Computers and Concrete, *Vol. 23, No. 4 (2019) 255-265*
DOI: https://doi.org/10.12989/cac.2019.23.4.255

Knowledge-based learning for modeling concrete compressive strength using genetic programming

Hsing-Chih Tsai* and Min-Chih Liao[a]

Department of Civil and Construction Engineering, National Taiwan University of Science and Technology, #43, Sec. 4, Keelung Rd., Taipei, Taiwan, R.O.C.

(Received June 26, 2018, Revised March 20, 2019, Accepted March 28, 2019)

Abstract. The potential of using genetic programming to predict engineering data has caught the attention of researchers in recent years. The present paper utilized weighted genetic programming (WGP), a derivative model of genetic programming (GP), to model the compressive strength of concrete. The calculation results of Abrams' laws, which are used as the design codes for calculating the compressive strength of concrete, were treated as the inputs for the genetic programming model. Therefore, knowledge of the Abrams' laws, which is not a factor of influence on common data-based learning approaches, was considered to be a potential factor affecting genetic programming models. Significant outcomes of this work include: 1) the employed design codes positively affected the prediction accuracy of modeling the compressive strength of concrete; 2) a new equation was suggested to replace the design code for predicting concrete strength; and 3) common data-based learning approaches were evolved into knowledge-based learning approaches using historical data and design codes.

Keywords: genetic programming; concrete compressive strength; design codes; functional mapping

1. Introduction

Concrete is a complex material that is widely used due to its strong resistance to compression. This compression resistance gives concrete a wide range of applications in construction and a particularly important role in civil engineering (Güllü and Girisken 2013, Güllü 2015, Güllü 2016, Güllü *et al*. 2017d, Canakci *et al*. 2018, Güllü *et al*. 2019). Determining the strength profile of specific concrete specimens, which are composed of varying ratios of various materials, requires many concrete cylinder tests. Portland cement, water, fine aggregates, and coarse aggregates (the four basic ingredients) in combination with fly ash, blast furnace slag, superplasticizer, and / or other supplementary materials are essential for making high-strength concrete (Hossain *et al*. 2006). However, identifying the specific parameters of concrete is difficult. Soft-computing is an alternative approach to predicting strengths that has been shown to achieve high levels of accuracy (Parichatprecha and Nimityongskul 2009, Bilgehan Turgut 2010, Ozbay *et al*. 2010).

Neural networks (NNs) are the most commonly used machine-learning method for inference tasks, from which many NN derivatives have been developed and applied (Tran *et al*. 2007, Mehrjoo*et al*. 2008, Behzad *et al*. 2009, Tsai 2009, Tsai 2010, Ismail and Jeng 2011, Muhammad *et al*. 2015, Olofintoye *et al*. 2016, Sonebi *et al*. 2016, Hossain *et al*. 2017, Ongpeng *et al*. 2017, Patel *et al*. 2017, Saha and Prasad 2017). However, NNs have been characterized as black-box models due to the extremely large number of nodes and connections within their structures. Since it was first proposed by Koza (1992), genetic programming (GP) has garnered considerable attention due to its ability to model nonlinear relationships for input-output mappings without assuming prior form. GP is sometimes called a grey-box model due to its ability to generate prediction equations against black-box models. Baykasoglu *et al*. (2008) compared a promising set of GP approaches, including Multi Expression Programming (MEP) (Oltean and Dumitrescu 2002), Gene Expression Programming (GEP) (Ferreira 2001), and Linear Genetic Programming (LGP) (Bhattacharya *et al*. 2001). Notably, LGP was the most efficient algorithm in case studies of limestone strengths. Differences between these algorithms are rooted in the methodology that is utilized to generate a GP individual. A chromosome representation, a tree topology, and a linear string are used by MEP, GEP, and LGP, respectively. Several studies have utilized GP derivatives to examine construction industry problems. Baykasoglu *et al*. (2009) applied GEP to determine concrete strength, cost, and slump. Güllü applied GEP to handle various structural or geotechnical problem (Güllü 2012, 2013, 2014, 2017a, 2017b, Güllü and Fedakar 2017). Yeh and Lien (2009) developed the GP derivative genetic operation tree (GOT) to investigate concrete strength. Although, some of the formulas that are generated by MEP, GEP, LGP and GOT have coefficients, all of these coefficients are fixed constants (2008). Coefficient constants do not frequently appear in formulas that are programmed using any of these GP models, therefore Giustolisi and Savic (2006) argued that GP is not very

*Corresponding author, Associate Professor
E-mail: tsaihsingchih@gmail.com
[a]Assistant Professor
E-mail: minchih.liao@mail.ntust.edu.tw

ISSN: 1598-8198 (Print), 1598-818X (Online)

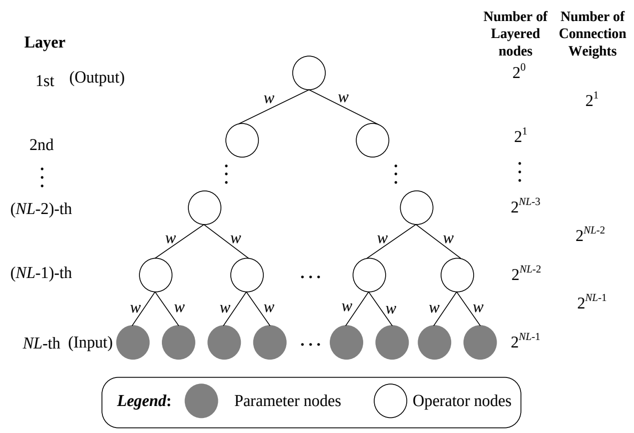

Fig. 1 Weighted genetic programming structure

powerful in finding constants. Consequently, Tsai (2011) proposed a weighted GP (WGP) to introduce weight coefficients into tree connections, generate a fully weighted formula, and provide coefficient constants for the obtained GP equations.

GP models obtain formulas against black-box models, with both grey-box and black-box models usually offering better estimation accuracies than white-box models (e.g., design codes). However, the knowledge inside the white-box models is no doubt significant to the engineering problem and was seldom found in previous grey-box models (Mousavi *et al*. 2012, Tsai and Lin 2011). The present paper attempts to integrate grey-box and white-box models in order to give consideration both to the good prediction accuracy of grey-box models and to the meaningful engineering properties of white-box models. In attempting this, the calculation results of design codes are considered as inputs for the GP model and take place possibly in the obtained GP formulas. Consequently, knowledge of the design codes is encapsulated in the resultant GP model in order to accomplish the knowledge-based learning model.

The remainder of the present paper is organized as follows. Section 2 presents the employed WGP model and associated candidate operators. Section 3 characterizes the details of developing knowledge-based WGP models for concrete compressive strength. Section 4 presents analytical results, comparisons, and discussions. Section 5 summarizes conclusions.

2. Genetic programming

GP is a subarea of evolutionary algorithms, which were inspired by Darwin's theory of evolution. GP, which is an extension of genetic algorithms (GAs), is defined as a supervised machine learning technique. Most GA operators may be implemented in GP executions. GP solutions are computer programs that are typically represented as tree structures and expressed as functional equations in order to describe input-output relationships.

