Computers and Concrete, *Vol. 11, No. 6 (2013) 603-620*
DOI: http://dx.doi.org/10.12989/cac.2013.11.6.603

Polynomial modeling of confined compressive strength and strain of circular concrete columns

Hsing-Chih Tsai*

Department of Construction Engineering, National Taiwan University of Science and Technology, Taiwan

(Received November 15, 2011, Revised August 5, 2012, Accepted December 17, 2012)

Abstract. This paper improves genetic programming (GP) and weight genetic programming (WGP) and proposes soft-computing polynomials (SCP) for accurate prediction and visible polynomials. The proposed genetic programming system (GPS) comprises GP, WGP and SCP. To represent confined compressive strength and strain of circular concrete columns in meaningful representations, this paper conducts sensitivity analysis and applies pruning techniques. Analytical results demonstrate that all proposed models perform well in achieving good accuracy and visible formulas; notably, SCP can model problems in polynomial forms. Finally, concrete compressive strength and lateral steel ratio are identified as important to both confined compressive strength and strain of circular concrete columns. By using the suggested formulas, calculations are more accurate than those of analytical models. Moreover, a formula is applied for confined compressive strength based on current data and achieves accuracy comparable to that of neural networks.

Keywords: genetic programming; weighted genetic programming; models; compressive strength; strain; concrete columns

1. Introduction

Soft-computing approaches include neural networks (NNs), fuzzy logic, support vector machines, genetic algorithms (GAs) 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 (Hossain *et al.* 2006, Parichatprecha and Nimityongskul 2009, Tsai 2009, Bilgehan and Turgut 2010, Ozbay *et al.* 2010, Tsai 2010). 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), GP has garnered considerable attention due to its ability to model nonlinear relationships for input-output mappings. Baykasoglu *et al.* (2008) compared a promising set of GP approaches, including Multi Expression Programming (MEP), Gene Expression Programming (GEP) and Linear Genetic Programming (LGP) (Oltean and Dumitrescu 2002, Ferreira 2001, Bhattacharya *et al.* 2001). Notably, LGP was the most efficient algorithm for studied limestone strengths. Differences between these algorithms are rooted in the

*Corresponding author, Assistant Professor, E-mail: tsaihsingchih@gmail.com

http://www.techno-press.org/?journal=cac&subpage=8 ISSN: 1598-8198(Print), 1598-818X (Online)

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 (Baykasoglu *et al*. 2008). Several studies have utilized GP derivatives for construction industry problems. Baykasoglu *et al*. (2009) applied GEP to determine concrete strength, cost and slump. Yeh and Lien (2009) 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 (2011) proposed a weighted GP (WGP) to introduce weight coefficients into tree connections and generate a fully weighted formula.

Confined concrete with lateral reinforcements increase its strength and ductility in axial compression. Many researchers have expended considerable effort to identify 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*. (1988) developed a novel equation with five parameters for confined concrete beams. Some studies followed the work by Mander *et al*. (1988) and analyzed different assumptions or parameters (Saatcioglu and Razvi 1992, Hoshikuma *et al*. 1997, Sakai *et al*. 2000, Penelis and Kappos 1997). However, analytical models are frequently limited by their calculation accuracy. Soft-computing approaches have potential to enhance prediction accuracy. Particularly, visible formulas can be provided by GP and its derivatives.

The main aims of this paper are as follows.

(1) Improve GP and WGP;

(2) Provide polynomials with a modified WGP, namely, soft-computing polynomials (SCP);

(3) Model confined compressive strength and strain of circular concrete columns with good prediction accuracy and visible formulas;

(4) Study parameter impact using sensitivity analysis;

(5) Prune techniques for compacting formulas.

The remainder of this paper is organized as follows. Section II presents the proposed GP, WGP, SCP methods and GAs. Section III characterizes confined compressive strength and strain of circular concrete columns. Section IV gives analytical results and discussions. Section V makes conclusions.

2. Genetic programming system

The proposed genetic programming system (GPS) is composed of three models, i.e., the GP, WGP and SCP models. Major improvements achieved include coefficient optimization for GP to provide weights and terminate operators for WGP to compact formulas. Moreover, the notion of SCP in this paper is novel.

2.1 Genetic programming

Genetic programming, a branch of GAs, was proposed by Koza (1992). The original GP is constructed by a GA string of numbers and forms an individual with a tree structure. Crossover and mutation are applied directly to GP trees. Following the study by Tsai (2011), this paper presents a GP 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. The concept of an attached *w* is novel in GP research. Therefore, the final GP results have optimized coefficients. This is why this work creates new weights for all parameter nodes.

$$x_i^{NL} = \text{one}\left(1 \quad P_1 \quad P_2 \quad \ldots \quad P_j \quad \ldots \quad P_{NI}\right), \qquad j = 0 \sim NI \tag{1}$$

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. 2). Each operation node *y* is operated by a set of defined functions with the two child nodal inputs of x_i and x_j.

$$y = F(x_i, x_j) = \text{one}\begin{cases} f_0 & = & T \\ f_1 & = & x_i + x_j \\ f_2 & = & x_i - x_j \\ f_3 & = & x_i x_j \\ f_4 & = & x_i / x_j \\ f_5 & = & \left|x_i\right|^{x_j} \\ f_6 & = & \sin(x_i) \\ f_7 & = & \cos(x_i) \\ f_8 & = & \exp(x_i) \\ f_9 & = & \log\left|x_i\right| \\ \ldots & \ldots & \ldots \\ f_{NF} & = & \dfrac{1}{\sin(x_i) + \cos(x_j)} \end{cases} \tag{2}$$

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 GP results when "*T*" is used; thus, "*T*" is a default operator function. Additionally, f_1 is the plus operator (+) and f_2 is the minus operator (-). Thus, f_3 and f_4 are multiplication (×) and division (/) operators, respectively. Further f_5 is a power (^) operator. All of these five operators are binary nodes. Besides, f_6, f_7, f_8 and f_9 are all unary functional operators—sin, cos, exp and log, respectively—and use their left-hand child nodes. Generally, each adopted function should be unique. However, exceptions are permitted based on user requirements. The last function in Eq. (2) is an example case in which a user has confidence in a lucky guess function. Although such a function can be reproduced by combinations of f_6, f_7 and f_1

