Advances in Engineering Software. 2012; 50: 82-96

# 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*

**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.

**Keywords**

Artificial intelligence; Genetic Programming; Structural engineering; Concrete; Shear strength; Regression analysis

## 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.

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.

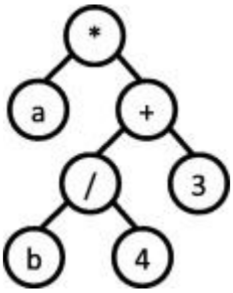
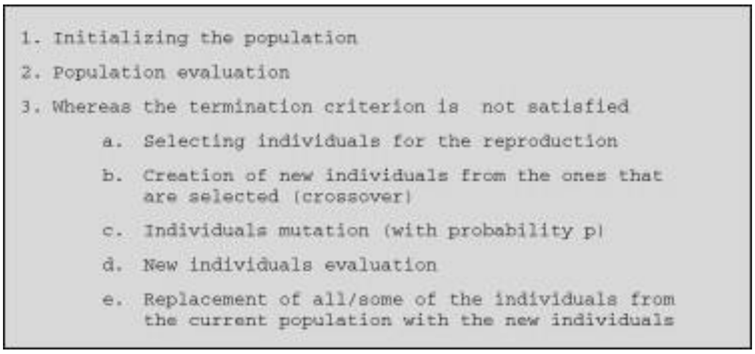


**Fig. 1**. Tree for the expression $a^{*}((b/4) + 3))$.

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 for all the negative numbers. For example, the protected natural logarithm operation could be implemented as the natural logarithm of the argument absolute value.

Each of these trees will be a possible solution to the problem in question. The fitness function is used to evaluate its goodness. In GP, each solution is called *individual*, and the set of individuals with whom it works is called *population*. This population, who is initially random, is made to evolve through a number of iterations that are called *generations* in which new individuals who will be part of the current population are created from the individuals of a previous generation. These new individuals are created combining the genetic material of some selected individuals, using the selection, crossover and mutation algorithms. In Fig. 2 it is described the GP general functioning. This outline is the same for any evolutionary algorithm.



```
1. Initializing the population
2. Population evaluation
3. Whereas the termination criterion is  not satisfied
        a.  Selecting individuals for the reproduction
        b.  Creation of new individuals from the ones that
            are selected (crossover)
        c.  Individuals mutation (with probability p)
        d.  New individuals evaluation
        e.  Replacement of all/some of the individuals from
            the current population with the new individuals
```

**Fig. 2**. Outline of an evolutionary algorithm.

## 3. Genetic Programming to improve well-known equations

The use of GP to develop mathematical expressions is probably its most extended application [15]. Its way of codifying allows these to be represented in an easy way, and they have been applied in a great deal of different fields related to science or engineering [1], [16], [17]. The results that were found have been very beneficial and the expressions that were achieved have improved in a great number of occasions compared to the previous ones existent in this field.

However, GP should not be useful only as a tool for the development of new equations in the development of mathematical expressions, but also it would be desirable to use GP to optimize one or several parts of a previously known equation. In previous works [18] GP is used to develop distinctive parts in a mathematical expression. However, in such study it is not started from a known equation, but different parts of an equation are made to evolve, which will eventually be combined linearly to give way to a more complex expression.

The developed algorithm that is detailed in this article is based on classic Genetic Programming techniques, but it has been specialized in the optimization of mathematical formulas. Thereby, algorithm development has been focused on the symbolic regression technique from learning patterns. Besides, a module has been added that makes it possible to incorporate expert knowledge to orientate the search process, by inserting different restrictions when creating, mutating and crossing individuals.

The algorithm that is developed allows the optimization of any mathematical expression at the points the expert selects. Given an equation *f* where *n* variables take part, it is expected to be optimized through a set of experimental data. This set of data can contain values for more variables than the ones taking part in the function. For instance, in Eq. (1) variables *a*, *b* and *c* take part, however in the set of experimental data, there are values for variables *a*–*e*. This means that it is possible that, apart from the variables *a*, *b* and *c*, which are in the function, variables *d* and *e* are also used in the optimization. The expert selects which points of the equation are desirable to optimize in order to orientate the solution, in this case four points (values in brackets of the equation: 2, 1/2, *a* + *c* and 5).

$$\mathbf{f}(\mathbf{a},\mathbf{b},\mathbf{c}) = [2]\left(\frac{1}{a}\right)^{[1/2]}\left(\frac{b}{[\mathbf{a}+\mathbf{c}]}\cdot c\right)^{[5]} \tag{1}$$

It is necessary to use a codification that allows us to develop the four branches that are desired to optimize for the optimization of this expression with GP. The solution to this problem will depend on a tree which, in this case, will contain four branches (see Fig. 3). Each branch represents the part of the formula that will be substituted in the original equation.

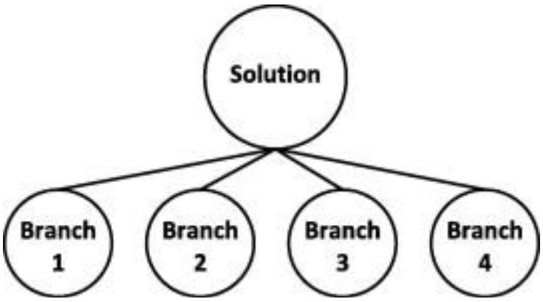


**Fig. 3**. Example of a solution.

It has been necessary to define what the solutions will be like in order to adapt the GP algorithm to the proposed problem. This means that it is necessary to give the genotype definition of the solution. To achieve this, it is started from the target function to be optimized. In Eq. (2) the resultant genotype of Eq. (1) is shown and the application way in the evaluation function. It is represented as a "*Branch*$_j$". Each of the solution branches that were obtained and the set of variables (*a*, *b*, *c*) are obtained from the data set.

$$Branch_1 \left(\frac{1}{a}\right)^{Branch_2} \left(\frac{b}{Branch_3} \cdot c\right)^{Branch_4} \tag{2}$$

As examples of restrictions, it could be mandatory that the use of a certain variable was not allowed to be used in a branch, or that only a subset of functions can be used (addition and subtraction). In this case, when creating, mutating or crossing an individual, if this does not satisfy the restrictions given by the expert; this individual would be "debugged" until all the restrictions are fulfilled. For example, in a branch where a variable cannot appear because of the given restriction, this is substituted by a constant value, in case a restricted variable appear as a result of a crossover or mutation operation. In Fig. 4 an example of a valid individual that optimizes Eq. (2) is shown. This individual satisfies the restrictions that have been imposed in Fig. 5 (Constrain Module).

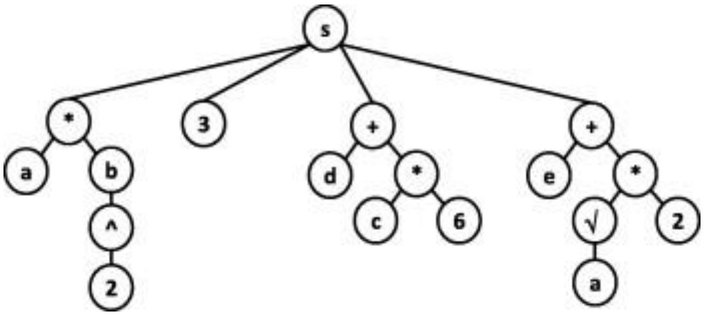


**Fig. 4**. Example of a result.

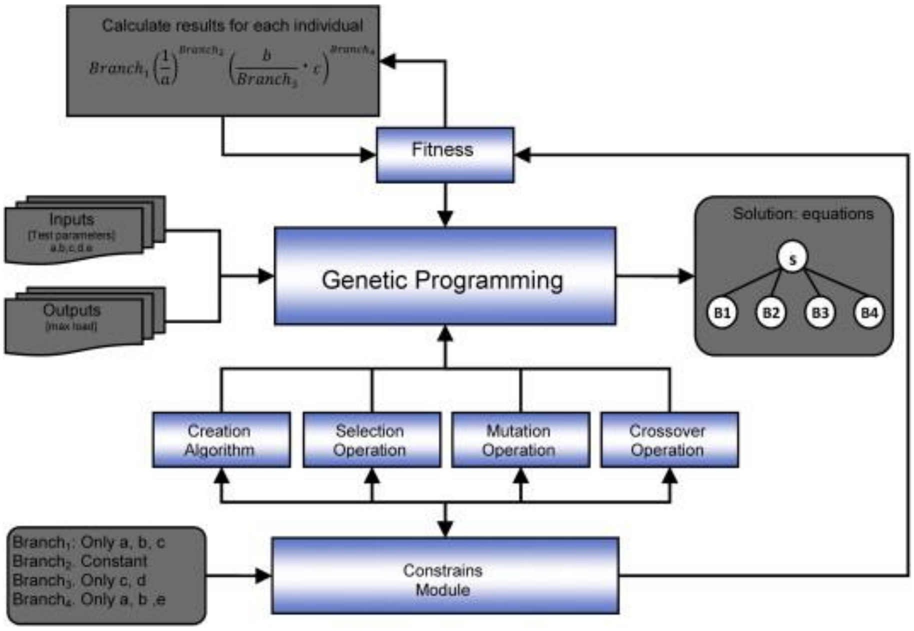


**Fig. 5**. Algorithm diagram.

Fig. 5 shows a diagram of the implemented algorithm. As it can be seen, the algorithm needs the evaluation function, the restrictions in each branch of the individuals, the input data and the typical configuration of the Genetic Programming to be defined (definition of terminal operators, functions, constants, crossover and mutation rates, size of initial population, maximum height, parsimony, etc.) For this, the expert will define the points where it is desired to optimize the equation, establishing, in this way, the number of children the solution will have. It is also necessary to establish the restrictions of each child (which variables are valid, if it must be one constant, etc.).

### *3.1. Fitness function*

The experimental results of shear failures of concrete beams without shear reinforcement show that, due to the great number of parameters taking part in the physical phenomenon and to concrete heterogeneity, there is certain variability. For this reason, the predictions that are made for this type of failure must take into account this variability and, despite this, they must obtain a safe result for most of the cases that are analyzed. In fact, the formulations designed to be used by engineers in national or international concrete codes must give priority to safety over precision, that is why oversizing can be suitable.

