Genet Program Evolvable Mach (2013) 14:221–243
DOI 10.1007/s10710-012-9176-3

Improving analytical models of circular concrete columns with genetic programming polynomials

Hsing-Chih Tsai · Chan-Ping Pan

Received: 24 May 2012 / Revised: 17 October 2012 / Published online: 10 November 2012

Abstract This study improves weighted genetic programming and uses proposed novel genetic programming polynomials (GPP) for accurate prediction and visible formulas/polynomials. Representing confined compressive strength and strain of circular concrete columns in meaningful representations makes parameter studies, sensitivity analysis, and application of pruning techniques easy. Furthermore, the proposed GPP is utilized to improve existing analytical models of circular concrete columns. Analytical results demonstrate that the GPP performs well in prediction accuracy and provides simple polynomials as well. Three identified parameters improve the analytical models—the lateral steel ratio improves both compressive strength and strain of the target models of circular concrete columns; compressive strength of unconfined concrete specimen improves the strength equation; and tie spacing improves the strain equation.

Keywords Genetic programming · Models · Compressive strength · Strain · Concrete columns · Polynomials

1 Introduction

Confined concrete with lateral reinforcements increases its strength and ductility in axial compression. Many researchers have expended considerable effort to identify

H.-C. Tsai (✉) · C.-P. Pan
Department of Construction Engineering, National Taiwan University of Science and Technology, #43, Sec. 4, Keelung Rd., Taipei, Taiwan 106, ROC
e-mail: tsaihsingchih@gmail.com

C.-P. Pan
e-mail: panpe@ms67.hinet.net

H.-C. Tsai
Ecological and Hazard Mitigation Engineering Researching Center, National Taiwan University of Science and Technology, #43, Sec. 4, Keelung Rd., Taipei, Taiwan 106, ROC

confinement mechanisms. Most of these studies are empirical or semi-empirical. Some assumptions are conventionally adopted to create an empirical or analytical equation, and unknowns in an equation are obtained by fitting data. Various analytical models have been applied to predict the compressive strength and strain of confined concrete beams. For instance, Mander et al. [1] developed a novel equation with five parameters for confined concrete beams. Some studies followed the work by Mander et al. [1] and analyzed different assumptions or parameters [2–5]. However, analytical models are frequently limited by their calculation accuracy. Soft-computing approaches have potential to enhance prediction accuracy.

Soft-computing approaches include neural networks (NNs), fuzzy logic, support vector machines, genetic algorithms (GAs), evolutionary polynomial regression (EPR), and genetic programming (GP). Each has unique benefits when applied to particular application categories. NNs are the most commonly used soft-computing approaches for inference tasks, from which many NN derivatives have been developed and applied [6–15]. 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 [16], GP has garnered considerable attention due to its ability to model nonlinear relationships for input–output mappings. Baykasoglu et al. [17] compared a promising set of GP approaches, including Multi Expression Programming (MEP), Gene Expression Programming (GEP), and Linear Genetic Programming (LGP) [18–20]. Notably, LGP was the most efficient algorithm for studied limestone strengths. Differences between these algorithms are rooted in the methodology utilized to generate a GP individual. A chromosome representation, a tree topology, and a linear string are used by MEP, GEP, and LGP, respectively. Although, some formulas generated by MEP, GEP, and LGP have coefficients, all coefficients are fixed constants [17]. Several studies have utilized GP derivatives for construction industry problems. Baykasoglu et al. [21] applied GEP to determine concrete strength, cost, and slump. Yeh and Lien [22] developed a genetic operation tree (GOT) to investigate concrete strength. The GOT uses a tree topology (as does GEP) and optimized coefficients that differ from other GP derivatives. Coefficients do not frequently appear in formulas programmed using any of these GP models. Tsai [23] proposed a weighted genetic programming (WGP) to introduce weight coefficients into tree connections and generate a fully weighted formula.

Giustolisi and Savic [24] proposed an evolutionary polynomial regression (EPR) and its applications had been validated in civil engineering [25, 26]. Giustolisi and Savic [24] argued that GP is not very powerful in finding constants and that it tends to produce functions that grow in length over time. This paper significantly improves GP in finding constants and controls formula length with layered tree structures and a terminal. Additionally, EPR has to set up the number of additive terms. Detailed comparisons on EPR and the proposed model will be revealed after resultant formulas are available.

The main aims of this paper are as follows

1. Improve previous WGP;
2. Model confined compressive strength and strain of circular concrete columns with good prediction accuracy and visible formulas;

3. Provide polynomials with a modified WGP, namely, genetic programming polynomials (GPP);
4. Improve analytical models with the GPP;
5. Study parameter impact using sensitivity analysis;
6. Prune techniques for compacting formulas.

The remainder of this paper is organized as: (1) the proposed WGP, GPP methods and GAs; (2) details of confined compressive strength and strain of circular concrete columns; (3) analytical results and discussions; and (4) conclusions.

2 Weighted genetic programming

Following the study by Tsai [23], this paper presents a WGP method with an *NL*-layered tree structure (Fig. 1). The eventual layer has 2^{NL-1} parameter nodes and each parameter node (x_i^{NL}) selects one input (including a unit parameter “1”). When a unit parameter is selected, the value of the parameter node uses its weight (i.e., value of *w* is not 1) to create a coefficient.

$$x_i^{NL} \in \left\{1\, P_1\, P_2 \ldots P_j \ldots P_{NI}\right\}, \quad j = 0 \sim NI \tag{1}$$

where x_i^{NL} represents nodes in the *NL*-th layer and *i* is a related node number; P_j is the *j*-th input parameter; and *NI* is the number of inputs. Each x_i^{NL} node selects one attached P_j. All nodes in the remaining layers are operator nodes, which use operators to calculate the values of parent nodes in a down-top order and are functions of child nodes (Fig. 1). Each operation node *y* is operated by a set of defined functions with the two child nodal inputs of x_i and x_j with weights of w_i and w_j, respectively.

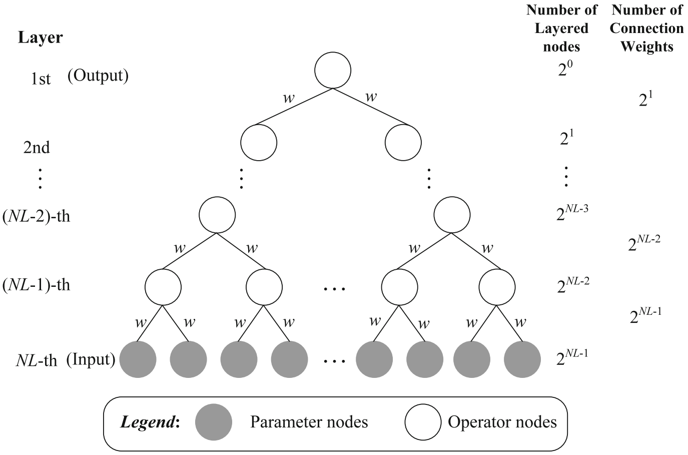


Fig. 1 Weighted genetic programming structure

$$y = F(w_i, w_j, x_i, x_j) \in \left\{ \begin{array}{ccc} f_0 & = & \mathrm{T} \\ f_1 & = & w_i x_i \\ f_2 & = & w_i x_i + w_j x_j \\ f_3 & = & (w_i x_i)(w_j x_j) \\ f_4 & = & (w_i x_i)/(w_j x_j) \\ f_5 & = & |w_i x_i|^{w_j x_j} \\ f_6 & = & \sin(w_i x_i) \\ f_7 & = & \cos(w_i x_i) \\ f_8 & = & \exp(w_i x_i) \\ f_9 & = & \log|w_i x_i| \\ \ldots & \ldots & \ldots \\ f_{NF} & = & \frac{1}{\sin(w_i x_i) + \cos(w_j x_j)} \end{array} \right\} \tag{2}$$

This paper adopts f_0 and nine functions in Eq. (2) for each operator selection F. A unique operator,f_0, is designed as a branch terminal ("T"). When "T" is selected for an operator node, it uses the value of the left-most parameter node as its nodal value directly. Therefore, some operator nodes do not exist in the final WGP results when "T" is used; thus, "T" is a default operator function and novel in WGP comparing to Tsai and Lin [27].f_1 is designed to inherit the child nodes on the left with w_i scaling, and is a unary operator (using "S"). This "S" does not exist in GP function sets. Although f_2 is a plus operator (+), the "−" operator in GP is absent in WGP, because f_2 fulfills negative weights. Thus,f_3,f_4,f_5,f_6,f_7,f_8, andf_9 are "×", "/", "^", "sin", "cos", "exp", and "log" with balanced weights, respectively. Furthermore, a lucky guess function may be utilized. Consequently, variables that must be optimized include (2^{NL-1} − 1) selected operators, (2^{NL-1}) selected parameters, and (2^{NL} − 2) optimized weights.