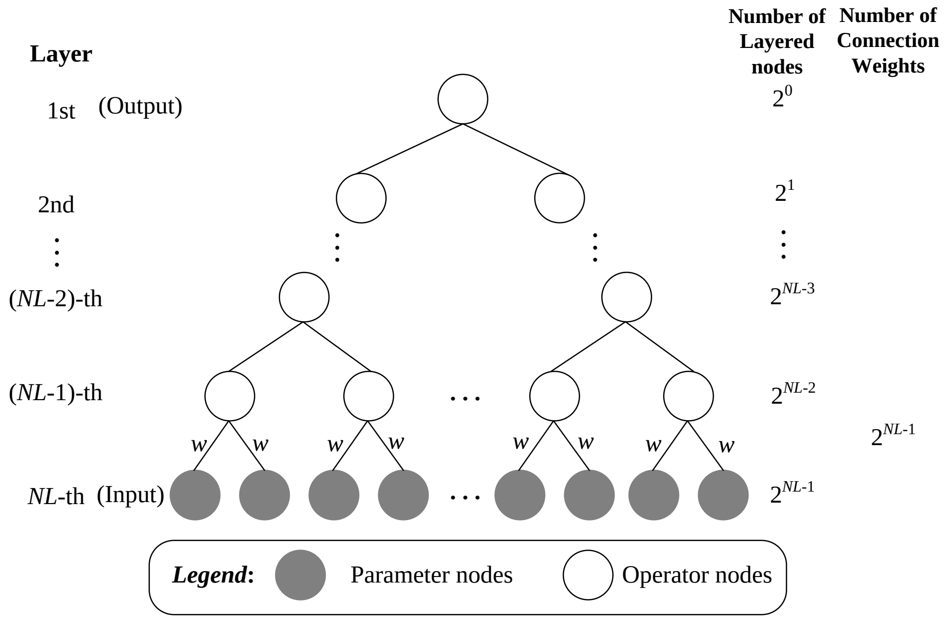

Fig. 1 Genetic programming structure

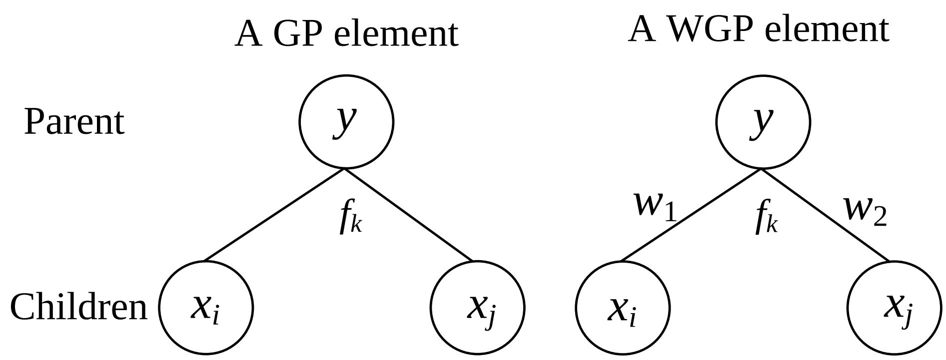

Fig. 2 Elements of GP and WGP

and assigning such an appropriate function as a candidate can markedly accelerate convergence or increase the probability that the function will appear in the final GP results. Consequently, variables that must be optimized include ($2^{NL\text{-}1}$-1) operator selections, ($2^{NL\text{-}1}$) parameter selections and ($2^{NL\text{-}1}$) weight optimizations (Table 1).

2.2 Weighted genetic programming

Although the aforementioned GP can generate coefficients, coefficient costs waste a branch of the GP structure (i.e., tree). Therefore, coefficients do not frequently occur in GP results. Tsai (2011) introduced a weighted balance for GP to create a WGP (Fig. 3). Similarly, each parameter

Table 1 Number of variables for GP and WGP

	GP				WGP			
	Number of				Number of			
NL	Operators	Parameters	Weights	All	Operators	Parameters	Weights	All
2	1	2	2	5	1	2	2	5
3	3	4	4	11	3	4	6	13
4	7	8	8	23	7	8	14	29
5	15	16	16	47	15	16	30	61
6	31	32	32	95	31	32	62	125

node x_i^{NL} selects one input (including a unit parameter, "1") following Eq. (3). When a unit parameter is selected in GP, its nodal value is a weight; however, WGP uses a value of 1 directly, as weights are applied by operators. Each operation node y is operated by a set of defined functions with two child nodal inputs of x_i and x_j with weights of w_i and w_j, respectively.

$$y = F(w_i, w_j, x_i, x_j) = \text{one}\begin{cases} f_0 & = & T \\ f_1 & = & w_i x_i \\ f_2 & = & w_i x_i + w_j x_j \\ f_3 & = & (w_i x_i)(w_j x_j) \\ f_4 & = & (w_i x_i)/(w_j x_j) \\ f_5 & = & \left|w_i x_i\right|^{w_j x_j} \\ f_6 & = & \sin(w_i x_i) \\ f_7 & = & \cos(w_i x_i) \\ f_8 & = & \exp(w_i x_i) \\ f_9 & = & \log\left|w_i x_i\right| \\ \cdots & \cdots & \cdots \\ f_{NF} & = & \dfrac{1}{\sin(w_i x_i) + \cos(w_j x_j)} \end{cases} \tag{3}$$

