Structural Engineering and Mechanics, Vol. 36, No. 2 (2010) 225-241
DOI: http://dx.doi.org/10.12989/sem.2010.36.2.225

Modeling of compressive strength of HPC mixes using a combined algorithm of genetic programming and orthogonal least squares

S.M. Mousavi[1], A.H. Gandomi*[2], A.H. Alavi[3a] and M. Vesalimahmood[4]

[1]Department of Civil Engineering, Sharif University of Technology, Tehran, Iran
[2]Department of Civil Engineering, Tafresh University, Tafresh, Iran
[3]School of Civil Engineering, Iran University of Science and Technology, Tehran, Iran
[4]School of Mathematics, Iran University of Science and Technology, Tehran, Iran

(Received December 2, 2009, Accepted June 17, 2010)

Abstract. In this study, a hybrid search algorithm combining genetic programming with orthogonal least squares (GP/OLS) is utilized to generate prediction models for compressive strength of high performance concrete (HPC) mixes. The GP/OLS models are developed based on a comprehensive database containing 1133 experimental test results obtained from previously published papers. A multiple least squares regression (LSR) analysis is performed to benchmark the GP/OLS models. A subsequent parametric study is carried out to verify the validity of the models. The results indicate that the proposed models are effectively capable of evaluating the compressive strength of HPC mixes. The derived formulas are very simple, straightforward and provide an analysis tool accessible to practicing engineers.

Keywords: high performance concrete; genetic programming; orthogonal least square; compressive strength; formulation.

1. Introduction

High performance concrete (HPC) is a class of concretes that provides superior performance than the conventional types. HPC is defined by the American concrete institute (ACI) as a concrete that meets special combinations of performance and uniformity requirements. These characteristics cannot always be achieved routinely using conventional constituents and normal mixing, placing and curing practices. A comprehensive definition for HPC is presented by Goodspeed *et al.* (1996) as "… concrete that attains mechanical, durability or construct ability properties exceeding those of normal concrete.". HPC is being extensively used in many countries of the world since it allows for the design of more elegant and lighter structural elements. HPC can provide increased durability which effectively increases service life and reduces maintenance. Decisions must be made for determining the HPC mix properties to meet the specified performance criteria at a reasonable cost while using locally available materials. This will require more trial mix batches and testing than are

*Corresponding author, Lecturer. E-mail: a.h.gandomi@gmail.com
[a]E-mail: ah_alavi@hotmail.com

necessary with the conventional concrete. In addition to the basic ingredients in the conventional concrete, the making of HPC needs to incorporate supplementary cementitious materials (e.g., fly ash and blast furnace slag) and chemical admixture (e.g., superplasticizer) (Domone and Soutsos 1994, Yeh 1998).

A key property of HPC mixes is their compression strength. The significance the compression strength in concrete technology is obvious. Developing accurate prediction models for the compression strength of HPC leads to saving costs, time and generating a successful concrete mixture. Empirical modeling based on statistical regression techniques may be used for this purpose (e.g., Yeh 1998, 2007). The regression-based analyses can have large uncertainties. Their major drawbacks pertain to idealization of complex processes, approximation and averaging widely varying prototype conditions. On the other hand, the current empirical equations presented in the codes and standards for estimating the compressive strength and workability are based on tests of concrete without supplementary cementitious materials. Hence, the validity of these relationships for concrete with supplementary cementitious materials (fly ash, blast furnace slag, etc.) should be investigated.

By extending developments in computational software and hardware, several alternative computer-aided data mining approaches have been developed. The idea is that a pattern recognition system learns adaptively from experience and extracts various discriminators. Artificial neural networks (ANNs) are the most widely used pattern recognition procedures. ANNs have been employed to assess different characteristics of concretes many times such as prediction of cement degree of hydration (Basma *et al.* 1999), concrete durability (Jepsen 2002), slump model (Yeh 2006a, 2007), and concrete mix proportion design algorithm (Ji *et al.* 2006). Also, many researchers applied ANNs to predict the compressive strength and slump flow of HPC mixes (Kasperkiewicz *et al.* 1995, Yeh 1998, Raghu Prasad *et al.* 2009). Rajasekaran and Amalraj (2002) and Rajasekaran *et al.* (2002) built empirical models for the strength of HPC mixes using sequential learning neural network (SLNN). In this connection, Rajasekaran and Lavanya (2007) utilized a wavelet neural network (WNN) method proposed by Salajegheh and Ali (2005). Despite the acceptable performance of ANNs, they do not usually give a definite function to calculate the outcome using the input values. Hence, they do not provide a better understanding of the nature of the derived relationships. The ANN approach is appropriate to be used as a part of a computer program and is not mostly suitable for practical calculations.

Genetic programming (GP) (Koza 1992, Banzhaf *et al.* 1998) is another alternative approach for behavior modeling of civil engineering problems. GP is a developing subarea of evolutionary algorithms (EAs) inspired from Darwin's evolution theory. The main advantage of the GP-based approaches is their ability to generate prediction equations without assuming prior form of the relationship. The developed equations can be easily manipulated in practical circumstances. GP was introduced by Koza (1992) as an extension of genetic algorithms (GAs). In GP, solutions are represented as tree structures and expressed in the functional programming language (Koza 1992). GP and its variants has successfully been applied to various kinds of civil engineering tasks (Johari *et al.* 2006, Alavi *et al*. 2010, Gandomi *et al.* 2010a,b).

Orthogonal least squares (OLS) algorithm (Billings *et al.* 1998, Chen *et al.* 1989) is an effective algorithm to determine which terms are significant in a linear-in-parameters model. The OLS algorithm introduces the error reduction ratio, which is a measure of the decrease in the variance of output by a given term. Madár *et al.* (2005a) combined GP and OLS to make a hybrid algorithm with better efficiency. It was shown that introducing this strategy into the GP process results in more robust and interpretable models (Madár *et al.* 2005a). GP/OLS is based on the data alone to determine the structure and parameters of the model. This technique has rarely been applied to the civil engineering problems (Gandomi and Alavi 2010, Gandomi *et al.* 2010c). The GP/OLS approach can be useful in deriving empirical models for characterizing the compressive strength behavior by directly extracting the knowledge contained in the experimental data.

The main purpose of this paper is to utilize GP/OLS to generate linear-in-parameters prediction models for the compressive strength of HPC mixes. Least square regression models are established to benchmark the proposed models. A reliable database of previously published compressive strength of HPC test results is utilized to develop the models.