Mechanisms that are configurable by the user have been established to be able to ensure the oversizing of the prediction obtained by the algorithm. The first one consists of locating the ratio target value, $V_{test}/V_{pred}$, equal to 1.1, against the value 1.0 that would be the one of the "exact" prediction. In this way, the model will be located slightly on the safety side (the value obtained will be lower than the value of the test result) obtaining a conservative prediction. The way of materializing it in Genetic Programming is through the fitness function. Eq. (3) shows how this has been defined. The value $l_{bias}$ corresponds to the bias, considered generally equal to 1.1 in this paper, as it has been previously pointed out. Besides, a penalization is added to prevent from the excessive size of the individuals, with $\alpha$ being the parsimony level and $s_i$ the individual size (number of nodes). Higher values of Eq. (3) represent a worst solution. With the parsimony factor ($\alpha \cdot s_i$), the algorithm may eliminate those expressions that have a similar adjustment to others but that they use a higher number of nodes.

$$fitness(i) = \frac{\sum_{i=0}^{n} \quad k_i \cdot \left| l_{bias} - \frac{V_{test,i}}{V_{pred,i}} \right|}{n} + \alpha \cdot s_i \tag{3}$$

In this type of models it is very important that most of the predictions are within a range, or several ranges of application, and penalize in a different way the errors that are very near of the target value against the ones that are very far from the target. For this reason, some intervals, which are configurable by the user, are defined and they adjust the error that is made regarding the obtained value. In the fitness function it is expressed as the factor $k$ that will adjust the error made in the prediction.

In Eq. (4) an example of the different values $k$ for each error interval is shown. As well as the number of intervals, the penalization value is configurable by the user and dependant on the problem to be optimized. In the example given in Eq. (4), the algorithm would penalize much more the predictions in the range $V_{test}/V_{pred}$ lower than 0.5 (non-conservative results) than the predictions in the range $V_{test}/V_{pred}$ greater than 1.5 (conservative results).

$$k = \begin{cases} 10, & \frac{V_{test}}{V_{pred}} < 0.5 \\ 1, & 0.5 \leqslant \frac{V_{test}}{V_{pred}} < 1.5 \\ 3, & \frac{V_{test}}{V_{pred}} \geqslant 1.5 \end{cases} \tag{4}$$

### *3.2. Methodology*

In Fig. 6 the process that was carried out to obtain the formulas is shown briefly.

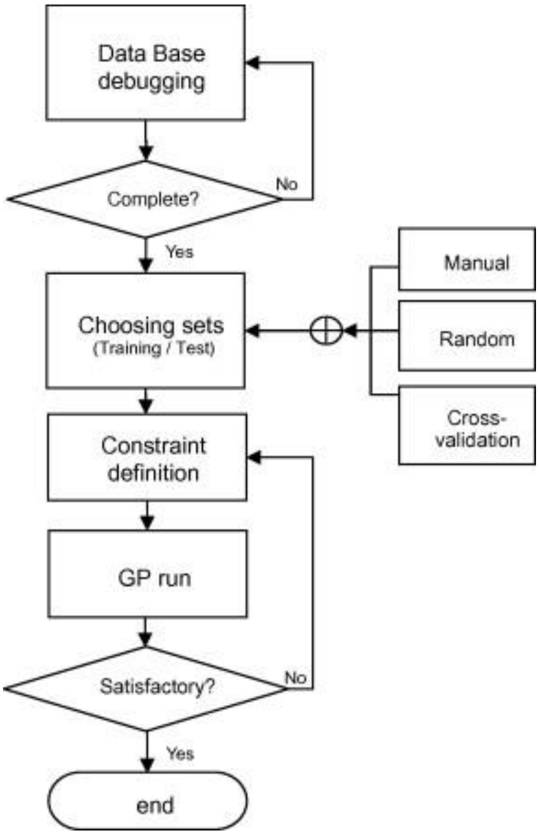


**Fig. 6**. Methodology used.

The first step, which is the database debugging, may seem trivial at first sight but it is essential to obtain coherent, unbiased results. Having a data base that is well verified and is as complete as possible, will be a solid base for the process. The data base must be representative enough in the data intervals to be studied. Should this condition not be fulfilled, a refinement must be made until a set that is representative enough is achieved.

Once a good set of data has been obtained, this set is divided into two subsets to see whether the results generalize correctly: one for training and the other for test. It is possible to opt for several methods that are subdivided regarding as to whether they are random or not. The most usual one would be to save a fixed percentage of patterns randomly to make the test. This method has the main drawback that the resultant sets can be very different (variable range, etc.). To avoid this, it is usual to resort to the cross-validation method of "*n*" levels that carry out the previous process "*n*" times, thus obtaining "*n*" pairs (training-test), and the result of the adjustment would be the mean of all [19], [20]. Finally, it can be decided to carry out the division process by hand, in this case it is the expert who is in charge of making the two sets. This last option is more complex but it guarantees that the sets are more representative.

Once the training/test sets have been established, they are placed at the disposal of establishing the formula or set of formulas that form the “frame” to make the tests. In each of these formulas it is necessary to determine the GP basic configuration. In this case, it is recommended to carry out different tests while varying only one parameter of the basic configuration (for example, % crossover, % mutations, selection method, etc.). Alongside it is necessary to determine the departure equation and which points of the equation are improvable. In this case the option to make several hypotheses may be given.

The definition of the starting equation (frame equation) is essential in guiding the search process of the new expression. From this moment onwards it begins an iterative process that is supervised by the expert and that consists of the definition of the applying restrictions, execution and analysis of results. This process will end when the expert’s demands are satisfied. However, the term “good solution” is not only based on a good adjustment of the formula that is found, but rather there is a “subjective” aspect that makes it possible for the solution which a priori has a better global adjustment, to be ruled out. For example, an expert can prefer a simple expression (with few terms) to a complicated one, but with a better adjustment. It is also possible that the adjustment of an expression in overall terms to be better but, when analysing different key subsets for the resolution of the problem in detail, it is proved that the expression offers unsatisfactory results for a certain subset; therefore the whole expression could be invalidated.

## 4. Case study

### *4.1. Problem description*

With the aim of proving the good functioning of the algorithm that has been developed in a real case, it has been used a problem that is enshrined within structural engineering: shear strength in concrete beams. This problem is one of the most controversial aspects linked to ultimate limit states in structural engineering, since the great complexity of the theoretical models makes it necessary to simplify in order to obtain standardized simple expressions. In fact, nowadays the current codes of practice propose very different formulations, most of them of empirical origin.

The phenomenon to be studied focuses on the shear strength mechanisms for beams without shear reinforcement. A report by the ASCE-ACI Committee 426 [21] in 1973 identified the following four shear strength mechanisms: shear stresses in uncracked concrete, shear transferred on the crack surface, known as aggregate interlock or shear friction, the dowel effect of longitudinal reinforcement, and arch action. In 1998, the report by the ASCE-ACI Committee 445 presented a new mechanism called residual tensile stresses, which are transmitted directly through the cracks. There are different opinions concerning the relative importance of each mechanism, leading to different models.

The international Eurocode 2 [22] was chosen to make the optimization of the shear model. Besides, the results obtained were compared to the ones given by the expressions 11-3 and 11-5 of the ACI Code 318-05 [23]. In Eq. (5) the formulation given in Eurocode 2 is shown, where a minimum for the value $V_{rd,c}$ is marked (Eq. (6)). Table 1 shows the variables used in the above equations.

$$V_{Rd,c} = 0.18 \cdot k \cdot (100 \cdot \rho_l \cdot f_c)^{1/3} \cdot b_w \cdot d \geqslant V_{Rd,Cmin} \tag{5}$$

$$V_{R,d,Cmin} = 0.035 \cdot k^{3/2} \cdot f_c^{1/2} \cdot b_w \cdot d \tag{6}$$

**Table 1**. Variables used in the Eurocode 2 formulation.

| Variables | Description | |
|---|---|---|
| $k$ | $k = 1 + \sqrt{\frac{200}{d}} \leqslant 200$ | |
| $\rho_l$ | Geometric amount of the longitudinal tensile reinforcement, anchored to a distance equal to or greater than d from the study selection | $\rho_l = \frac{A_S}{b_w \cdot d}$ |
| $A_s$ | Tensile reinforcement (mm$^2$) | |
| $f_c$ | Compression concrete project strength (N/mm$^2$) | $f_c \leqslant 90$MPa |
| $b_w$ | Minimum thickness web | |
| $d$ | Piece useful depth | |
| $Y_c$ | Security factor (=1 in this job) | |

Kani showed the importance of arch action in not very slender beams [24]. Its importance is inversely proportional to the relationship between the shear span and the effective depth, *a*/*d*. In beams with a coefficient *a*/*d* lower than 2.5, shear cracks are developed and, after an internal redistribution of stress, the beams are capable of resisting a significant load increase because the applied strength can be transmitted directly to the supports through the appearance of compressed struts in concrete. In the case of beams with *a*/*d* greater than or equal to 2.5, this effect loses importance, as can be observed in Fig. 7. For this reason this paper deals only with slender beams, beams with *a*/*d* equal to or greater than 2.5.

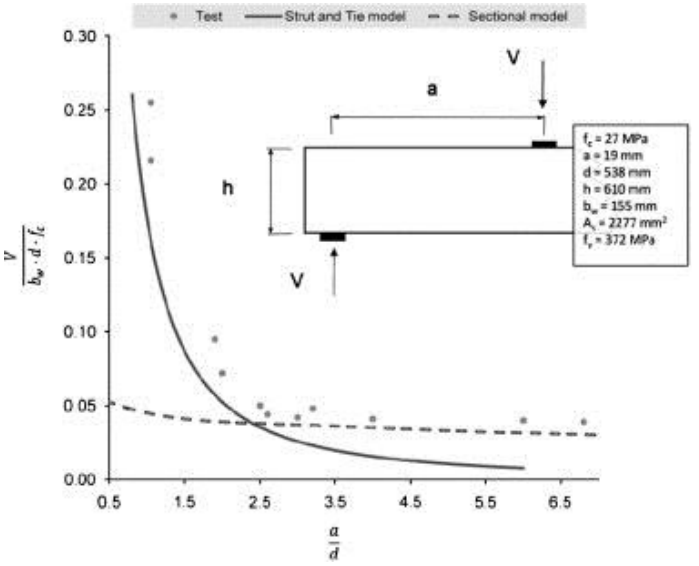


**Fig. 7**. Strength calculated and observed in concrete beams tested by Kani [24].

The treatment of the influence of the longitudinal reinforcement also varies noticeably from one code of practice to another. The formulation given by the Eurocode 2 (EC-2) propounds that shear strength is proportional to the amount of longitudinal reinforcement. However, other models propose that shear strength is proportional to the $\rho_l \cdot V \cdot d/M$ value, as in the case of one of the methods proposed in the ACI Code [23].