Additionally, f_0 is utilized in WGP. Default "*T*" is novel in WGP comparing to Tsai and Lin (2011). 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 and f_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 (Table 1). The difference in number of variables for GP and WGP is (2^{NL-1}-2) optimized weights. Fig. 4 shows differences between WGP and GP with a four-layer tree. Analytical outputs (*O*) are programmed as follows

$$O_{\mathrm{GP}} = 3.1 P_2^{3.4} - \frac{P_3 + 3.6}{\sin(P_4)} \tag{4}$$

$$\begin{aligned} O_{\mathrm{WGP}} &= 1.1(2.1 \times 2.2 \times (3.3P_2)^{3.4}) - 1.2\frac{2.3(-3.5P_3 + 3.6)}{-2.4\sin(3.7P_4)} \\ &= 294.4P_2^{3.4} - \frac{4.03P_3 - 4.14P_2}{\sin(3.7P_4)} \end{aligned} \tag{5}$$

The coefficient term in Eq. (4) is produced by replacing an input parameter with a unit parameter "1"; however, this wastes a parameter node. Notably, all child nodes in WGP have weights to produce coefficients. Additionally, when the optimized coefficient exceeds the search domain, GP has substitutions for recovering such, which are to waste additional parameter nodes using unit parameters. Fortunately, WGP may produce a large coefficient by combining a large number of weight coefficients. Thus, WGP is naturally equipped to produce weight coefficients. Furthermore, GP obviously has finite combinations under a fixed *NL* layer; however, WGP has infinite possibilities. Even when *NL* is 2, the GP result is $y=x_i+x_j$ and the WGP result is $y=C_i \times x_i+C_j \times x_j$ with a large number of combinations. Although a WGP should contribute on optimizing (2^{NL-1}-2) weights additionally, applying weighted balancing for each nodal connection is effective.

2.3 Soft computing 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

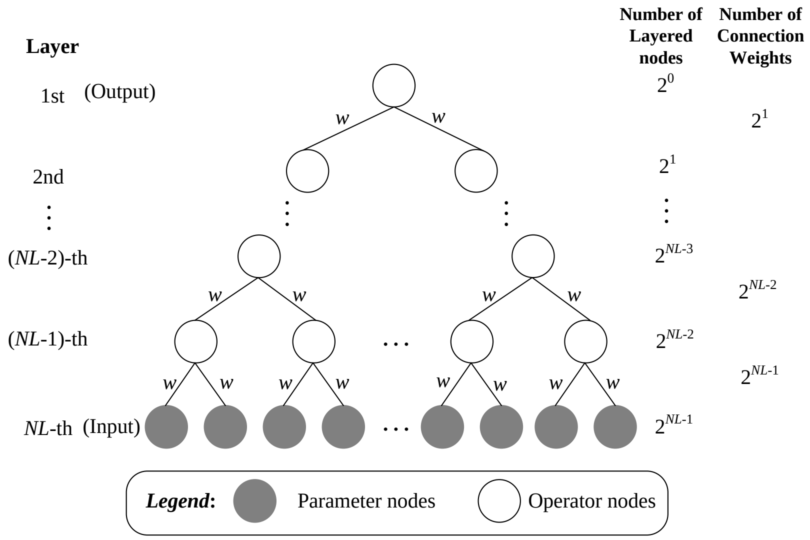

Fig. 3 Weighted genetic programming structure

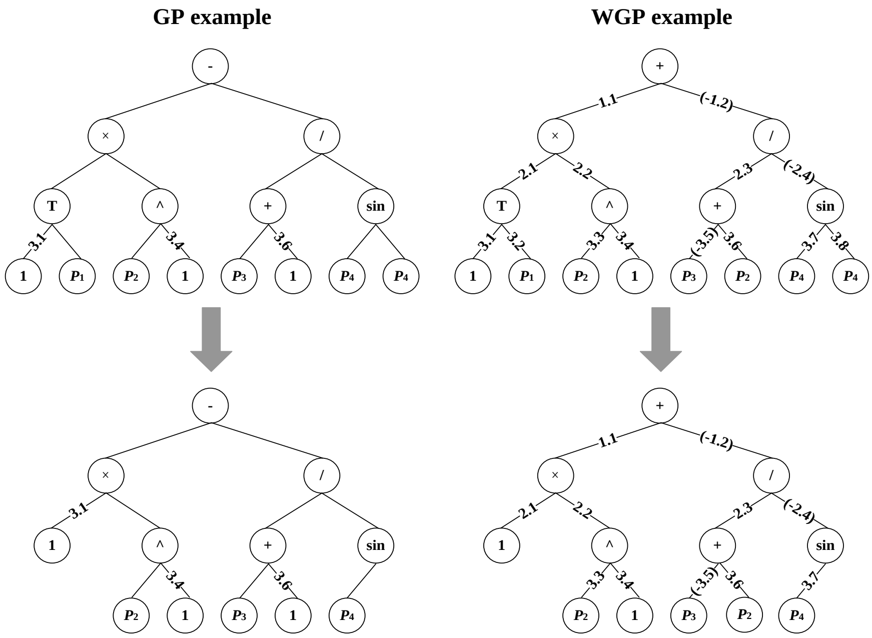

Fig. 4 Examples of WGP and GP

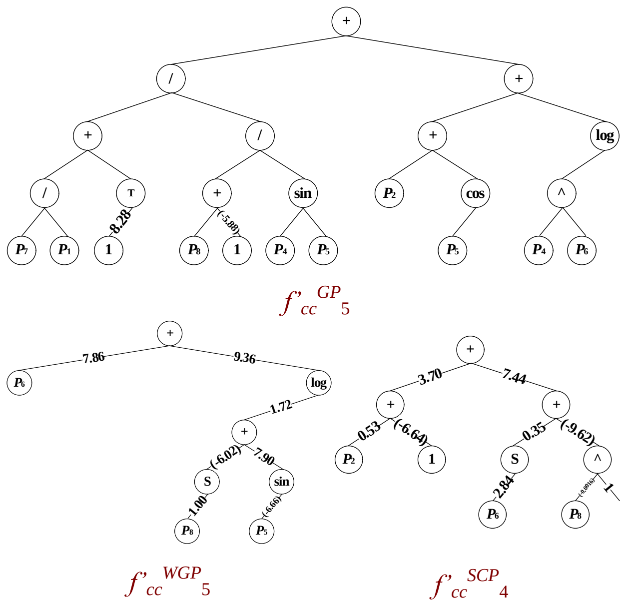

Fig. 5 Tree structures of $f'_{cc}{}^{GP}{}_5$, $f'_{cc}{}^{WGP}{}_5$ and $f'_{cc}{}^{SCP}{}_4$