2.1 Genetic programming polynomials

Many operator functions can be created to model problems. If a lucky function is suggested, training performance will be improved markedly. However, such good luck is rare. Thus, another choice for modeling problems in a concise and simple format seems to be a good idea. Polynomials exist in a wide range of disciplines, including mathematics and the sciences. Mathematically, a polynomial is an expression of finite length constructed from variables and constants using only addition, subtraction, multiplication, and non-negative integer exponents. This paper further designs a novel WGP derivative to provide polynomials for modeling engineering problems, namely, genetic programming polynomials (GPP). Therefore, a new function set is designed for WGP to create GPP:

$$y = F(w_i, w_j, x_i, x_j) \in \left\{ \begin{array}{ccc} f_0 & = & \mathrm{T} \\ f_1 & = & w_i x_i \\ f_2 & = & w_i x_i + w_j x_j \\ f_3 & = & (w_i x_i)(w_j x_j) \\ f_4 & = & (w_i x_i)^{w_j} \end{array} \right\} \tag{3}$$

where f_1, f_2, f_3, and f_4 are "S", "+", "×", and "^", respectively; the exponent term, w_j, in f_4 uses integers 1–10 to create positive integers as exponent terms for polynomials. The division function (f_4) in Eq. (2) is not allowed for polynomials. All materials for the WGP and GPP are considered ready, with the exception of operator selection, parameter selection, and weight optimization. Obviously, optimization generates difficult challenges for the WGP and GPP when a large *NL* is used. For an *NL*-layered WGP or GPP, variables that must be optimized include ($2^{NL-1} - 1$) selected operators, (2^{NL-1}) selected parameters, and ($2^{NL} - 2$) optimized weights, and typed of integers, integers and floats, respectively (Table 1). The global search for ($2^{NL+1} - 3$) variables is performed by means of GA [28]. Standard GA uses a binary alphabet to form chromosomes. Instead, float GA coding is used here to determine integer and float variables simultaneously. Each GA individual has a double vector varied within 0–1, containing parameter selections (one x_i^{NL} node selects an attached P_j); operator selections (F) and weights (w). Values are linearly transferred to boundaries of x_i^{NL}, F, and w at integer 0–*NI*, integer 0–9 for WGP (integer 0–4 for GPP), and float -10 to 10 respectively. In this study, the fitness function was directly set as inverse of the training root mean square error (RMSE). A larger fitness value indicates a healthier individual. Procedures of WGP and GPP are demonstrated in Fig. 2. Only one difference exists among WGP and GPP procedures, i.e. the boundaries of operator selections (step 4 in Fig. 2).

Table 1 Number of variables for WGP and GPP

	Number of			
NL	Operators	Parameters	Weights	All
2	1	2	2	5
3	3	4	6	13
4	7	8	14	29
5	15	16	30	61
6	31	32	62	125

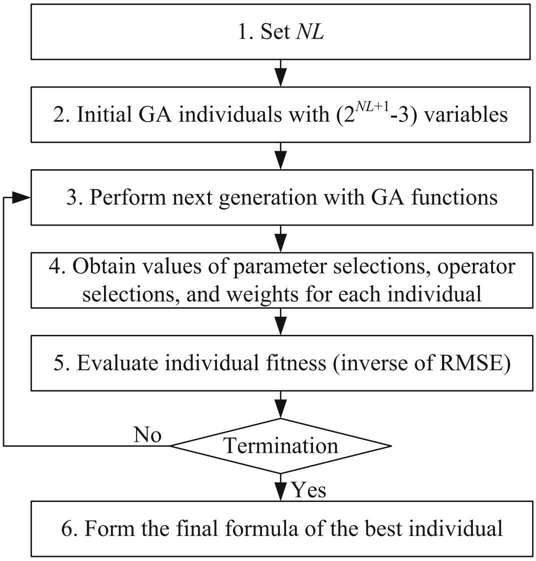

1. Set NL
2. Initial GA individuals with (2^NL+1-3) variables
3. Perform next generation with GA functions
4. Obtain values of parameter selections, operator selections, and weights for each individual
5. Evaluate individual fitness (inverse of RMSE)
Termination
No
Yes
6. Form the final formula of the best individual

Fig. 2 Procedures for WGP and GPP

The following GA parameters are also used in both WGP and GPP implementations [23]:

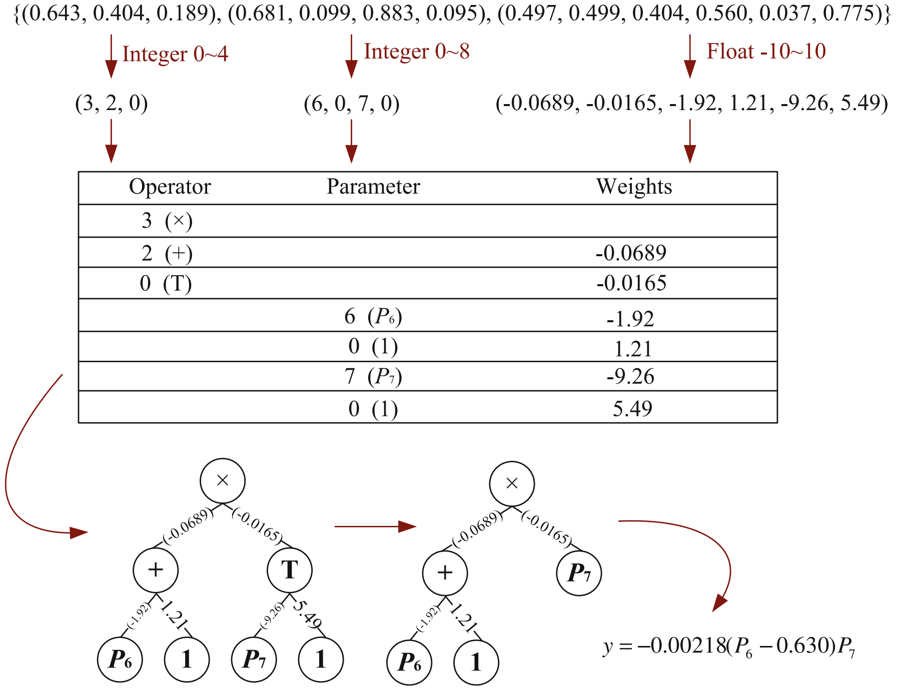
1. Scattered crossover at a rate of 0.8.
2. Uniform mutation at a rate of 0.05.
3. Uniform selection stochastically.
4. 200 individuals.
5. 5,000 iterations without any other terminate criterion.

For instance, a 3-layed GPP needs 13 GA float variables ranged within 0–1 to identify three used operators, four selected parameters, and six balanced weights (Fig. 3). Each child node has an attached weight connecting to its parent node. Sequentially, a corresponding GPP tree structure can be obtained. Finally, the associated formula can be found.

3 Confined compressive strength and strain of circular concrete columns

3.1 Data for confined compressive strength and strain of circular concrete columns

Confined concrete is subjected to transverse reinforcement in the form of closed hoops or spirals to prevent lateral swelling. Such confinement increases the



Operator	Parameter	Weights
3 (×)		
2 (+)		-0.0689
0 (T)		-0.0165
	6 (P_6)	-1.92
	0 (1)	1.21
	7 (P_7)	-9.26
	0 (1)	5.49

$$y = -0.00218(P_6 - 0.630)P_7$$

Fig. 3 An example to transfer GA variables into a GPP formula ($NL = 3$)

compressive strength of concrete and enhances its ductility. Many studies have focused on developing analytical models and identifying factors affecting confined concrete. Oreta et al. [29] adopted the three datasets for circular concrete columns from Mander et al. [1], Sakai et al. [30], and Sakai [4]. Eight factors were listed for compressive strength of confined concrete specimen f'_{cc} and confined strain at peak stress, ε_{cc}. These eight factors are: (1) compressive strength of an unconfined concrete cylinder, f'_c; (2) compressive strength of an unconfined concrete specimen with the same size and geometry, f'_{co}; (3) core diameter of a circular column, d; (4) column height, H; (5) yield strength of lateral or transverse reinforcements, f_{yh};(6) ratio of volume of a lateral reinforcement to volume of a confined concrete core, ρ_s; (7) spacing of a lateral reinforcement or spiral pitch, s; and, (8) ratio of longitudinal steel to the area of a core of section, ρ_{cc}. Although f'_{co} was not used in NN prediction by Oreta et al. [29], it is retained in this work. Of 38 column experiments, 29 were used for training and 9 for testing (Table 2). An NN result set (N7-4-2) was adopted from the work by Oreta et al. [29]. Their attached RMSE were calculated as 1.68 MPa and 0.054 for f'_{cc} and ε_{cc}, respectively. Such analytical results provide references for prediction accuracy. Therefore, this work attempts to achieve prediction accuracies as good as those of NNs and provide visible formulas for f'_{cc} and ε_{cc} against black-box NNs.

