Advances in Engineering Software 50 (2012) 82–96

Contents lists available at SciVerse ScienceDirect

Advances in Engineering Software

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

Optimization of existing equations using a new Genetic Programming algorithm: Application to the shear strength of reinforced concrete beams

Juan L. Pérez [a], Antoni Cladera [b,*], Juan R. Rabuñal [c], Fernando Martínez-Abella [d]

[a] School of Building Engineering and Technical Architecture, University of A Coruña, Spain
[b] Department of Physics, University of the Balearic Islands, Spain
[c] Department of Information and Communication Technologies, University of A Coruña, Spain
[d] Department of Construction Technology, University of A Coruña, Spain

ARTICLE INFO

Article history:
Available online 14 March 2012

Keywords:
Artificial intelligence
Genetic Programming
Structural engineering
Concrete
Shear strength
Regression analysis

ABSTRACT

A method based on Genetic Programming (GP) to improve previously known empirical equations is presented. From a set of experimental data, the GP may improve the adjustment of such formulas through the symbolic regression technique. Through a set of restrictions, and the indication of the terms of the expression to be improved, GP creates new individuals. The methodology allows us to study the need of including new variables in the expression. The proposed method is applied to the shear strength of concrete beams. The results show a marked improvement using this methodology in relation to the classic GP and international code procedures.

1. Introduction

On certain occasions there are contrasted theoretical formulations that allow finding a solution to a particular engineering problem, but there is not often a proven theoretical solution and it is necessary to resort to empirical formulations that are inferred from experimental results. The evolutionary computation is a tool that is capable to solve on its own and from experimental results, numerous problems in different fields as, for example, in Civil Engineering [1]. In this study field it appears different interests where artificial intelligence techniques can help to the science enrichment. In most of the problems a physical phenomenon is abstracted in a mathematical problem to simulate and predict such phenomenon. Since in most of the case study there has already been some available knowledge about a particular phenomenon, that is, there have already been different models that try to adjust the physical/chemical behavior through equations, the use of artificial intelligence techniques is of great interest for the optimization or improvement, if anything, of such models.

In scientific literature there are numerous approximations for the optimization of several processes. If we concentrate on the example field (structural engineering), most of the optimization processes are focused on the resource optimization, that is, on the execution of a specific element with the minimum of resources that are used but always guarantying the element security. An example is the job made by Perera and Vique [2]. In this paper the authors use the genetic algorithms for automatically producing optimal strut-and-tie models for the design of reinforced concrete beams. For this, they look for minimizing the axial force product, the length and axial strain of the truss elements.

Another example to quote is the one developed by Sonebi and Cevik [3]. In this case the authors use the Genetic Programming technique to find an equation for modelling the fresh properties and the compressive strength of self-compacting concrete (SCC) containing pulverized fuel ash (PFA), highlighting the obtaining of good results in spite of the fact that there are available few data.

As well as the evolutionary computation techniques, the artificial neural networks (ANNs) can be used to improve the physical model involved in a process. In this aspect, it is important to point out the job of Cladera and Marí [4], who uses the ANN for the analysis of the shear strength in concrete beams without shear reinforcement. In this case, and afterwards the training and verification process, the ANNs were used as a virtual laboratory, predicting test values that were not made physically. With the one that was developed, they get to study the dependence type facing each of the variables, finally formulating two design expressions that improve noticeably any of the ones developed by other authors or by other national or international codes. The main inconvenient in the use of ANN is the impossibility to give expression explicitly to the result, that is, the result that was obtained through the learning is a data recorder which only gives results according to the input stimulus, without relating explicitly the input values to the output values at no time. On the other hand, it is impossible to apply restrictions as the ones presented in the article through the use of ANN.

* Corresponding author.
E-mail address: antoni.cladera@uib.es (A. Cladera).

0965-9978/$ - see front matter © 2012 Civil-Comp Ltd and Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2012.02.008

Nearer to our case study, it is found the job made by Ashour et al. [5]. In this case they obtain an expression through GP that, from the previously standardized variables, is capable of predicting the shear strength in concrete beams. This example differ mainly from the one presented here in two questions. In the first place, the variables have been standardized. In the second place, the search process is not directed anyway. Although it is obtained better results with a priori standardized data, it would entail not being able to apply the resultant formula immediately since it would be necessary to apply the standardization to the data. In any case, they get good adjustments from a database of only 141 beams tests indexed to scientific literature, although they do not compare them to the current codes of practice in spite of mentioning them.