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

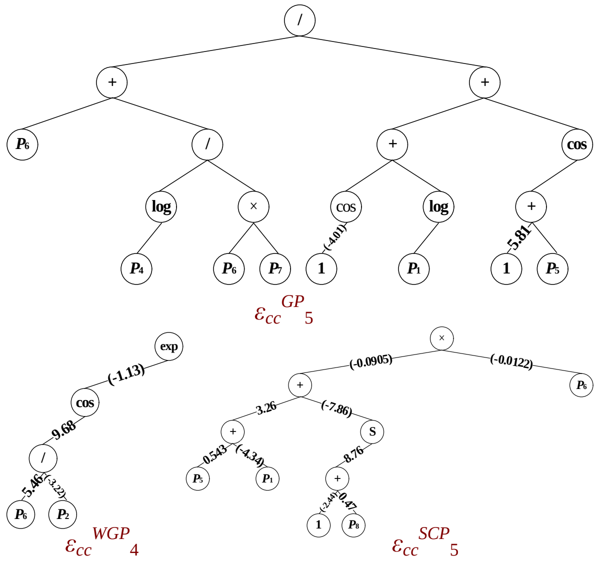

Fig. 6 Tree structures of $\varepsilon_{cc}{}^{GP}{}_5$, $\varepsilon_{cc}{}^{WGP}{}_4$ and $\varepsilon_{cc}{}^{SCP}{}_5$

constants using only addition, subtraction, multiplication and non-negative integer exponents. This paper further designs a novel WGP to provide polynomials for modeling engineering problems, namely, soft-computing polynomials (SCP). Therefore, a new function set is designed for WGP to create SCP

$$y = F(w_i, w_j, x_i, x_j) = \text{one}\begin{cases} f_0 = & T \\ f_1 = & w_i x_i \\ f_2 = & w_i x_i + w_j x_j \\ f_3 = & (w_i x_i)(w_j x_j) \\ f_4 = & (w_i x_i)^{w_j} \end{cases} \tag{4}$$

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. (3) is not allowed for polynomials. All materials for the GPS, which is composed of GP, WGP and SCP, are considered ready, with the exception of operator selection, parameter selection and weight optimization. Obviously, optimization generates difficult challenges for the proposed GPS when a large *NL* is used. Further, trying to optimize all variable simultaneously is very difficult. A minor discrepancy in operator selection will result in very different optimized weights. Fortunately, GA is powerful for global optimization.

2.4 Genetic algorithms

The GA which imitates parts of the natural evolution process, were first proposed by Holland (1975). GA is a stochastic search approach inspired by natural evolution that involves crossover, mutation and evaluation of survival fitness. Genetic operators work from initial generation to offspring in order to evolve an optimal solution through generations. Each individual of a generation generates a result for the problem, which is represented as a string-named chromosome. This relatively straightforward and simple implementation procedure gives GAs exceptional flexibility to hybridize with domain-dependent heuristics to create effective implementation solutions tailored to specific problems. Based on these merits, the potential to use GAs in optimization has been studied intensively (Gen and Cheng 1997). However, simple GA is difficult to apply directly and successfully to a large range of difficult-to-solve optimization problems (Michalewicz 1996).

MATLAB was employed in this study because it is a powerful tool that incorporates various function sets, including genetic algorithm (*ga* function). To study how the MATLAB *ga* function works is essential due to the numerous parameter settings that must be set properly in order to ensure correct results. The five basic steps to use GA are:

(1) Initialize population – Initial individuals in a population are randomly generated in types of "doubleVector" varied within 0~1, containing parameter selections (one x_i^{NL} node selects an attached P_j); operator selection (F) and weights (w). Values are linearly transferred to boundaries of x_i^{NL}, F and w at integer 0~NI, integer 0~9 and float -10~10 respectively.

(2) Evaluate individuals – Fitness is a major index used to evaluate individual status, with decreasing fitness values correlated to increasing degrees of achievement of the model objective. 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.

(3) Perform crossover – A positive scalar should be set for parts of the population to perform crossover. To create crossover children, a function must be selected. This study performed a scattered (function of @corssoverscattered in MATLAB) crossover with a crossover rate of 0.8.

(4) Perform mutation – A mutation produces spontaneous random changes in chromosomes. The MATLAB function @mutationuniform with a mutation rate at 0.05 was used herein.

(5) Select individuals – The @selectionstochinif was used herein to select parents of crossover and mutation children. In additional, two elitist individuals were guaranteed to survive to the next generation herein.

Prior to model execution, two other additional major parameters that should be set include population size and iteration number. Two parameters are set on case by case in accordance with the experienced input of experts. The population size chosen for this study was 200 and 5,000 iterations, respectively.

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

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 compressive strength of concrete and enhances its ductility. Many studies have focused on developing analytical models and identifying factors affecting confined concrete. Oreta and Kawashima (2003) adopted the three datasets for circular concrete columns from (Mander *et al.* 1998, Sakai *et al.* 2000, Sakai 2001). 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*. (2003), 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*. (2003). 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.