3.2 Analytical models for confined compressive strength and strain of circular concrete columns

Various analytical models have been applied to calculate compressive strength, f'_{cc}, and the corresponding strain, ε_{cc}, of confined concrete columns. However, these models usually have disadvantages in terms of prediction accuracy and advantages in reasonable/visible formulas comparing to NN learning. This study improves analytical models with formulas via soft computing approaches. Since both WGP and GPP models can provide visible formulas, two analytical models are chosen to be improved.

Table 2 Detailed values of inputs and outputs

Factors	Lower bound	Upper bound	Avg.	Std.
P_1: f'_c (MPa)	19.45	33	25.1	5.03
P_2: f'_{co} (MPa)	21	32	25.2	4.19
P_3: d (mm)	185	438	329	98.8
P_4: H (mm)	600	1500	1096	359
P_5: f_{yh} (MPa)	307	376	352	19.0
P_6: ρ_s (%)	0.28	2.5	1.51	0.63
P_7: s (mm)	20	240	72.5	46.1
P_8: ρ_{cc} (%)	1.18	4.8	1.97	0.74
f'_{cc} (MPa)	19.3	54	35.6	11.0
ε_{cc} (%)	0.24	1.38	0.58	0.25

Hoshikuma et al. [3] stated:

$$\begin{cases} f'^{HOS}_{cc} = f'_{co} + 0.0383\rho_s f_{yh} \\ \varepsilon^{HOS}_{cc} = 000218 + 0000332\rho_s f_{yh}/f'_{co} \end{cases} \tag{4}$$

Sakai [4] provided:

$$\begin{cases} f'^{SAK}_{cc} = f'_{co}(0.94 + 0.047C) \\ \varepsilon^{SAK}_{cc} = 0.0024 + 0.00064C \end{cases} \quad \text{where} \quad \begin{cases} C = K_s[\rho_s f_{yh}/(2f'_{co})] \\ K_s = [1 - s/(d\tan 30)] \geq 0 \end{cases} \tag{5}$$

where f'^{HOS}_{cc} and ε^{HOS}_{cc} are models developed by Hoshikuma et al. [3]; and f'^{SAK}_{cc} and ε^{SAK}_{cc} were developed by Sakai [4]. Their corresponding RMSEs can be calculated using aforementioned datasets. They are 10.2 MPa and 0.155 for f'^{HOS}_{cc} and ε^{HOS}_{cc}, and 6.24 MPa and 0.159 for f'^{SAK}_{cc} and ε^{SAK}_{cc}, respectively. The calculation accuracy of analytical models also provides references for soft computing approaches and will be improved by the proposed GPP later.

4 Results and discussion

4.1 Predictions and visible formulas

This paper utilizes *NL*s in the range of 2–6 to model circular concrete columns. Tables 3 and 4 list statistical results of 10 runs for f'_{cc} and ε_{cc}, respectively, with concerns of the use of the "T" operator. Results focus on training/testing RMSE, execution time, and "Count". The "Count" is used to count the number of activated operators. For instance, a fully linked four-layer tree has 7 operator nodes; thus, the "Count" is 7. When a "T" function is used in the third layer, the "Count" is 6. As a "T" occurs in the second layer, three operator nodes are eliminated and the "Count" is 4. Therefore, the "Count" is designed to represent the number of operator nodes needed to model f'_{cc} and ε_{cc}. The "Count" is markedly related to prune the complexity of GP trees and effect on conciseness of resulted formulas. Additionally, the use of one-handed operators also positively decreases tree complexity. As *NL* increases, computational time increases, the complexity of the tree structure increases, and the accuracy of computational results increases. Theoretically, both WGP and GPP may achieve better prediction accuracy without using "T" than those with employments of "T". However, when a tree structure uses "T" operators, parts of its variables do not impact on the target function and need not be optimized. That eases the loading of optimization and may lead results to further prediction accuracy easily. Additionally, it seems that both WGP and GPP achieve results under acceptable accuracies with the uses of "T" operator. Therefore, they are primarily compared hereafter. Overall four-layer tree structures are sufficient for achieving good prediction accuracy (Tables 3, 4), as in WGP findings obtained by in Tsai and Lin [27]. Obviously, WGP gives better accuracy than GPP because WGP uses more complicated functions for operators. However, GPP always adopts simpler trees to model circular concrete columns than WGP and

Table 3 Results for confined compressive strength of circular concrete columns

NL		Without "T"						With "T"					
		WGP			GPP			WGP			GPP		
		Training/testing RMSE (MPa)	Time (h)	Count	Training/testing RMSE (MPa)	Time (h)	Count	Training/testing RMSE (MPa)	Time (h)	Count	Training/testing RMSE (MPa)	Time (h)	Count
2	AVG	4.14/4.55	0.32	1	4.02/4.63	0.31	1	3.85/4.99	0.31	1	3.91/4.81	0.32	1
	STD	0.34/0.35		0	0.00/0.01		0	0.29/0.58		0	0.24/0.42		0
	Selected	4.02/4.62		1	4.02/4.62		1	4.03/4.62		1	4.02/4.62		1
3	AVG	2.60/3.38	0.50	3	2.20/2.84	0.49	3	2.35/2.91	0.50	2.6	2.09/2.45	0.49	2.7
	STD	0.50/0.98		0	0.37/1.08		0	0.38/0.80		0.5	0.21/0.53		0.5
	Selected	2.13/2.59		3	1.97/2.07		3	1.83/2.25		3	2.14/1.66		2
4	AVG	2.16/2.72	0.87	7	2.06/2.67	0.84	7	2.48/2.98	0.87	6.6	2.25/2.54	0.85	5.0
	STD	0.23/0.52		0	0.26/0.25		0	0.47/0.63		0.8	0.51/0.45		1.2
	Selected	2.09/2.14		**7**	1.85/2.52		**7**	1.98/2.22		4	**2.00/1.92**		**5**
5	AVG	2.32/2.75	1.62	15	1.93/2.50	1.54	15	2.10/2.71	1.62	11.1	2.05/2.57	1.55	7.1
	STD	0.32/0.46		0	0.19/0.18		0	0.33/0.50		4.0	0.13/0.55		3.3
	Selected	1.86/2.20		15	1.67/2.25		15	1.49/2.19		8	2.01/1.87		2
6	AVG	1.82/2.48	3.11	31	1.91/2.26	3.01	31	1.95/2.72	3.07	23.3	2.14/2.47	3.01	7.9
	STD	0.46/0.63		0	0.31/0.36		0	0.50/0.91		6.6	0.23/0.44		5.1
	Selected	1.44/1.90		31	1.99/1.62		31	1.40/1.86		27	2.03/1.86		2