One of the limitations that EC-2 poses for elements without shear reinforcement is the fact that the EC-2 procedure does not take into account the bending moment-shear force interaction, except for the need to check that the longitudinal tension reinforcement is able to resist the additional tensile force caused by shear. In a given section, according to the EC-2 formulation, shear strength is independent of the concomitant bending moment if the last is kept away from which produces the yielding of the longitudinal reinforcement. On the other hand, more complex models such as the Modified Compression Field Theory (MCFT) [12] predict a reduction in shear strength as the concomitant bending moment increases for any value of the bending moment.

### 4.2. Equation to be optimized

Eqs. (7), (8) have been established to make the optimization through the algorithm that has been developed:

$$V_c = Branch_1\left(1+\left(\frac{Branch_2}{d}\right)^{Branch_2}\right)(100\cdot\rho_l)^{Branch_4}\cdot f_c^{Branch_5}\cdot b_w\cdot d \tag{7}$$

$$V_c = Branch_1\left(1+\left(\frac{Branch_2}{d}\right)^{Branch_2}\right)(100\cdot\rho_l)^{Branch_4}\cdot f_c^{Branch_5}\cdot (Branch_6)^{Branch_7}\cdot b_w \cdot d \tag{8}$$

Eq. (7) comes directly from the generalization of the Eurocode 2 procedure (5), and it was previously optimized in Ref. [11]. However, in Eq. (8) a new term has been added, $Branch_6^{Branch_7}$ . The aim of introducing a new term is to introduce the influence of the relationship between shear load and the concomitant bending moment.

The choice of these equations for optimization, based on Eurocode 2, is justified on the fact that the formulation given by EC-2 for beams without shear reinforcement is simple and generally offers a good correlation with experimental results, although it does have some gaps, as explained above. In addition, this formulation has been adopted by different national regulations, such as Spanish Code EHE-08 [25], so it is well known to structural engineers. In any case, this paper explains in detail the methodology used to make it possible to replicate the study with using other initial equations.

### 4.3. Database

For the execution of the tests for the study of the shear crack phenomenon, a data base compiled by Collins [26] was mainly used. It has been specifically included 1149 experimental results of beams whose relation *a*/*d*, where *a* is the shear span and *d* is the effective depth (see Fig. 8), is greater than or equal to 2.5 and where the failure of the beams was identified as shear failure.

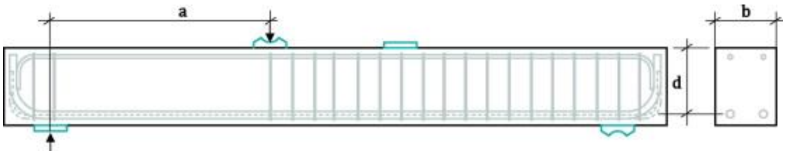


**Fig. 8**. Beam parameters.

### *4.3.1. Available variables*

According to Eurocode 2 procedure, the variables that influence shear strength are beam width, $b_w$, mechanical depth, $d$, compression concrete strength, $f_c$, and the geometric amount of longitudinal tensile reinforcement, $\rho_l$. The result variable is the experimental shear strength, $V_u$.

Besides, there are also available other two variables in the data base: the maximum aggregate size, $a_g$, and the variable $V \cdot d/M$, which takes into account the relation among the shear force and the concomitant bending moment in the critical section and that it is easily deduced from the shear span of the test and the type of load.

The data base was divided into two sets to conduct this study. Five-hundred and eighteen beams were used for the training process. The remaining 631 cases were left for the test process. Table 2 shows the set characteristics of the data used. In Fig. 9, the distribution of the values $d$, $f_c$ and $\rho_l$ for the data base beams is shown in a graphic way, depending on whether they belong to the training or verification set. Division of the beams into two sets was random, but it was compulsory in the training set to included beams whose variables belong to the range ends observed in Table 2. In this way, and as observed in Fig. 9, for most of the usual variable values, less than 50% of the beams were part of the training set but, for the most extreme variable values, most of the beams were introduced in the training set. In this way, an attempt was made to minimize the lack of homogeneity of the database used.

**Table 2**. Data set used

| Data set | Amount | $b_w$ (mm) | | $d$ (mm) | | $f_c$ (MPa) | | $\rho_l$ (%) | | $V \cdot d/M$ (–) | | $a_g$ (mm) | | $Vu$ (kN) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | Min | Max | Min | Max | Min | Max | Min | Max | Min | Max |
| Training | 518 | 21 | 3000 | 41 | 3000 | 6.1 | 127.5 | 0.14 | 9.5 | 0.077 | 0.904 | 2 | 50 | 1.9 | 1575.00 |
| Test | 631 | 21 | 1829 | 41 | 2000 | 6.3 | 127.5 | 0.17 | 9.42 | 0.071 | 0.953 | 1 | 50 | 2.1 | 789.00 |

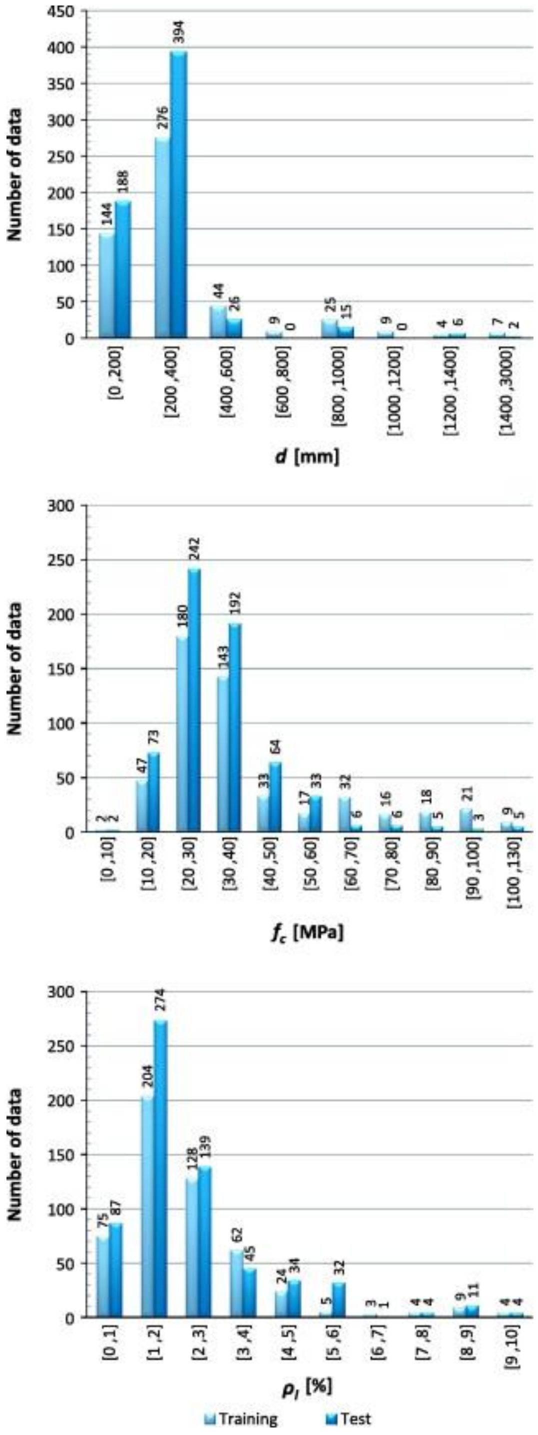


**Fig. 9**. Distribution of the essential variables from EC-2 formulation for the database beams that were used.

*4.3.1.1. Variables that were selected*

As mentioned at the beginning of this section, it is expected to optimize Eqs. (7), (8) which are based on Eurocode 2. A subset of variables ($d$, $f_c$ and $\rho_l$) were used to optimize Eq. (7). The decision of selecting only this set is due to the fact that the current EC-2 formula only uses such variables in the equations points that are proposed to be improved (Eq. (11)).

On the other hand, the variables ($d$, $fc$, $\rho_l$ and $V \cdot d/M$) were used in the optimization of Eq. (8). The use of the last variable depends on being able to study the influence of the concomitant moment and failure shear relationship.

At a data later than the development of the equations that were proposed in this article, a study was conducted as to the necessity to incorporate, or not, the maximum aggregate size, $a_g$. The results are presented in Section 6 and they led us not to use this variable.

### *4.4. Fitness function*

Regarding the definition of the parameters $k$ and $l_{bias}$ of the fitness function (see Eq. (9)), tests using classic GP techniques were carried out. It was decided to opt for using 1.0, 1.1 and 1.3 for the execution of these tests. The value of $V_{test}/V_{pred}$ of 1.0 represents the exact value of the prediction, against the values 1.1 or 1.3 in which it is oversized. In this way the model will be located slightly on the safety side (the value of the formula that has been obtained will be inferior to the test value) with the aim of obtaining a conservative prediction.

Besides, the tests were carried out with or without the error adjustment regarding the interval. In Eq. (9) the value $k$ that was inspired in the use of the "demerit points" technique and used by Collins to categorize the results is shown [27].

In total, five sets of different tests were carried out as a whole in order to determine the best value of $l_{bias}$ and the necessity, or not, to use the value $k$ (aim 1.0 with or without penalization; aim 1.1 with or without penalization; and aim 1.3 with penalization) when using the same configuration. This was: population of 1000 individuals, crossover rate of 80%, mutation rate of 20%, selection algorithm by tournament and maximum height of seven depth nodes. This configuration was chosen after the execution of some qualifying tests, since it was the configuration that produced better results. Besides, the tests were repeated using different parsimony values: a set of tests without taking into account the solution size and other two groups with parsimony values fixed at 0.01 and 0.000001. One-hundred and fifty runs were made as a whole.

$$k = \begin{cases} 10, & \dfrac{V_{test}}{V_{pred}} < 0.5 \\ 5, & 0.5 \leqslant \dfrac{V_{test}}{V_{pred}} < 0.67 \\ 3, & 0.67 \leqslant \dfrac{V_{test}}{V_{pred}} < 0.85 \\ 1, & 0.85 \leqslant \dfrac{V_{test}}{V_{pred}} < 1.3 \\ 2, & 1.3 \leqslant \dfrac{V_{test}}{V_{pred}} < 2 \\ 3, & \dfrac{V_{test}}{V_{pred}} \geqslant 2 \end{cases} \tag{9}$$

Fig. 10 shows the demerit point values (DP) and the coefficient of variation (COV) of the resultant expressions for the different executions against the experimental data of training and verification. In Appendix B the calculation formulas that were considered for the COV and the DP are presented. As can be observed, it can be corroborated, after the tests are made, that the best results are obtained when using aim value 1.1 and the error adjustment factor (value *k*), both in training and in test (see Fig. 10).