4. Results and comparisons

4.1 GPS 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 20 runs for f'_{cc} and ε_{cc}, respectively. 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 (Fig. 4), 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. Overall five-layer tree structures are sufficient for achieving good prediction accuracy (Tables 3 and 4), as in WGP findings obtained by in Tsai and Lin (2011). At *NL*=2, both WGP and SCP achieve better accuracy than GP because GP does not have weighted balancing, as do WGP and SCP. Moreover, WGP has more operator functions to select than SCP. As *NL* increases, the complexity of the tree structure increases and prediction results have good accuracy. Among all models, GP and WGP provide similar accuracy as NNs; however, SCP generates slightly worse accuracy that is still acceptable. Additional execution time is needed to optimize WGP or SCP coefficients. The run time with WGP is slightly longer than that with SCP due to the complexity of operator functions used by WGP. In terms of "Count", SCP always uses fewer operators to model f'_{cc} and ε_{cc} than do GP and WGP. This reduces the prediction accuracy of SCP but results in concise SCP formulas. Finally, a best trial is selected among the 20 runs according to minimum training and testing RMSE summations. The final f'_{cc} and ε_{cc} formulas are defined as $f'_{cc}{}^{GP}{}_{NL}$ and $\varepsilon_{cc}{}^{GP}{}_{NL}$ for WGP; $f'_{cc}{}^{WGP}{}_{NL}$ and $\varepsilon_{cc}{}^{WGP}{}_{NL}$ for WGP; and $f'_{cc}{}^{SCP}{}_{NL}$ and $\varepsilon_{cc}{}^{SCP}{}_{NL}$ for SCP. These formulas are listed as follows

$$f'_{cc}{}^{GP}{}_{2} = P_2^{1.18} \tag{5}$$

$$f'_{cc}{}^{GP}{}_{3} = 1.28 P_6^{0.32} P_2 \tag{6}$$

$$f'_{cc}{}^{GP}{}_{4} = P_2 + P_8 + 6.18 P_6 - \sin(P_4)\log(P_7) \tag{7}$$

$$f'_{cc}{}^{GP}{}_{5} = \frac{\sin(P_4)(\frac{P_7}{P_1} + 8.28)}{P_8 - 5.88} + P_2 + \cos(P_5) + \log|P_4{}^{P_6}| \tag{8}$$

$$f'_{cc}{}^{GP}{}_{6} = P_2 + \cos(P_6 P_7) - 0.14 P_2 \sin(P_4) + \log(|P_4 P_8|^{P_6}) - \sin(P_1)\sin(P_8) + \sin(P_7 P_8) \tag{9}$$

$$f'_{cc}{}^{WGP}{}_{2} = 1.02 P_2 + 7.4 P_6 \tag{10}$$

$$f'_{cc}{}^{WGP}{}_{3} = 7.81 P_6 + 24.7 - 9.28 \sin(-4.75 P_5) \tag{11}$$

$$f'_{cc}{}^{WGP}{}_{4} = -7.1 P_2 + 7.85 P_6 - 20.1 \tag{12}$$

$$f'_{cc}{}^{WGP}{}_{5} = 7.86 P_6 + 9.36 \log|-10.35 P_8 + 13.6 \sin(-6.66 P_5)| \tag{13}$$

$$f'_{cc}{}^{WGP}{}_{6} = \log|53215(639\exp(6.14 P_6))^{2.26\exp(-9.95(3.14)^{-0.13 P_2})} \log|9.64 P_2 P_8| \cos(-9.78 + 32.3 P_6)| \tag{14}$$

$$f'_{cc}{}^{SCP}{}_{2} = 7.67 P_6 + P_2 \tag{15}$$

$$f'_{cc}{}^{SCP}{}_{3} = -4.14 P_2 - 25.8 + 6.64 P_6 \tag{16}$$

$$f'_{cc}{}^{SCP}{}_{4} = 1.96 P_2 - 24.6 + 7.40 P_6 + 0.115 P_8 \tag{17}$$

$$f'_{cc}{}^{SCP}{}_{5} = 13.6 P_2 + 97.4 + 51.1 P_6 \tag{18}$$

$$f'_{cc}{}^{SCP}{}_{6} = -56.6 - 22.9 P_6 + 18 P_2 \tag{19}$$

$$\varepsilon_{cc}{}^{GP}{}_{2} = 0.38 P_6 \tag{20}$$

Table 3 GPS results for confined compressive strength of circular concrete columns

No. of Layers *NL*		GP			WGP			SCP		
		Training/Testing RMSE (MPa)	Time (hr)	Count	Training/Testing RMSE (MPa)	Time (hr)	Count	Training/Testing RMSE (MPa)	Time (hr)	Count
2	AVG	5.73/6.22	0.26	1	3.85/4.99	0.31	1	3.91/4.81	0.32	1
	STD	0.21/0.24		0	0.29/0.58		0	0.24/0.42		0
	Selected	5.47/5.88		1	4.03/4.62		1	4.02/4.62		1
3	AVG	3.54/3.94	0.35	3	2.35/2.91	0.50	2.6	2.09/2.45	0.49	2.7
	STD	0.78/0.54		0	0.38/0.80		0.5	0.21/0.53		0.5
	Selected	2.50/3.58		3	1.83/2.25		3	2.14/1.66		2
4	AVG	2.53/3.17	0.52	6.9	2.48/2.98	0.87	6.6	2.25/2.54	0.85	5.0
	STD	0.51/0.84		0.3	0.47/0.63		0.8	0.51/0.45		1.2
	Selected	1.74/1.64		7	1.98/2.22		4	**2.00/1.92**		**5**
5	AVG	2.37/3.06	0.87	14.2	2.10/2.71	1.62	11.1	2.05/2.57	1.55	7.1
	STD	0.49/0.84		0.8	0.33/0.50		4.0	0.13/0.55		3.3
	Selected	**1.49/1.79**		**13**	**1.49/2.19**		**8**	2.01/1.87		2
6	AVG	1.72/2.68	1.55	26.6	1.95/2.72	3.07	23.3	2.14/2.47	3.01	7.9
	STD	0.31/0.69		3.2	0.50/0.91		6.6	0.23/0.44		5.1
	Selected	1.26/1.73		23	1.40/1.86		27	2.03/1.86		2

$$\varepsilon_{cc}{}^{GP}{}_{3} = \frac{5.84P_6}{-9.89+P_2} \tag{21}$$

$$\varepsilon_{cc}{}^{GP}{}_{4} = \frac{(\exp(P_6)+1.41)(P_3+P_5)}{P_1P_3} \tag{22}$$