2. Genetic programming

GP is a symbolic optimization technique that creates computer programs to solve a problem using the principle of Darwinian natural selection (Koza 1992). GP may generally be defined as a supervised machine learning technique that searches a program space instead of a data space (Banzhaf *et al.* 1998). In GP, a random population of individuals (computer programs) is created to achieve high diversity. The symbolic optimization algorithms present the potential solutions by structural ordering of several symbols. A population member in GP is a hierarchically structured tree comprising functions and terminals. The functions and terminals are selected from a set of functions and a set of terminals. For example, the function set F can contain the basic arithmetic operations (+, −, ×, /, etc.), Boolean logic functions (AND, OR, NOT, etc.), or any other mathematical functions. The terminal set T contains the arguments for the functions and can consist of numerical constants, logical constants, variables, etc. The functions and terminals are chosen randomly and constructed together to form a computer model in a tree-like structure with a root point with branches extending from each function and ending in a terminal. An example of a simple tree representation of a GP model is illustrated in Fig. 1.

The creation of the initial population is a blind random search for solutions in a large space of possible solutions. Once a population of models has been randomly created, the GP algorithm evaluates the individuals, selects individuals for reproduction, reproduction, and generates new individuals by mutation, crossover, and direct reproduction (Koza 1992).

During the crossover procedure, a point on a branch of each solution (program) is randomly

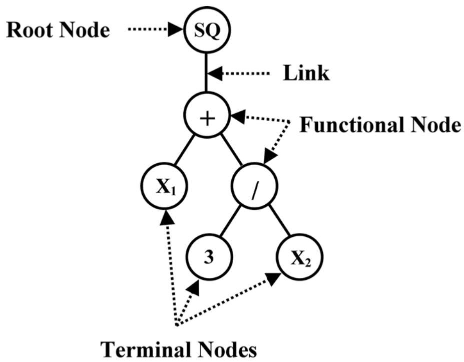

Fig. 1 The tree representation of a GP model $(X_1 + 3/X_2)^2$

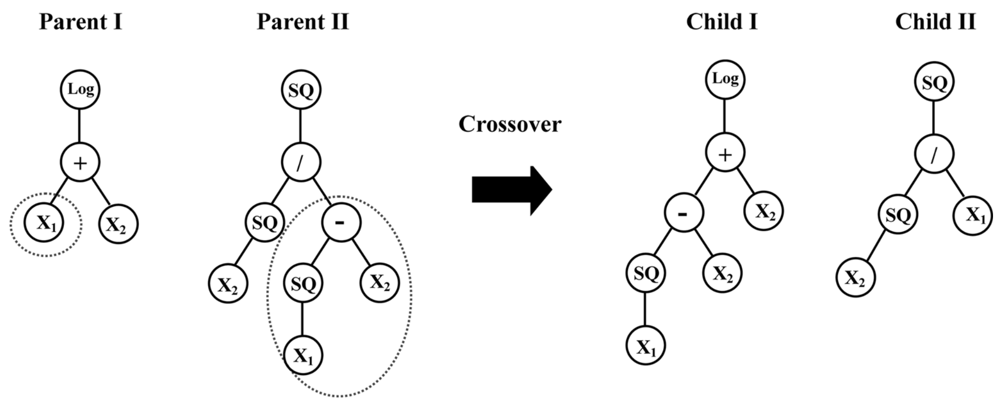

Fig. 2 Typical crossover operation in genetic programming

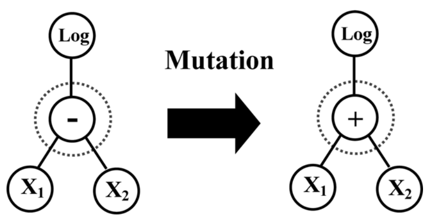

Fig. 3 Typical mutation operation in genetic programming

selected and the set of terminals and/or functions from each program are then swapped to create two new programs (see Fig. 2). The evolutionary process continues by evaluating the fitness of the new population and starting a new round of reproduction and crossover. During this process, the GP algorithm occasionally selects a function or terminal at random from a model and mutates it (see Fig. 3). The best program that appeared in any generation, the best-so-far solution, defines the output of the GP algorithm (Koza 1992).

2.1 Genetic programming for linear-in-parameters models

In general, GP creates not only nonlinear models but also linear-in-parameters models. In order to avoid parameter models, the parameters must be removed from the set of terminals. That is, it contains only variables: $T = (x_0(k), ..., x_i(k)\}$, where $x_i(k)$ denotes the ith repressor variable. Hence, a population member represents only F_i nonlinear functions (Pearson 2003). The parameters are assigned to the model after "extracting" the F_i function terms from the tree, and determined using a least square (LS) algorithm (Reeves 1997). A simple technique for the decomposition of the tree into function terms can be used. The sub-trees, representing the F_i function terms, are determined by decomposing the tree starting from the root as far as reaching nonlinear nodes (nodes which are not "+" or "–"). As can be seen in Fig. 4, the root node is a "+" operator; therefore, it is possible to decompose the tree into two sub-trees of "A" and "B". The root node of the "A" tree is a new a linear operator; therefore, it can be decomposed into "C" and "D" trees. As the root node of the "B" tree is a nonlinear node (/), it cannot be decomposed. The root nodes of "C" and "D" trees are also nonlinear. Consequently, the final decomposition procedure results in three sub-trees: "B", "C", and

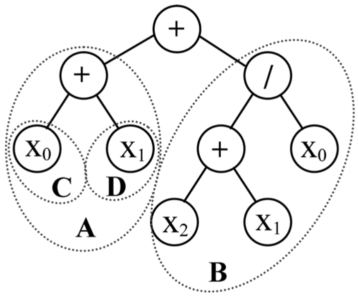

Fig. 4 Decomposition of a tree to function terms (Madár *et al.* 2004)

"D". According to the results of the decomposition, it is possible to assign parameters to the functional terms represented by the obtained sub-trees. The resulted linear-in-parameters model for this example is y: $p_0 + p_1(x_2 + x_1)/x_0 + p_2x_0 + p_3x_1$.

GP can be used for selecting from special model classes, such as a polynomial model. To achieve it, the set of operators must be restricted and some simple syntactic rules must be introduced. For instance, if the set of operators is defined as F = {×, +} and there is a syntactic rule that exchanges the internal nodes that are below a "×"-type internal nodes to "×"-type nodes, GP will generate only polynomial models (Koza 1992, Madár *et al.* 2004).