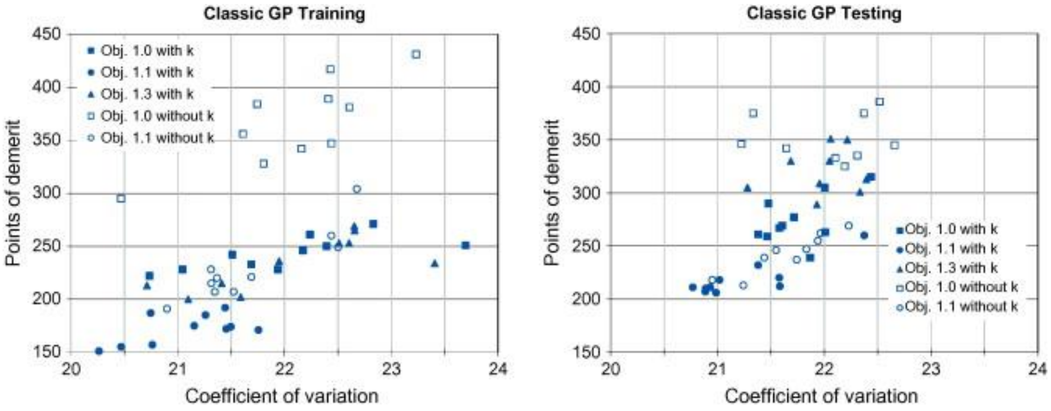


**Fig. 10**. Training and test results without parsimony.

### 4.5. GP application

Several sets of tests were established, following the aim of leading the search space according to expert indications in the job area. Table 3 shows the default settings used in the runs of the implementation of the algorithm that has been developed. These parameters were chosen due to the fact that they were the ones that gave the best results in the initial tests. The input data to the algorithm were not standardized, with the aim of the resultant formulas being directly applicable.

**Table 3**. Parameters used

| Configuration parameters | Default values | Other values |
|---|---|---|
| Population size | 1000 | |
| Crossover rate | 80% | |
| Non-terminal selection rate | 90% | |
| Mutation probability | 20% | |
| Algorithms | Selection: Tournament | |
| | Initialization: Ramped Half & Half | |
| | Mutation & Crossover: Subtree | |
| Elitist strategy | Yes | |
| Parsimony | 0 | 0.1, 0.01, 0.001, 0.0001, $1 \times 10^{-7}$ or $1 \times 10^{-10}$ |
| Initial tree depth | 3 | 6 |
| Maximum tree depth | 6 | 8 or 10 |
| Maximum mutation depth | 3 | 7 or 8 |

Addition, subtraction, product and protected division were chosen as default operators and non-terminal nodes. This new definition of the division operator widens the application dominion to all the real numbers for the purpose of ensuring the individual evaluation. In this case, the behavior chosen is shown in Eq. (10). Variables belonging to the data set ($d$, $f_c$ and $\rho_l$) for the terminal nodes, random constants from the interval (−1, 1) with the integer numbers between the intervals [−10, 10] are used.

$$ProtectedDivision(a,b) = \begin{cases} 1, & b = 0 \\ \frac{a}{b}, & b \neq 0 \end{cases} \qquad (10)$$

*4.5.1. Restrictions*

Table 4 shows the most representative restrictions made using Eq. (7) as base. The symbol "Cst" indicates that any constant is allowed to be used (generated through GP, containing operations such as addition, subtraction, product and protected division of constants: real numbers in the interval (−1, 1) and integer numbers in the interval [−10, 10]). With the symbols $d$, $f_c$ o $\rho_l$, or any combination between them, it is indicated that such variables from the data base can be used or any constant that is related through the operators of addition, subtraction, product and protected division. Finally, when a value appears, it indicates that only this value will be used in that branch.

**Table 4**. Restrictions applied in Eq. (7).

| Restriction | $Branch_1$ | $Branch_2$ | $Branch_3$ | $Branch_4$ | $Branch_5$ |
|---|---|---|---|---|---|
| A | Cst | $d,f_c,\rho_l$ | $d,f_c,\rho_l$ | $d,f_c,\rho_l$ | $d,f_c,\rho_l$ |
| B | Cst | $d,f_c,\rho_l$ | Cst | Cst | Cst |
| C | Cst | Cst | $d,f_c,\rho_l$ | Cst | Cst |
| D | Cst | Cst | Cst | $d,f_c,\rho_l$ | Cst |
| E | Cst | Cst | Cst | Cst | $d,f_c,\rho_l$ |
| F | Cst | Cst | $0.40 + 0.001f_c$ | 0.37 | Cst |
| G | Cst | Cst | Cst | Cst | Cst |

Once the tests have been made, using Eq. (7) as base and after the experience that was obtained, only two restriction sets were established for Eq. (8). In Table 5 these restrictions are shown.

**Table 5**. Restrictions applied in Eq. (8).

| Restriction | $Branch_1$ | $Branch_2$ | $Branch_3$ | $Branch_4$ | $Branch_5$ | $Branch_6$ | $Branch_7$ |
|---|---|---|---|---|---|---|---|
| H | Cst | 1600 | 0.42 | Cst | 1/3 | $V \cdot d/M$ | Cst |
| I | Cst | Cst | $f_c$ | Cst | Cst | $V \cdot d/M$ $V/M$, $d,\rho_l$ | Cst |

## 5. Results

In the following tables it is shown the adjustments obtained by the current standardized formulations and the equations proposed in this article against the database used. The equations that have been developed in this research work have been subdivided into three categories. The first one represents the equations that were generated through the algorithm developed in this article; the second one corresponds to the equations that were generated with classic GP; and, finally, the equations obtained with classic GP with fitness error. For each category the best value is shown in bold type, highlighting the best value as a whole (underlined). In each table the equation name appears (Formula_Base + Name_Restriction + Solution_Number); for example, equation 7A3 means Equation base 7, restrictions "A", selected formula number "3". The $V_{test}/V_{pred}$ term represents the average quotient of the value obtained in the real test into the value that was predicted through the different models. Appendix B presents the coefficient of variation value COV. $R^2$ is the square of the correlation coefficient from the Pearson product. Finally, there are the Mean Squared Error (MSE), the Mean Average Error (MAE) and the demerit points (DPs). Appendix B presents the formulas that were used for the calculation.

In Table 6 the results for the training set are shown. Regarding the criterion to follow, it would be obtained that equations 7A1, 7A3, 8H1 and 8I1 (see Appendix A) are the best ones to predict shear strength for this data set. If the demerit points are taken as a reference, the best function would be 7A3. Instead, if the MSE is taken as a reference, the best model would be 7A1.

**Table 6**. Training results

| | | COV | $V_{test}/V_{pred}$ | $R^2$ | MSE | MAE | DP |
|---|---|---|---|---|---|---|---|
| International models | EC2 | **23.40** | 1.10 | **0.960** | 2165.38 | **17.84** | **229** |
| | ACI 11-3 | 33.23 | 1.44 | 0.867 | 18820.12 | 41.11 | 636 |
| | ACI 11-5 | 28.72 | 1.30 | 0.876 | 18962.56 | 38.13 | 551 |
| Models generated by the developed GP algorithm | 7A1 | 17.32 | 1.09 | 0.978 | **387.92** | **11.36** | 106 |
| | 7A3 – GP-3 | 16.51 | 1.10 | 0.972 | 542.31 | 11.84 | **72** |
| | 7B1 | 17.88 | 1.11 | 0.960 | 806.80 | 13.30 | 116 |
| | 7C1 | 18.37 | 1.09 | 0.968 | 767.09 | 12.55 | 122 |
| | 7D1 | 17.94 | 1.09 | 0.976 | 433.11 | 11.89 | 123 |
| | 7E1 | 17.61 | 1.10 | 0.977 | 446.50 | 12.03 | 114 |
| | 7F1 – GP-2 | 18.06 | 1.09 | **0.978** | 410.72 | 11.78 | 125 |
| | 7G1 – GP-1 | 18.06 | 1.09 | 0.975 | 445.70 | 11.97 | 124 |
| | 8H1 – GP-4 | 16.21 | 1.09 | 0.972 | 696.41 | 12.51 | 130 |
| | 8I1 | **15.91** | 1.09 | 0.975 | 528.54 | 11.85 | 104 |
| Classic GP | P1.0 | **18.40** | 0.98 | **0.967** | 944.29 | 13.37 | 273 |
| | P1.1 | 18.59 | 1.08 | 0.965 | **803.00** | **13.60** | **165** |
| Classic GP with error adjustment (*k*) | P1.0K | 18.49 | 1.03 | 0.948 | 2552.57 | 15.78 | 199 |
| | P1.1K | **18.33** | 1.10 | 0.961 | 705.53 | 13.94 | 145 |
| | P1.1K-C | 16.78 | 1.10 | **0.971** | **687.33** | **12.58** | **95** |

Nevertheless, it has special importance to study the response of the formulations that were obtained for the test set, since they are beams that were not used in the learning process. In this case (see Table 7), the function that has the greater generalization ability is 8I1, followed by 8H1.