2.1 Weighted genetic programming

Tsai (2011) introduced a weighted balance for tree-based GP to create weighted genetic programming (WGP). Weights are attached to all of the branches of the WGP tree structure in order to balance the impacts of the two front nodes (Fig. 1). The WGP uses parameter selection to adopt inputs from the bottom layer, executes operator selection to determine operators for nodes above the bottom layer, and then outputs functional programs from the top node. The parameter set (*PS*) includes all input parameters (*P*) and a unit parameter "1", which produces a constant for the branch. Users select the operator set (*OS*) for specific purposes such as generating polynomial functions. Thus, the present paper adopts *PS* and *OS* as

$$PS = \{1 \quad P_1 \quad P_2 \quad ... \quad P_{NI}\} \tag{1}$$

$$OS = \{T \quad N \quad S \quad B \quad + \quad \times \quad / \quad ^\wedge\} \tag{2}$$

where *NI* is the number of input parameters and each parameter selection *PS* selects the most suitable parameter from the *NI*+1 candidates. The first four *OS* operators are designed primarily to cut the tree topology, with the remaining four operators providing polynomial-like equations, as polynomials are a kind of mathematical expression that is frequently adopted to describe engineering problems. Details on the performance of the eight *OS* operators are provided in the following

$$y = f(w_i, w_j, x_i, x_j, x_{end}) = \begin{cases} x_{end} & ,OS = T \\ x_i & ,OS = N \\ w_i x_i & ,OS = S \\ x_i + w_i & ,OS = B \\ w_i x_i + w_j x_j & ,OS = + \\ w_i x_i \times w_j x_j & ,OS = \times \\ w_i x_i \,/\, w_j x_j & ,OS = / \\ w_i x_i^{\,p} & ,OS = ^\wedge \end{cases} \tag{3}$$

where nodal output y is the function of nodal values (x_i and x_j) of two front nodes, connection weights (w_i and w_j), and, occasionally, the most left-hand side branch end (x_{end}; Fig. 2); *T* operator is a terminate operator that directly adopts the most left-hand side branch end x_{end}; *N* operator is a next-layer operator that directly inherits x_i; *S* operator handles scaling for x_i; *B* operator tackles shifting for the x_i; and the last four operators deal with summation, multiplication, division, and power operations, respectively. In engineering, large values are infrequently adopted for exponents. Therefore, candidate exponents for the p of the "^" operator were considered [-2, -1.5, -1, -0.5, 0, 0.5, 1, 1.5, 2] (Berardi *et al*. 2008), as determined by the transformation of w_j.

2.2 Knowledge-based WGP learning model

Using the previously mentioned WGP model, a polynomial equation may be produced to describe the studied engineering problem. The obtained WGP polynomials are frequently compared with those obtained by black-box and white-box models without further hybridization of the models. WGP polynomials are typically

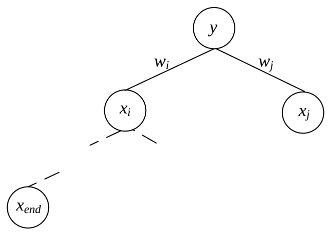

Fig. 2 Nodal value of a WGP node

superior to white-box models in terms of prediction accuracy, and WGP offers polynomial equations that are not obtainable using black-box models. Thus, the present paper aims to integrate the knowledge in white-box models into grey-box models in order to improve WGP learning. As soon as the target problem is ready to execute WGP learning, the calculated results of white-box models (design codes) should be available. The present paper suggests considering the calculation results of design codes as input parameters in the WGP model. Thus, Eq. (1) should be rewritten as

$$PS = \{1\ P_1\ P_2 ... P_j ... P_{NI}\ code_1 ... code_k\} \tag{4}$$

where *code* is the calculation result of a specific design code. When a particular design code is useful for improving the final accuracy of WGP learning, it may take place in the WGP equation and knowledge of the design code will be attached to the equation. Thus, the knowledge of white-box models will participate thoroughly in the process of grey-box learning.

3. Model development

3.1 Design codes for concrete strength

Abrams proposed a set of water-cement ratio formulas to calculate concrete compressive strength. Under fixed curing age and temperature conditions, Abrams suggested a relationship between concrete strength and water-cement ratio that may be represented as

$$f_c' = \frac{A}{B^z} \tag{5}$$

$$z = W / C \tag{6}$$

where *fc*' is concrete compressive strength; *A* and *B* are experimental constants that are determined by a given age; *z* is the water-cement ratio; *W* is water content; and *C* is cement content. Under normal temperature and moisture curing conditions, Abrams gave Portland cement concrete 7-day strength (fc'_7) and 28-day strength (fc'_{28}) as (Oluokun 1994)

$$f_c'_7 = \frac{63.45}{14^z} \text{ MPa} \tag{7}$$

$$f_c'_{28} = \frac{96.55}{8.2^z} \text{ MPa} \tag{8}$$

Under Abrams' law, concrete strength is a function of water cement ratio only. Materials such as fly ash, blast furnace slag, and superplasticizer, while absent in Abrams' formulas, are widely used today as concrete admixtures. Many studies have since been done to extend, modify, and generalize Abrams' formulas (Babu and Rao 1996, Popovics 1990, Nagaraj and Banu 1996).

3.2 Experimental database

Yeh (1998) created open-source concrete compressive strength datasets. These include 1,030 concrete cylinder specimens and use 8 quantitative input variables to estimate concrete strength. Among the 1,030 specimens, 126 are categorized for 7-day strength and 425 are categorized for 28-day concrete strength (Table 1). A total of 101 of the 126 and 340 of the 425, respectively, were selected at random as the training and testing datasets. The present paper aimed to model 7-day and 28-day concrete strength separately. Therefore, only the first seven factors (P_1-P_7 in Table 1) were treated as influenced variables.

3.3 Parameter settings for WGP models

MATLAB was employed in the present study because it is a powerful tool that incorporates various function sets, including genetic algorithm (GA). The MATLAB GA function was used in the present study to implement WGP learning. Initial individuals in a population are randomly generated in "doubleVector" types that vary between 0-1, containing parameter selections *PS*, operator selection *OS*, and weight coefficients. Each *PS* selects suitable parameters from 8 candidates, including the first 7 concrete parameters in Table 1 and the calculation results of Abrams' codes (*code*); each *OS* picks operators from the 8 candidates, as shown in Eq. (4); weight coefficients are transferred linearly into the range -10-10. Fitness is a major index that is used to evaluate individual status, with decreasing fitness values correlated with increasing degrees of achievement of the model objective. In the present study, the fitness function was directly set as the inverse of the training root mean square error (*RMSE*), with larger fitness values indicating a healthier individual. The index of the *RMSE* and coefficient of determination (R^2) were used to evaluate model performance, which were given in the form of the following relationships

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(h_i - t_i)^2}{n}} \tag{9}$$

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

in which h_i and t_i are, respectively, the actual and predicted outputs for the *i*th item; $\bar{h}$ is the average of the actual outputs (Babanajad *et al*. 2013, Fiore *et al*. 2016). The population size chosen for the present study was 200 and 5,000 iterations, respectively, by trial and error.