2.2 Orthogonal least squares algorithm

The great advantage of using linear-in-parameter models is that the LS method can be used for identifying the model parameters. This is much less computationally demanding than other nonlinear optimization algorithms since the optimal $p = [p_1,..., p_m]^T$ parameter vector can analytically be calculated

$$p = (U^{-1}U)^T U_y \tag{1}$$

in which $y = [y(1),..., y(N)]^T$ is the measured output vector and the U regression matrix is

$$U = \begin{pmatrix} U_1(x(1)) & \dots & U_M(x(1)) \\ \vdots & \ddots & \vdots \\ U_1(x(N)) & \dots & U_M(x(N)) \end{pmatrix} \tag{2}$$

The OLS algorithm (Billings *et al.* 1988, Chen *et al.* 1989) is an effective algorithm for determining which terms are significant in a linear-in-parameters model. The OLS technique introduces the error reduction ratio (*err*), which is a measure of the decrease in the variance of output by a given term. The matrix form corresponding to the linear-in-parameters model is

$$y = U_p + e \tag{3}$$

where the U is the regression matrix, p is the parameter vector, and e is the error vector. The OLS method transforms the columns of the U matrix into a set of orthogonal basis vectors to inspect the individual contributions of each term (Cao *et al.* 1999). It is assumed in the OLS algorithm that the regression matrix U can be orthogonally decomposed as $U = WA$, where A is a M by M upper triangular matrix (i.e., $A_{ij} = 0$ if $i > j$). W is a N by M matrix with orthogonal columns in the sense that $WTW = D$ is a diagonal matrix (N is the length of the y vector and M is the number of repressors). After this decomposition, the OLS auxiliary parameter vector g can be calculate as

$$g = D^{-1}W^T y \tag{4}$$

where g_i represents the corresponding element of the OLS solution vector. The output variance $(y^T y)/N$ can be described as

$$y^T y = \sum_{i=1}^{M} g_i^2 w_i^T w_i + e^T e \tag{5}$$

Therefore, the error reduction ratio $[err]_i$ of the U_i term can be expressed as

$$[err]_i = \frac{g_i^2 w_i^T w}{y^T y} \tag{6}$$

This ratio offers a simple mean for order and selects the model terms of a linear-in-parameters model on the basis of their contribution to the performance of the model.

2.3 Hybrid genetic programming-orthogonal least squares algorithm

The application of OLS in the GP algorithm leads to significant improvements in the performance of GP. The main feature of this hybrid approach is to transform the trees to simpler trees which are more transparent, but their accuracies are close to the original trees. In this coupled algorithm, GP generates a lot of potential solutions in the form of a tree structure during the GP operation. These trees may have better and worse terms (sub-trees) that contribute more or less to the accuracy of the model represented by the tree. OLS is used to estimate the contribution of the branches of the tree to the accuracy of the model, whereas, using the OLS, one can select the less significant terms in a linear regression problem. According to this strategy, terms (sub-trees) having the smallest error reduction ratio are eliminated from the tree (Pearson 2003). This "tree pruning" approach is realized in every fitness evaluation before the calculation of the fitness values of the trees. Since GP works with the tree structure, the further goal is to preserve the original structure of the trees as far as it possible. The GP/OLS method always guarantees that the elimination of one or more function terms of the model can be done by pruning the corresponding sub-trees, so there is no need for structural rearrangement of the tree after this operation. The way the GP/OLS method works on its basis is simply demonstrated in Fig. 5. Assume that the function which must be identified is $y(x) = 0.8u(x-1)^2 + 1.2y(x-1) - 0.9y(x-2) - 0.2$. As can be seen in Fig. 5, the GP algorithm finds a

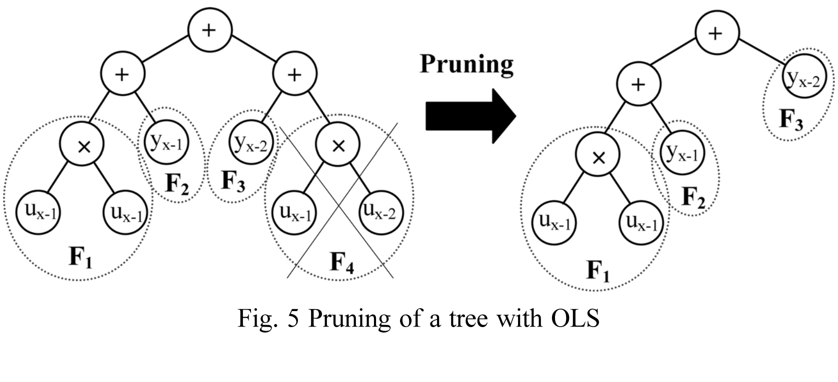

Fig. 5 Pruning of a tree with OLS

solution with four terms: u (x − 1)^2, y (x − 1), y (x − 2), u (x − 1) × u (x − 2). Based on the OLS algorithm, the sub-tree with the least error reduction ratio ($F_4 = u\ (x-1) \times u\ (x-2)$) is eliminated from the tree. Subsequently, the error reduction ratios and mean square error values (and model parameters) are calculated again. The new model (after pruning) may have a higher mean square error but it obviously has a more adequate structure.

3. Modeling of compressive strength of HPC mixes

The performance characteristics of HPC are major concerns in construction of civil engineering applications. The enhanced performance characteristics of HPC are generally achieved by addition of various cementitious materials and chemical and mineral admixtures to the conventional concrete mix designs. Advances in recent years have been assisted by the use and understanding of chemical admixtures, notably superplasticizers, and cement replacement materials, notably fly ash, blast furnace slag, etc. The use of fly ash and slag plays an important role in contributing to a better workability and low slump loss rates of HPC. This is due to the mutual containment with surface lubrication and the ball-bearing effects among the fly ash and micro fine materials. In many cases, there is also the economic benefit of the price differential between cement and the supplementary cementitious material. Additionally, partial replacement of cement nearly always allows a significant reduction in the dosage of the superplasticizer, which is a particularly expensive ingredient (Yeh 1998).

In its current state, the behavior modeling of the compressive strength of HPC containing these additives is a difficult task. In this study, the GP/OLS approach was utilized to obtain meaningful relationships between the compressive strength of HPC mixes and the influencing variables. The predictor variables were chosen on the basis of an extensive trial study and literature review (Yeh 1998, Chen 2003, Yeh 2006a, b, Chen and Wang 2010, Yeh and Lien 2009). After developing and controlling several models with different combinations of the input parameters, two models were selected and presented as the optimal models. The compressive strength (σ) formulations were considered to be as follows

$$\sigma_I = f\left(K, \frac{CA}{FA}, Ln(A)\right) \tag{7}$$

and

$$\sigma_{II} = f\,(K, Ln(A)) \tag{8}$$

where,

K: Ratio of water and superplasticizer summation to binder ($(W+S)/B$)
B: Binder content ($C+BF+F$)
W (kg/m^3): Water content
C (kg/m^3): Cement content
BF (kg/m^3): Blast furnace slag content
F (kg/m^3): Fly ash content
S (kg/m^3): Superplasticizer content
CA (kg/m^3): Coarse aggregate content
FA (kg/m^3): Fine aggregate content
A (day): Age of specimens

