The 2018 Structures Congress (Structures18)
Songdo Convensia, Incheon, Korea, August 27 - 31, 2018

Using genetic programming to predict torsional strength of reinforced concrete beams

*Hsing-Chih Tsai[1)]

[1)] *Department of Civil and Construction Engineering, National Taiwan University of Science and Technology, Taiwan.*
[1)] *tsaihsingchih@gmail.com*

ABSTRACT

The potential of using genetic programming to predict engineering data has caught the attention of researchers in recent years. This paper utilizes a derivative of genetic programming to model the torsional strength of reinforced concrete beams using polynomial-like equations. The prediction results include an accurate estimation and a functional input-output mapping relationship, i.e. polynomial-like equations. The proposed polynomial link the torsional strength to the area enclosed by the center of the stirrups, the cross sectional area of one leg of a closed stirrup over the spacing of the stirrups, the yield strength of the closed stirrups, and the compressive strength of concrete. A comparative study is conducted to evaluate the proposed polynomial versus other artificial intelligent models and building codes. In a summary, the proposed polynomial has a simple formulation and provide better prediction performance than the other models.

1. INTRODUCTION

Monolithic reinforced concrete (RC) structures are subject to significant torques, which affect their strength and may cause case deformation. Many articles have been published on the behavior of RC components under pure tension and under tension in combination with axial loads, shear loads, bending moments, and other load scenarios. However, developing an efficient model to describe the actual physical behavior of RC components continues to pose a significant challenge for researchers. The complex nature of RC components continues to make the torsional design of these components a complicated process that requires significant engineering experience and expertise (Cevik et al., 2010). The torsional strength of RC beams is currently estimated using predictions generated by analytical models, finite element analysis, or artificial intelligence (AI). Analytical models such as the skew-bending theory and the space-truss analogy are widely used to handle the torsional strength design of RC beams, and the latter is widely incorporated into building codes worldwide. However, variations in

---

[1)] Associate Professor

building codes may generate widely divergent values (Fiore et al., 2012). Finite element analysis seems an appropriate tool for crack estimation in RC components. Nevertheless, the simulations necessary to crack concrete and to test the bonding stress between steel bars and concrete are difficult to conduct and are subject to computational problems that are caused by nonlinearities (Gandomi and Roke, 2015). Artificial intelligence (AI) predictions such as those made by artificial neural networks (ANN) commonly offer results that are significantly more accurate than those generated by either analytical models or finite element analysis. However, ANN have been characterized as “black-box” models due to the extremely large number of nodes and connections within their layered structures (Tsai, 2009; Tsai, 2010).

Since it was first proposed by Koza (1992), genetic programming (GP) has garnered considerable attention due to its ability to model nonlinear relationships for input-output mappings without assuming the prior form of these relationships. GP is sometimes called a grey-box model due to its ability to generate prediction equations against black-box models. Baykasoglu et al. (2008) compared a promising set of GP approaches, including Multi Expression Programming (MEP) (Oltean and Dumitrescu, 2002), Gene Expression Programming (GEP) (Ferreira, 2001), and Linear Genetic Programming (LGP) (Bhattacharya et al., 2001). Notably, LGP has proven the most efficient GP algorithm in case studies of limestone strength. Differences between these algorithms are rooted in the methodology that is utilized to generate a GP individual. A chromosome representation, a tree topology, and a linear string are used by MEP, GEP, and LGP, respectively. Although some of the formulas that are generated by MEP, GEP, LGP and GOT use coefficients, all of these coefficients are fixed constants (Baykasoglu et al., 2008). As coefficient constants are not often appeared in formulas that are programmed using any of these GP models, Giustolisi and Savic (2006) argued that GP is not very powerful in finding constants. Consequently, Tsai (2011) proposed a weighted GP (WGP) to introduce weight coefficients into tree connections, generate a fully weighted formula, and provide coefficient constants for the obtained GP equations. This paper attempts to utilize the WGP to predict the torsional strength of reinforced concrete beams.

2. THEORIES OF TORSIONAL STRENGTH AND TORSION IN BUILDING STANDARDS

The skew-bending theory and the space truss analogy are the two theoretical models used to develop the RC beam torsional strength requirements in various national building codes. The former was the basis of the American building code up to 1995 and the latter is the basis of the building codes that are currently in force in America and Europe (Fiore et al., 2012).