Table 1 Input and output parameters for 7-day concrete strength

Variables	7-day				28-day			
	Min.	Max.	Mean	SD	Min.	Max.	Mean	SD
P_1: Cement (kg/m^3)	102	540	313	105	102	540	265	105
P_2: Blast Furnace Slag (kg/m^3)	0	359	93.3	100	0	359	86.7	87.4
P_3: Fly Ash (kg/m^3)	0	141	13.5	36.5	0	200	63.3	65.8
P_4: Water (kg/m^3)	126	228	183	18.8	122	247	183	19.3
P_5: Superplasticizer (kg/m^3)	0	32.2	4.46	6.31	0	32.2	7.24	5.08
P_6: Coarse Aggregate (kg/m^3)	822	1134	984	81.4	801	1145	956	93.8
P_7: Fine Aggregate (kg/m^3)	594	993	768	81.8	594	993	764	73.1
P_8: Age (day)	7	7	7	0	28	28	28	0
S: Concrete Strength (MPa)	126				425			

Table 2 Knowledge-based WGP learning for 7-day concrete strength

# layers	Min.	Training	Testing	Training	Testing	Active # operator nodes	Active # parameter nodes
		RMSE (MPa)		R^2			
2	Avg.	8.42	8.77	68.4%	49.9%	1.00	1.40
	Best	7.88	8.69	72.5%	50.9%	1	2
3	Avg.	7.69	8.10	73.3%	56.8%	2.23	1.73
	Best	5.86	6.14	84.8%	75.4%	3	2
4	Avg.	7.26	7.77	76.1%	60.2%	3.73	2.20
	Best	5.54	6.24	86.3%	74.6%	5	3
5	Avg.	6.82	7.44	78.8%	63.4%	5.70	3.17
	Best	5.37	5.90	87.2%	77.3%	5	3
6	Avg.	6.44	7.36	80.8%	64.1%	9.33	5.10
	Best	5.26	6.04	87.7%	76.2%	13	7

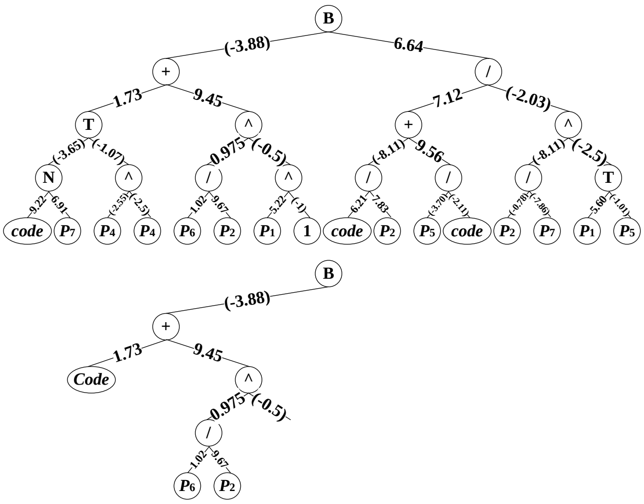

Fig. 3 Structures of the best run of a 5-layered WGP

4. Results and discussion

4.1 Knowledge-based WGP learning for concrete strength

Although WGP offers operators like T and N, which may be used to reduce the size of tree-based structures, a large WGP tree may be easy to obtain owing to the good prediction accuracy of WGP. However, the associated polynomial equation may be complicated, and the obtained result may be considered as over-trained. Therefore, the present paper performed and compared WGP from 2 to 6 layers, and each statistical result adopted 30 runs. Table 2 relates to the WGP estimation for 7-day concrete strength using tree structures of 2 to 6 layers. The average and best results are presented, with the best results determined by summing training and testing *RMSE*. In terms of the best *RMSE* results, it seems that the results obtained by 4-layered WGP already approximates those obtained by 6-layered WGP. In addition, WGP is able to generate accurate predictions without using all of the parameter and operator nodes. For instance, the best run of the 5-layered WGP (15 operator nodes and 16 parameter nodes) was structured as shown in Fig. 3. After WGP learning, the WGP tree was dramatically pruned, with the effects of B and T operators shown in Fig. 3. Finally, only 5 of the 15 operators, including the T operator in the third layer, remained active, and only 3 of the 16 parameters participated in WGP learning. This good situation shortened and simplified the final WGP polynomial significantly. In looking at the pruned WGP tree, as shown in Fig. 3, the best runs of the 5-layered WGP may be presented as the functional mapping from concrete parameter inputs to concrete strength. In the same manner, the 2- to 6-layered WGP polynomials for 7-day concrete strength may be formed as

$$O_2 = 1.67 \times code + 0.568 P_5 \tag{11}$$

$$O_3 = 1.61 \times code + 0.744 \sqrt{P_2} \tag{12}$$

$$O_4 = 1.84 \times code + 59.8 \frac{P_2}{P_6} - 4.27 \tag{13}$$

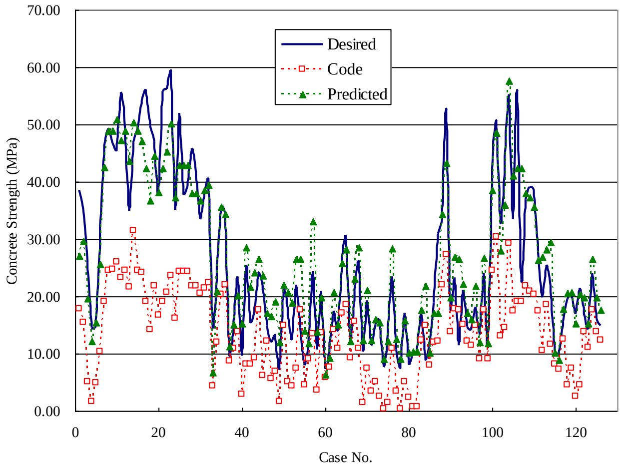

Fig. 4 Concrete strength of desired experiments, design codes, and S_7 predictions

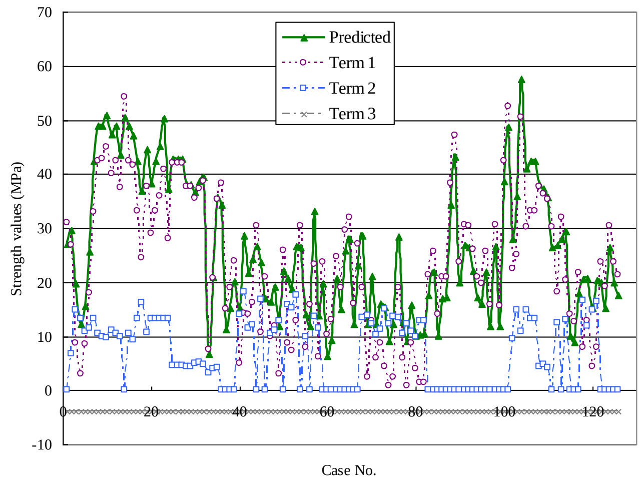

Fig. 5 Strength values of S_7

$$O_5 = 1.73 \times code + 29.5\sqrt{\frac{P_2}{P_6}} - 3.88 \quad (14)$$

$$O_6 = 1.82 \times code + 31.5\sqrt{\frac{P_2}{P_6}} + \frac{code}{0.344P_4 - 0.998} - 5.70 \quad (15)$$

in which, O is the output WGP equation, the sub-index of the O is the number of WGP tree layers, and *code* represents the $f_c{}'_7$. It is particular noteworthy that the *code* of Eq. (7) joins all of the 5 above equations, indicating that the knowledge of the design code indeed improves WGP learning in terms of improved prediction accuracy. Furthermore, the above 5 equations perform similarly, including the participation of *code*, the constant scaling factor around 1.7, and the impacts of P_2 and P_6. All of the above equations perform in the forms of the linear modes of *code*, which appear suitable for studying the improvements of *code*. Although the scaling factors of the linear equations are restricted to constants, a constant scaling seems to be the best choice for the linear models. The shifting terms of the linear equations are all dominated by P_2 and P_6, which provides evidence that the P_2 (blast furnace slag) and P_6 (coarse aggregate) affect primarily the early-age strength of the improvements of the Abrams' *code*. Notably, instead of the direct estimates for concrete strength commonly seen in various literatures, the effects of P_2 and P_6 focus on improving *code*.