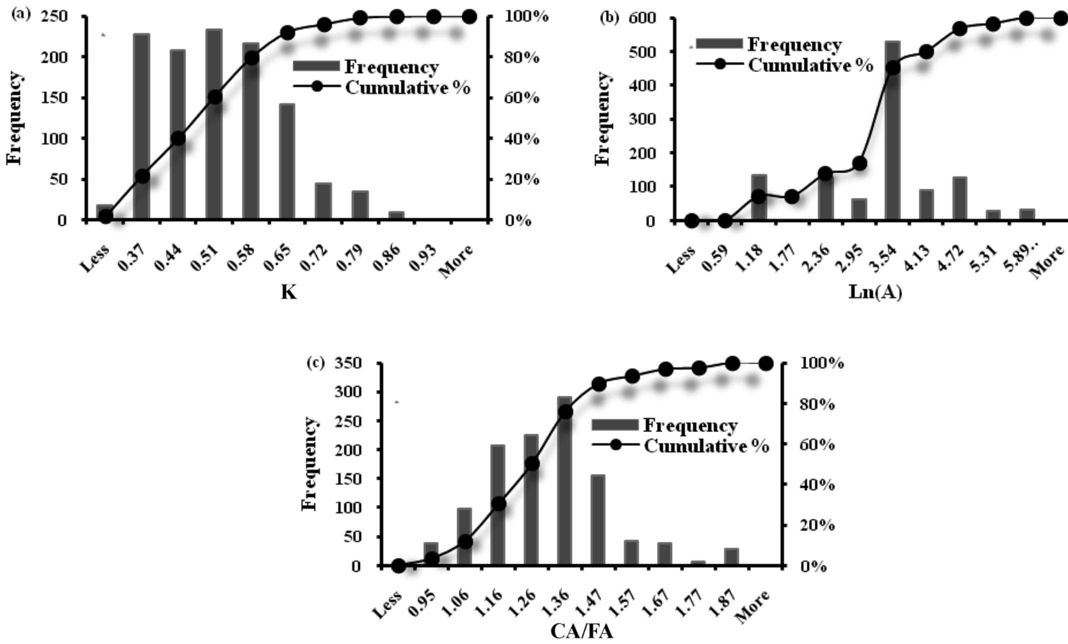

Fig. 6 Histograms of the variables used in the model development

Table 1 Descriptive statistics of the variables used in the model development

Variable	Mean	Standard Deviation	Sample Variance	Kurtosis	Skewness	Range	Min	Max	Confidence (95%)
Water (%)	7.8515	1.1106	1.2335	0.0746	0.2561	6.0831	5.1390	11.2222	0.0647
Cement (%)	11.7824	4.2725	18.2542	-0.5786	0.4731	18.0591	4.4815	22.5406	0.2490
Blast Furnace Slag (%)	3.1886	3.6149	13.0672	-0.5435	0.7555	15.0339	0.0000	15.0339	0.2107
Fly Ash (%)	2.6965	3.0866	9.5272	-0.8526	0.6297	11.2652	0.0000	11.2652	0.1799
Super-plasticizer (%)	0.2722	0.2429	0.0590	1.0700	0.7403	1.3149	0.0000	1.3149	0.0142
Coarse Aggregate (%)	41.2553	3.2501	10.5634	-0.4130	-0.3986	16.2504	31.7342	47.9846	0.1895
Fine Aggregate (%)	32.9535	3.2951	10.8577	0.0455	-0.1940	16.6176	24.7971	41.4147	0.1921
Age (day)	44.0565	60.4413	3653.154	13.8117	3.4696	364.0000	1.0000	365.0000	3.5232
Compressive Strength (MPa)	35.8380	16.1005	259.2264	-0.1564	0.4224	80.2674	2.3318	82.5992	0.9385

3.1 Experimental database

A reliable database was obtained from the literature to develop the models. The database contains 1133 compressive strength (σ) of HPC test results presented by Yeh (2006a, b). The database includes measurements of water (W), cement (C), blast furnace slag (BF), fly ash (F), superplasticizer (S), coarse aggregate (CA), fine aggregate (FA), age of specimens (A) and compressive strength (σ) of HPC mixes. To visualize the distribution of the samples, the data are presented by frequency histograms (Fig. 6). The descriptive statistics of the variables used are given in Table 1.

For the GP/OLS analysis, the database was randomly divided into learning, validation and testing subsets. The learning data were taken for training (genetic evolution).The validation data were used to specify the generalization capability of the models on data they did not train on (model selection). Thus, both of the learning and validation data were involved in the modeling process and were categorized into one group referred to as "training data". The testing data were further employed to measure the performance of the optimal models on data that played no role in building the models. In order to obtain a consistent data division, several combinations of the training and testing sets were considered. The selection was such that the maximum, minimum, mean and standard deviation of parameters were consistent in the training and testing data sets. For the analysis, 907 values (80%) of the data were taken for the training process (800 sets for learning and 107 sets for validation) and the rest of the values (20%) were used for the testing of the generalization capability of the models.

3.2 Performance measures

The best GP/OLS models were chosen on the basis of a multi-objective strategy as follows:

i. The simplicity of the model, although this was not a predominant factor.

ii. Providing the best fitness value on the learning set of data.

iii. Providing the best fitness value on a validation set of data.

The first objective can be controled by the user through the parameter settings (e.g., maximum tree depth). For the other objectives, the following objective function (OBJ) was constructed as a measure of how well the model predicted output agrees with the measured output. The selection of the best models was deduced by the minimization of the following function (Gandomi *et al.* 2010b)

$$OBJ = \left(\frac{No._{Learning} - No._{Validation}}{No._{Training}}\right)\frac{MAE_{Learning}}{R^2_{Learning}} + \frac{2No._{Validation}}{No._{Training}}\frac{MAE_{Validation}}{R^2_{Validation}} \tag{9}$$

where $No._{Learning}$, $No._{Validation}$ and $No._{Train}$ are respectively the number of learning, validation and training data. R and MAE are respectively correlation coefficient and mean absolute error given in the form of formulas as follows

$$R = \frac{\sum_{i=1}^{n}(h_i - \bar{h}_i)(t_i - \bar{t}_i)}{\sqrt{\sum_{i=1}^{n}(h_i - \bar{h}_i)^2 \sum_{i=1}^{n}(t_i - \bar{t}_i)^2}} \tag{10}$$

$$MAE = \frac{\sum_{i=1}^{n}|h_i - t_i|}{n} \tag{11}$$

in which h_i and t_i are respectively the experimental and calculated outputs for the *i*th output, $\bar{h}_i$ is the average of the experimental outputs, and n is the number of sample. It is well known that the R value alone is not a good indicator of prediction accuracy of a model. This is because that by shifting the output values of a model equally, the R value will not change. The constructed objective function takes into account the changes of R and MAE together. Higher R values and lower MAE values result in lower OBJ and, consequently, indicate a more precise model. In addition, the above function considers the effects of different data divisions for the learning and validation data.