**Table 7**. Test results

| | | COV | $V_{test}/V_{pred}$ | $R^2$ | MSE | MAE | DP |
|---|---|---|---|---|---|---|---|
| International models | EC2 | **25.80** | 1.14 | **0.928** | **627.68** | **13.23** | **267** |
| | ACI 11-3 | 33.26 | 1.53 | 0.829 | 1902.22 | 25.62 | 639 |
| | ACI 11-5 | 25.36 | 1.35 | 0.845 | 1778.20 | 21.98 | 488 |
| Models generated by the developed GP algorithm | 7A1 | 20.11 | 1.11 | 0.941 | 548.76 | 12.18 | 195 |
| | 7A3 – GP-3 | 20.07 | 1.14 | **0.962** | 497.30 | 12.50 | 177 |
| | 7B1 | 22.44 | 1.16 | 0.931 | 673.01 | 13.73 | 213 |
| | 7C1 | 20.20 | 1.11 | 0.942 | 527.19 | 11.95 | 195 |
| | 7D1 | 20.56 | 1.12 | 0.950 | 536.59 | 12.59 | 221 |
| | 7E1 | 20.90 | 1.13 | 0.944 | 631.20 | 13.28 | 221 |
| | 7F1 – GP-2 | 20.78 | 1.11 | 0.950 | 567.72 | 12.89 | 219 |
| | 7G1 – GP-1 | 20.64 | 1.12 | 0.949 | 546.25 | 12.73 | 212 |
| | 8H1 – GP-4 | 17.75 | 1.12 | 0.946 | 501.52 | 11.78 | 153 |
| | 8I1 | **17.01** | 1.11 | 0.949 | **466.99** | **11.26** | **136** |
| Classic GP | P1.0 | 21.23 | 1.01 | 0.934 | 507.54 | 11.65 | 346 |
| | P1.1 | **20.95** | 1.12 | **0.956** | **477.34** | 12.49 | **218** |
| Classic GP with error adjustment (*k*) | P1.0K | 21.46 | 1.08 | 0.922 | 623.14 | **12.02** | 259 |
| | P1.1K | **20.58** | 1.13 | **0.959** | **503.57** | 12.73 | **213** |
| | P1.1K-C | 22.67 | 1.14 | 0.944 | 804.65 | 14.28 | 229 |

More information can be obtained regarding the correlations of formulas 7G1, 7F1, 7A3 and 8H1 in Ref. [11], under the name GP-1, GP-2, GP-3 and GP-4 respectively. These equations were initially selected; especially GP-1 and GP-4 for their simplicity of use, although equation 8I1 should also be taken into account for further studies.

It can be observed that the equations that are generated by the present algorithm offer better results than the ones that are generated by classic-GP techniques, especially when checking the correlation regarding the test set. Besides, as can be seen in Appendix A, it is possible to obtain equations with great simplicity, as is the case of expressions 7D1, 7E1, 7F1, 7G1, 8H1 and 8I1. Furthermore, the equations that were developed by the proposed algorithm improve the EC2 correlation and the ACI318-05 code formulations (equations 11-3 and 11-5).

In addition, it can also be observed that the hypothesis that was made by the expert team when defining Eq. (8), which makes it possible to include variable $V \cdot d/M$, is suitable since the two equations obtained by this procedure are the ones that give better results.

It is worth noting the safety rise that was achieved both with the skew “1.1” that was applied and the factor “*k*” of error adjustment. This effect can be seen in Fig. 11 which shows the correlation of Eurocode 2 and equation 8I1. The shady area corresponds to the unsafe values of the predictions.

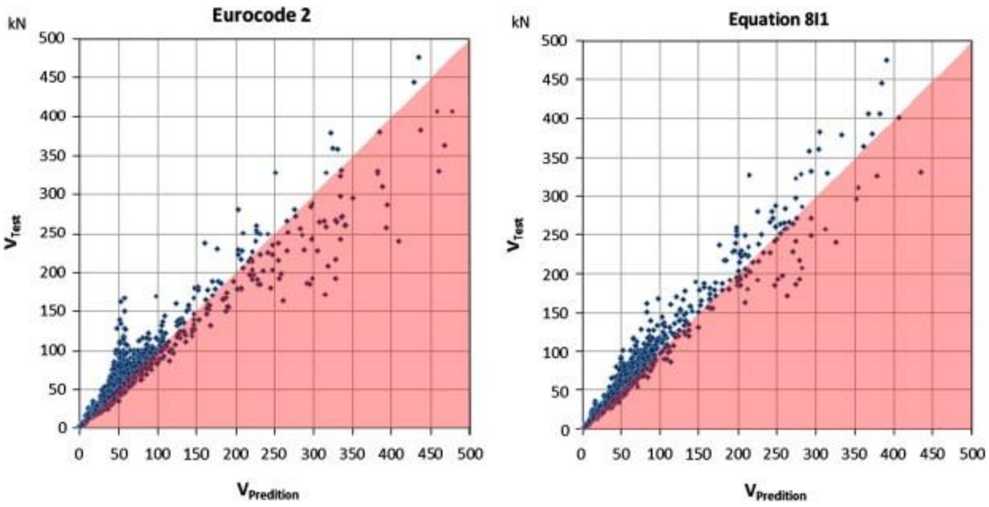


**Fig. 11**. Correlation between the test value and the prediction value.

Fig. 12 shows the coefficient of variation of each model and equation. It can be observed that the international models are the ones that classify the worst, followed by classic GP which – taking into account the worst equation P1.0 – obtained better results than any of the international models. These results improve when using factor *k* of error adjustment. And, finally, when applying the algorithm that has been developed in this article, it is possible to refine even more and make fewer errors in the prediction.

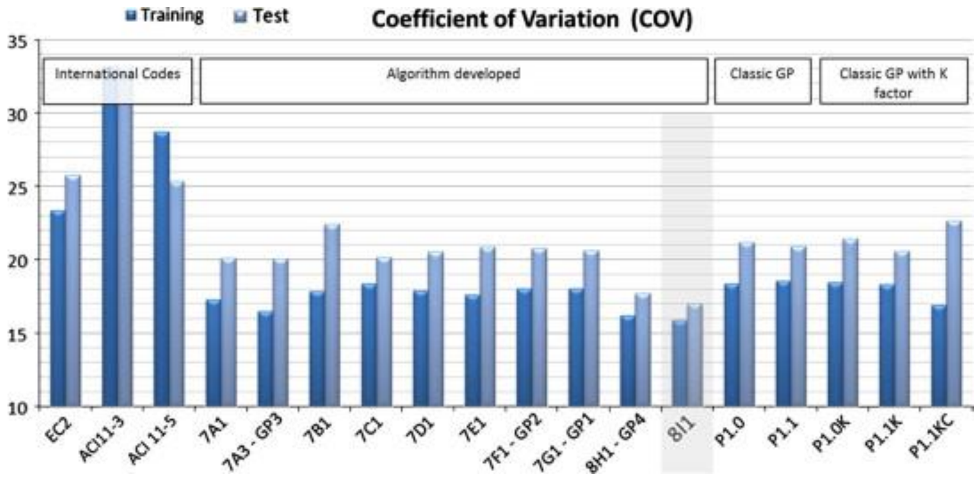


**Fig. 12**. Coefficient of variation of the formulas.

## 6. Influence of the maximum aggregate size

The design procedures based on the Modified Theory of Compression Field [12] include as a design variable the maximum aggregate size, $a_g$. Its influence on shear strength is based on the fact that the aggregate size is a main parameter in the shear-friction mechanism, which is, at the same time, one of the key mechanisms of shear strength. Its influence was inferred, at that moment, from the maximum aggregate size influence in the shear-friction response in type Z elements [28]. Its influence is usually related to the size effect. Summing up, and for strength conventional concretes, the greater the maximum aggregate size, the more shear-friction and, therefore, the more shear strength.

Once the previous expressions were obtained, expression 7G1 was taken as the departure reference to make a specific study of the maximum aggregate size influence on real beam tests. This expression was taken since it has great simplicity and considers the same variables as the Eurocode 2 formulation. In this expression it was allowed to optimize two branches, as shown in

$$V_c = Branch_0 \cdot Branch_1 \cdot (100_{pl})^{0.37} \cdot f_c^{1/3} \cdot b_w \cdot d \qquad (11)$$

The study of the influence of the variable $a_g$ by GP techniques was carried out in two phases. In both of them departure expression (11), where the following restrictions were imposed: *Branch*$_0$ must be a constant and *Branch*$_1$ is a function that simulates the size effect and the possible maximum aggregate size influence. In the first phase of the study a penalization in the adjustment due to the tree size (parsimony) was not included; that is why expressions from *Branch*$_1$ could be very complex, with the imposition of the following restrictions:

(I) *Branch*$_1$ depends on $d$.
(II) *Branch*$_1$ depends on $d$ and $a_g$.
(III) *Branch*$_1$ depends on $s_{xe}$, a parameter defined in Eq. (12).
(IV) *Branch*$_1$ depends on $d$ and $f_c$.
(V) *Branch*$_1$ depends on $d$, $a_g$ and $f_c$.

$$s_{xe} = \frac{36}{16 + a_g} \cdot 0.9d \qquad (12)$$

In the second phase of this study, the same general restrictions were kept but it was opted to set parsimony at a value of 0.00001 and the maximum height of the tree equal to 6. This configuration forced GP to give very simplified expressions.

Table 8 compares the expressions obtained in the first phase, without using the parsimony term, for a whole number of 50 runs for each case. It can be observed that for the training set, a certain improvement is produced when considering variable $a_g$ in the size effect term (*Branch*$_1$) and, above all, some results are even better when considering variables $a_g$ and $f_c$. The results worsen slightly when considering variable $s_{xe}$ instead of variable $d$. However, when analyzing the results for the test set, it is proved that the best correlations in the training set are due to a certain overtraining, since these improvements become inexistent when comparing the formulations obtained with the tests of beams that were not used in the training process.

**Table 8**. Average results and better ones for 50 executions – phase 1

| Case | Variables *Branch*$_1$ | Training set | | | | Testing set | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Average COV | Minimum COV | Average DP | Minimum DP | Average COV | Minimum COV | Average DP | Minimum DP |
| I | $d$ | 19.39 | 18.12 | 132.76 | 115 | 20.44 | 19.57 | 207.02 | 185 |
| II | $d,a_g$ | 18.34 | 16.99 | 112.66 | 92 | 20.68 | 19.62 | 204.12 | 185 |
| III | $s_{xe}$ | 19.71 | 18.39 | 128.76 | 117 | 20.84 | 19.89 | 233.46 | 210 |
| IV | $d,f_c$ | 19.40 | 17.50 | 128.64 | 106 | 20.60 | 19.62 | 208.16 | 186 |
| V | $d,a_g,f_c$ | 18.45 | 16.37 | 112.40 | 90 | 30.35 | 19.54 | 209.86 | 192 |

In the second phase, while limiting the complexity of $Branch_1$ a lot, each of the case studies executed a total of 50 times, and three formulas were analyzed with a lower coefficient of variation for each of the cases. In Table 9 there is an outline of the results in the second phase of the study (parsimony 0.00001, maximum height 6), in which it is clearly observed that when limiting the complexity of the solution, Genetic Programming tends to remove variable, $a_g$, which only arises in one of the six possible cases. So, GP tends to disregard the influence of that variable; which is why it omits it, since as a whole the contribution of $a_g$ is minimal compared to the contribution that other variables may have, like the one from $d$.