Among Eqs. (11)-(15), the O_5 was selected as the final result for improving the 7-day *code* (using S_7 to represent the O_5 hereafter), especially due to good performance on testing *RMSE* evaluation. The first 101 cases in Fig. 4 are training results and remainders are testing patterns. As can be observed from Fig. 4, the S_7 greatly improve *code* to fit experiment values. In order to discuss the S_7 in more details, Fig. 5 shows the values of the three terms on the right-hand side of Eq. (14). The first term performs the scaling effects on *code* and affects the S_7 the most severely. The other two terms are combined to be treated as the shifting terms of *code*. The last one is a constant minus shifting term and the middle one is a parametric term function dominated by P_2 and P_6. The middle one sometime influences the S_7 a lot and is as significant as the first one. Special findings the S_7 different to other literatures may fall into improvements of *code* with effects of P_2 and P_6.

Knowledge-based WGP learning may be performed in a similar manner on 28-day concrete strength. Table 3 shows statistical results of 30 WGP runs from 2 to 6 layers. As the strength of 28-day concrete is higher than 7-day concrete, the RMSE prediction results for 28-day concrete strength were reasonably higher than those for 7-day concrete strength. However, even though still acceptable, the results

Table 3 Knowledge-based WGP learning for 28-day concrete strength

# layers	Min.	Training *RMSE* (MPa)	Testing *RMSE* (MPa)	Training R^2	Testing R^2	Active # operator nodes	Active # parameter nodes
2	Avg.	11.59	10.79	39.2%	40.1%	1.00	1.00
	Best	11.47	10.34	40.5%	45.0%	1	1
3	Avg.	10.53	9.57	49.8%	52.9%	2.00	1.03
	Best	9.07	8.10	62.8%	66.3%	2	2
4	Avg.	10.25	9.31	52.2%	55.2%	3.37	1.50
	Best	8.16	7.38	69.9%	72.0%	7	4
5	Avg.	10.06	9.12	53.8%	56.8%	4.20	1.57
	Best	8.04	7.23	70.8%	73.1%	8	4
6	Avg.	9.51	8.67	58.4%	60.7%	5.70	2.23
	Best	7.72	7.03	73.0%	74.6%	6	2

Table 4 A comparison of concrete strength prediction using different methods

Reference	Method	*RMSE* (MPa)		Age
		Training	Testing	
Peng *et al.* (2009)	Genetic operation trees	9.10	10.80	All
	Regression models	9.05	15.45	All
	Back-propagation networks	8.09	8.83	All
Tasi (2016)	Higher-order neural networks	4.74	6.13	7-day
		8.03	6.78	28-day
This paper	Abrams' codes in Eq. (7)	15.73	15.09	7-day
	Abrams' codes in Eq. (8)	18.23	18.08	28-day
	S_7 in Eq. (14)	5.37	5.90	7-day
	S_{28} in Eq. (20)	7.72	7.03	28-day

Table 5 Common WGP learning for 7-day concrete strength

# layers	Min.	Training	Testing	Training	Testing	Active # operator nodes	Active # parameter nodes
		RMSE (MPa)		R^2			
2	Avg.	12.47	11.50	28.3%	11.9%	1.00	1.47
	Best	7.81	8.33	72.9%	54.8%	1	2
3	Avg.	9.65	8.94	57.5%	47.2%	2.47	2.00
	Best	6.79	7.82	79.5%	60.2%	2	2
4	Avg.	7.73	7.69	72.6%	61.0%	4.24	2.90
	Best	6.05	6.28	83.7%	74.3%	4	4
5	Avg.	8.62	8.32	65.0%	53.6%	5.93	3.07
	Best	5.75	6.17	85.3%	75.2%	8	4
6	Avg.	7.84	8.09	72.7%	57.4%	5.00	1.00
	Best	5.46	6.38	86.8%	73.5%	10	6

of the correlations coefficient of 28-day concrete strength were worse than those of 7-day concrete strength. The best runs of 2- to 6-layered WGP may be obtained and formed as the following equations

$$O_2 = 2.33\sqrt{P_1} \tag{16}$$

$$O_3 = 7.43\sqrt{code + 0.0602P_2} \tag{17}$$

$$O_4 = 9.62\sqrt{code} + 74.8\frac{P_2 - 5.67}{P_6 - 9.78} - 8.02 \tag{18}$$

$$O_5 = 0.985 \times code - \frac{70.3}{P_2 + 4.88} + 21.6 \tag{19}$$

$$O_6 = 1.76 \times code + 0.962\sqrt{P_2} - 3.94 \tag{20}$$

in which, *code* represents the $f_c'{}_{28}$ in Eq. (8). Different to Eqs. (11)-(15) for 7-day concrete strength, the above equations do not all perform in a similar format. Only Eq. (16) does not contain the effects of *code*. Therefore, *code* still works to improve the prediction accuracy of the 28-day concrete strength. As shown in Table 3, when the number of WGP layers exceeds 4, the attached *RMSE* results are good and the obtained equations may be selected as representative of 28-day concrete strength. The O_5 and O_6 perform generally like a linear model as in Eqs. (11)-(15). Ultimately, O_6 was selected to represent WGP learning for 28-day concrete strength (using S_{28} to represent the O_6 hereafter). Furthermore, S_{28} has a constant scaling factor of approximately 1.7 in accordance with the previous findings of the S_7 and the shifting movement of S_{28} is affected by P_2.

4.2 Comparative study

In additional to current knowledge-based WGP learning, the database that was adopted in the present study has been implemented several times in the literature to model concrete strength. In order to evaluate the capabilities of the knowledge-based WGP equations, *RMSE* was used in comparisons. The statistical results from published studies that used the same database are summarized in Table 4. The S_7 of Eq. (14) and S_{28} of Eq. (20) outperformed the regression models and the two black-box methods that were employed in Peng (2009). This result is reasonable due to the large numbers of cases and various concrete curing ages used in these published studies. S_7 and S_{28} both performed on par with the grey-box methods, *i.e.*, the higher-order neural networks used in Tsai (2016). In a summary, results in these two references point out that S_7 and S_{28} offer good prediction results in terms of prediction accuracy. Compared to the results of Abrams' codes, the S_7 improved *RMSE* values from 15 MPa to 5 MPa and the S_{28} decreased *RMSE* values from 18 MPa to 7 MPa. In additional to achieving a reasonable level of accuracy, the knowledge WGP equations in linear mode form are remarkably simple and meaningful, with scaling factors of around 1.7 and shifting terms affected by P_2 and/or P_6. This is an unexpected finding of this paper, but outstandingly good in terms of offering good accuracy and studying advanced improvements of *code*.

