Applied Soft Computing 19 (2014) 112–120

Contents lists available at ScienceDirect

Applied Soft Computing

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

Linear genetic programming for shear strength prediction of reinforced concrete beams without stirrups

Amir H. Gandomi[a,*,1], Danial Mohammadzadeh S.[b], Juan Luis Pérez-Ordóñez[c], Amir H. Alavi[d,2]

[a] Department of Civil Engineering, The University of Akron, Akron, OH, USA
[b] Civil Engineering Department, Ferdowsi University of Mashhad, Mashhad, Iran
[c] School of Building Engineering and Technical Architecture, University of A Coruña, Spain
[d] Department of Civil and Environmental Engineering, Engineering Building, Michigan State University, East Lansing, MI 48824, USA

ARTICLE INFO

Article history:
Received 25 January 2013
Received in revised form 9 December 2013
Accepted 6 February 2014
Available online 15 February 2014

Keywords:
Shear strength
Reinforced concrete beam
Linear genetic programming
Design equation

ABSTRACT

A new design equation is proposed for the prediction of shear strength of reinforced concrete (RC) beams without stirrups using an innovative linear genetic programming methodology. The shear strength was formulated in terms of several effective parameters such as shear span to depth ratio, concrete cylinder strength at date of testing, amount of longitudinal reinforcement, lever arm, and maximum specified size of coarse aggregate. A comprehensive database containing 1938 experimental test results for the RC beams was gathered from the literature to develop the model. The performance and validity of the model were further tested using several criteria. An efficient strategy was considered to guarantee the generalization of the proposed design equation. For more verification, sensitivity and parametric analysis were conducted. The results indicate that the derived model is an effective tool for the estimation of the shear capacity of members without stirrups ($R = 0.921$). The prediction performance of the proposed model was found to be better than that of several existing buildings codes.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Although several research programs have been conducted to predict the shear capacity of concrete (e.g. [1]), there is still no clear expression to predict the shear failure mechanisms of concrete elements. Most of the available shear design expressions have different forms and do not provide a consistent factor of safety against shear failure. Thus, the behavior of concrete beam has been extensively investigated during the last three decades. Numerous theoretical models have been established in recent years to investigate the interaction between several forces including axial, shear, bending, and torsion [2,3].

Recently, application of machine learning has attracted much attention for solving structural engineering problems. The machine learning systems are powerful tools for design of computer programs. They automatically learn from experience and extract various discriminators [4]. Artificial neural networks (ANNs) are the most widely used branch of machine learning. There have been some researches focusing on the application of ANNs to the evaluation of the shear strength of reinforced concrete beams without reinforcement. Recently, Choi et al. [5] used another machine learning method, namely fuzzy logic (FL) for the modeling of the shear strength of slender reinforced concrete beams. Although ANNs and FL are successful in prediction, they are not usually able to produce practical prediction equations. Furthermore, for the ANN-based modeling, the structure of the network should be identified a priori. Besides, determination of the fuzzy rules in FL is a difficult task. These methods are mostly appropriate to be used as a part of a computer program [6]. Genetic programming (GP) [7] is a developing subarea of the machine learning techniques. GP is known as an extension genetic algorithm (GA) where the solutions are computer programs rather than fixed length binary strings [6]. Classical (standard) GP and its variants have been recently employed to derive greatly simplified formulations for structural engineering problems and especially concrete structures modeling (e.g. [8–10]). Linear genetic programming (LGP) [11] is a new subset of GP. LGP operates on computer programs that are represented as linear sequences of instructions of an imperative programming language

* Corresponding author. Tel.: +1 234 788 0619; fax: +1 330 972 2060.
E-mail addresses: a.h.gandomi@gmail.com, ag72@uakron.edu (A.H. Gandomi), jlperez@udc.es (J.L. Pérez-Ordóñez), ah_alavi@hotmail.com, alavi@msu.edu (A.H. Alavi).
[1] http://gozips.uakron.edu/~ag72/.
[2] http://www.egr.msu.edu/~alavi/.

http://dx.doi.org/10.1016/j.asoc.2014.02.007
1568-4946/© 2014 Elsevier B.V. All rights reserved.

[6]. In contrast with ANNs, GA and classical GP, application of LGP in the field of civil engineering is quite new and restricted to a few areas [6,12–16]. It is worth mentioning that classical GP and new variants of GP have been, respectively, used by Ashour et al. [17] and Gandomi et al. [18,19] to predict the load capacity of RC deep beams. Gandomi et al. [20] have applied LGP to the modeling of fibrous RC beams. Recently, Kara [21] employed GP for the prediction of the shear strength of FRP-reinforced concrete beams without stirrups upon a limited number of experimental test results. Moreover, Pérez et al. [22] applied GP to the optimal adjustment of EC-2 shear formulation for RC beams without web reinforcement.

However, applications of the GP-based approaches to directly obtain a simple formula to predict the shear strength of RC beams are conspicuous by its absence. There are approaches which present simple formulation, but based on other advanced approaches [23]. The main purpose of this study is to utilize the LGP technique to build a simple predictive model for the shear strength of RC beams without stirrups. The shear strength was formulated in terms of shear span to depth ratio, concrete cylinder strength at date of testing, amount of longitudinal reinforcement, lever arm and maximum specified size of coarse aggregate. The predictions made by models were further compared with those provided by several well-known building codes.