The skew-bending theory considers the assumption of a skew-failure surface. The failure of a rectangular section in torsion occurs because of bending around an axis that is parallel to the wide face of the section and inclined at about 45 degrees to the longitudinal axis of the beam. Prior versions of the ACI code (1971 - 1989) used the skew-bending theory beam to calculate torsional strength (T_n). This theory considers two parts: the effects of concrete (T_c) and the contributions of reinforcement (T_s). Hsu (1968) compared hollow and solid rectangular beams, and calculated that the concrete core does not contribute to torsional strength due mainly to the shear resistance of the diagonal concrete struts.

ACI-318-2005 (2005) radically changed considerations of T_n based on the space truss analogy, which was first proposed by Rausch (1929). This analogy holds that concrete is separated by cracks into a series of constant-angle helical members, which are assumed to interact with the longitudinal steel bars and the stirrups to form a space truss. The circulatory shear stresses that develop in the cross-section of the space truss produce an internal torque capable of resisting the applied torsional moment. As the concrete core does not influence torsional strength (Hsu, 1968), the beam may be considered as the equivalent of an ideal tubular element.

According to the ACI code (ACI-318-2005, 2005), closed stirrups and longitudinal steel bars only contribute to T_n, while the contribution of concrete is ignored. The torsional strength T_n is given as follows:

$$T_n = \frac{2A_0 A_{st} f_{yv}}{s}\cot\theta \text{ ,(1)}$$

$$\cot\theta = \sqrt{\frac{A_{sl} f_{yl} s}{A_{st} f_{yv} p_h}} \text{ ,(2)}$$

where A_0 is the gross area enclosed by the shear flow path that is equal to $0.85A_{sh}$; A_{sh} is the area enclosed by the center of the stirrups; A_{st} is the cross sectional area of one leg of a closed stirrup; f_{yv} is the yield strength of the closed stirrups; θ is the angle of the compression diagonals; s is the spacing of the stirrups; A_{sl} is the total area of the longitudinal torsional reinforcement; f_{yl} is the yield strength of the torsional reinforcement; and p_h is the perimeter of the centerline of the outmost transverse torsional reinforcement.

Under Australian Standard AS3600 (2001) and Canadian Standard CSA (1994), the torsional strength T_n is expressed using the same equation as that used in ACI-318-2005 (2005). However, British Standard BS8110 (1985) calculates torsional strength T_n differently, as:

$$T_n = \frac{0.8x_1 y_1 (0.87 f_{yv}) A_{sv}}{s} \text{ ,(3)}$$

where x_1 and y_1 are the center-to-center distances of the shorter and longer stirrup legs, respectively, and their product equals A_{sh}, i.e., x_1y_1=A_{sh}. A_{sv} is the area of the two legs of the stirrups in a section and is equal to twice the value of A_{st}, i.e., A_{sv}=$2A_{st}$.

Under European Standard Eurocode 2 (2002), the torsional strength T_n is furnished in three different expressions that represent the torsional resistances of concrete, shear reinforcement, and longitudinal steel bars, respectively. The minimum result among the three is selected as the final T_n, which is calculated under the concept of the equivalent thin-walled section, as follows:

$$T_n = 1.2(1-\frac{f_c}{250}) f_c A_k t_{ef} \sin\theta \cos\theta \text{ ,(4)}$$

$$T_n = \frac{2A_k A_{st} f_{yv}}{s} \cot\theta \text{ ,(5)}$$

$$T_n = \frac{2A_k A_{sl} f_{yl}}{\mu_k} \cot\theta \text{ ,(6)}$$

where f_c is the compressive strength of concrete; A_k is the area enclosed by the centerline of the connecting walls, which may be assumed to equal $x_1 y_1$ (Fig. 1); t_{ef}, the effective wall thickness, may be calculated as A/u, with A equal to the total area ($x_0 y_0$) and u equaling the outer circumference of the cross-section ($2x_0+2y_0$); and μ_k is the perimeter of area A_k ($2x_1+2y_1$).