Table 6 Common WGP learning for 28-day concrete strength

# layers	Min.	Training	Testing	Training	Testing	Active # operator nodes	Active # parameter nodes
		RMSE (MPa)		R^2			
2	Avg.	11.88	10.78	35.5%	39.6%	1.00	1.10
	Best	11.47	10.34	40.5%	45.0%	1	1
3	Avg.	12.00	10.91	33.3%	37.0%	2.10	1.40
	Best	8.01	7.38	71.0%	72.0%	3	3
4	Avg.	10.98	9.88	44.5%	48.8%	3.63	1.87
	Best	8.01	7.41	71.0%	71.8%	4	3
5	Avg.	10.41	9.35	50.4%	54.6%	5.50	2.71
	Best	7.63	7.15	73.7%	73.7%	7	6
6	Avg.	9.38	8.34	60.2%	64.2%	9.00	4.00
	Best	7.63	6.92	73.7%	75.4%	11	5

4.3 WGP learning without inputs of design codes

Although most previous WGP equations consider the effects of *code*, it is quite easy to obtain WGP equations without considering these effects. Two simple alternatives may be deployed. One is to remove design codes from inputs and the other is to set an extreme penalty number for the occurrences of *code* when evaluating individual fitness for maintaining the consistency of model developments. Similarly, the 7-day and 28-day concrete strengths may be learned using WGP without the effects of the design codes. The statistical results for learning 7-day and 28-day concrete strengths are listed in Table 5 and Table 6, respectively. In comparing Table 2 with Table 5 and Table 3 with Table 6, common WGP learning did not provide results that were more accurate than knowledge-based WGP learning. This explains why most previous knowledge-based WGP equations consider the effects of *code*. Five WGP equations that ignore the effects of *code* may be obtained for the best runs in Table 5, which are

$$O_2 = 0.0846P_1 + 0.680P_5 \tag{21}$$

$$O_3 = 0.0832P_1 + 1.10P_5 - 3.84 \tag{22}$$

$$O_4 = 49.2\frac{P_2}{P_6} + 18.2\frac{P_1}{P_4} - 9.98 \tag{23}$$

$$O_5 = \frac{42.8P_1}{P_7 - 9.90} + 5.79\sqrt{P_5} - 2.33 \tag{24}$$

$$O_6 = 18.5\frac{P_1}{P_4} + \frac{1.46\sqrt{P_1P_2}}{-1.74P_3 + 2.04P_4} - 12.2 \tag{25}$$

Without the effects of *code*, more parameters were involved in learning 7-day concrete strength. Consequently, the above O_5 was selected as the final representation of 7-day concrete strength with the effects of P_1, P_5, and P_7 due to good testing accuracy. One may argue that more parameters or particular parameters should participate in the final WGP equation. Therefore, for instance, taking the average of Eq. (23) and Eq. (25) includes the effects of five parameters, yielding

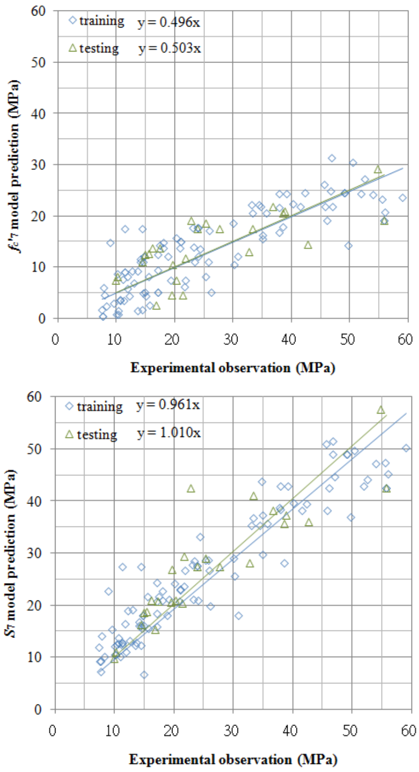

Fig. 6 Linear regression of predictions and targets for 7-day concrete strength

$$O = \frac{O_4 + O_6}{2} = 24.6\frac{P_2}{P_6} + 18.35\frac{P_1}{P_4} + \frac{0.73\sqrt{P_1P_2}}{-1.74P_3 + 2.04P_4} - 11.09 \tag{26}$$

The above equation is able to simply and accurately predict 7-day concrete strength. Moreover, adopting other WGP equations or different ratios is still a possible scenario. Certainly, in a similar manner, WGP equations that lack the effects of *code* for the best runs in Table 6 may be obtained as well for 28-day concrete strength.

$$O_2 = 2.33\sqrt{P_1} \tag{27}$$

$$O_3 = \frac{20.0P_1 + 15.0P_2}{P_4} \tag{28}$$

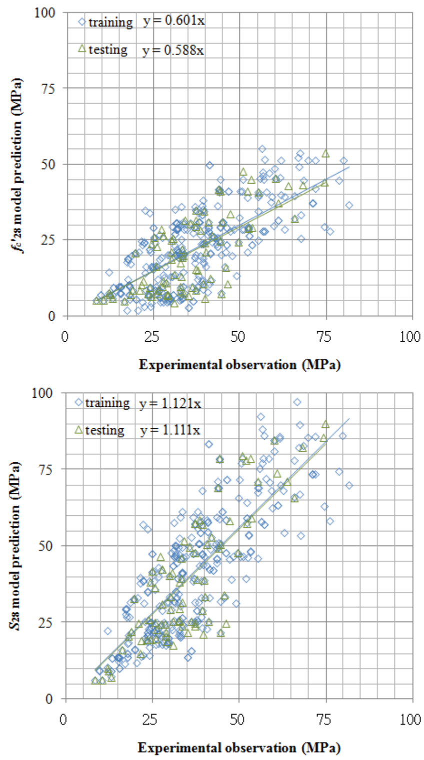

Fig. 7 Linear regression of predictions and targets for 28-day concrete strength

$$O_4 = \frac{19.9P_1 + 15.4P_2}{P_4} \tag{29}$$

$$O_5 = 34.9\sqrt{\frac{P_1}{P_4}} + \frac{87.5P_4 - 64.5P_2}{2.50P_1 + 2.11P_3} \tag{30}$$

$$O_6 = \frac{19.4P_1 + 14.7P_2 + 5.90P_3 - 58.8}{P_4} \tag{31}$$

4.4 Proposed equations for improving the Abrams' design code

Rather than achieving extremely high prediction accuracy, this paper aims to obtain simple polynomial-like equations that provide an acceptable level of prediction accuracy. Consequently, the S_7 and S_{28} were the equations proposed to improve the Abrams' *code* for 7-day and 28-day concrete strength, respectively. For 7-day concrete strength, Fig. 6 shows model predictions of the original $f_c{'}_7$ and S_7 versus the experimental values. Fig. 7 describes the same over the $f_c{'}_{28}$ and S_{28} for 28-day concrete strength. The diamond and triangle marks denote training and testing patterns, respectively. Apparently, the Abrams' code greatly underestimates the concrete strength, with slopes of the least square fit lines of around 0.5 to 0.6. The two proposed equations effectively improve the original codes to approach the ideal 45-degree line and the prediction accuracy significantly improved, as shown in Table 4.