Table 4 Results for strain of circular concrete columns

NL		Without "T"				With "T"			
		WGP		GPP		WGP		GPP	
		Training/testing RMSE (MPa)	Count	Training/testing RMSE (MPa)	Count	Training/testing RMSE (MPa)	Count	Training/testing RMSE (MPa)	Count
2	AVG	0.127/0.111	1	0.197/0.132	1	0.151/0.116	1	0.197/0.134	1
	STD	0.028/0.004	0	0.004/0.020	0	0.040/0.016	0	0.004/0.018	0
	Selected	0.118/0.109	1	0.192/0.100	1	0.119/0.091	1	0.192/0.114	1
3	AVG	0.116/0.106	3	0.154/0.126	3	0.128/0.099	2.6	0.181/0.126	2.6
	STD	0.039/0.018	0	0.043/0.030	0	0.034/0.012	0.5	0.022/0.021	0.7
	Selected	0.103/0.071	3	0.101/0.088	3	0.089/0.098	3	0.158/0.067	3
4	AVG	0.114/0.093	7	0.155/0.121	7	0.116/0.101	6.4	0.154/0.124	5.3
	STD	0.045/0.032	0	0.037/0.024	0	0.043/0.041	1.1	0.039/0.026	1.7
	Selected	0.079/0.055	**7**	0.114/0.103	**7**	0.079/0.069	7	**0.104/0.080**	**7**
5	AVG	0.100/0.107	15	0.148/0.115	15	0.088/0.082	12.0	0.150/0.124	8.6
	STD	0.025/0.026	0	0.052/0.013	0	0.015/0.015	4.0	0.042/0.024	3.9
	Selected	0.095/0.098	15	0.114/0.109	15	0.075/0.063	15	0.101/0.102	5
6	AVG	0.094/0.097	31	0.131/0.102	31	0.102/0.100	24.7	0.167/0.128	8.3
	STD	0.018/0.018	0	0.043/0.012	0	0.016/0.046	4.2	0.037/0.050	7.5
	Selected	0.084/0.098	31	0.119/0.086	31	0.088/0.073	20	0.122/0.087	13

provide compact formulas (see “Count” in Tables 3, 4). Moreover, the accuracy of GPP remains acceptable. Finally, a best trial is selected among the 10 runs according to minimum training and testing RMSE summations. The final f'_{cc} and ε_{cc} formulas are defined as $f'^{WGP}_{cc\,NL}$ and $\varepsilon^{WGP}_{cc\,NL}$ for WGP; and $f'^{GPP}_{cc\,NL}$ and $\varepsilon^{GPP}_{cc\,NL}$ for GPP. They are listed as follows:

$$f'^{WGP}_{cc\,2} = 1.02P_2 + 7.4P_6 \tag{6}$$

$$f'^{WGP}_{cc\,3} = 7.81P_6 + 24.7 - 9.28\sin(-4.75P_5) \tag{7}$$

$$f'^{WGP}_{cc\,4} = -7.1P_2 + 7.85P_6 - 20.1 \tag{8}$$

$$f'^{WGP}_{cc\,5} = 7.86P_6 + 9.36\log|-10.35P_8 + 13.6\sin(-6.66P_5)| \tag{9}$$

$$\begin{aligned} f'^{WGP}_{cc\,6} = {} & \log|53215(639\exp(6.14P_6))^{2.26\exp(-9.95(3.14)^{-0.13P_2})}\log|9.64P_2P_8|\cos(-9.78 \\ & + 32.3P_6)| \end{aligned} \tag{10}$$

$$f'^{GPP}_{cc\,2} = 7.67P_6 + P_2 \tag{11}$$

$$f'^{GPP}_{cc\,3} = -4.14P_2 - 25.8 + 6.64P_6 \tag{12}$$

$$f'^{GPP}_{cc\,4} = 1.96P_2 - 24.6 + 7.40P_6 + 0.115P_8 \tag{13}$$

$$f'^{GPP}_{cc\,5} = 13.6P_2 + 97.4 + 51.1P_6 \tag{14}$$

$$f'^{GPP}_{cc\,6} = -56.6 - 22.9P_6 + 18P_2 \tag{15}$$

$$\varepsilon^{WGP}_{cc\,2} = 10\frac{P_6}{P_2} \tag{16}$$

$$\varepsilon^{WGP}_{cc\,3} = \frac{0.405P_6}{P_2\log(0.12P_2)} \tag{17}$$

$$\varepsilon^{WGP}_{cc\,4} = \exp\left(-1.13\cos\left(-16.4\frac{P_6}{P_2}\right)\right) \tag{18}$$

$$\varepsilon^{WGP}_{cc\,5} = |6.19\sin(-0.5P_2) - 25.9\sin\left(1.33^{P_2}/_{P_6}\right)|^{-0.029P_7} \tag{19}$$

$$\varepsilon^{WGP}_{cc\,6} = \exp\left(-0.595\log\left|4.95\cos\left(44.7\frac{(2.84P_6 - 8.69P_1)}{P_2}\right)\right|\right) \tag{20}$$

$$\varepsilon^{GPP}_{cc\,2} = 0.31P_6 + 0.125 \tag{21}$$

$$\varepsilon^{GPP}_{cc\,3} = 0.0016(P_5 - 0.097P_4)(P_6 + 0.31P_8) \tag{22}$$

$$\varepsilon^{GPP}_{cc\,4} = 0.63P_6 + 0.003P_6^5 - 0.00022P_1^2P_6^2 \tag{23}$$

$$\varepsilon^{GPP}_{cc\,5} = 0.0011P_6(1.77P_5 - 14.1P_1 + 168 - 32.4P_8) \tag{24}$$

$$\varepsilon^{GPP}_{cc\,6} = 0.512P_6 - 1.51(0.0116P_6P_1 + 0.00546P_8^2)^4 \tag{25}$$

.

Although WGP usually provides better prediction accuracy than GPP, WGP formulas are much complex than those of GPP. Obviously, some WGP operators pick functions of "sin", "cos", "exp", and "log". However, their significances are hard to be identified. Thereby, the functions are suggested for specific problems, when particular attributes are determined or some of their parameters are angles (or periodical). When the background knowledge in detail is not available, the proposed GPP is more applicable than WGP to model problems with polynomials. Finally, GPP formulas are suggested in terms of good prediction accuracy and simplicity of formulas. Of which, f'^{GPP}_{cc4} is proposed for f'_{cc} with accuracy at 2.00 MPa/1.92 MPa (associated value of coefficient of determination R^2 are 0.966/0.969) (bold values in Table 3); and ε^{GPP}_{cc4} for ε_{cc} with 0.104/0.080 accuracy (0.842/0.754 in R^2) (bold values in Table 4).

4.2 Results of *k*-fold cross validations and comparisons with neural networks

As previous results, a four-layer GPP structure is enough for modeling circular concrete columns. A *k*-fold cross validation method is further used to evaluate performance of the four-layer GPP model. Fivefold cross validations are conducted. Source data are partitioned into five equal-sized random datasets. Four folds are used to train the GPP model and a fifth fold for testing. Each experience conducts 10 runs and is repeated five times such that each of five folds is used for testing the resultant model trained by the other four. The uses of random folds are primary for measuring model stability, especially where further datasets are costly to collect. Additionally, in order to compare with neural network results [29], which used 7 parameters excepting f'_{co}, GPP results with 7 parameters are also provided. Results of cross validations are shown in Table 5. With 8 parameters, it seems that results of f'_{cc} are not as stable as those of ε_{cc}; but they performance well in terms of R^2 (over 0.9) and are acceptable. When f'_{co} is withdrawn, RMSE of f'_{cc} results are obviously increased, which indicates the significance of f'_{co} to f'_{cc}. Comparing to neural network results (1.68 MPa for f'_{cc} and 0.054 for ε_{cc}), the best runs of GPP give 2.31/1.74 MPa (0.957/0.970 in R^2) for f'_{cc} and 0.112/0.068 (0.816/0.773 in R^2) for ε_{cc}. Although, GPP does not provide results as accurate as black-box neural networks, the prediction accuracy is still acceptable in terms of R^2 and visible formulas can be obtained against black-box approaches. Moreover, it is a fact that that ε_{cc} measurements are not as accurate as that of f'_{cc}. It may be the reason that GPP always gives better R^2 values for f'_{cc} than those for ε_{cc}.