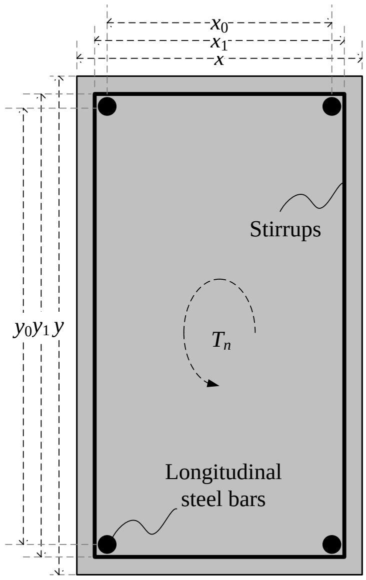

Fig. 1 The cross-section of a rectangular RC beam

3. DATABASE FOR RC BEAM TORSIONAL STRENGTH

Cevik et al. (2010) gathered 76 datasets for RC beam torsional strength from 5 articles. Same as the original suggestions, of the 76 datasets, 61 were used for training and 15 were selected for testing. The test specimens were solid, non-deep-beam rectangular beams that were subjected to pure torsion. Twelve influenced parameters were identified to determine the T_n. The experimental database may be obtained through either Cevik et al. (2010) or Fiore et al. (2012), with the data ranges listed in Table 1. The values for A_{sh} were obtained by multiplying x_1 and y_1 and ρ_t and ρ_l as the steel ratios, respectively, for stirrups and longitudinal reinforcement. However, instead of adopting the twelve influenced parameters as the input parameters for the GP training, this paper followed the suggestions of Fiore et al. (2012) by adopting five parameters as the candidate explanatory variables for T_n.

Table 1 Data ranges of influenced parameters and prediction targets for the database of RC beam torsional strength

Parameter	Min.	Max.	Mean	S.D.
x (mm)	160	350	253	60.8
y (mm)	275	508	404	89.4
x_1 (mm)	130	300	214	54.7
y_1 (mm)	216	470	344	87.6
f_c (MPa)	25.6	110	46.9	23.5
s (mm)	50.0	216	102	34.6
A_{st} (mm^2)	71.3	217	97.2	26.8
f_{yv} (MPa)	319	672	411	117
A_{sl} (mm^2)	381	3438	1231	720
f_{yl} (mm^2)	310	638	412	111
ρ_t (%)	0.220	2.56	1.20	0.510
ρ_l (%)	0.300	3.51	1.36	1.01
P_1: A_{sh} (10^4×mm^2)	3.19	13.5	7.74	3.63
P_2: A_{st}/s (mm)	0.330	2.34	1.07	0.494
P_3: A_{sl}/ρ_l (10^2×mm^2/%)	4.35	17.6	10.7	4.47
P_4: f_{yv} (10^2×MPa)	3.19	6.72	4.11	1.17
P_5: f_c (10^2×MPa)	0.256	1.10	0.469	0.235
T_n (kN×m)	11.3	239	55.3	47.1

4. WEIGHTED GENETIC PROGRAMMING

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

Tsai (2011) introduced a weighted balance for tree-based GP to create weighted genetic programming (WGP). Weights are attached to all of the branches of the WGP tree structure in order to balance the impacts of the two front nodes (Fig. 2). The WGP uses parameter selection to adopt inputs from the bottom layer, executes operator selection to determine operators for nodes above the bottom layer, and then outputs functional programs from the top node. The parameter set (*PS*) includes all input parameters (*P*) and a unit parameter "1", which produces a constant for the branch. Various functions, including transcendental ones, may be selected for the operator set (*OS*) based on the specific demands of individual users. In contrast, this paper adopted basic functions for the *OS* while aiming at generating polynomial-like equations. Thus, the proposed weighted genetic programming polynomial model adopts *PS* and *OS* as:

$$PS = \{1 \quad P_1 \quad P_2 \quad \ldots \quad P_{NI}\} \text{ ,(7)}$$

$$OS = \{T \quad N \quad S \quad B \quad + \quad \times \quad / \quad \wedge\} \text{ ,(8)}$$