Table 2 Parameter settings for the GP/OLS algorithm

Parameter	Settings
Function set	+, −, ×, /
Population size	1000
Maximum tree depth	10
Maximum number of evaluated individuals	2500
Generation	150
Type of selection	Roulette-wheel
Type of mutation	Point-mutation
Type of crossover	One-point (2 parents)
Type of replacement	Elitist
Probability of crossover	0.5
Probability of mutation	0.5
Probability of changing terminal-non-terminal nodes (vice versa) during mutation	0.25

3.3 Development of empirical models using GP/OLS

Three input parameters, K, CA/FA and Ln(A), were used to create the GP/OLS models. The GP/OLS method creates numerous, even millions of linear and nonlinear, randomly formed functions and selects the one that best fits the results for each run. Various parameters are involved in the GP/OLS predictive algorithm. The parameter selection significantly affects the generalization capability of the GP/OLS-based models. The parameter settings for the GP/OLS algorithm are shown in Table 2. In order to obtain simple and straightforward formulas, four basic arithmetic operators (+, −, ×, /) were utilized in the analysis. Also, the maximum tree depth was limited to 10. The number of programs in the population that GP/OLS will evolve is set by the population size. A run will take longer with a larger population size. The number of generation sets the number of levels the algorithm will use before the run terminates. The proper numbers of population and generation depend on the number of possible solutions and complexity of the problem. In this study, a fairly large number of initial population and generations were tested to find models with minimum error. The program was run until the runs terminated automatically. Mutation and crossover rate are the probabilities that an offspring will be subjected to the mutation and crossover operations, respectively. The values of both of these parameters for the optimal models were 50%. The other involved parameters values were selected based on some previously suggested values (Madár et al. 2005b) and also after a trial study. The GP/OLS approach was implemented using MATLAB® software.

3.3.1 GP/OLS-based formulation for compressive strength of HPC mixes

The GP/OLS-based formulations of the compressive strength (σ) in terms of K, CA/FA and Ln(A) are as given below

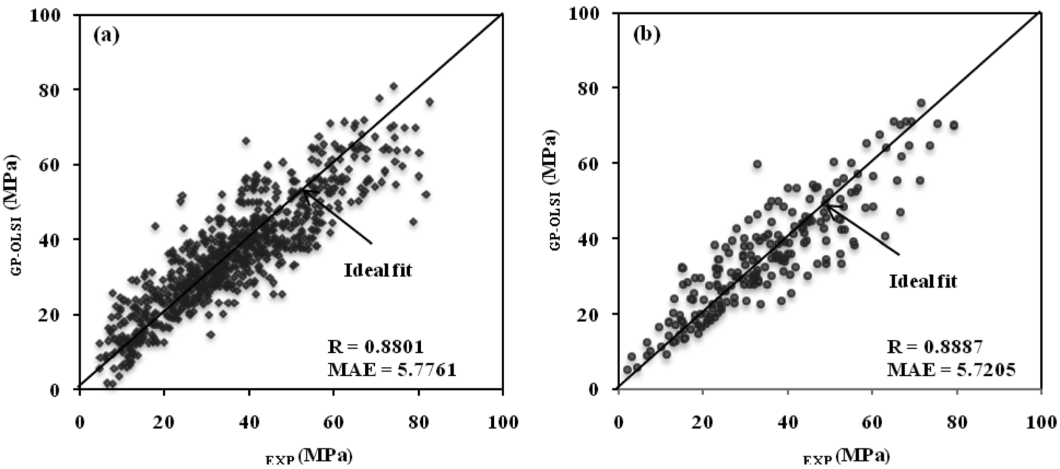

Fig. 7 Predicted versus experimental compressive strength values using the GP/OLS model (Model I) (a) training data, (b) testing data

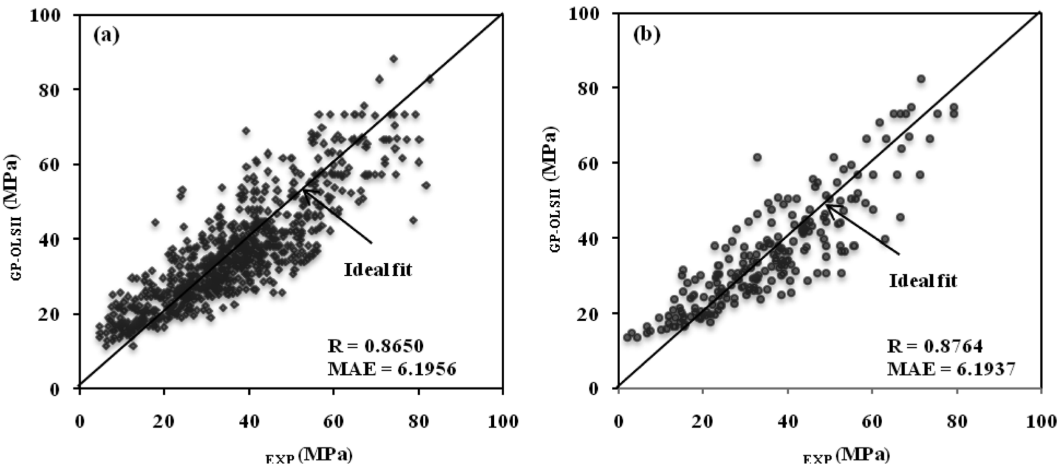

Fig. 8 Predicted versus experimental compressive strength values using the GP/OLS model (Model II) (a) training data, (b) testing data

$$\sigma_{GP/OLS,I}(\text{MPa}) = -36.36K - 3.94\left(\frac{CA}{FA} - \frac{Ln(A)}{K}\right) + 30.88 \tag{12}$$

$$\sigma_{GP/OLS,II}(\text{MPa}) = 6.46\left(\frac{Ln(A)}{K} - K\right) + 11.37 \tag{13}$$

Comparisons of the predicted versus experimental compressive strength of HPC are shown in Figs. 7 and 8. As it is seen, Eq. (12) with higher R and lower MAE values outperforms Eq. (13).

3.4 Development of empirical model using regression analysis

In the conventional material modeling process, regression analysis is an important tool for building a model. In this study, a multivariable least squares regression (LSR) (Ryan 1997) analysis was performed to have an idea about the predictive power of the GP/OLS technique, in comparison with a classical statistical approach. The LSR method is extensively used in regression analysis primarily because of its interesting nature. LSR minimizes the sum-of-squared residuals for each equation, accounting for any cross-equation restrictions on the parameters of the system. If there are no such restrictions, this technique is identical to estimating each equation using single-equation ordinary least squares. Eviews software package (Maravall and Gomez 2004) was used to perform the regression analysis. The LSR-based formulations of the compressive strength, σ, in terms of the influencing variables are as given below

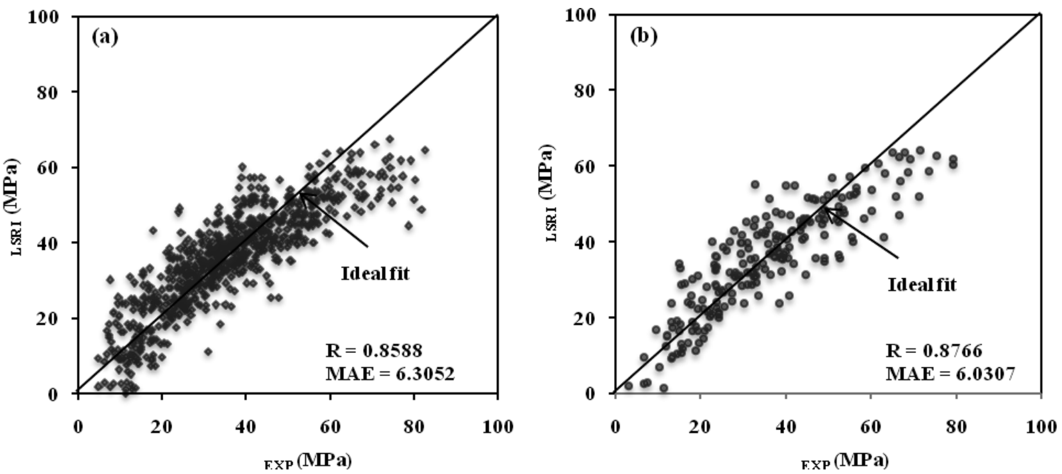

Fig. 9 Predicted versus experimental compressive strength values using the LSR model (Model I) (a) training data, (b) testing data