4.3 Comparisons on EPR and GPP

The proposed WGP and GPP are models of symbolic regression. EPR is another symbolic data-driven technique. One of the general model structures of EPR is formed as:

Table 5 Four-layer GPP results for fivefold cross validations

Fold		8 parameters			7 parameters		
		RMSE (MPa)		Count	RMSE (MPa)		Count
		Training	Testing		Training	Testing	
f'_{cc}	1	1.97	2.63	4.6	2.81	2.92	4.6
	2	2.17	1.74	5.4	2.96	2.38	4.6
	3	1.73	2.80	4.4	3.00	2.61	4.2
	4	1.99	2.51	3.7	2.88	2.77	4.8
	5	2.30	1.11	3.9	2.77	2.50	4.3
	All	2.03	2.16	4.4	2.88	2.64	4.5
ε_{cc}	1	0.128	0.116	5.6	0.116	0.118	4.6
	2	0.109	0.118	5.6	0.121	0.113	5.2
	3	0.095	0.120	5.1	0.093	0.130	5.0
	4	0.093	0.125	5.3	0.101	0.133	5.8
	5	0.106	0.099	4.7	0.115	0.086	5.4
	All	0.106	0.116	5.3	0.109	0.116	5.2

$$y = a_0 + \sum_{j=1}^{m} a_j Z_j \tag{26}$$

$$Z_j = \prod_{k=1}^{NI} P_k^{ES_{jk}} \tag{27}$$

where a is a coefficient term, m is the number of non-constant terms, P is a input parameter (candidate-independent input therein), ES is a NI by m exponent matrix valued within user-defined bounds, and Z is a product of inputs. Since ERP and GPP are innately different, this paper compares them with GPP resultant formulas. In terms of Eq. (24), four terms are included and the coefficients are products of several GPP weights. To obtain the same polynomial of Eq. (24), EPR needs settings: four coefficient terms ranged within [−0.0356, 0.185], and a 4 by 8 ES matrix valued at −1, 0, or 1. In Eq. (25), there are 6 terms after simplification. Using EPR to obtain Eq. (25), the maximum value of a should be ranged within $[-1.51 \times 0.00546^4, 0.512]$, and a 6 by 8 ES matrix valued from −8 to 8 should be used. As aforementioned comparisons, GPP shows differences to EPR in settings of resultant polynomial, high flexibilities in polynomial forms, and products of weights to provide extreme coefficient values. Additionally, GPP improves GP in finding constants and does WGP in resulting polynomials. Furthermore, an extended use of GPP is revealed in the coming sub-section.

4.4 Improving analytics models

The abilities of GPP were proved in aforementioned descriptions. Another use is to improve analytical models using the proposed GPP. The idea of improving

Table 6 Improving analytical models of circular concrete columns with IGPP

NL		f'^{HOS}_{cc}		ε^{HOS}_{cc}		f'^{SAK}_{cc}		ε^{SAK}_{cc}	
		Training/testing RMSE	Count	Training/testing RMSE	Count	Training/testing RMSE	Count	Training/testing RMSE	Count
2	AVG	4.29/4.91	1	0.102/0.085	1	3.23/2.37	1	0.150/0.125	1
	STD	0.08/0.18	0	0.003/0.004	0	0.15/0.35	0	0.001/0.010	0
	Selected	4.18/4.80	1	0.099/0.088	1	3.01/2.07	1	0.148/0.102	1
3	AVG	2.30/2.51	2.9	0.098/0.081	2.8	2.81/1.76	2.9	0.122/0.085	3
	STD	0.32/0.80	0.3	0.006/0.009	0.4	0.19/0.56	0.2	0.006/0.013	0
	Selected	2.21/1.88	3	**0.089/0.059**	**2**	2.69/1.05	3	0.123/0.061	3
4	AVG	2.23/2.58	4.9	0.094/0.081	5.7	2.83/1.74	4.8	0.121/0.081	5.9
	STD	0.38/0.64	1.4	0.005/0.011	1.6	0.14/0.59	1.8	0.005/0.017	1.1
	Selected	**1.97/2.03**	**6**	0.089/0.060	7	**2.62/1.16**	**7**	**0.118/0.071**	**4**
5	AVG	2.30/2.81	7.3	0.096/0.082	8.8	2.87/1.79	7.1	0.121/0.083	9.5
	STD	0.59/0.66	2.3	0.006/0.013	3.1	0.19/0.55	3.4	0.011/0.029	3.3
	Selected	1.59/2.39	8	0.086/0.057	10	2.80/1.04	7	0.115/0.047	10
6	AVG	2.21/2.595	8.1	0.095/0.083	14.5	2.89/1.79	11.8	0.118/0.076	11.9
	STD	0.27/0.45	3.4	0.006/0.009	6.0	0.22/0.35	6.7	0.009/0.019	5.9
	Selected	2.20/1.93	6	0.089/0.060	22	2.63/1.07	13	0.114/0.049	13

analytical models can be found in Tsai [23], which used previous WGP (without "T" operators). The idea is to make a prediction based on a summation of calculations by analytical models and GPP prediction to fit the targets. Therefore, the GPP makes predictions basing on analytical models and improves analytical models, namely, improving GPP (IGPP). Following a process similar to that of Tables 3, 4, results of improving $f_{cc}^{\prime HOS}$, ε_{cc}^{HOS}, $f_{cc}^{\prime SAK}$, and ε_{cc}^{SAK} can be made in Table 6. The accuracy of NN results can be treated as references to IGPP still; however, the original accuracy achieved by analytical models is the most important object to this IGPP learning. The IGPP markedly improves $f_{cc}^{\prime HOS}$ from 10.21 MPa to 1.97 MPa/2.03 MPa (0.967/0.966 in R^2); and ε_{cc}^{HOS} from 0.155 to 0.089/0.059 (0.883/0.866 in R^2). Improvements for $f_{cc}^{\prime SAK}$ and ε_{cc}^{SAK} are not outstanding but are good. Improving $f_{cc}^{\prime SAK}$ are from 6.24 MPa to 2.62 MPa/1.16 MPa (0.941/0.989 in R^2), and ε_{cc}^{SAK} from 0.159 to 0.118/0.071 (0.795/0.806 in R^2). The final IGPP formulas are $f_{cc\,NL}^{\prime IGPP}$ and $\varepsilon_{cc\,NL}^{IGPP}$. The IGPP formulas are as follows:

$$f_{cc\,2}^{\prime IGPP} = f_{cc}^{\prime HOS} - 2.84P_6 - 0.0139P_5 \tag{26}$$

$$f_{cc\,3}^{\prime IGPP} = f_{cc}^{\prime HOS} - 6.06P_6 + 1.07P_2 - 27.1 \tag{27}$$

$$f_{cc\,4}^{\prime IGPP} = f_{cc}^{\prime HOS} - 6.19P_6 + 0.937P_2 + 0.0103P_8^4 - 23.9 \tag{28}$$

$$f_{cc\,5}^{\prime IGPP} = f_{cc}^{\prime HOS} - 6.10P_6 + (1.86P_4 - 26.5P_1)(98.8P_2P_8 + 23.3P_4 - 98.2P_5) \times 10^{-7} \tag{29}$$

$$f_{cc\,6}^{\prime IGPP} = f_{cc}^{\prime HOS} - 5.93P_6 + 1.02P_2 - 26.0 \tag{30}$$

$$\varepsilon_{cc\,2}^{IGPP} = \varepsilon_{cc}^{HOS} - 0.00287P_1P_6 \tag{31}$$

$$\varepsilon_{cc\,3}^{IGPP} = \varepsilon_{cc}^{HOS} - 0.00218(P_6 - 0.630)P_7 \tag{32}$$

$$\varepsilon_{cc\,4}^{IGPP} = \varepsilon_{cc}^{HOS} - 0.00215(P_6 - 0.622)P_7 \tag{33}$$

$$\varepsilon_{cc\,5}^{IGPP} = \varepsilon_{cc}^{HOS} - 6.56 \times 10^{-6} \times (P_6 - 0.63)P_5P_7 \tag{34}$$

$$\varepsilon_{cc\,6}^{IGPP} = \varepsilon_{cc}^{HOS} - 0.00211(P_6 - 0.610)P_7 \tag{35}$$

.

To reveal the improvements of the aforementioned equations upon $f_{cc}^{\prime HOS}$ and ε_{cc}^{HOS}, Figs. 4 and 5 compare results of experiments, analytical models, and IGPP formulas. It can be easily obtained that $f_{cc}^{\prime HOS}$ and ε_{cc}^{HOS} frequently overestimate the strength and strain based on experimental results. Therefore, all IGPP formulas generally contribute negative impacts on improving $f_{cc}^{\prime HOS}$ and ε_{cc}^{HOS}. Comparing all IGPP formulas in terms of result accuracy and formula simplicity (Table 6), Eqs. (28) and (32) are selected in improving $f_{cc}^{\prime HOS}$ and ε_{cc}^{HOS}, respectively, and both of them have good consistency with experimental results, especially Eq. (28). Since these formulas are visible, they render parameter studies easy. Table 7 shows the occurrences of input parameters in IGPP formulas. For instance, $f_{cc\,4}^{\prime IGPP}$ results are more accurate than $f_{cc\,3}^{\prime IGPP}$ with presence of P_8 and $f_{cc\,5}^{\prime IGPP}$ makes further improvements with three additional input parameters. Obviously, P_6 is significant