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

$$y = f(w_i, \quad w_j, \quad x_i, \quad x_j, \quad x_{end}) = \begin{cases} x_{end} & , OS = E \\ x_i & , OS = N \\ w_i x_i & , OS = S \\ x_i + w_i & , OS = B \\ w_i x_i + w_j x_j & , OS = + \\ w_i x_i \times w_j x_j & , OS = \times \\ w_i x_i / w_j x_j & , OS = / \\ w_i x_i{}^p & , OS = \wedge \end{cases} \quad ,(9)$$

where nodal output y is the function of the nodal values (x_i and x_j) of the two front nodes, the connection weights (w_i and w_j), and, occasionally, the branch end on the extreme left-hand side (x_{end}; Fig. 3); the *E* operator is a terminate operator that directly adopts the branch end on the extreme left-hand side x_{end}; the *N* operator is a next-layer operator that directly inherits x_i; the *S* operator handles scaling for x_i; the *B* operator tackles shifting for the x_i; and the last four operators deal with summation, multiplication, division, and power operations, respectively. In engineering, large values are infrequently adopted for exponents. Therefore, candidate exponents for the p of the "^" operator were considered [-2, -1.5, -1, -0.5, 0, 0.5, 1, 1.5, 2] (Berardi et al., 2008), as determined by the transformation of w_j.

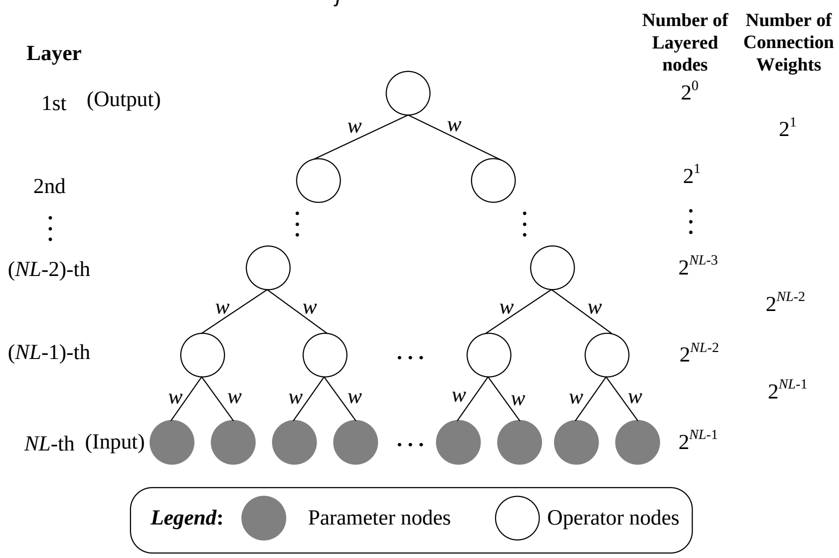

Fig. 2 Weighted Genetic Programming Structure

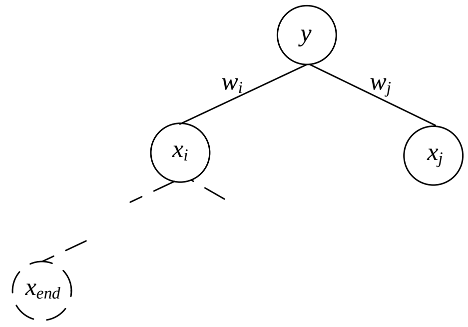

Fig. 3 Nodal value of a WGP node

Table 2 Statistical results of WGP learning for RC beam torsional strength

# layers		Training	Testing	Training	Testing	Active # operator nodes	Active # parameter nodes
		RMSE (kN×m)		R^2(%)			
2	Avg.	25.14	17.98	73.7	72.0	1.00	1.77
	Best	25.32	12.87	73.5	86.6	1	1
3	Avg.	21.87	17.30	79.6	73.4	2.57	2.57
	Best	11.33	15.25	94.7	81.2	3	4
4	Avg.	18.82	15.89	84.4	77.5	4.70	3.20
	Best	11.76	8.93	94.3	93.6	6	5
5	Avg.	17.04	15.07	87.2	79.8	6.97	3.90
	Best	11.76	8.93	94.3	93.6	6	5

5. RESULTS AND DISCUSSION

Although WGP offers operators like *E* and *N*, which may be used to reduce the size of tree-based structures, a large WGP tree may be easy to obtain owing to the good prediction accuracy. However, the associated polynomial equation may be complicated and thus may not reflect the input-output mapping relationship in meaningful ways. Therefore, the present paper performed and compared WGP from 2 to 5 layers in order to figure out a suitable tree size, and each statistical result adopted 30 runs. Table 2 shows the results of 2 to 5 WGP layers in modeling torsional strength of reinforced concrete beams. The average and best results are presented, with the best results determined by summing the training and testing root mean square errors (RMSE). In terms of the best RMSE results, the results obtained by 4-layered WGP trees approximate those obtained by 5-layered WGP trees. Therefore, 4-layered WGP trees may provide accurate predictions with relatively less computational effort and yield concise polynomials. In addition, the *PS* and *OS* nodes are not fully activated to generate accurate predictions. For instance, the best run of the 4-layered WGP was greatly pruned to 6 active operator nodes and 5 active parameter nodes from its original 7 operator and 8 parameter nodes (see Table 2 and Fig. 3). The bottom tree in Fig. 3 shows a pruned version of the top tree with active attributes. The 4-layered WGP tree was greatly pruned by *N* and ^ operators, which made the final polynomial equations significantly more concise. In looking at the pruned WGP tree as shown in