**Table 9**. Selection of GP variables when limiting the complexity of *Branch*$_1$.

| Case | Allowed variables *Branch*$_1$ | Variables that really appear in *Branch*$_1$ | | |
|---|---|---|---|---|
| | | Eq. (1) | Eq. (2) | Eq. (3) |
| I | $d$ | $d$ | $d$ | $d$ |
| II | $d,a_g$ | $d$ | $d$ | $d$ |
| III | $s_{xe}$ | $s_{xe}$ | $s_{xe}$ | $s_{xe}$ |
| IV | $d,f_c$ | $d$ | $d,f_c$ | $d$ |
| V | $d,a_g,f_c$ | $d,a_g$ | $d$ | $d$ |

Finally, it has been studied the Eq. (11) from case V (Table 9), the only equation in which $a_g$ appears to analyze the importance that the formulation that was proposed by GP gives to maximum aggregate size, $a_g$ (Eq. (13)):

$$Branch_0 = 195/196$$
$$Branch_1 = \frac{7680 + 480 \cdot d + 160 \cdot a_g}{1440 \cdot d + d^2} \tag{13}$$

The graphical representation of the maximum aggregate size influence in the tensile crack is presented in Fig. 13, for a beam of 300 mm of effective depth and for another beam of 1000 mm of depth. The rest of the variables were kept constant. It is observed that increasing the maximum aggregate size from 5 mm to 50 mm has a small influence on shear strength: approximately 4.7% for a beam with small depth ($d$ = 300 mm) and 1.5% for a beam with large depth ($d$ = 1000 mm). This influence is small, taking into account that it is about extreme values of the variables, and it is much lower than the errors that were obtained by all the calculation formulations presented. So, taking into account the great variability in experimental shear crack results, it is estimated that it is not necessary to include the maximum aggregate size variable, $a_g$, for the study of shear crack in real beams. Besides, it must be pointed out that the maximum aggregate size is a parameter that concrete supplying companies do not always respect in building practice. Finally, it is worth highlighting the physical sense of the result that was obtained through GP techniques: the more effective the depth, the lower the influence of maximum aggregate size.

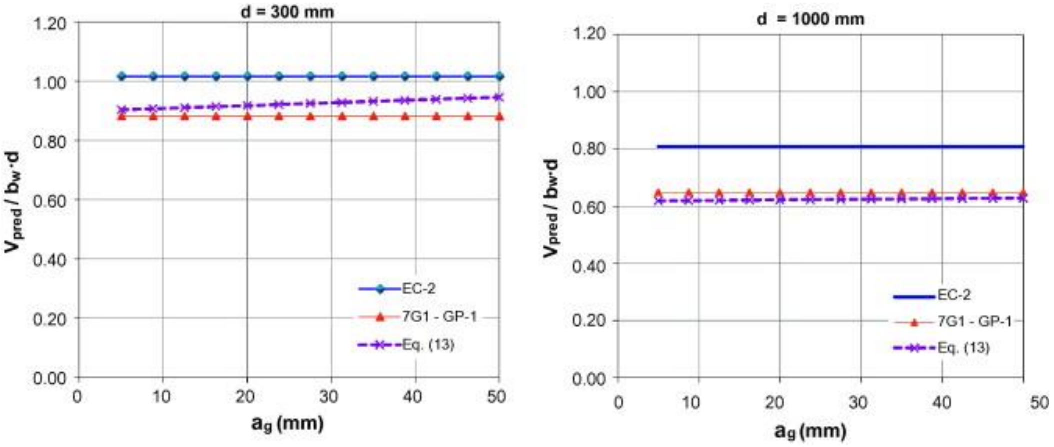


**Fig. 13**. Tensile shear crack versus effective depth, $d$, and maximum aggregate size, $a_g$.

## 7. Bending moment-shear force interaction

The proposed 8H1 and 8I1 equations introduce the term of $V \cdot d/M$ or $V/M$, which reflects the influence of the relationship between the concomitant moment and shear force. Fig. 14 presents the diagram of ultimate bending moment-shear force interaction obtained with Eurocode 2, the proposed equations 8H1 and 8I1, equation 7G1 and through the software Response-2000 [29], based on the Modified Compression Field Theory [12].

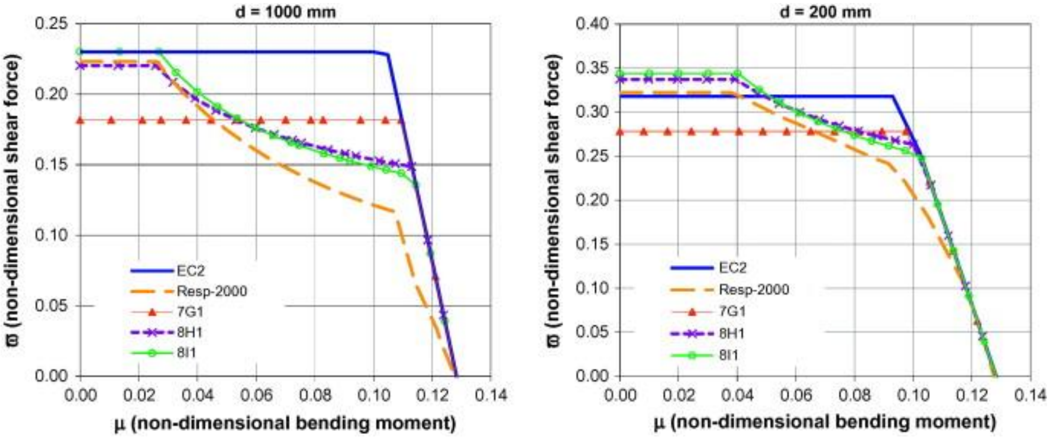


**Fig. 14**. Bending moment-shear force interaction according to EC-2, Response-2000 (Resp-2000) and proposed equations 7G1, 8H1 and 8I1, as a function of beam depth ($b_w$ = 250 mm, $f_c$ = 25 MPa, $\rho_l$ = 0.689%).

The non-dimensional values of Fig. 14 were obtained using

$$\mu = \frac{M_u}{b \cdot d^2 \cdot f_c}$$
$$\omega = \frac{V_u}{b \cdot d \cdot f_c^{1/3}} \tag{14}$$

In this section, the limits given by Eq. (15) were used for the graphical representations of the results given by equations 8H1 and 8I1. These limits do not apply to any of the beams from the database, but it is necessary to fix it to extend the formulation to short beams with a small *a*/*d* ratio.

$$\frac{V \cdot d}{M} \ngtr 1.0; \frac{V}{M} \ngtr d^{-1} \tag{15}$$

It can be observed that Eurocode 2 does not present ultimate moment-ultimate shear interaction until high values of the bending moment. On the other hand, the simple formulas 8H1 and 8I1 present interaction for any value of the concomitant bending moment and their behavior is similar to the one given by the sophisticated model Modified Compression Field Theory (Response-2000). The values obtained by equation 7G1 represent an average of the Response-2000 results, being slightly conservative for low non-dimensional bending moments and somehow non-conservative for high non-dimensional moments. The effect of beam depth increase may be clearly seen in Fig. 15 for the different procedures. EC-2 shear procedure is, compared to the Response-2000 results, unsafe for high concomitant bending moments. This effect is especially remarkable for large beams. In Fig. 15 the graph for equation 8I1 is not given, but it will be very similar to equation 8H1.