$$\varepsilon_{cc}{}^{GP}{}_{5} = \frac{P_6+\log(P_4)/P_6P_7}{-0.65+\log(P_1)+\cos(5.81+P_5)} \tag{23}$$

$$\varepsilon_{cc}{}^{GP}{}_{6} = \cos(\log|\log|P_6-P_2||)\exp(\sin(\log(P_2)))\left(\frac{P_2}{P_6(P_5-P_2)}+P_6\right) \tag{24}$$

$$\varepsilon_{cc}{}^{WGP}{}_{2} = 10\frac{P_6}{P_2} \tag{25}$$

$$\varepsilon_{cc}{}^{WGP}{}_{3} = \frac{0.405P_6}{P_2\log(0.12P_2)} \tag{26}$$

$$\varepsilon_{cc}{}^{WGP}{}_{4} = \exp(-1.13\cos(-16.4\frac{P_6}{P_2})) \tag{27}$$

$$\varepsilon_{cc}{}^{WGP}{}_5 = |6.19\sin(-0.5P_2) - 25.9\sin(1.33P_2/P_6)|^{-0.029P_7} \quad (28)$$

$$\varepsilon_{cc}{}^{WGP}{}_6 = \exp(-0.595\log|4.95\cos(44.7\frac{(2.84P_6 - 8.69P_1)}{P_2})|) \quad (29)$$

$$\varepsilon_{cc}{}^{SCP}{}_2 = 0.31P_6 + 0.125 \quad (30)$$

$$\varepsilon_{cc}{}^{SCP}{}_3 = 0.0016(P_5 - 0.097P_4)(P_6 + 0.31P_8) \quad (31)$$

$$\varepsilon_{cc}{}^{SCP}{}_4 = 0.63P_6 + 0.003P_6{}^5 - 0.00022P_1{}^2P_6{}^2 \quad (32)$$

$$\varepsilon_{cc}{}^{SCP}{}_5 = 0.0011P_6(1.77P_5 - 14.1P_1 + 168 - 32.4P_8) \quad (33)$$

$$\varepsilon_{cc}{}^{SCP}{}_6 = 0.512P_6 - 1.51(0.0116P_6P_1 + 0.00546P_8{}^2)^4 \quad (34)$$

In terms of prediction accuracy and formula compactness, this paper suggests using $f'_{cc}{}^{GP}{}_5$, $f'_{cc}{}^{WGP}{}_5$ and $f'_{cc}{}^{SCP}{}_4$ for predicting f'_{cc} in RMSEs of 1.49/1.79 MPa, 1.49/2.19 MPa and 2.00/1.92 MPa; and $\varepsilon_{cc}{}^{GP}{}_5$, $\varepsilon_{cc}{}^{WGP}{}_4$ and $\varepsilon_{cc}{}^{SCP}{}_5$ for ε_{cc} at 0.084/0.092 MPa, 0.079/0.069 MPa and 0.101/0.102 MPa, respectively. Although SCP has shortcomings in prediction accuracy, the forms of SCP formulas are good and simple to read and study. Therefore, $f'_{cc}{}^{SCP}{}_4$ and $\varepsilon_{cc}{}^{SCP}{}_5$ are finally proposed for circular concrete columns (Figs. 5 and 6).

Table 4 GPS results for confined strain of circular concrete columns

No. of Layers *NL*		GP			WGP			SCP		
		Training/testing RMSE	Time (hr)	Count	Training/testing RMSE	Time (hr)	Count	Training/testing RMSE	Time (hr)	Count
2	AVG	0.231/0.167	0.27	1	0.151/0.116	0.32	1	0.197/0.134	0.32	1
	STD	0.032/0.021		0	0.040/0.016		0	0.004/0.018		0
	Selected	0.201/0.147		1	0.119/0.091		1	0.192/0.114		1
3	AVG	0.136/0.112	0.36	2.8	0.128/0.099	0.50	2.6	0.181/0.126	0.50	2.6
	STD	0.036/0.014		0.5	0.034/0.012		0.5	0.022/0.021		0.7
	Selected	0.092/0.095		3	0.089/0.098		3	0.158/0.067		3
4	AVG	0.136/0.141	0.54	6.6	0.116/0.101	0.86	6.4	0.154/0.124	0.85	5.3
	STD	0.039/0.055		1.1	0.043/0.041		1.1	0.039/0.026		1.7
	Selected	0.102/0.079		7	**0.079/0.069**		**7**	0.104/0.080		7
5	AVG	0.110/0.108	0.90	12.7	0.088/0.082	1.58	12.0	0.150/0.124	1.56	8.6
	STD	0.027/0.031		2.1	0.015/0.015		4.0	0.042/0.024		3.9
	Selected	**0.084/0.092**		**12**	0.075/0.063		15	**0.101/0.102**		**5**
6	AVG	0.110/0.115	1.52	26.9	0.102/0.100	3.04	24.7	0.167/0.128	2.97	8.3
	STD	0.032/0.032		3.4	0.016/0.046		4.2	0.037/0.050		7.5
	Selected	0.086/0.086		24	0.088/0.073		20	0.122/0.087		13

Table 5 Parameter impacts for modeling confined compressive strength of circular concrete columns

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
$f'_{cc}{}^{GP}{}_2$		■						
$f'_{cc}{}^{GP}{}_3$		■				■		
$f'_{cc}{}^{GP}{}_4$		■		■		■	■	■
$f'_{cc}{}^{GP}{}_5$	■	■		■	■	■	■	■
$f'_{cc}{}^{GP}{}_6$	■	■		■		■	■	■
$f'_{cc}{}^{WGP}{}_2$		■				■		
$f'_{cc}{}^{WGP}{}_3$		■			■			
$f'_{cc}{}^{WGP}{}_4$		■				■		
$f'_{cc}{}^{WGP}{}_5$					■	■		■
$f'_{cc}{}^{WGP}{}_6$		■				■		■
$f'_{cc}{}^{SCP}{}_2$		■				■		
$f'_{cc}{}^{SCP}{}_3$		■				■		
$f'_{cc}{}^{SCP}{}_4$		■				■		■
$f'_{cc}{}^{SCP}{}_5$		■				■		
$f'_{cc}{}^{SCP}{}_6$		■				■		

Since the GPS formulas are visible, they make parameter studies easy. Tables 5 and 6 show the number of times input parameters occur in GPS formulas. Thus, P6 is significant when modeling f'cc and εcc. Although P1 and P2 are independent parameters, they can be substituted for each other (same as findings in Oreta and Kawashima 2003). Additionally, P1 and P2 are important when modeling f'_{cc} and ε_{cc}. Conversely, P_3 has almost no effect on f'_{cc} and ε_{cc} according to current datasets. The remaining parameters occur sometimes in the formulas. Additional conformation is needed because the number of datasets is not large enough to omit specific parameters. However, the significances of P_1, P_2 and P_6 are checked with current datasets.