Fig. 3, the resultant model may be presented as polynomial-like equations. Following the procedure shown in Fig. 3, the best runs of the 2- to 5-layered WGP equations for the RC beam torsional strength with the effects of 5 input parameters may be obtained as:

$T_2 = 2.47P_1^{1.5}$, (10)

$T_3 = (1.27P_1 - 2.71) \times (3.32P_2 + 9.14P_5)$, (11)

$T_4 = 0.142P_1^{2.5} + P_5(78.4P_2 - 7.10P_4)$, (12)

$T_5 = 9.00P_1^{1.125} - \dfrac{1.17P_1 + 7.05P_5}{P_5(P_2 + 0.00210)}$, (13)

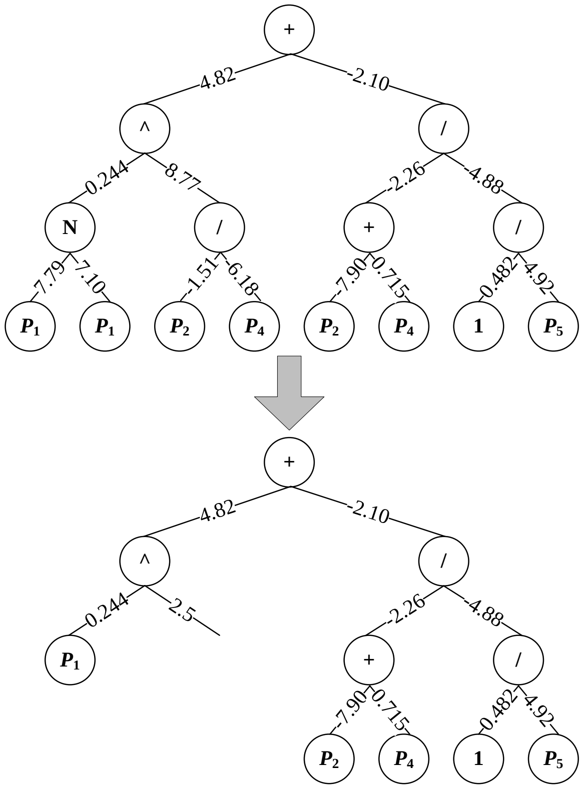

Fig. 3 Structures of the best run of a 4-layered WGP

in which, T represents the polynomial equation for RC beam torsional strength and the sub-index of T represents the number of WGP tree layers. Apparently, both T_4 and T_5 offer good prediction accuracy and they are influenced by P_1, P_2, P_4, and P_5. Consequently, T_4 was selected as the proposed WGP polynomial for modeling the torsional strength of RC beams. In comparing the resultant T_4 with the experiment values, the first 61 cases in Fig. 4 were the training sets and the remaining 15 were the testing sets. The T_4 fits the experiment values much better than building codes. Table 3 gives comparative results for the proposed equation and the three building codes. The proposed equation is a simple equation in polynomial forms and is able to be applied to calculate the torsional strength of RC beams with good accuracy.