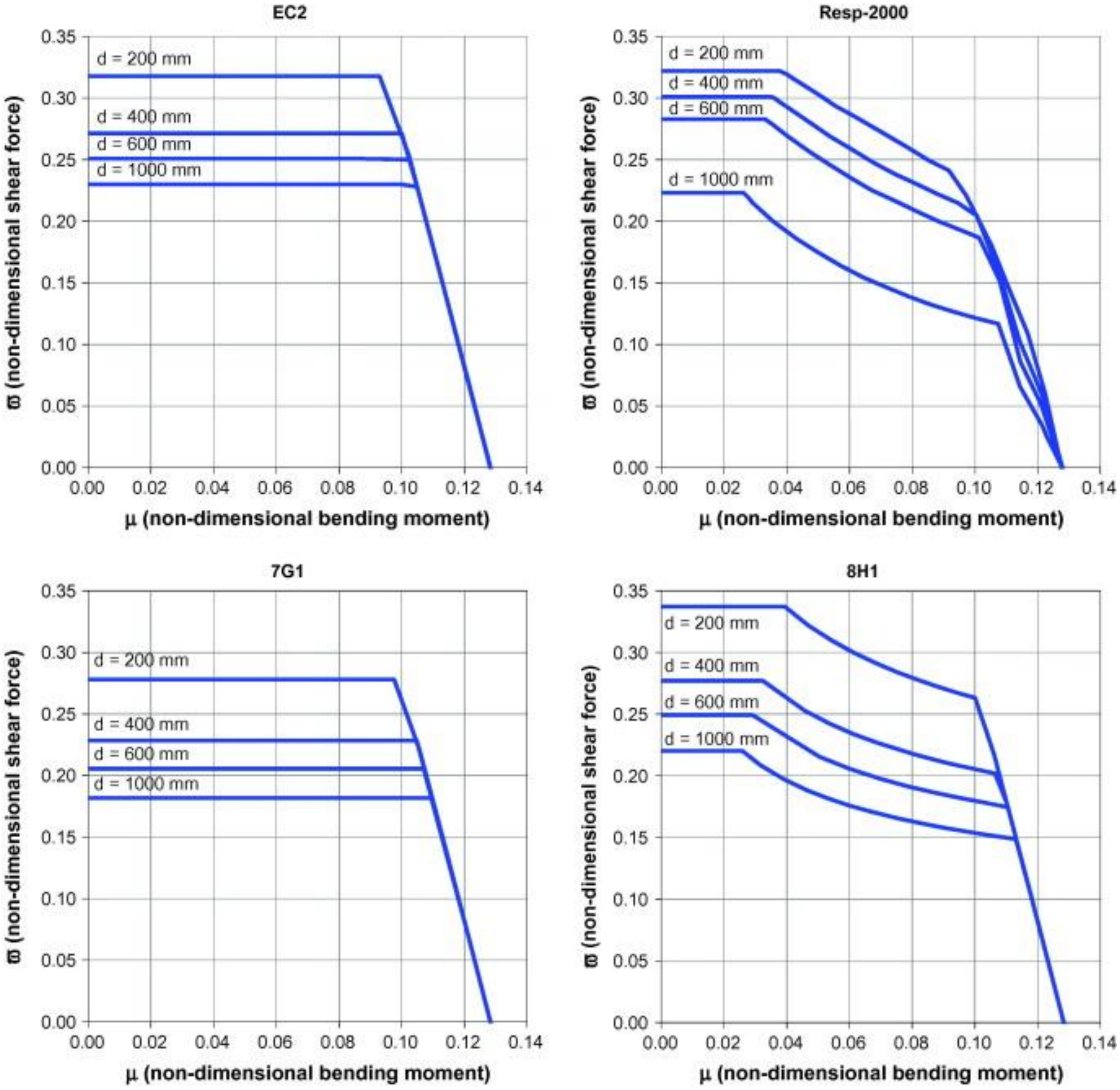


**Fig. 15**. Bending moment-shear force interaction diagrams as a function of beam depth.

The fact of taking into account the bending moment-shear force interaction has effects on the influence on the amount of longitudinal reinforcement. Fig. 16 presents the influence of $\rho_l$ according to different shear procedures. Fig. 16a shows the evolution of the predictions of shear failure stress when increasing the value of $\rho_l$ and keeping the rest of the parameters constant. In this case, the represented equations (EC-2, 7G1, 8H1, 8I1) show a similar behavior for low values of longitudinal reinforcement. For values higher than 2% of longitudinal amount, EC-2 does not consider increases in the value of the ultimate shear, whereas this keeps increasing for the formulations proposed through GP techniques.

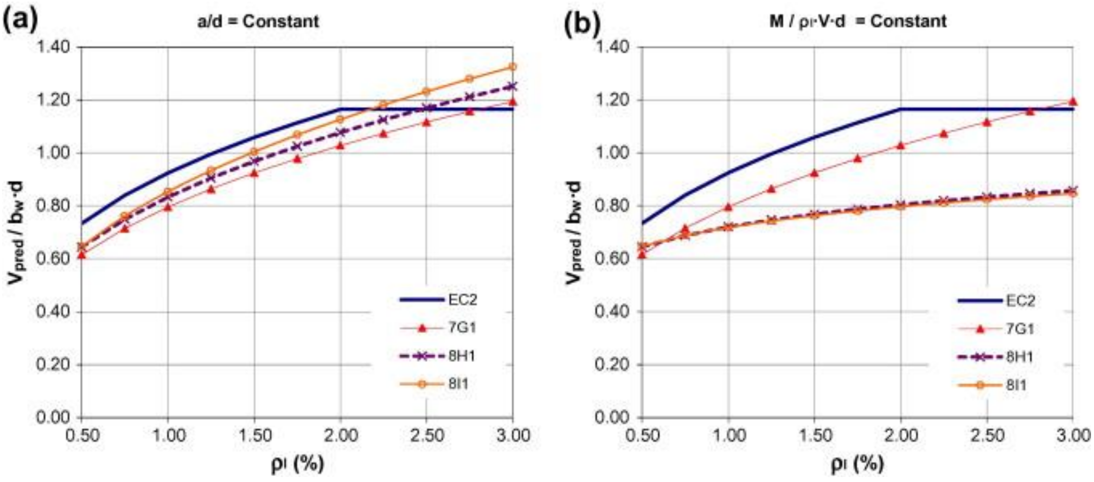


**Fig. 16**. Influence of the amount of longitudinal reinforcement. (a) Beams with *a*/*d* constant. (b) Beams with $M/\rho_l \cdot V \cdot d$ constant.

In Fig. 16b the shear spans of the beams was increased as the value of the longitudinal reinforcement amount increased, which leads to the value of the stress in the reinforcement remaining constant, or what is equivalent, the value of the parameter $M/\rho_l \cdot V \cdot d$ keeps constant. It is observed that the predictions of EC-2 and 7G1 are identical to the ones given in Fig. 16a. However, for the 8H1 and 8I1 formulations, ultimate shear increases to a lesser extent when increasing the amount of longitudinal reinforcement.

## 8. Conclusions

The algorithm that has been presented is a valid method for improving existing expressions in certain points that were chosen by the experts with different possible restrictions. The restrictions can refer both to the choice of the data set variables that can be used, and what operators (both terminal and non-terminal) are allowed in the generation of new individuals. With the proposed solution it is gone beyond mere numerical value adjustments within an expression. Although this case would be a subset of problems that it is possible to solve through the algorithm presented, it would be more suitable to use the genetic algorithms, which are another evolutionary computation technique.

With the set of tests that were made, the validity of the algorithm developed is shown, obtaining better results than the international Codes of Practice that were analyzed. Although with this method good adjustments are achieved with relatively complex formulas (equation 7A1), this one is too complex from the point of view of a standardized approach of structural engineering. The power of the method is highlighted when obtaining formulas of great simplicity, as in the case of equations 7F1, 7G1, 8H1 and 8I1, which could be taken into account to incorporate them to the Code of Practice as a prediction model of shear strength for concrete beams without shear reinforcement. Within the four formulas, 8I1 stands out, as it obtains very good results with a maximum simplicity; or the set that is made up of equations 7G1 and 8H1, since they are alike and the difference lies in taking into account 8H1 or not 7G1 the phenomenon of the bending moment-shear force interaction.

Restriction refinement was essential in obtaining the final results, thanks to the experience of the Civil Engineers involved in the tests development.

Although an almost similar adjustment can be achieved with classic Genetic Programming techniques, it is not managed to satisfy the field expert demands, due mainly to two reasons. Firstly, with the same level of adjustment, the resultant formula of classic GP is more complex; secondly, the great difference there is between the formulas obtained by classic GP as opposed to the internationally well-known models. In fact, structure designers are generally reluctant to important changes in a standardized model even though a significant improvement is shown; that is why it is easier for them to accept the improvement of a model that keeps great resemblance with the original, already known formulation.

**Acknowledgements**

This work was partially supported by the Spanish Ministry of Science and Innovation (Refs. BIA2007-60197 and BIA2010-21551) and Grants from the Ministry of Economy and Industry (Consellería de Economía e Industria) of the Xunta de Galicia (Refs. 08TMT005CT and 10TMT034E).

## Appendix A. Resulting equations

$$7A1 = 0.81994081 \cdot \left(1 + \frac{abs\left(\frac{14.2569}{\left(\frac{4}{10 \cdot \rho_l} - 8\right)} - f_c\right)}{d}\right)^{\frac{d}{8*\left(f_c - \frac{5000 \cdot \rho_l}{\frac{d}{16} - 3} + 50\right)}} (100\rho_l)^{0.37058625} \cdot f_c^{-\frac{1}{\frac{5 \cdot f_c}{7} - \frac{d}{252} + \frac{3}{20}}} \cdot b_w \cdot d$$

$$7A3 = 1 \cdot \left(1 + \left(\frac{Branch_2}{d}\right)^{-\frac{-\frac{1}{320}(-1+\frac{1}{10}\cdot f_c)*(100\rho_l - \frac{1}{3}\cdot f_c) + \frac{1}{30}\cdot d}{-f_c + 3 + 100 \cdot \rho_l}}\right) (100\rho_l)^{Branch_4} \cdot f_c^{\frac{1\ \frac{1}{10} + \frac{3}{100}\cdot d}{10 \quad 4 - f_c}} \cdot b_w \cdot d$$

where:

$$Branch_2 = \frac{\frac{\frac{\frac{-4 + f_c - 100\rho_l}{d - 12 - 10 \cdot f_c}}{65 - \frac{1}{3} \cdot d}}{\frac{1}{3} \cdot \frac{d}{100\rho_l - 0.63} - \frac{31}{3} \cdot f_c + \frac{\frac{1}{27} \cdot \left(36 f_c - \frac{1}{3} \cdot d + 10\right) \cdot d}{\frac{1}{3} \cdot d - \frac{36 + 10 \cdot f_c}{100\rho_l}}}}{\frac{1 - \frac{9}{10} \cdot f_c - \frac{9}{d}}{(-\frac{5}{9} - \frac{1}{3} \cdot d) \cdot \left(d + \frac{10}{d} \cdot (f_c + 3)\right) - 105.6838 - \frac{1}{3} \cdot d}}$$

$$Branch_4 = \frac{8f_c - 45 - \frac{1}{100} \cdot \frac{d^2}{(100\rho_l - f_c) \cdot (f_c - 5) - 100} + \frac{\frac{\frac{300}{100\rho_l - 0.63} - \frac{7}{3} \cdot d + 3}{-\frac{50}{69} \cdot d}}{\frac{9}{50} \cdot f_c - \frac{63}{250} + \frac{1}{30} \cdot d}}{\frac{\frac{\frac{0.63 - \frac{d}{3}}{-10} - \frac{1}{3} \cdot d}{\frac{100 - \frac{1}{3} \cdot d}{f_c}}}{\frac{8}{9} - \frac{1}{90} \cdot d} + 18 \cdot f_c}$$

$$7B1 = \left(\frac{24}{49} + \frac{24}{49} \cdot \left(\frac{abs\left(\frac{10000}{3} \cdot \rho_l - 10f_c + \frac{10}{3} + \frac{1}{6} \cdot d\right)}{d}\right)^{\frac{9}{19}}\right) \cdot \frac{(100\rho_l)^{\frac{5}{12}}}{f_c^{\frac{1}{168}}} \cdot b_w \cdot d$$

$$7C1 = 0.4617 \cdot \left(1 + (\frac{3.67}{d})^{\frac{\left(\frac{d}{9}\right)+\left(\frac{f_c-6}{-3}\right)}{\cdot}\left(\frac{f_c-100\rho_l}{0.23087253}\right)+\left(7+(\frac{d}{9})\right)}\right)(100\rho_l)^{0.3565} \cdot f_c^{0.0823} \cdot b_w \cdot d$$

$$7D1 = 0.0918912\left(1 + \left(\frac{2424.57}{d}\right)^{0.392987}\right)(100\rho_l)^{0.392987} \cdot f_c^{0.31492} \cdot b_w \cdot d$$