Regarding the tendencies in the field of Genetic Programming, related to the orientation of the search process, they are synthesized in syntactic restrictions. For example, Koza uses this type of restrictions when generating new individuals [6]. There is a mechanism developed likewise by Koza [7], called "Automatically Defined Functions (ADFs)", that it could be explained as a particular case of syntactic restrictions, since the ADF are functions or subroutines that are "reusable" by the Genetic Programming algorithm of a fixed structure that can evolve. Another type of restrictions would be the ones that involve the type of data, or the dimensional coherence of a result. In this case, Montana [8] proposes a "Strongly Type Genetic Programming method (STGP)" with it is achieved, for example, that the operator "sine" is only applied to variables that contain angles. Finally, there are the techniques based on "Grammar Guided Genetic Programming (GGGP)", in which the genetic operations are conditioned by grammar that is defined by the user. In this grammar, called "Context Free Grammar (CFG)", it lies the expert knowledge in the study area. For example, García-Arnau et al. [9] develops a method called "Grammar Based Initialization Method (GBIM)" that he uses with GGGP for classification tasks in Breast Cancer. More related to the case study of this job, Ralte and Sebag [10] use GGGP to create a behavior model of a material from experimental data.

Pérez et al. [11] have presented an algorithm that allows to improve a mathematical expression that is controlled by an expert on the basis of experimental data, leading the search process through restrictions given by the expert in the creation of new solutions. In the current article it is carefully presented the followed methodology, and it is compared to the results that would be obtained with classic techniques of GP. Besides, it is proposed a methodology to study the necessity or not to include certain variables that were not considered in the initially chosen formulation to be optimized. As an example, and as an illustration of its functioning, it has been chosen a problem that is enshrined within the structural engineering: the shear strength phenomenon in concrete beams. Besides, in the article it is presented how the consideration of a variable that was not initially included, the relation among the shear force and the concomitant bending moment allows to establish shear-moment interaction diagrams through two simple expressions, obtaining results that have a lot in common to the ones given by one of the most developed and complex theoretical models, the Modified Compression Field Theory [12].

2. Genetic Programming

Genetic Programming is a subset of solution search techniques enshrined within the term of evolutionary computation (EC). EC includes a set of methods based on models that emulate certain characteristics of nature, mainly the capacity that living beings possess to adapt themselves to their environment. This feature of living beings had been captured by Charles Darwin to make his theory of evolution according to the species natural selection principle [13]. Darwin holds that those individuals in a population who possess the most advantageous characters will leave proportionally more descendants in the following generation, and if such characters are due to genetic differences that can be transmitted to the descendants, the genetic composition of the population will tend to change, raising the number of individuals with such characteristics. In this way, the complete population of living beings adapt themselves to the changeable circumstances of their environment. The final result is that living beings tend to perfect themselves in relation to the circumstances that surround them.

John Holland was the first to develop this type of techniques that, in a first moment, he called them *reproductive plans*, but he became popular under the name of genetic algorithm (GA) after the publication of his book "Adaptation in Natural and Artificial Systems" in 1975 [14]. Nowadays the GA is being used mainly to develop solutions to parameterized problems (optimization problems). But it was John Koza who laid the foundations in 1992 of what has been known from that moment onwards as Genetic Programming [6]. The GP arises as an evolution of the traditional GA, keeping the same principle of natural selection. With this technique the aim is to provide solutions to problems through the program induction and the algorithms that solve them. They are used in several science fields such as electronic circuit design, pattern recognition, and symbolic regression.

In GP, an analogy between the set of solutions to a problem and the set of individuals in a natural population is established, codifying the information of each solution through a tree-shaped structure. In this codification two types of nodes are differentiated. The first type is the *non-terminal* nodes or *functions* where the operators of the algorithm that is wanted to develop are lodged (for example addition, subtraction, etc.). They are characterized because they always have one or more children. The second type is the *terminal* nodes or tree leaves, where the constant values and the previously defined variables are located. These nodes have not got children. For example, Fig. 1 represents a possible solution to a problem where it is desired to relate the input variables (a, b) to the output ones $f(a, b)$ through the expression $f(a, b) = a^*((b/4) + 3)$. In this example, the non-terminal nodes or functions would correspond to the product, the addition and the division, whereas the terminal nodes would be the values 3 and 4, together with the variables a and b. Therefore, a fundamental part of the GP configuration for its execution is the specification of the terminal and non-terminal element set before the beginning of the evolutionary process, since the algorithm will build the trees with the nodes that are specified to it.

Since in the execution of GP it will be created a great deal of trees in which it will not be controlled the node disposition, it is possible that operations that are not valid are generated, for example, that a value is divided into 0. In general terms, it is widened the dominion of application in each operator to avoid possible errors in the application of the operators. This new operator is called protected operation. For example, the natural logarithm dominion of application is the set of positive real numbers including zero. In this case, it will be necessary to widen the dominion of application

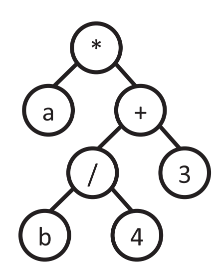

Fig. 1. Tree for the expression $a^*((b/4) + 3)$.