4.1 GPS Predictions and visible formulas

Sensitivity analysis can be applied to assess the impact of parameters (Scardi and Harding 1999). 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.

$$SA^2{}_i = V(M-\delta_i) - V(M) \tag{36}$$

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 (Table 2) and V is an output value calculated by SCP formulas. Therefore, $SA^1{}_i$ is a sensitivity measurement with a positive variation on the i-th input and $SA^2{}_i$ with a negative variation. Table 7 lists the SCP sensitivity results. When an input parameter is linear in an SCP formula, the values of $SA^1{}_i$ and $SA^2{}_i$ are the same. As the effect of the i-th input on output increases, $SA^1{}_i$ or $SA^2{}_i$ increase. When an input is insensitive to outputs, it can be omitted or replaced. For instance, P_8 can be omitted to compact $f'^{SCP}_{cc\ 4}$. Consequently, only P_2 and P_6 impact on all SCP formulas for f'_{cc}. Notably, P_1 and P_6, especially P_6, are important to SCP formulas for ε_{cc}. Although parts of aforementioned results have been discussed in Section 4.1, sensitivity analysis provides convincing evidence of the impact of parameters.

4.2 Sensitivity analysis

Another method for studying the impact of parameters is pruning technique (Peng *et al.* 2009), which replaces an input with a fixed value (e.g., a mean or median) to determine the effect of

Table 6 Parameter impacts for modeling confined strain of circular concrete columns

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
$\varepsilon_{cc\ 2}^{GP}$						■		
$\varepsilon_{cc\ 3}^{GP}$		■				■		
$\varepsilon_{cc\ 4}^{GP}$	■		■		■	■		■
$\varepsilon_{cc\ 5}^{GP}$	■	■		■	■	■	■	
$\varepsilon_{cc\ 6}^{GP}$		■			■	■		
$\varepsilon_{cc\ 2}^{WGP}$		■				■		
$\varepsilon_{cc\ 3}^{WGP}$		■				■		
$\varepsilon_{cc\ 4}^{WGP}$		■				■		
$\varepsilon_{cc\ 5}^{WGP}$		■				■	■	
$\varepsilon_{cc\ 6}^{WGP}$	■	■				■		
$\varepsilon_{cc\ 2}^{SCP}$						■		
$\varepsilon_{cc\ 3}^{SCP}$				■	■	■		■
$\varepsilon_{cc\ 4}^{SCP}$	■					■		
$\varepsilon_{cc\ 5}^{SCP}$	■				■	■		■
$\varepsilon_{cc\ 6}^{SCP}$	■					■		■

Table 8 Pruning and compacting $f'^{SCP}_{cc\ 4}$

No.	Remove	$f'^{SCP}_{cc\ 4}$ formula	RMSE (MPa)
(a)	--	$1.96P_2-24.6+7.40P_6+0.115P_8$	2.14/1.66
(b)	P_2	$24.8+7.40P_6+0.115P_8$	7.90/8.38
(c)	P_6	$1.96P_2-13.4+0.115P_8$	4.82/4.70
(d)	P_8	$1.96P_2-24.4+7.40P_6$	2.02/1.90
(e)	P_8	$AP_2-B+CP_6=1.88P_2-22.4+7.48P_6$	1.97/1.97
(f)	P_2, P_6	$36.0+0.115P_8$	10.7/11.3
(g)	P_2, P_8	$25.0+7.4P_6$	7.94/8.38
(h)	P_6, P_8	$1.96P_2-13.2$	4.85/4.72

the input on outputs. Furthermore, the pruning technique can be used to prune/compact formulas. Selecting $f'_{cc}{}^{SCP}{}_4$ as an example, it has training/testing RMSEs at 2.14/1.66 MPa originally (step (a) in Table 8). The process continues replacing parameters one by one. As P_2 is replaced in $f'_{cc}{}^{SCP}{}_4$, RMSEs increase to 7.90/8.38 MPa. Thus, P_2 cannot be replaced by its mean.

Another method for studying the impact of parameters is pruning technique (Peng *et al.* 2009), 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 $f'_{cc}{}^{SCP}{}_4$ as an example, it has training/testing RMSEs at 2.14/1.66 MPa originally (step (a) in Table 8). The process continues replacing parameters one by one. As P_2 is replaced in $f'_{cc}{}^{SCP}{}_4$, RMSEs increase to 7.90/8.38 MPa. Thus, P_2 cannot be replaced by its mean. After steps (b)–(d), P_8 is a good candidate for removal from $f'_{cc}{}^{SCP}{}_4$ with an accuracy at 2.02/1.90 MPa. However, coefficients in the formula of step (d) are not yet optimized. The GAs can 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 8) remove two parameters at a time. Finally, the formula from step (e) is applied to prune $f'_{cc}{}^{SCP}{}_4$. Furthermore, the pruned and optimized $f'_{cc}{}^{SCP}{}_4$ 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 SCP. Finally, the formula in step (e) (Table 8) is applied to model f'_{cc} and two parameters are involved. The formula is presented below with an accuracy of 1.97/1.97 MPa, indicating that f'_{co} and ρ_s are essential to model f'_{cc}.

