimental results obviously indicate that S_GEP outperforms classical GEP with less computational effort and higher effectiveness.

2. Preliminary material

2.1. Brief overview of symbolic regression

In detail, the task of regression is to identify the variables (inputs) in the data that are related to the changes in the important control variables (outputs), to express these relationships in mathematical models, and to analyze the quality and generality of the constructed models [14]. Symbolic regression differs from traditional regression since it does not rely on a specific a priori determined model structure. The only assumption made in symbolic regression is that the response surface can be described by an algebraic expression. Instead of the traditional approach where the model structure is fixed and the remaining free parameters are optimized, symbolic regression reformulates the regression problem as a search problem for the optimal model structure. Once a model structure of sufficient quality is found, traditional techniques can be used to find the optimal coefficients [15].

2.2. Brief overview of GEP

As the case with GP and GA, when using GEP to solve a problem, the main algorithm description is similar with GP and GA. And generally the five components: function set, terminal set, fitness function, GEP control parameters, and stop condition, all need to be specified.

Chromosomes of GEP are composed of one or several genes by connecting with operators. The gene is made up of a head and a tail with the head encompassing functions and terminals, and the tail containing only terminals, of which function set is formed by all the function symbol needed while solving problems. The terminals set is made up of the known symbols, variable or constant describing the problems. And the head length h, tail length t and the largest number of function arguments n must meet the following relations:

$$t = h \times (n-1)+1 \tag{1}$$

where, h is determined by the users according to the problems to be solved. The relationship ensures that no matter how the head is composed, genes can be decoded with the effective semantic meaning and valid expression. First, GEP codes the individuals into the linear string of fixed length. It encodes the operands to form the genome while doing optimization. Its coding rules can be simply described as the following: the algorithm builds the expression tree (ET) with genome according to their semantic, and traverses the ET from top to bottom, then from left to right, and the derived symbol serials are effective parts of the gene code (known as K-expression). This process is called decoding chromosomes. Some tail nodes may not be seen in the expression tree. Such redundant nodes accommodate the future operations of the genetic changes in the structure which may have left space in terms of the expression imposed on the string of genetic operations; we will resolve the semantic effective expression tree. All the GEP individuals are effective candidates of solution [1]. After the results of each generation evolution undergo the moderate evaluation of fitness function, high fitness individuals are retained, and have a higher chance to breed future generations by moving in cycles. The algorithm will not stop until a satisfactory solution or expected evolution algebra have been achieved.

The group in the evolutionary process needs to evaluate the individual with fitness. Generally, the traditional method of fitness calculation needs to transfer the representative individuals of chromosomes or genomes into the expression tree, and then traverse the expression tree and derive the corresponding mathematical expression to compute the value of the K-expression, finally, evaluate the fitness according to the fitness function. In each generation evolution, high fitness individuals are preserved with more chances to breed future generations, and repeatedly in this method. The algorithm will not stop until a satisfactory solution or expected evolution generations have been derived.

2.3. Constants

It is assumed that the creation of floating-point constants is necessary to do symbolic regression in general [16,18]. Ferreira introduced two approaches for symbolic regression in the original GEP [17]. The first algorithm does not include any constants in the terminal set, but relies on the spontaneous emergence of necessary constants through the evolutionary process of GEP. Whereas the second algorithm involves the ability to explicitly manipulate random constants by adding a random constant domain Dc at the end of the gene. Experiments have shown that the first algorithm is more efficient in terms of both accuracy of the evolved models and computational time for solving problems.

Because in many experiments, the explicit use of random constants results in considerably worse performance. Ferreira gives advice that GEP find or compose the most suitable combination of constants itself [17]. And some research presents some GEP-RNC variant method to fine-tuned constants [4,17,18,21]. But using them, many extra computational resources are required.

2.4. Traditional method of decoding the GEP

In GEP, the group in the evolutionary process needs to calculate the individual fitness to evaluate. But the traditional method of calculate the individual fitness value needs to transform the genotype of chromosome into expression tree, and then traverses expression tree and attains the corresponding mathematical calculation. Finally, we need to compute the fitness according to the fitness function. According to the basic principles of GEP, the usual practice is to read from left to right one by one of the corresponding genes in the chromosome of the characters with the first character as the tree root, and the remaining follow-up node structure of the tree as a branch added to the parent node. The level of the corresponding ET will be built with the top-down, from left to right. Then according to their semantic level traverse the nodes in ET, we will gain the valid string of gene encoding, then calculate the value of K-expression in the first order traverse of ET (this is the value of the gene, if this chromosome is one with a single gene, the value of the chromosome is the value of the gene; while the chromosome is the one with couple genes, the value of the chromosome gene is the value of each gene calculated according to certain rules), and finally in accordance with the established fitness function to evaluate fitness value of the chromosome. For example, the following is a GEP chromosome