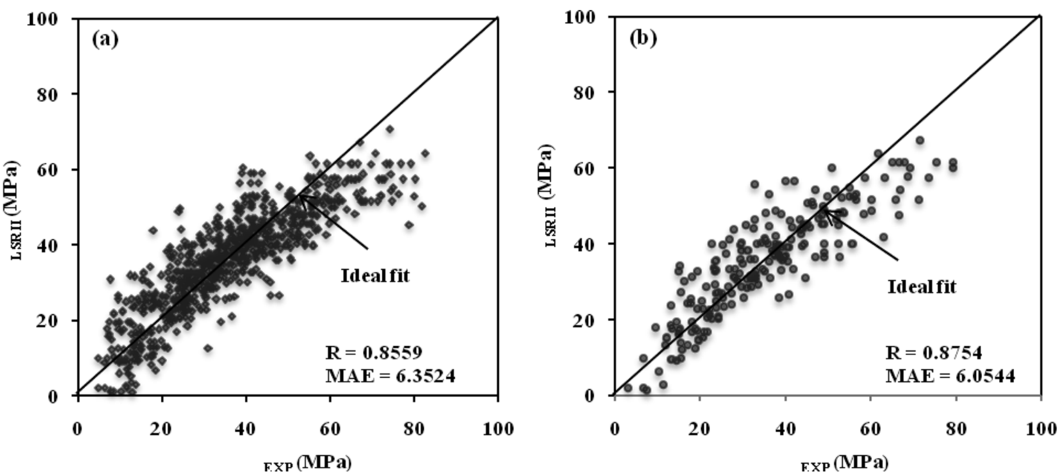

Fig. 10 Predicted versus experimental compressive strength values using the LSR model (Model II) (a) training data, (b) testing data

$$\sigma_{LSR,I}(\text{MPa}) = -90.5636K + 8.4978\frac{CA}{FA} - 6.0809Ln(A) + 60.1093 \quad (14)$$

$$\sigma_{LSR,II}(\text{MPa}) = -89.2885K + 8.42952Ln(A) + 51.9932 \quad (15)$$

Comparisons of the predicted versus experimental compressive strength of HPC are shown in Figs. 9 and 10. As it is seen, LSR, Model I with higher R and lower MAE values outperforms LSR, Model II.

4. Discussion

The results presented in Figs. 7-10 clearly indicate that the GP/OLS-based models outperform the LSR models. Comparisons of the compressive strength predictions obtained by these models on the whole of data are visualized in Fig. 11. No rational model to predict the compressive strength of HPC mixes has been developed yet that would encompass the influencing variables considered in this study. Therefore, it was not possible to conduct a comparative study between the results of this research and those of previous studies.

In addition to the reasonable performance of the GP/OLS-based equations, they are simple and can be used for routine design practice via hand calculations. Although the proposed regression-based models yield relatively good results for the current database, empirical modeling based on statistical regression techniques has significant limitations. Unlike GP/OLS, the regression-based analyses model the nature of the corresponding problem by a pre-defined equation, either linear or nonlinear.

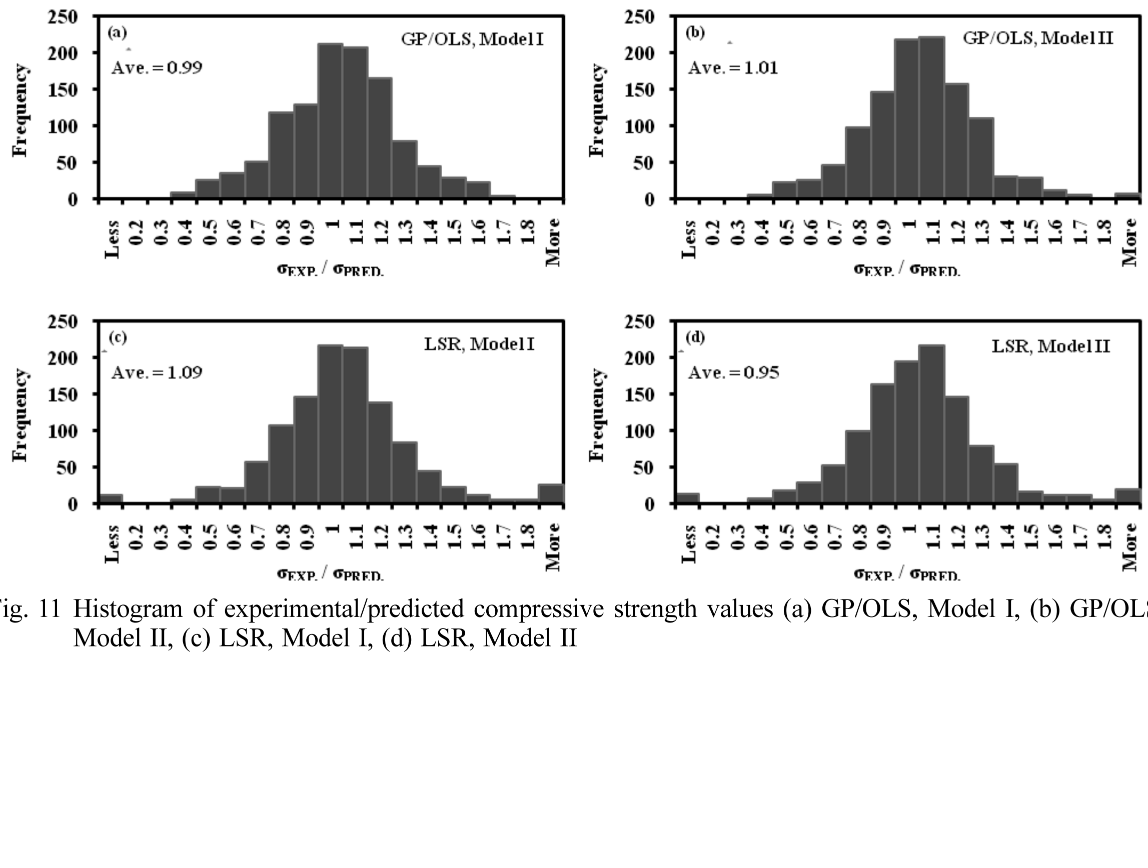

Fig. 11 Histogram of experimental/predicted compressive strength values (a) GP/OLS, Model I, (b) GP/OLS, Model II, (c) LSR, Model I, (d) LSR, Model II