$$f'_{cc}{}^{SCP} = 1.88P_2 - 22.4 + 7.48P_6 \tag{37}$$

Other $f'_{cc}{}^{SCP}{}_{NL}$ formulas may still follow the pruning steps (Table 8) and achieve accurate and compact formulas. Of course, pruning technique can also be applied to $\varepsilon_{cc}{}^{SCP}{}_5$. When P_5 and P_8 are replaced by their means, the associated RMSEs are 0.126/0.115. The pruned $\varepsilon_{cc}{}^{SCP}{}_5$ is shown in Eq. (38). Sequentially, coefficients in Eq. (38) can be optimized by the GAs; the final formula is in Eq. (39) with RMSEs of 0.116/0.115.

$$\varepsilon_{cc}{}^{SCP}{}_5 = 0.0011P_6(-14.1P_1 + 727) \tag{38}$$

$$\varepsilon_{cc}{}^{SCP}{}_5 = 0.00107P_6(-19.7P_1 + 870) \tag{39}$$

Using a final formula for ε_{cc} that is the same as that in Eq. (37) is not suitable, as formats of $\varepsilon_{cc}{}^{SCP}{}_{NL}$ formulas are not in common. However, Eq. (39) states that f'_c and ρ_s are significant when modeling ε_{cc} in accordance with current data and achieve 0.116/0.115 in accuracy.

5. Conclusions

This paper proposes a novel GPS, composed of GP, WGP and SCP, for modeling confined compressive strength and strain of circular concrete columns. All models in GPS have good prediction accuracy and visible formulas for target strength and strain. The most significant findings in work are as follows.

1. Improvements to GP and WGP are achieved by attaching weight coefficients to parameter nodes and a terminate operator to decrease tree complexity and make formulas compact, respectively.

2. The SCP provides polynomials for compressive strength and strain of circular concrete columns. This is a good approach for generating simple solutions to problems.

3. Visible formulas are bonus production of GPS models comparing black-box NN approaches. Unlike analytical models, a prior equation format should be assigned. The proposed GPS models provide functional relationships directly for confined strength and strain based on fitting data.

4. Visible formulas increase the ease of parameter studies, sensitivity analysis, and application of pruning techniques.

5. Eq. (37) was applied to model confined compressive strength based on current data. Compressive strength of unconfined concrete specimens of the same size and geometry f'_{co}, and ratio of volume of lateral reinforcement to volume of confined concrete core, ρ_s, are significant when modeling f'_{cc}. Additionally, compressive strength of unconfined concrete specimens of the same size and geometry, f'_c, and ratio of volume of lateral reinforcements to the volume of confined concrete core, ρ_s, are significant when modeling ε_{cc}. Therefore, concrete compressive strength and lateral steel ratio are important to both confined compressive strength and strain of circular concrete columns.

References

Baykasoglu, A., Güllü, H., Çanakçi, H. and Ozbakir, L. (2008), "Prediction of compressive and tensile strength of limestone via genetic programming", *Expert 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", *Expert Syst. Appl.*, **36**(3), 6145-6155.

Bhattacharya, M., Abraham, A. and Nath, B. (2001), "A linear genetic programming approach for modeling electricity demand prediction in Victoria", *Proceeding 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. Concrrete*, **7**(3), 271-283.

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

Gen, M. and Cheng, R. (1997), *Genetic algorithms and engineering design*, New York, Wiley.

Holland, J.H. (1975), *Adaptation in neural and artificial systems, The University of Michigan Press*, Ann Arbor.

Hoshikuma, K., Kawashima, K., Nagaya, K. and Taylor, A.W. (1997), "Stress-strain model for confined reinforced concrete in bridge piers", *J. Struct. Eng.*, **123**(5), 624-633.

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.

Koza, J.R. (1992), *Genetic programming: On the programming of computers by means of natural selection*, MIT Press.

Mander, J.B., Priestley, M.J.N. and Park, R. (1988), "Observed stress-strain behavior of confined concrete", *J. Struct. Eng.*, **114**(8), 1827-1849.

Michalewicz, Z. (1996), *Genetic algorithms + data structures = evolution programs*, 3rd edn, Springer, New York.

Oltean, M. and Dumitrescu, D. (2002), "Multi expression programming, technical report, UBB-01
-2002", Babes-Bolyai University, Cluj-Napoca, Romania.

Oreta, A.W.C. and Kawashima, K. (2003), "Neural network modeling of confined compressive strength and strain of circular concrete columns", *J. Struct. Eng.*, **129**(4), 554-561.

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.

Penelis, G.G. and Kappos, A.J. (1997*), Earthquake-resistant concrete structures*, E&FN Spon, London, Sec. **7**(4), 177-196.

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.

Saatcioglu, M. and Razvi, S.R. (1992), “Strength and ductility of confined concrete”, *J. Struct. Eng.*, **118**(6), 1590-1607.

Sakai, J. (2001), *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.

Sakai, J., Kawashima, K., Une, H. and Yoneda, K. (2000), “Effect of tie spacing on stress-strain relation of confined concrete”, *J. Struct. Eng.*, **46**(3), 757-766.

Scardi, M. and Harding, L.W. (1999), “Developing an empirical model of phytoplankton primary production: a neural network case study”, *Ecol. Model.*, **120**(2), 213-223.

Tsai, H.C. (2009), “Hybrid high order neural networks”, *Appl. Soft. Comput*., **9**(3), 874-881.

Tsai, H.C. (2010), “Predicting strengths of concrete-type specimens using hybrid multilayer perceptrons with center-unified particle swarm optimization”, *Expert Syst. Appl.*, **37**(2), 1104-1112.

Tsai, H.C. (2011), “Using weighted genetic programming to program squat wall strengths and tune associated formulas”, *Eng. Appl. Artif. Intell.*, **24**(3), 526-533.

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. and Lien, L.C. (2009), “Knowledge discovery of concrete material using genetic operation trees”, *Expert Syst. Appl.*, **36**(3), 5807-5812.

CC