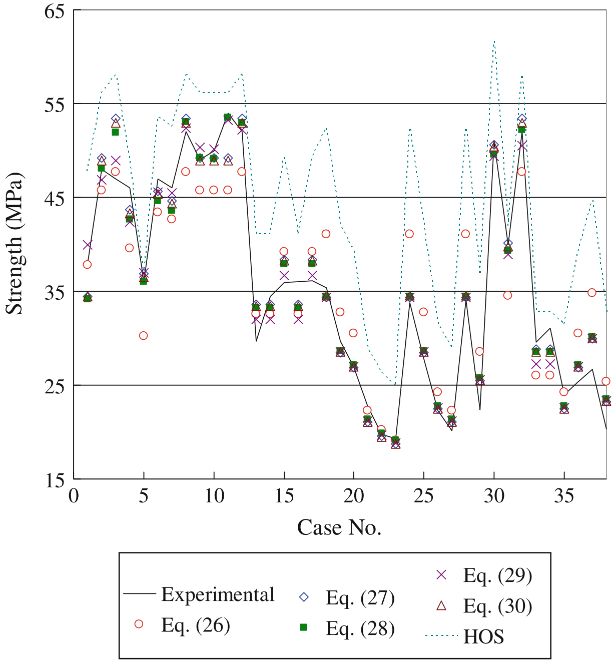


Fig. 4 Results of improving f'^{HOS}_{cc} with IGPP

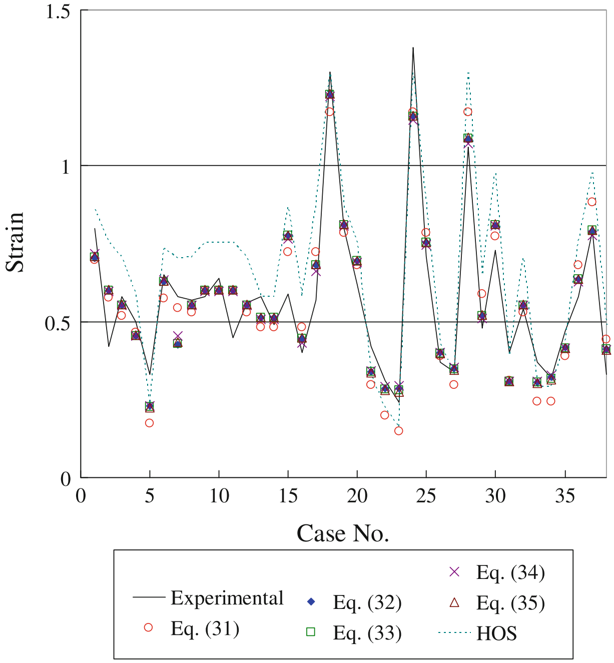


Fig. 5 Results of improving ε^{HOS}_{cc} with IGPP

Table 7 Result accuracy, formulas simplicity, and parameter impacts for improving analytical models of Hoshikuma et al. [3]

	Formula ranking		P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
	Accuracy	Simplicity								
$f'^{IGPP}_{cc\,2}$	5	1					■	■		
$f'^{IGPP}_{cc\,3}$	3	2		■				■		
$f'^{IGPP}_{cc\,4}$	2	4		■				■		■
$f'^{IGPP}_{cc\,5}$	1	5	■	■		■	■	■		■
$f'^{IGPP}_{cc\,6}$	4	2		■				■		
$\varepsilon^{IGPP}_{cc\,2}$	5	1	■					■		
$\varepsilon^{IGPP}_{cc\,3}$	2	2						■	■	
$\varepsilon^{IGPP}_{cc\,4}$	3	2						■	■	
$\varepsilon^{IGPP}_{cc\,5}$	1	5					■	■	■	
$\varepsilon^{IGPP}_{cc\,6}$	4	2						■	■	

in improving f'^{HOS}_{cc} and ε^{HOS}_{cc}, P_2 improves f'^{HOS}_{cc}, P_7 improves ε^{HOS}_{cc}, and P_3 does nothing. The same process can be applied to Eqs. (6)–(25) and the parameters will impact on modeling the confined compressive strength and strain of circular concrete columns instead of improving analytical models.

There are many manners to improve prediction errors of Eqs. (4) and (5). For instance, GA can be directly applied to optimize coefficients of analytical equations, which can be formed as:

$$\begin{cases} f'^{HOS}_{cc} = c_1 f'_{co} + c_2 \rho_s f_{yh} = 105 f'_{co} + 00198 \rho_s f_{yh} \\ \varepsilon^{HOS}_{cc} = c_3 + c_4 \rho_s f_{yh} / f'_{co} = 000249 + 0000273 \rho_s f_{yh} / f'_{co} \end{cases} \tag{36}$$

$$\begin{cases} f'^{SAK}_{cc} = f'_{co}(c_5 + c_6 C) = f'_{co}(117 + 0.0426C) \\ \varepsilon^{SAK}_{cc} = c_7 + c_8 C = 0.00265 + 0.000500C \end{cases} \tag{37}$$

where c_1–c_8 are coefficients for optimization. Same GA settings are used and 10 runs for each equation. The best trial results, according to minimization of summation of training and testing errors, can be found. The prediction RMSE of f'^{HOS}_{cc} is improved from 10.2 MPa to 4.41/4.89 MPa (training/testing RMSE); ε^{HOS}_{cc} from 0.155 to 0.111/0.0792; f'^{SAK}_{cc} from 6.24 MPa 3.98/4.17 MPa; and ε^{SAK}_{cc} from 0.159 to 0.138/0.148. Although the improvement in prediction accuracy is significant, IGPP still provide better results. Two major differences existed between the aforementioned manner and IGPP. One is that IGPP does not have to set a fixed format for the studied equation. Therefore, better prediction accuracy can be expected for IGPP and users need not concern about which coefficient should be optimized (c_1 in Eq. (36) or their exponent coefficients). The other difference is that the analytical model is completely kept in the obtained IGPP formulas. Improvements are directly achieved upon analytical models. The manner of tuning coefficients breaks the original model, but it is useful for figuring out an empirical model under prerequisites of influenced parameter relationships.

4.5 Sensitivity analysis

Sensitivity analysis can be applied to assess the impact of parameters [31]. A common approach in sensitivity analysis is to change one factor at a time to determine its effect on output. This work adopts mean values for all inputs and standard deviation of a targeted input is treated as the variation of the factor changed.

$$SA_i^1 = V(M + \delta_i) - V(M) \tag{38}$$

$$SA_i^2 = V(M - \delta_i) - V(M) \tag{39}$$

where M represents mean values of all inputs, δ_i is the standard deviation of the i-th input and impacts the mean value of the i-th input (found in Table 2), and V is an output value calculated by IGPP formulas. Therefore, SA_i^1 is a sensitivity measurement with a positive variation on the i-th input and SA_i^2 with a negative variation. Table 8 lists the IGPP sensitivity results. When an input parameter is linear in an IGPP formula, the values of SA_i^1 and SA_i^2 are the same. As the effect of the i-th input on output increases, SA_i^1 or SA_i^2 increase. When an input is insensitive to outputs, it can be omitted or replaced. For instance, P_8 can be omitted to compact $f'^{IGPP}_{cc\,4}$. Notably, P_2 and P_6, especially P_6, are important to IGPP formulas for f'_{cc}.

4.6 Pruned IGPP formulas

Another method for studying the impact of parameters is pruning technique [32], which replaces an input with a fixed value (e.g., a mean or median) to determine the effect of the input on outputs. Furthermore, the pruning technique can be used to prune/compact formulas. Selecting $f'^{IGPP}_{cc\,4}$ as an example, it has training/testing RMSEs at 1.97/2.03 MPa originally (step (a) in Table 9). The process continues replacing parameters one by one. As P_2 is replaced in $f'^{IGPP}_{cc\,4}$, RMSEs increase to 4.38/5.07 MPa. Thus, P_2 cannot be replaced by its mean. After steps (b)–(d), P_8 is a good candidate for removal from $f'^{IGPP}_{cc\,4}$ with an accuracy at 2.25/2.01 MPa. However, coefficients in the formula of step (d) are not yet optimized. The GAs can

Table 8 Results of sensitivity analysis for f'^{IGPP}_{cc} formulas with SA_i^1 and SA_i^2 (kN)