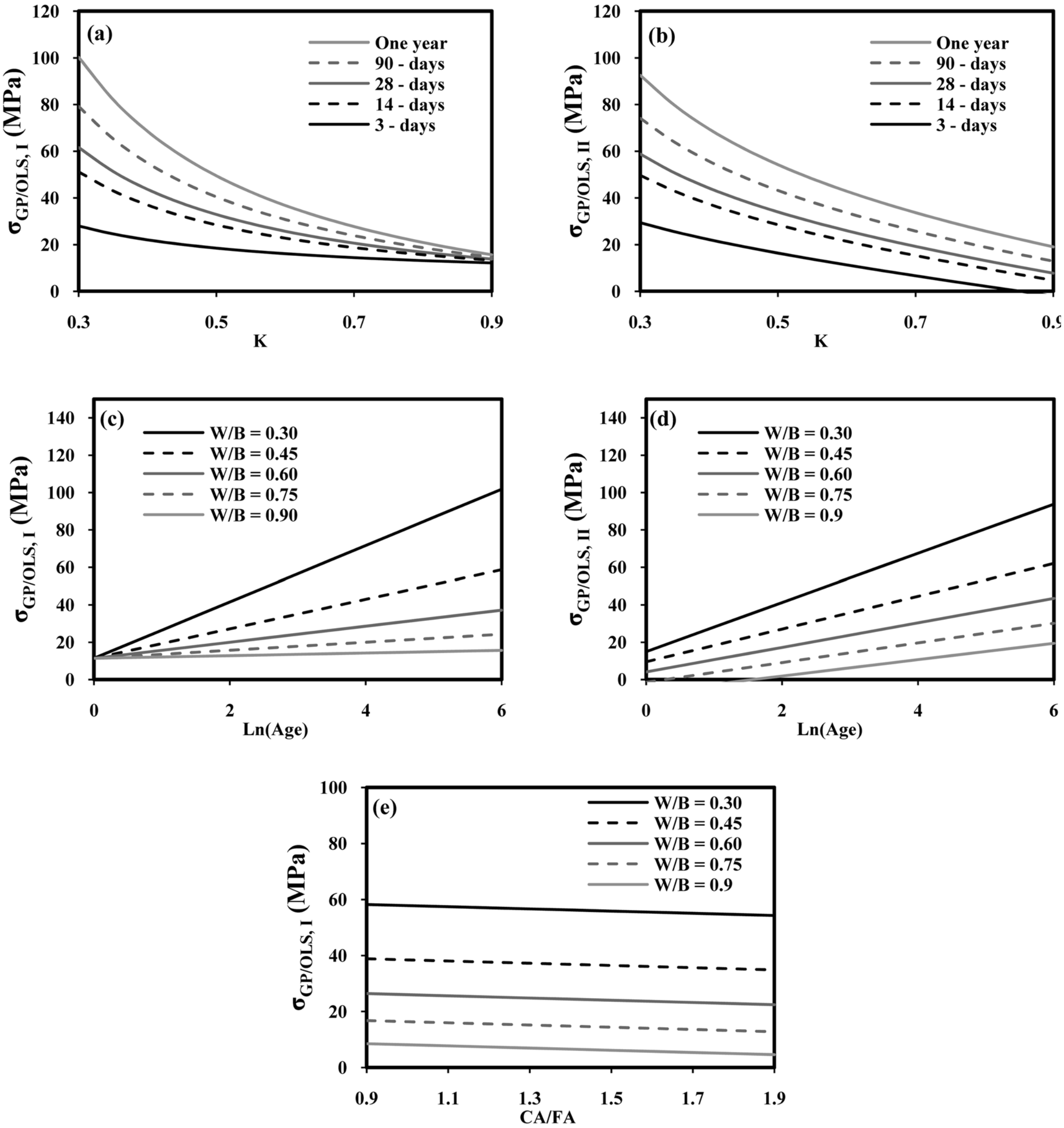

Fig. 12 Parametric analysis of the compressive strength in the best GP/OLS models

5. Parametric analysis

For further verification of the best GP/OLS models, a parametric analysis was performed in this study. The main goal was to find the effect of each parameter on the values of compressive strength of HPC. The methodology was based on the change of only one input variable at a time while the other input variable is kept constant at the average values of its entire data set. Fig. 12 presents the tendency of the compressive strength predictions to the variations of the ratio of water and superplasticizer to binder (K), ratio of coarse aggregate to fine aggregate (CA/FA) and natural logarithm of age of specimens (Ln (A)). As shown in this figure, σ continuously decreases due to increasing K and increases with increasing Ln (A) in both of the proposed models. This is an expected case from structural engineering viewpoint (Yeh 1998). The results of the parametric study also indicate that σ is not sensitive to the changes in the values of CA/FA. It was previously shown in Figs. 8 and 9 that incorporating the effect of CA/FA into the model development does not significantly improve the performance of the models.

6. Conclusions

In the study, a combined GP and OLS algorithm, called GP/OLS, was utilized to formulate the compressive strength of HPC mixes. Two simplified formulas were obtained for the compressive strength. A reliable database of previously published test results was used for the training and testing of the prediction models. The GP/OLS-based models were benchmarked against a multivariable linear regression model. It was observed that the GP/OLS-based models are capable of predicting the compressive strength of HPC mixtures with high accuracy. Due to nonlinearity in the compressive strength behavior, the nonlinear GP/OLS models produced considerably better outcomes over the developed linear regression-based models. The proposed models simultaneously take into account the role of several important factors representing the compressive strength behavior. The developed models can be used for routine design practice in that they were derived from tests on mixtures with a wide range of aggregate gradation and properties.

A major advantage of GP/OLS for determining the compressive strength lies in its powerful ability to model the mechanical behavior without any need to pre-defined equations. Another major distinction of the GP/OLS approach is that for a specific type of HPC mixture, the compressive strength can accurately be estimated without carrying out destructive, sophisticated and time-consuming laboratory tests. As more data become available, including those for other HPC mixtures and test conditions, the proposed models can be improved to make more accurate predictions for a wider range.

Acknowledgements

The authors are thankful to Engineer Parvin Arjmandi and Dr. Mohammad G Sahab (Tafresh University), and Prof. I-Cheng Yeh (Chung-Hua University) for their supports and providing the experimental database.

References

Alavi, A.H., Gandomi, A.H., Sahab, M.G. and Gandomi, M. (2010), "Multi expression programming: a new approach to formulation of soil classification", *Eng. Comput.*, **26**(2), 111-118.

Banzhaf, W., Nordin, P., Keller, R. and Francone, F. (1998), *Genetic Programming – An Introduction. on the Automatic Evolution of Computer Programs and Its Application,* San Francisco: (The Morgan Kaufmann Series in Artificial Intelligence), Morgan Kaufmann Publishers, Heidelberg.