$$7E1 = 0.3571\left(1 + \left(\frac{9.8250}{d}\right)^{0.1765}\right)(100\rho_l)^{0.4006} \cdot \frac{f_c^{16-300\rho_l+2f_c}}{d + 3 + 8f_c} \cdot b_w \cdot d$$

$$7F1 = 0.1235(1 + (\frac{1600}{d})^{0.40+\frac{f_c}{1000}})(100\rho_l)^{0.37} \cdot f_c^{1/4} \cdot b_w \cdot d$$

$$7G1 = 0.094\left(1 + \left(\frac{1600}{d}\right)^{0.42}\right)(100\rho_l)^{0.37} f_c^{1/3} \cdot b_w \cdot d$$

$$8H1 = 0.114\left(1 + \left(\frac{1600}{d}\right)^{0.42}\right)(100\rho_l)^{0.37} \cdot f_c^{1/3}\left(\frac{V \cdot d}{M}\right)^{0.21} \cdot b_w \cdot d$$

$$8I1 = 1.75 \cdot (100\rho_l)^{0.4} f_c^{\frac{2}{7}} \cdot (\frac{V}{M})^{\frac{1}{4}} \cdot b_w \cdot d$$

$$P1.0 = \frac{10f_c + b_w d}{-2(100\rho_l) - 0.8612} + (100\rho_l) \cdot b_w \cdot f_c + b_w \cdot f_c + b_w \cdot d + \frac{-5b_w f_c + 2b_w d}{b_w d + f_c^2} \cdot (b_w + b_w \cdot f_c)$$

$$P1.1 = \frac{b_w^2 \cdot f_c + b_w \cdot d(d + f_c)}{-50 \cdot (100\rho_l) \cdot f_c - d - b_w} + (100\rho_l) \cdot b_w \cdot f_c + b_w \cdot d - 20f_c - 10 \cdot (100\rho_l)^2 + 10b_w$$

$$P1.0K = \frac{b_w \cdot d + f_c^2}{-3(100\rho_l) - 0.9119} + (-10 + b_w) \cdot (100\rho_l) \cdot f_c + b_w \cdot f_c + b_w \cdot d + \frac{-3b_w \cdot f_c + b_w \cdot d - 9b_w}{b_w \cdot d + f_c^2}(b_w \cdot f_c - 10d)$$

$$P1.1K = d \cdot \frac{-5d + d \cdot b_w}{-42(100\rho_l) \cdot f_c - \frac{2}{3}b_w - d - 10 \cdot f_c} + b_w \cdot (100\rho_l)\frac{f_c + 5}{-\frac{1}{10}d + 1} + b_w \cdot (100\rho_l) \cdot (f_c + 4) + d \cdot b_w$$

$$P1.1KC = \frac{b_w^2}{b_w + 42} + 4d + 10b_w - 46 + (100\rho_l) + b_w \cdot (100\rho_l) \cdot f_c + d \cdot b_w + 18 \cdot \frac{b_w}{5(100\rho_l) \cdot f_c - 35} + \frac{5}{18} \cdot (100\rho_l) \cdot f_c \cdot \frac{d}{3 + d + (100\rho_l) - f_c} + \frac{36}{7} \cdot \frac{d}{-18 + 6(100\rho_l)} - \frac{\frac{f_c + 10d}{b_w}}{(100\rho_l)} + \frac{(100\rho_l) - f_c - d}{(100\rho_l) - 4} - 7 \cdot (100\rho_l)^2 \cdot f_c + \frac{\alpha + (b_w + 2)b_w \cdot (100\rho_l) + b_w \cdot (100\rho_l) \cdot d + 81b_w - 324 - 72f_c - d \cdot b_w(-9 + b_w + d) + \frac{b_w + d}{(100\rho_l) - 2} \cdot (9d + (100\rho_l))}{\frac{10}{9}d - 7 + 2b_w + 2 \cdot (100\rho_l) + \frac{-5 + d}{2f_c + 5} + (33 - 0.2273f_c)(\cdot (100\rho_l)^2 \cdot f_c + (100\rho_l) \cdot f_c)}$$

where: $\alpha = 2((100\rho_l) \cdot f_c + b_w + d) \cdot (100\rho_l) \cdot f_c \cdot \frac{d}{-\frac{3}{2}d + (100\rho_l)f_c + f_c}$

## Appendix B. Equations for error calculation

$$DemeritPoints(i) = \sum_{i=0}^{n} \quad k_i, \mathbf{k} = \begin{cases} 10, & \frac{V_{real}}{V_{pred}} < 0.5 \\ 5, & 0.5 \leqslant \frac{V_{real}}{V_{pred}} < 0.67 \\ 2, & 0.67 \leqslant \frac{V_{real}}{V_{pred}} < 0.85 \\ 1, & 1.3 \leqslant \frac{V_{real}}{V_{pred}} < 2 \\ 2, & \frac{V_{real}}{V_{pred}} \geqslant 2 \end{cases}$$

$$\mathrm{COV} = \frac{\sqrt{\frac{\sum_{i=1}^{n} \quad (x_i - \bar{x})^2}{n - 1}}}{\frac{\sum_{i=1}^{n} \quad (x_i)}{n}} 100$$

$$R^2 = \left( \frac{\sum_{i=1}^{n} \quad (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} \quad (x_i - \bar{x})^2 (y_i - \bar{y})^2}} \right)^2$$

$$\mathrm{MSE}(X,Y) = \frac{1}{n} \sum_{i=1}^{n} \quad (x_i - y_i)^2$$

$$\mathrm{MAE}(X,Y) = \frac{1}{n} \sum_{i=1}^{n} \quad abs(x_i - y_i)$$

## References

[1] Kicinger R, Arciszewski T, Jong K. Evolutionary computation and structural design: a survey of the state-of the-art. Comput. Struct. 2005;83(23–24):1943–78.

[2] Perera R, Vique J. Strut-and-tie modelling of reinforced concrete beams using genetic algorithms optimization. Constr Build Mater 2009;23(8):2914–25.

[3] Sonebi MC, Cevik A. Genetic programming based formulation for fresh and hardened properties of self-compacting concrete containing pulverised fuel ash. Constr Build Mater 2009;23(7):2614–22.

[4] Cladera A, Marí AR. Shear design procedure for reinforced normal and high-strength concrete beams using artificial neural networks. Part I: beams without stirrups. Eng Struct 2004;26(7):917–26.

[5] Ashour AF, Álvares LF, Toropov VV. Empirical modelling of shear strength of RC deep beams by genetic programming. Comput Struct 2003;81:331–8.

[6] Koza JR. Genetic programming: on the programming of computers by means of natural selection. Cambridge: MIT Press; 1992.

[7] Koza JR. Genetic programming II: automatic discovery of reusable programs. Cambridge, MA: MIT Press; 1994.

[8] Montana DJ. Strongly typed genetic programming. Evol Comput 1995:199–230.

[9] García-Arnau M, Manrique D, Ríos J, Rodríguez-Patón A. Initialization method for grammar-guided genetic programming. Knowl-Based Syst 2007;20:127–33.

[10] Ratle A, Sebag M. Grammar-guided genetic programming and dimensional consistency: application to non-parametric identification in mechanics. Appl Soft Comput 2001;1(1):105–18.

[11] Pérez JL, Cladera A, Rabuñal JR, Martínez-Abella F. Optimal adjustment of EC-2 shear formulation for concrete elements without web reinforcement using genetic programming. Eng Struct 2010;32(11):3452–66.

[12] Vecchio FJ, Collins MP. The modified compression field theory for reinforced concrete elements subjected to shear. ACI Struct J 1986;86(2):219–31.

[13] Darwin CR. On the origin of species by means of natural selection. London: John Murray; 1859.

[14] Holland JH. Adaptation in natural and artificial systems. Michigan: University of Michigan Press; 1975.

[15] Sette S, Boullart L. Genetic programming: principles and applications. Eng Appl Artif Intell 2001;14:727–36.

[16] Shaw D, Miles J, Gray A. Genetic programming within civil engineering. In: Organisation of the adaptive computing in design and manufacture 2004 conference, Engineers House, Clifton, Bristol, UK; 2004.

[17] Pérez JL. Metodología para orientar procesos de extracción de conocimiento basados en Computación Evolutiva. Aplicación al desarrollo de modelos y formulaciones en el ámbito del hormigón estructural. PhD thesis, Department of Information and Communication Technologies, University of A Coruña; 2010.

[18] Rodríguez-Vazquez K, Oliver-Morales C. Function approximation by means of multi-branches genetic programming. In: Maarten Keijzer, editor. Late breaking papers at the 2004 genetic and evolutionary computation conference, Seattle, Washington, USA; 2004.

[19] Cantú-Paz EK, Kamath C. An empirical comparison of combinations of evolutionary algorithms and neural networks for classification problems. IEEE Trans Syst, Man Cybernet Part B – Cybernet 2005;35(5):915–27.

[20] Dietterich TG. Approximate statistical tests for comparing supervised classification learning algorithms. Neural Comput 1998;10(7):1895–924.

[21] ASCE-ACI Committee 426. The shear strength of reinforced concrete members. J Struct Div 1973; 99(6): 1091–1187.

[22] Eurocode 2. Design of concrete structures part I: general rules and rules for buildings. European Committee for Standardization, Brussels, EC-2; 2002.

[23] ACI Committee 318. Building code requirements for structural concrete (ACI 318-05) and commentary (ACI 318R-05), USA; 2005.

[24] Kani GNJ. The riddle of shear failure and its solution. ACI J 1964;61(2):441–67.

[25] Comisión Permanente de Hormigón. Code on structural concrete EHE-08, Ministerio de Fomento, 2008. Available in English in http://www.fomento.gob.es/MFOM/LANG_CASTELLANO/ORGANOS_COLEGIADOS/CPH/instrucciones/EHE08INGLES/.

[26] Collins MP, Bentz EG, Sherwood EG, Xie L. An adequate theory for the shear strength of reinforced concrete structures. In: Morley symposium on concrete plasticity and its application. University of Cambridge: Cambridge; 2008.

[27] Collins MP. A evaluation of shear design procedures for concrete structures. A report prepared for the CSA technical committee on reinforced concrete design, Canada; 2001.

[28] Walraven JC. Fundamental analysis of aggregate interlock. J Struct Div Proc ASCE 1981;107(ST11):2245–70.

[29] Bentz EC. Sectional analysis of reinforced concrete members. PhD thesis, Department of Civil Engineering, University of Toronto; 2000.