The original Abrams' codes involved only two parameters, P_1 and P_4, which refer to the water cement ratio. In S_7, P_2 and P_6 are further included. Therefore, Fig. 8 specifically compares the $f_c{'}_7$ with S_7 over the four parameters. In the same way, the diamond and triangle marks represent the results of training and testing patterns, respectively. The solid lines plot model predictions with a particular parameter, while other parameters directly use their mean values. The distribution of the marks of the S_7 are widely varied within the range of 0 to 60 MPa, which suggests that the modified equation offers various predictions for specimens with a specific value of the parameter. The solid lines in Fig. 8 reveal the trend of 7-day concrete strength over a specific parameter. The $f_c{'}_7$ includes the effects of P_1 and P_4 on 7-day concrete strength. The concrete strength increases when P_1increases or P_4 decreases, while the S_7 broadens the effects of these two parameters. With regard to the influence degree of parameters, the strength values vary from 0 to 25 MPa when considering the lower and upper bound values of P_1. The S_7 severely changes the strength values into ranges between 7 to 51 MPa, which makes P_1 the most significant parameter affecting S_7. The trend line of the $f_c{'}_7$ over P_4 varies within 22 to 9 MPa, and that of the S_7 changes within 44 to 22 MPa. The aforementioned changes to P_1 and P_4 achieved by the S_7 are dominated by the scaling magnification to $f_c{'}_7$. Additionally, whereas S_7 considers the effects of P_2 and P_6 on concrete strength, $f_c{'}_7$ does not. The trend line of S_7 over P_2 is between 20 to 40 MPa, which shows that P_2 is actually not negligible to 7-day concrete strength. However, the trend line over P_6 is between 28 to 30 MPa, indicating that 7-day concrete strength is not particularly sensitive to P_6, as the prediction accuracy of RMSE in this paper was around 5 to 7 MPa. Consequently, this paper concludes P_1, P_2, and P_4 as the significant parameters affecting 7-day concrete strength. In terms of work to improve Abrams' code, the effects of constant scaling, P_2, and constant shifting should be considered.

In a similar manner, Fig. 9 compares $f_c{'}_{28}$ and S_{28} over the three parameters influencing Eq. (20). The S_{28}increases significantly the impacts of P_1 and P_4 on $f_c{'}_{28}$. The trend curves in Fig. 9 seem quite similar to those in Fig. 8. Therefore, P_1, P_2, and P_4 are also concluded as significant parameters affecting 28-day concrete strength. In summary, when excluding the effects of P_6 on S_7, the mean value of P_6 may substitute itself, and the S_7 becomes

$$S_7 = 1.73 \times code + 0.940\sqrt{P_2} - 3.88 \tag{32}$$

The above equation is quite similar to S_{28}, with scaling factors of around 1.7, coefficients of about 0.9 for the square root of P_2, and a shifting constant. Finally, this paper suggests the following new equation for the Abrams' code

$$\begin{aligned} f_c{'} &= a_1 \times code + a_2\sqrt{P_2} + a_3 \\ &= \frac{b_1}{b_2^{\,P_4/P_1}} + b_3\sqrt{P_2} + b_4 \end{aligned} \tag{33}$$

in which, a_1, a_2, and a_3 are three regression constants, which may approximate those in Eq. (32) to perform the scaling and shifting on the original Abrams' *code*. Alternatively, Eq. (5) may be improved using the effects of three parameters

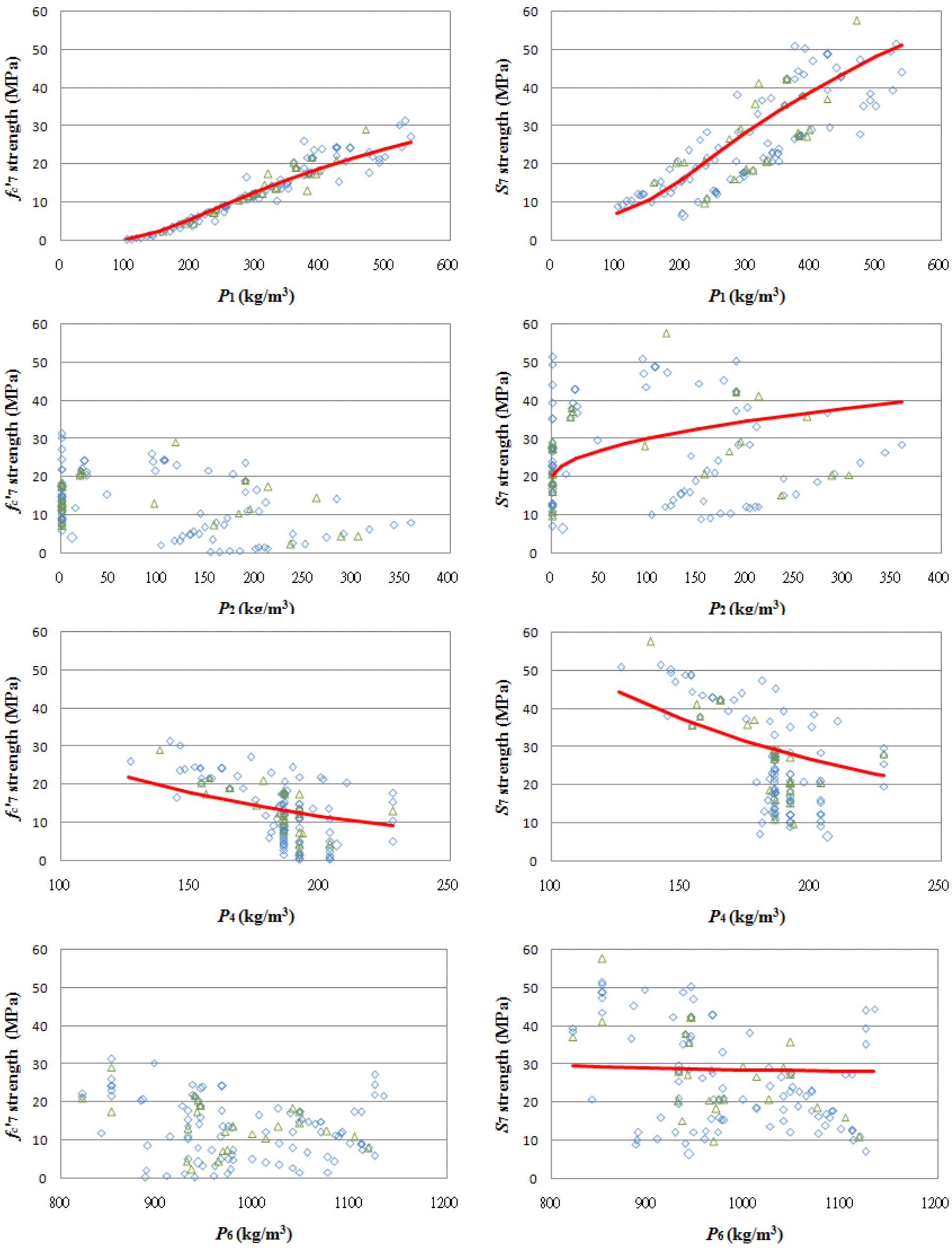

Fig. 8 Effects of the four parameters on 7-day concrete strength

and four constants. The constants b_1 and b_2 are experimental constants determined by a given age, while b_3 and b_4 are regression constants.