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
$f'^{IGPP}_{cc\,2}$	–	–	–	–	−0.28	−1.8	–	–
					0.28	1.8		
$f'^{IGPP}_{cc\,3}$	–	4.6	–	–	–	−3.7	–	–
		−4.6				3.7		
$f'^{IGPP}_{cc\,4}$	–	4.0	–	–	–	−3.8	–	0.5
		−4.0				3.8		−0.1
$f'^{IGPP}_{cc\,5}$	0.2	0.4	–	4.1	−0.9	−3.7	–	0.9
	−0.2	−0.4		−0.7	0.9	3.7		−0.9
$f'^{IGPP}_{cc\,6}$	–	4.3	–	–	–	−3.7	–	–
		−4.3				3.7		

Table 9 Pruning and compacting $f'^{IGPP}_{cc\,4}$

Step	Remove	$f'^{ISCP}_{cc\,4}$ formula	RMSE (MPa)
(a)	–	$f'^{HOS}_{cc} - 6.19P_6 + 0.937P_2 + 0.0103P_8^4 - 23.9$	1.97/2.03
(b)	P_2	$f'^{HOS}_{cc} - 6.19P_6 + 0.0103P_8^4 - 0.3$	4.38/5.07
(c)	P_6	$f'^{HOS}_{cc} + 0.937P_2 + 0.0103P_8^4 - 33.2$	4.16/4.97
(d)	P_8	$f'^{HOS}_{cc} - 6.19P_6 + 0.937P_2 - 23.7$	2.25/2.01
(e)	P_8	$f'^{HOS}_{cc} - AP_6 + BP_2 - C = f'^{HOS}_{cc} - 5.89P_6 + 1.05P_2 - 26.7$	2.20/1.91
(f)	P_2, P_6	$f'^{HOS}_{cc} + 0.0103P_8^4 - 9.62$	4.38/5.51
(g)	P_2, P_8	$f'^{HOS}_{cc} - 6.19P_6 - 0.133$	4.85/5.08
(h)	P_6, P_8	$f'^{HOS}_{cc} + 0.937P_2 - 33.1$	4.06/4.91

be applied again to optimize constants of A, B, and C in the formula of step (e). The accuracy after step (e) is improved again relative to that in step (d). Sequentially, steps (f)–(h) (Table 9) remove two parameters at a time. Finally, the formula from step (e) is applied to prune $f'^{IGPP}_{cc\,4}$. Furthermore, the pruned and optimized $f'^{IGPP}_{cc\,4}$ is very similar to those in Eqs. (12) and (16), although they differ innately. This work considers above evidences as the reliability of the proposed GPP. Finally, the formula in step (e) (Table 9) is applied to model f'_{cc} and two parameters are involved. The formula is listed below with an accuracy of 2.20/1.91 MPa, indicating that f'_{co} and ρ_s are essential to improve f'^{HOS}_{cc}.

$$f'_{cc} = f'^{HOS}_{cc} - 5.89P_6 + 1.05P_2 - 26.7 \tag{40}$$

Equations (32), (33), and (35) have similar formats but have entirely different tree structures (Fig. 6). The evidence of reliable GPP is therefore proved again. A formula with two variables can be assigned to GAs to improve ε^{HOS}_{cc}.

$$\varepsilon_{cc} = \varepsilon^{HOS}_{cc} - A(P_6 - B)P_7 = \varepsilon^{HOS}_{cc} - 0.00215(P_6 - 0.619)P_7 \tag{41}$$

Variables A and B in Eq. (41) can be easily obtained by GAs. When optimization is achieved, Eq. (41) can be treated as the ultimate IGPP solution for improving ε^{HOS}_{cc} with RMSEs at 0.089/0.059. Moreover, ρ_s and s are identified important when improving ε^{HOS}_{cc}. Furthermore, this work follows same steps in Table 9 for $\varepsilon^{IGPP}_{cc\,5}$, and the same formula in Eq. (41) can be obtained. Certainly, such a pruning process can be applied to Eqs. (6)–(25). Summarily, as visible formulas can be obtained, extended uses can be easily carried out.

To combine Eq. (4), (40), and (41) together, improved f'^{HOS}_{cc} and ε^{HOS}_{cc} can be formed as:

$$\begin{cases} f'_{cc} = [f'_{co} + 0.0383\rho_s f_{yh}] - 5.89\rho_s + 105f'_{co} - 267 \\ \varepsilon_{cc} = [0.00218 + 0.000332\rho_s f_{yh}/f'_{co}] - 0.00215(\rho_s - 0619)s \end{cases} \tag{42}$$

The proposed formulas attempt to modify the influences of f'_{co} and ρ_s on confined compressive strength f'_{cc} and achieve higher calculation accuracy based on current datasets. Although eight input parameters are used herein, no new parameter is altered to impact the improved f'^{HOS}_{cc}. A new parameter, *i.e.* s, is used to improve ε^{HOS}_{cc}. The

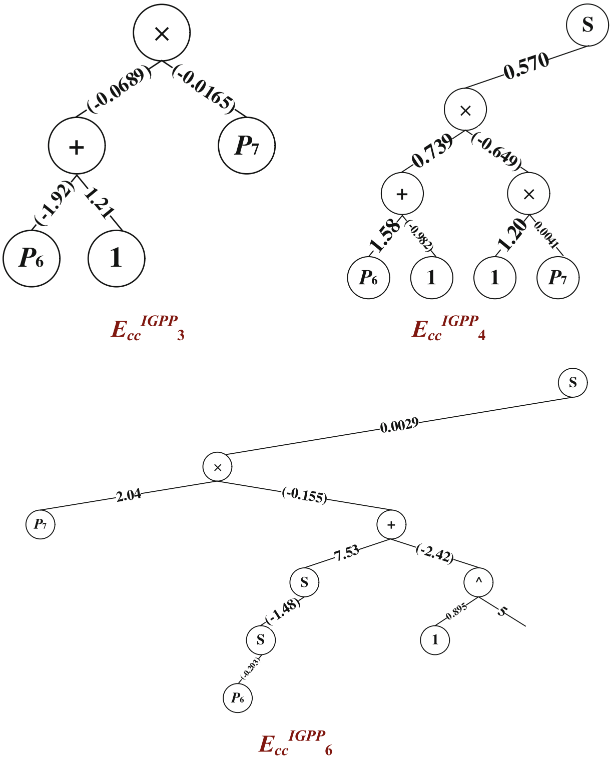


Fig. 6 Tree structures of $\varepsilon_{cc\,3}^{IGPP}$, $\varepsilon_{cc\,4}^{IGPP}$, and $\varepsilon_{cc\,6}^{IGPP}$

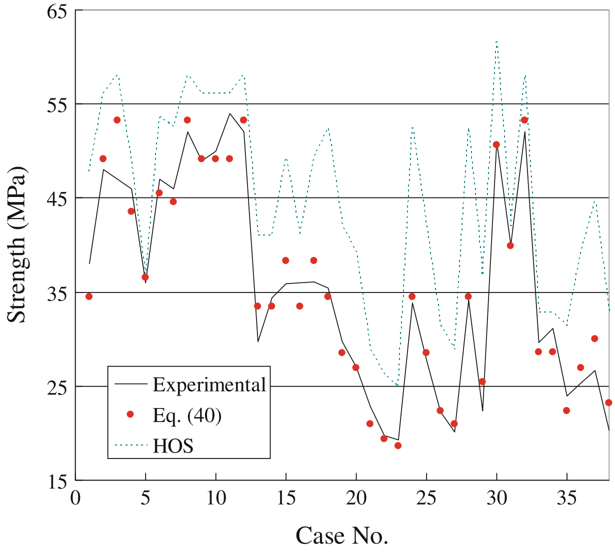


Fig. 7 Final f'_{cc} results

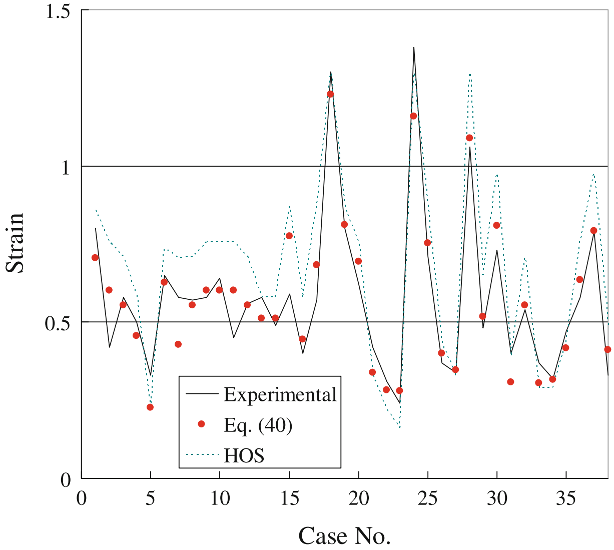


Fig. 8 Final ε_{cc} results

occurrence of s can be justified by Sakai [4] (see Eq. (5)). Owing to the coefficients in the improved terms are numerical results not analytical, they are not meaningful in structural mechanism and are changed when using different datasets. In spite of lacking complicated structural mechanism, the improved formulas for confined compressive strength and strain of circular concrete columns achieve better accuracy than those of the original model (Figs. 7, 8). The influenced parameters are key results of this paper to identify potential parameters on improving analytical models.