2. Machine learning

Machine learning is a branch of artificial intelligence essentially inspired from biological learning. The machine learning approach deals with the design of computer models that are able to automatically learn with experience [4,24]. The machine learning methods extract knowledge and complex patterns from machine readable data [4]. The major focus of the machine learning research is on data mining problems, difficult-to-program applications, and software applications customizing to the individual user's preferences [4,25].

2.1. Linear genetic programming

GP is a machine learning technique, inspired by biological evolution, to find computer programs that solve a problem. It uses the principle of Darwinian natural selection to evolve a program [26]. GP is a specialization of GA. The main difference between GP and GAs is the representation of the solution. GA creates a string of numbers that represent the solution. The GP solutions are computer programs [6]. GP works with population of individuals (computer programs) that are created randomly. The classical GP technique is also referred to as standard tree-based GP [6]. A population member in standard GP is a hierarchically structured tree comprising functions and terminals. In addition to traditional tree-based GP, there are other types of GP where programs are represented in different ways.

LGP is a linear variant of GP. LGP is a subset of GP with a linear representation of individuals. The main characteristic of LGP in comparison with traditional tree-based GP is that expressions of a functional programming language (like LISP) are substituted by programs of an imperative language (like C/C++) [6,27]. Fig. 1 presents a comparison of the program structures in LGP and tree-based GP. A linear genetic program can be seen as a data flow graph generated by multiple usage of register content. That is, on the functional level the evolved imperative structure denotes a special directed graph. In the tree-based GP, the data flow is more rigidly determined by the tree structure of the program [6,27].

In the LGP system described here, an individual program is interpreted as a variable-length sequence of simple C instructions. The instruction set or function set of LGP consists of arithmetic operations, conditional branches, and function calls. The terminal set of

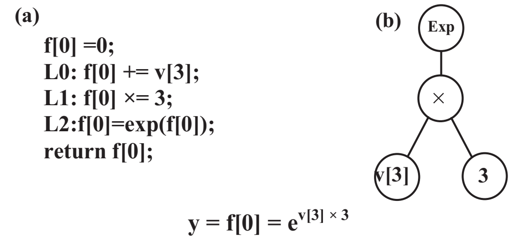

Fig. 1. Comparison of the GP program structures. (a) LGP and (b) tree-based GP.

the system is composed of variables and constants [6]. The instructions are restricted to operations that accept a minimum number of constants or memory variables, called registers, and assign the result to a destination register.

Automatic Induction of Machine code by Genetic Programming (AIMGP) is a particular form of LGP. The programs evolved by AIMGP are stored as linear strings of native binary machine code and directly executed by the processor during fitness calculation. The absence of an interpreter and complex memory handling results in a significant speedup in the AIMGP execution compared to tree-based GP [6,11,27]. This machine-code-based LGP approach searches for the computer program and the constants at the same time. Here are the steps the machine-code-based LGP follows for a single run [11]:

I. Initializing a population of randomly generated individuals (programs) and calculating their fitness values.
II. Running a tournament. In this step four individuals are selected from the population randomly. They are compared and based on their fitness, two individuals are picked as the best adapted (less fitness value) and two as the worst adapted (more fitness value).
III. Transforming the best adapted individuals (winner programs). After that, best adapted individuals are copied and transformed probabilistically as follows:
   - Parts of the best adapted individuals are exchanged with each other to create two new individuals (crossover); and/or
   - Each of the tournament best adapted individuals is transformed randomly to create two new individuals (mutation).
IV. Replacing the worst adapted individuals (loser programs) in the tournament with the transformed adapted individuals. Elitist strategy is used, i.e., the best adapted of the tournament remain without change [6].
V. Repeating steps two through four until termination or convergence conditions are satisfied.

Crossover occurs between instruction blocks. Fig. 2 demonstrates a two-point linear crossover used in LGP for recombining two tournament winners [20]. As it is seen, a segment of random position and arbitrary length is selected in each of the two parents and exchanged. If one of the two children would exceed the

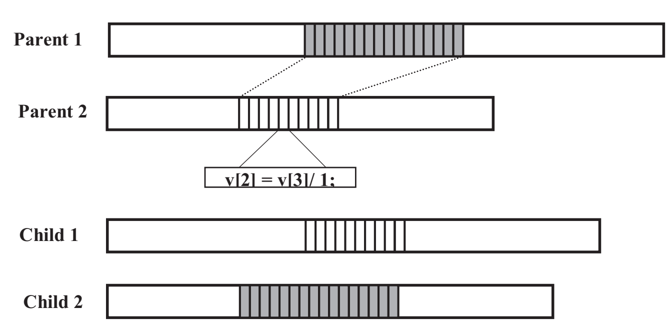

Fig. 2. Crossover in LGP.

Download English Version:

https://daneshyari.com/en/article/495556

Download Persian Version:

https://daneshyari.com/article/495556

Daneshyari.com