5. Conclusions

The present study used WGP, a robust variant of GP, to formulate concrete strength. Using particular settings for the operator candidates obtained equations that were polynomial-like, simple, and accurate. The proposed WGP model was validated by comparing the results of this model with those of two studies from the literature. When treating the results of design codes as inputs, the knowledge that is inherent to white-box models is introduced into WGP learning, which potentially improves prediction accuracy. Furthermore, final WGP equations that incorporate the design code may further improve the design code. Thus, the advantages of knowledge-based WGP learning models include accurate prediction, functional input-output relationships, and possible new approaches to studying the improvement of the design code. Consequently, the solid, engineering-related findings of the present paper include the proposed equation to replace the Abrams' design code. Altering the effects of P_2 and adding two regression constants are the two proposed changes to improve the Abrams' design code.

Acknowledgments

The research presented in this paper was financially

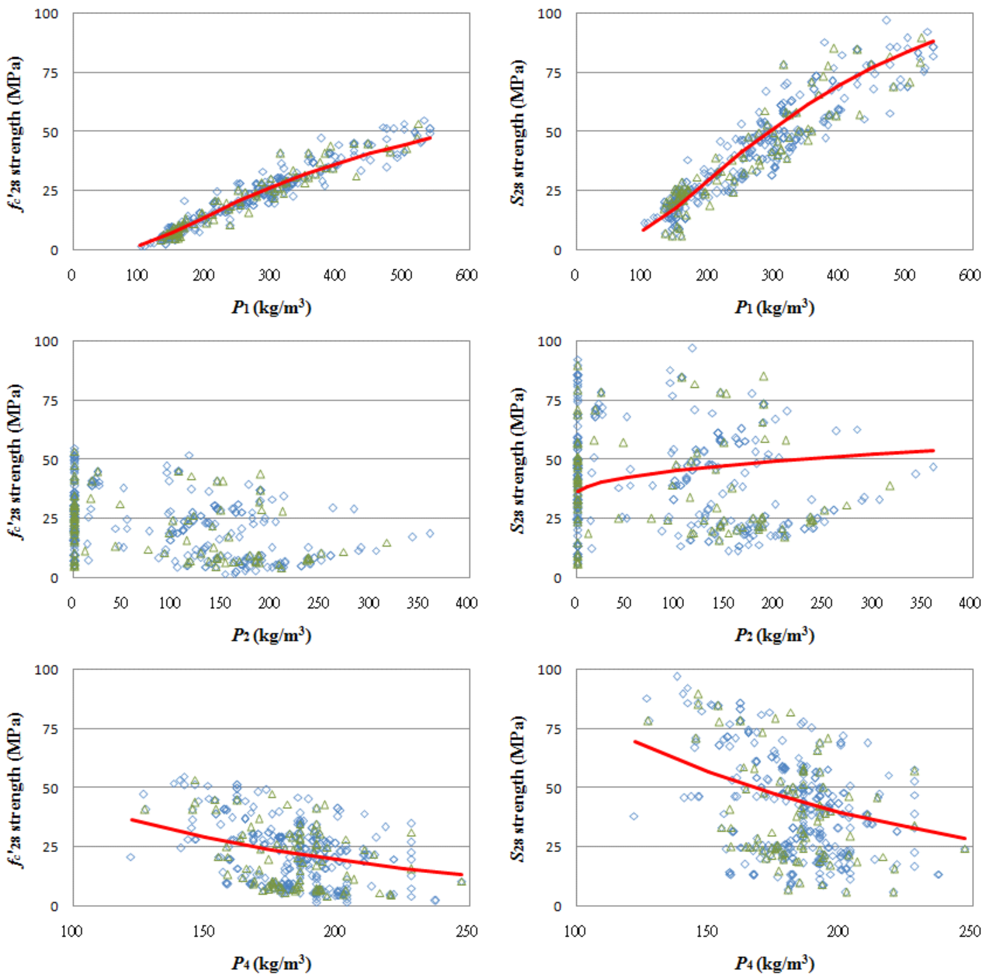

Fig. 9 Effects of the three parameters on 28-day concrete strength

supported by Ministry of Science and Technology, Taiwan, under grant MOST 106-2221-E-011-019.

References

Babanajad, S.K., Gandomi, A.H., Mohammadzadeh, S.D. and Alavi, A.H. (2013), "Numerical modeling of concrete strength under multiaxial confinement pressures using linear genetic programming", *Autom. Constr*., **36**, 136-144.

Babu, K.G. and Rao, G.S.N. (1996), "Efficiency of fly ash in concrete with age", *Cement Concrete Res*., **26**(3), 465-474.

Baykasoglu, A., Güllü, H., Çanakçi, H. and Ozbakir, L. (2008) "Prediction of compressive and tensile strength of limestone via genetic programming", *Exp. Syst. Appl*., **35**(1-2), 111-123.

Baykasoglu, A., Oztas, A. and Ozbay, E. (2009), "Prediction and multi-objective optimization of high-strength concrete parameters via soft computing approaches", *Exp. Syst. Appl*., **36**(3), 6145-6155.

Behzad, M., Asghari, K., Eazi, M. and Palhang, M. (2009), "Generalization performance of support vector machines and neural networks in runoff modeling", *Exp. Syst. Appl*., **36**(4), 7624-7629.

Berardi, L., Kapelan, Z., Giustolisi, O. and Savic, D. (2008), "Development of pipe deterioration models for water distribution systems using EPR", *J. Hydroinf*., **10**(2), 113-126.

Bhattacharya, M., Abraham, A. and Nath, B. (2001), "A linear genetic programming approach for modeling electricity demand prediction in Victoria", *Proceedings of the Hybrid Information Systems, First International Workshop on Hybrid Intelligent Systems*, Adelaide, Australia, 379-393.

Bilgehan, M. and Turgut, P. (2010), "The use of neural networks in concrete compressive strength estimation", *Comput. Concrete*, **7**(3), 271-283.

Canakci, H., Güllü, H. and Dwle, M.I.K. (2018), "Effect of glass powder added grout for deep mixing of marginal sand with clay", *Arab. J. Sci. Eng*., **43**(4), 1583-1595.

Ferreira, C. (2001), "Gene expression programming: A new adaptive algorithm for solving problems", *Complex Syst*., **13**(2), 87-129.

Fiore, A., Quaranta, G., Marano, G.C. and Monti, G. (2016), "Evolutionary polynomial regression-based statistical determination of the shear capacity equation for reinforced concrete beams without stirrups", *J. Comput. Civil Eng*., **30**(1), 04014111.

Giustolisi, O. and Savic, D.A. (2006), "A symbolic data-driven technique based on evolutionary polynomial regression", *J. Hydroinf*., **8**(3), 207-222.