5 Conclusions

This study improves previous WGP with a terminate operator, which reduces tree complexity and makes formulas compact, and proposes a novel GPP, which allow GP approaches providing polynomials for problems. Engineering applications focus on not only modeling compressive strength and strain, but also particularly improving analytical models with GPP. Employments of sensitivity analysis and pruning techniques can also be found in this study. Both WGP and GPP achieve good prediction accuracy and visible formulas for target strength and strain. The significant findings of this study are as follows.

1. Visible formulas are bonus production of the WGP and GPP compared with black-box approaches. Unlike analytical models, the proposed WGP and GPP directly provide functional relationships for strength and strain based on fitting data without a prior setting of equation formats.
2. The GPP provides polynomials for compressive strength and strain of circular concrete columns. This approach provides simple solutions for problems.

3. Improving analytical models with soft computing approaches is a novel idea. This IGPP may open a new avenue for improving existing models using soft computing approaches. Both WGP and GPP, even EPR, can be implemented to proceed same manner on improving various analytical models.
4. Visible formulas make parameter studies, sensitivity analysis, and application of pruning techniques easy.
5. Equations (38) and (39) improve the analytical equations developed by Hoshikuma et al. [3]. The ratio of lateral reinforcement volume to confined concrete core volume, ρ_s, impacts improvements to both confined compressive strength and strain equations; the compressive strength of an unconfined concrete specimen of the same size and geometry, f'_{co}, improves the strength equation and the spacing of lateral reinforcement or spiral pitch, s, in the strain equation.
6. The proposed GPP is a data-driven approach. Owing to only 38 datasets are used herein, there is possibility of over-fitting in the obtained formulas although cross-validation method is conducted.

References

1. J.B. Mander, M.J.N. Priestley, R. Park, Observed stress-strain behavior of confined concrete. J. Struct. Eng. **114**(8), 1827–1849 (1988)
2. M. Saatcioglu, S.R. Razvi, Strength and ductility of confined concrete. J. Struct. Eng. **118**(6), 1590–1607 (1992)
3. K. Hoshikuma, K. Kawashima, K. Nagaya, A.W. Taylor, Stress-strain model for confined reinforced concrete in bridge piers. J. Struct. Eng. **123**(5), 624–633 (1997)
4. J. Sakai, *Effect of Lateral Confinement of Concrete and Varying Axial Load on Seismic Response of Bridges*. Doctor of Engineering Dissertation, Dept. of Civil Engineering, Tokyo Institute of Technology, Tokyo (2001)
5. G.G. Penelis, A.J. Kappos, *Earthquake-Resistant Concrete Structures*, (E&FN Spon, London, 1997) Sec. 7.4, pp. 177–196
6. M. Mehrjoo, N. Khaji, H. Moharrami, A. Bahreininejad, Damage detection of truss bridge joints using Artificial Neural Networks. Expert Syst. Appl. **35**(3), 1122–1131 (2008)
7. E. Mesbahi, Y. Pu, Application of ANN-based response surface method to prediction of ultimate strength of stiffened panels. J. Struct. Eng. **134**(10), 1649–1656 (2008)
8. Y.Q. Ni, H.F. Zhou, J.M. Ko, Generalization capability of neural network models for temperature-frequency correlation using monitoring data. J. Struct. Eng. **135**(10), 1290–1300 (2009)
9. H.-C. Tsai, Hybrid high order neural networks. Appl. Soft Comput. **9**, 874–881 (2009)
10. N.B. Dash, S.N. Panda, R. Remesan, N. Sahoo, Hybrid neural modeling for groundwater level prediction. Neural Comput. App. **19**(8), 1251–1263 (2010)
11. H.-C. Tsai, Predicting strengths of concrete-type specimens using hybrid multilayer perceptrons with center-unified particle swarm optimization. Expert Syst. Appl. **37**, 1104–1112 (2010)
12. A. Nazari, G. Khalaj, S. Riahi, ANFIS-based prediction of the compressive strength of geopolymers with seeded fly ash and rice husk-bark ash. Neural Comput. App. 1–13 (2011)
13. S. Cawley, F. Morgan, B. McGinley, S. Pande, L. McDaid, S. Carrillo, J. Harkin, Hardware spiking neural network prototyping and application. Genet. Program. Evol. M. **12**(3), 257–280 (2011)
14. H. Yaprak, A. Karaci, I. Demir, (2011) Prediction of the effect of varying cure conditions and w/c ratio on the compressive strength of concrete using artificial neural networks. Neural Comput. App. 1–9 (2011)
15. H.-C. Tsai, Y.-W. Wu, Y.-Y. Tyan, Y.-H. Lin, *Programming Squat Wall Strengths and Tuning Associated Codes with Pruned Modular Neural Network*. Neural Comput App, Accepted (2012)

16. J.R. Koza, *Genetic programming: On the programming of computers by means of natural selection* (MIT Press, Massachusetts, 1992)
17. A. Baykasoglu, H. Güllü, H. Çanakçi, L. Ozbakir, Prediction of compressive and tensile strength of limestone via genetic programming. Expert Syst. Appl. **35**(1–2), 111–123 (2008)
18. M. Oltean, D. Dumitrescu, Multi expression programming, technical report, UBB-01-2002, Babes-Bolyai University, Cluj-Napoca, Romania (2002). www.mep.cs.ubbcluj.ro
19. C. Ferreira, Gene expression programming: A new adaptive algorithm for solving problems. Compl. Syst. **13**(2), 87–129 (2001)
20. M. Bhattacharya, A. Abraham, B. Nath, A linear genetic programming approach for modeling electricity demand prediction in Victoria. In *Proceedings of the hybrid information systems, first international workshop on hybrid intelligent systems*, Adelaide, Australia, pp 379–393 (2001).
21. A. Baykasoglu, A. Oztas, E. Ozbay E, Prediction and multi-objective optimization of high-strength concrete parameters via soft computing approaches. Expert Syst. Appl. **36**(3), 6145–6155 (2009)
22. I.-C. Yeh, L.-C. Lien, Knowledge discovery of concrete material using genetic operation trees. Expert Syst. Appl. **36**(3), 5807–5812 (2009)
23. H.-C. Tsai, Using weighted genetic programming to program squat wall strengths and tune associated formulas. Eng. Appl. Artif. Intell. **24**, 526–533 (2011)
24. O. Giustolisi, D.A. Savic, A symbolic data-driven technique based on evolutionary polynomial regression. J. Hydroinf. **8**(3), 207–222 (2006)
25. L. Berardi, Z. Kapelan, O. Giustolisi, D. Savic, Development of pipe deterioration models for water distribution systems using EPR. J. Hydroinf. **10**(2), 113–126 (2008)
26. A. Doglioni, D. Mancarella, V. Simeone, O. Giustolisi, Inferring groundwater system dynamics from hydrological time-series data. Hydrol. Sci. J. **55**(4), 593–608 (2010)
27. H.-C. Tsai, Y.-H. Lin, Predicting high-strength concrete parameters using weighted genetic programming. Eng. Comput. **27**(4), 347–355 (2011)
28. J.H. Holland, *Adaptation in neural and artificial systems* (The University of Michigan Press, Ann Arbor, 1975)
29. A.W.C. Oreta, K. Kawashima, Neural network modeling of confined compressive strength and strain of circular concrete columns. J. Struct. Eng. **129**(4), 554–561 (2003)
30. J. Sakai, K. Kawashima, H. Une, K. Yoneda, Effect of tie spacing on stress-strain relation of confined concrete. J. Struct. Eng. **46A**(3), 757–766 (2000)
31. M. Scardi, L.W. Harding, Developing an empirical model of phytoplankton primary production: a neural network case study. Ecol. Modell. **120**(2), 213–223 (1999)
32. C.-H. Peng, I.-C. Yeh, L.-C. Lien, Modeling strength of high-performance concrete using genetic operation trees with pruning techniques. Comput. Concrete **6**(3), 203–223 (2009)