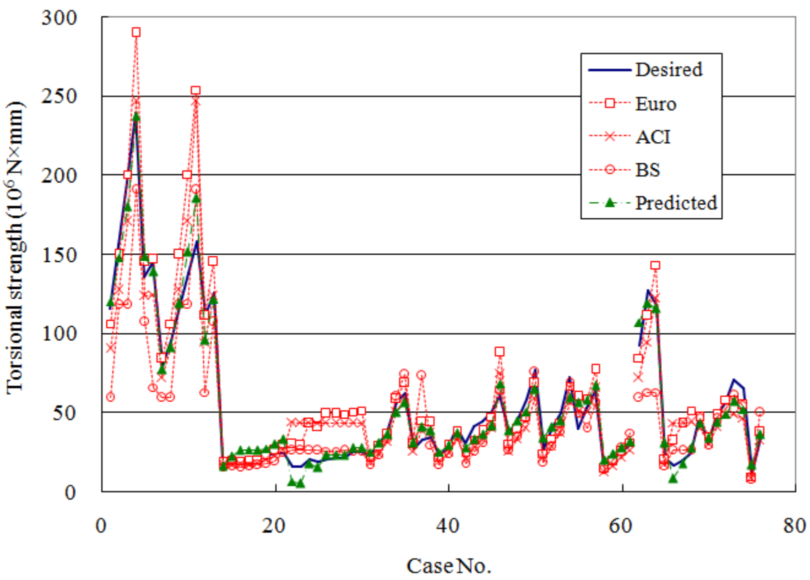

Fig. 4 Torsional strength of desired experiments, European building code, and predicted T_4 values

Table 3 Results of overall R^2, RMSE for the database of RC beam torsional strength

Capacity equation	RMSE (kN×m)	R^2 (%)
ACI-318-2005, Eq. (1-2)	17.7	85.6
BS8110, Eq. (3)	22.7	76.4
Eurocode-2, Eq. (4-6)	19.1	83.3
T_4 in Eq. (12)	11.3	94.2

3. CONCLUSIONS

This study develops WGP, a robust variant of GP, to model RC beam torsional strength with polynomial-like equations. In this regard, a database consisting of 76 specimens are employed. The following conclusions were drawn from this investigation:

- The proposed WGP equation can predict the torsional strength of RC beams easily and acurately.
- Prediction accuracy of the proposed equation is much more accuracy than those of building codes.
- The proposed equation identified that A_{sh}, A_{st}/s, f_{yv}, and f_c impact on the thorsional strength of RC beams. The parameter of A_{sl}/ρ_l is considered negligible for the torsional strength, however additional datasets need to be gathered to confirm this point.
- Civil engineers may use the proposed equation to predict the torsional strength of RC beams and to avoid conducting costly tests.

REFERENCES

ACI 318-2005, 2005. Building code requirements for structural concrete (ACI 318-05) and commentary (318R-05). Farmington Hills, Mich.: American Concrete Institute.

AS3600, 2001. Concrete structures. Standards Association of Australia.

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

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

Bhattacharya M., Abraham A., Nath B., (2001), "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, 379–393.

BS8110, 1985. Structural use of concrete. Part 2. British Standards.

Cevik, A., Arslan, M.H., Köroglu, M.A., (2010), "Genetic-programming-based modeling of RC beam torsional strength," KSCE J Civ Eng, **14**(3), 371–384.

CSA, 1994. Design of concrete structures: structure design. CSA Standard, A23-3-94. Rexdale (Ontario): Canadian Standard Association.

Eurocode 2, 2002. Eurocode 2. Design of concrete structures, prEN, 1992-1-1, Draft for stage 49, Commission of the European communities, European Committee for Standardization.

Ferreira C., (2001), "Gene Expression Programming: A New Adaptive Algorithm for Solving Problems," Complex Syst, **13**(2), 87–129.

Fiore, A., Berardi, L., Marano, G.C., (2012), "Predicting torsional strength of RC beams by using evolutionary polynomial regression," Adv Eng Software, **47**(1), 178–187.

Gandomi, A.H., Roke, D.A., (2015), "Assessment of artificial neural network and genetic programming as predictive tools," Adv Eng Software, **88**, 63–72.

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

Hsu, T.T.C., (1968), "Torsion of structural concrete-behavior of reinforced concrete rectangular members," Torsion of structural concrete SP-18, ACI, Farmington Hills. Mich; pp. 261–306.

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

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

Rausch E., (1929), Design of reinforced concrete in torsion. Berlin: Technische Hochschule.

Tsai H.C., (2009), "Hybrid High Order Neural Networks," Appl Soft Comput, **9**, 874–881.

Tsai H.C., (2010), "Predicting Strengths of Concrete-type Specimens Using Hybrid Multilayer Perceptrons with Center-Unified Particle Swarm Optimization," Expert Syst Appl, **37**, 1104–1112.

Tsai H.C., (2011), "Using Weighted Genetic Programming to Program Squat Wall Strengths and Tune Associated Formulas," Eng Appl Artif Intell, **24**, 526–533.