Güllü, H. (2012), "Prediction of peak ground acceleration by genetic expression programming and regression: A comparison using likelihood based measure", *Eng. Geol*., **141-142**, 92-113.

Güllü, H. (2013), "On the prediction of shear wave velocity at local site of strong ground motion stations an application using artificial intelligence", *B. Earthq. Eng*., **11**(4), 969-997.

Güllü, H. (2014), "Function finding via genetic expression

programming for strength and elastic properties of clay treated with bottom Ash”, *Eng. Appl. Artif. Intel*., **35**, 143-157.
Güllü, H. (2015), “Unconfined compressive strength and freeze-thaw resistance of fine-grained soil stabilised with bottom ash, lime and superplasticizer”, *Road Mater. Pavement*, **16**(3), 608-634.
Güllü, H. (2016), “Comparison of rheological models for jet grout cement mixtures with various stabilizers”, *Constr. Build. Mater*., **127**, 220-236.
Güllü, H. (2017a), “A new prediction method to rheological behavior of grout with bottom ash for jet grouting columns”, *Soil. Found*., **57**(3) 384-396.
Güllü, H. (2017b), “A novel approach to prediction of rheological characteristics of jet grout cement mixtures via genetic expression programming”, *Neural Comput. Appl*., **28**, 407-420.
Güllü, H. and Fedakar, H.I. (2017c), “On the prediction of unconfined compressive strength of silty soil stabilized with bottom ash, jute and steel fibers via artificial intelligence”, *Geomech. Eng*., **12**(3), 441-464.
Güllü, H. and Girisken, S. (2013), “Performance of fine-grained soil treated with industrial wastewater sludge”, *Environ. Earth Sci*., **70**, 777-788.
Güllü, H., Canakci, H. and Al Zangana, I.F. (2017d), “Use of cement based grout with glass powder for deep mixing”, *Constr. Build. Mater*., **137**, 12-20.
Güllü, H., Cevik, A., Al-Ezzi, K.M. and Gülsan, M.E. (2019), “On the rheology of using geopolymer for grouting: A comparative study with cement-based grout included fly ash and cold bonded fly ash”, *Constr. Build. Mater*., **196**, 594-610.
Hossain, K.M.A., Lachemi, M. and Easa, S.M. (2006), “Artificial neural network model for the strength prediction of fully restrained RC slabs subjected to membrane action”, *Comput. Concrete*, **3**(6), 439-454.
Hossain, M.S., Ong, Z.C., Ismail, Z., Noroozi, S. and Khoo, S.Y. (2017), “Artificial neural networks for vibration based inverse parametric identifications: A review”, *Appl. Soft Comput*., **52**, 203-219.
Ismail, A. and Jeng, D.S. (2011), “Modelling load-settlement behaviour of piles using high-order neural network (HON-PILE model)”, *Eng. Appl. Artif. Intell*., **24**(5), 813-821.
Koza, J.R. (1992), *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, A Bradford Book, The MIT Press.
Mehrjoo, M., Khaji, N., Moharrami, H. and Bahreininejad, A. (2008), “Damage detection of truss bridge joints using Artificial Neural Networks”, *Exp. Syst. Appl*., **35**(3), 1122-1131.
Mousavi, S.M., Aminian, P., Gandomi, A.H., Alavi, A.H. and Bolandi, H. (2012), “A new predictive model for compressive strength of HPC using gene expression programming”, *Adv. Eng. Softw*., **45**(1), 105-114.
Muhammad, K., Mohammad, N. and Rehman, F. (2015), “Modeling shotcrete mix design using artificial neural network“, *Comput. Concrete*, **15**(2), 167-181.
Nagaraj, T. and Banu, Z. (1996), “Generalization of Abrams’ law”, *Cement Concrete Res*., **26**(6), 933-942.
Olofintoye, O., Otieno, F. and Adeyemo, J. (2016), “Real-time optimal water allocation for daily hydropower generation from the Vanderkloof dam, South Africa”, *Appl. Soft Comput*., **47**, 119-129.
Oltean, M. and Dumitrescu, D. (2002), “Multi expression programming”, Technical Report, UBB-01-2002, Babes-Bolyai University, Cluj-Napoca, Romania.
Oluokun, F.A. (1994), “Fly ash concrete mix design and the water-cement ratio law”, *ACI Mater. J*., **91**(4), 362-371.
Ongpeng, J., Soberano, M., Oreta, A. and Hirose, S. (2017), “Artificial neural network model using ultrasonic test results to predict compressive stress in concrete“, *Comput. Concrete*, **19**(1), 59-68.
Ozbay, E., Oztas, A. and Baykasoglu, A. (2010), “Cost optimization of high strength concretes by soft computing techniques”, *Comput. Concrete*, **7**(3), 221-237.
Parichatprecha, R. and Nimityongskul, P. (2009), “An integrated approach for optimum design of HPC mix proportion using genetic algorithm and artificial neural networks”, *Comput. Concrete*, **6**(3), 253-268.
Patel, K.A., Chaudhary, S. and Nagpal, A.K. (2017), “Neural network based approach for rapid prediction of deflections in RC beams considering cracking“, *Comput. Concrete,* **19**(3), 293-303.
Peng, C.H., Yeh, I.C. and Lien, L.C. (2009), “Modeling strength of high-performance concrete using genetic operation trees with pruning techniques”, *Comput. Concrete*, **6**(3), 203-223.
Popovics, S. (1990), “Analysis of the concrete strength versus water-cement ratio relationship”, *ACI Mater. J*., **87**(5), 517-529.
Saha, P., Prasad, M.L.V. and RathishKumar, P. (2017), “Predicting strength of SCC using artificial neural network and multivariable regression analysis“, *Comput. Concrete,* **20**(1), 31-38.
Sonebi, M., Grünewald, S., Cevik, A. and Walraven, J. (2016), “Modelling fresh properties of self-compacting concrete using neural network technique”, *Comput. Concrete*, **18**(4), 903-921.
Tran, D.H., Ng, A.W.M. and Perera, B.J.C. (2007), “Neural networks deterioration models for serviceability condition of buried storm water pipes”, *Eng. Appl. Artif. Intell*., **20**(8), 1144-1151.
Tsai, H.C. (2009), “Hybrid high order neural networks”, *Appl. Soft Comput*., **9**, 874-881.
Tsai, H.C. (2010), “Predicting strengths of concrete-type specimens using hybrid multilayer perceptrons with center-unified particle swarm optimization”, *Exp. Syst. Appl*., **37**, 1104-1112.
Tsai, H.C. (2011), “Using weighted genetic programming to program squat wall strengths and tune associated formulas”, *Eng. Appl. Artif. Intell*. **24**, 526-533.
Tsai, H.C. (2016), “Modeling concrete strength with high order neural networks”, *Neural Comput. Appl*., **27**, 2465-2473.
Tsai, H.C. and Lin, Y.H. (2011), “Predicting high-strength concrete parameters using weighted genetic programming”, *Eng. Comput*., **27**(4) 347-355.
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. and Lien, L.C. (2009), “Knowledge discovery of concrete material using genetic operation trees”, *Exp. Syst. Appl*., **36**(3), 5807-5812.

CC