Basma, A.A., Barakat, S. and Oraimi, S.A. (1999), "Prediction of cement degree of hydration using artificial neural networks", *Mater. J.*, **96**(2), 166-172.

Billings, S., Korenberg, M. and Chen, S. (1988), "Identification of nonlinear outputaffine systems using an orthogonal least-squares algorithm", *Int. J. Syst. Sci.*, **19**(8), 1559-1568.

Cao, H., Yu, J., Kang, L. and Chen, Y. (1999), "The kinetic evolutionary modelling of complex systems of chemical reactions", *Comput. Chem. Eng.*, **23**(1), 143-151.

Chen, L. (2003), "A study of applying macroevolutionary genetic programming to concrete strength estimation", *Expert. Syst. Appl.,* **17**(4), 290-294.

Chen, L. and Wang, T. (2010), "Modeling strength of high-performance concrete using an improved grammatical evolution combined with macro genetic algorithm", *J. Comput. Civil Eng.*, **24**(3), 281-288.

Chen, S., Billings, S. and Luo, W. (1989), "Orthogonal least squares methods and their application to non-linear system identification", *Int. J. Control*, **50**(5), 1873-1896.

Domone, P. and Soutsos, M. (1994), "An approach to the proportioning of high-strength concrete mixes", *Concrete Int*., **16**(10), 26-31.

Gandomi, A.H., Alavi, A.H. and Sahab, M.G. (2010a), "New formulation for compressive strength of CFRP Confined concrete cylinders using linear genetic programming", *Mater. Struct*., **43**(7), 963-983.

Gandomi, A.H., Alavi, A.H., Mirzahosseini, M.R. and Moghadas Nejad, F. (2010b), "Nonlinear genetic-based models for prediction of flow number of asphalt mixtures", *J. Mater. Civil Eng.* (ASCE), DOI: 10.1061/(ASCE)MT.1943-5533.0000154 (in press).

Gandomi, A.H. and Alavi, A.H. (2010c), "Hybridizing genetic programming with orthogonal least squares for modeling of soil liquefaction", *Computational Collective Intelligence and Hybrid Systems Concepts and Applications*, IGI Global Publishing (in press).

Gandomi, A.H., Alavi, A.H. and Arjmandi, P. (2010a), "Genetic programming and orthogonal least squares: a hybrid approach to modeling of compressive strength of CFRP-confined concrete cylinders" *J. Mech. Mater. Struct.* (in press).

Gandomi, A.H., Alavi, A.H., Sahab, M.G. and Arjmandi, P. (2010b), "Formulation of elastic modulus of concrete using linear genetic programming", *J. Mech. Sci. Tech.,* **24**(6), 1011-1017.

Goodspeed, C.H., Vanikar, S. and Cook, R. (1996), "High-performance concrete (HPC) defined for highway structures", *Concrete Int.*, **18**(2), 62-67.

Jepsen, M.T. (2002), "Predicting concrete durability by using artificial neural network", *Published in a Special NCR-publication*, ID. 5268.

Ji, T., Lin, T. and Lin, X. (2006), "A concrete mix proportion design algorithm based on artificial neural networks", *Cement Concrete Res.*, **36**(7), 1399-1408.

Johari, A., Habibagahi, G. and Ghahramani, A. (2006), "Prediction of soil-water characteristic curve using genetic programming", *J. Geotech. Geoenviron. Eng.,* **132**(5), 661-665.

Kasperkiewicz, J., Racz, J. and Dubrawski, A. (1995), "HPC strength prediction using artificial neural network", *J. Comput. Civil Eng.*, **9**(4), 279-284.

Koza, J.R. (1992), *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge.

Madár, J., Abonyi, J. and Szeifert, F. (2005a), "Genetic programming for the identification of nonlinear input-output models", *Indian Eng. Chem. Res.*, **44**(9), 3178-3186.

Madár, J., Abonyi, J. and Szeifert, F. (2005b), "Genetic programming for the identification of nonlinear input-output models", White Paper.

Madár, J., Abonyi, J. and Szeifert, F. (2004), "Genetic programming for system identification", *Proceedings of the Intelligent Systems Design and Applications (ISDA 2004) Conference*, Budapest, Hungary.

Maravall, A. and Gomez, V. (2004), EViews Software, Ver. 5, Quantitative Micro Software, LLC, Irvine CA.

Pearson, R.K. (2003), "Selecting nonlinear model structures for computer control", *J. Process Contr.*, **13**(1), 1-26.

Raghu Prasad, B.K., Eskandari, H. and Venkatarama Reddy, B.V. (2009), "Prediction of compressive strength of SCC and HPC with high volume fly ash using ANN", *Constr. Build. Mater.*, **23**(1), 117-128.

Rajasekaran, S. and Amalraj, R. (2002), "Predictions of design parameters in civil engineering problems using SLNN with a single hidden RBF neuron", *Comput. Struct.*, **80**(31), 2495-2505.

Rajasekaran, S. and Lavanya, S. (2007), "Hybridization of genetic algorithm with immune system for

optimization problems in structural engineering", *Struct. Multidiscip. O.*, **34**(5), 415-429.
Rajasekaran, S., Suresh, D. and Pai, G.A.V. (2002), "Application of sequential learning neural networks to civil engineering modeling problems", *Eng. Comput.,* **18**, 138-147.
Reeves, C.R. (1997), "Genetic algorithm for the operations research", *Inf. J. Comput.,* **9**, 231-250.
Ryan, T.P. (1997), *Modern Regression Methods*, Wiley, New York.
Salajegheh, E. and Ali, H. (2005), "Optimum design of structures against earthquake by wavelet neural network and filter banks", *Earthq. Eng. Struct. D.*, **34**(1), 67-82.
Yeh, I. and Lien, L. (2009), "Knowledge discovery of concrete material using genetic operation trees", *Expert. Syst. Appl.*, **36**, 5807-5812.
Yeh, I.C. (1998), "Modeling of strength of high-performance concrete using artificial neural networks", *Cement Concrete Res.*, **28**(12), 1797-808.
Yeh, I.C. (1998), "Modeling of strength of high-performance concrete using artificial neural networks", *Cement Concrete Res.*, **28**(12), 1797-1808.
Yeh, I.C. (2006a), "Exploring concrete slump model using artificial neural networks", *J. Comput. Civil Eng.*, **20**(3), 217-221.
Yeh, I.C. (2006b), "Analysis of strength of concrete using design of experiments and neural networks", *J. Mater. Civil Eng.*, **18**(4), 597-604.
Yeh, I.C. (2006c), "Generalization of strength versus water-cementations ratio relationship to age", *Cement Concrete Res.*, **36**(10), 1865-1873.
Yeh, I.C. (2007), "Modeling slump flow of concrete using second-order regressions and artificial neural networks", *Cement Concrete Comp.*, **29**, 474-480.