World Applied Sciences Journal 5 (2): 215-223, 2008
ISSN 1818-4952

A Multi Expression Programming Application to High Performance Concrete

[1]A.A.R. Heshmati, [1]H. Salehzade, [1]A.H. Alavi, [2]A.H. Gandomi and [3]M. Mohammad Abadi

[1]College of Civil Engineering, Iran University of Science and Technology, Tehran, Iran
[2]College of Civil Engineering, Tafresh University, Tafresh, Iran
[3]Islamic Azad University, Gonabad Branch, Iran

Abstract: High performance concrete (HPC) is a class of concretes that provides superior performance than those of conventional types. The enhanced performance characteristics of HPC are generally achieved by the addition of various cementitious materials and chemical and mineral admixtures to conventional concrete mix designs. These additives considerably influence the compressive strength and workability properties of HPC mixes. To avoid testing several mix proportions to generate a successful mix and also to simulate the behaviour of strength and workability improvement efficiently that often lead to saving in cost and time, it is idealistic to develop predictive models so that the performance characteristics of HPC mixes can be evaluated from the influencing parameters. Accordingly, the main focus of the present study is to propose new formulations of compressive strength and slump flow of HPC mixes for the first time in the literature by means of a promising variant of genetic programming (GP) which is known as multi expression programming (MEP). The models are developed using an experimental database including compressive strength and slump flow of HPC test results obtained from the previously published literature. The results of proposed formulations are compared with other existing models and formulas found in the literature. The results demonstrate that the formulas obtained by the MEP method are able to predict the strength and slump flow to high degree of accuracy.

Key words: High performance concrete · multi expression programming · Formulation · Compressive strength · Workability

INTRODUCTION

Different factors influence the compressive strength and workability properties of high performance concrete (HPC) mixes. A range of materials such as portland cement, silica fume, superplasticizer, fly ash, fine and coarse aggregates and ground-granulated blast furnace slag and combinations of these can be used in order to obtain concrete mix designs with superior performance. Using these material considerably influence the compressive strength and workability properties of HPC mixes. A deep knowledge of the nature of the relationship between these interrelated parameters and the resulting mix is required to provide an effective mix design procedure [1]. The complex behavior of strength and workability improvement and a need to avoid trying several mix proportions to produce a successful mix suggest the necessity to develop comprehensive mathematical models to be able to evaluate the performance characteristics of HPC mixes with high accuracy.

Genetic programming (GP) [2, 3] is a developing subarea of evolutionary algorithms inspired from Darwin's evolution theory. GP may be defined generally as a supervised machine learning technique that searches a program space instead of a data space [3]. Recently, a particular variant of GP that uses a linear representation of chromosomes namely, multi expression programming (MEP) have been proposed. MEP [4] has a special ability to encode multiple computer programs of a problem in a single chromosome. Based on the numerical experiments MEP approach has the ability to significantly outperform similar techniques and can be utilized as an efficient alternative to the traditional Koza's tree-based GP [5]. Despite the significant advantages of MEP, there has been just some little scientific effort directed at applying it to civil engineering tasks [6].

The main purpose of this paper is to utilize MEP technique to obtain formulas for the determination of compressive strength and slump flow of HPC mixes. To our knowledge, this is the first time in the literature to utilize this approach to introduce explicit formulations of

Correspondig Author: H. Salehzade, College of Civil Engineering, Iran University of Science and Technology, Tehran, Iran

the performance characteristics of HPC mixes. In order to evaluate the prediction quality, a comparison between the proposed formulations results, as well as existing models found in the literature, was conducted considering the influencing parameters such sand cement ratio, coarse aggregate cement ratio, water cement ratio, percentage of silica fume and percentage of superplasticizer. A reliable database including previously published compressive strength and slump flow of HPC test results was utilized to develop the models.

Review of Previous Studies: In composite materials like HPC changes in constituent properties of the cementitious materials and chemical admixtures may extremely influence the advantageous performance characteristics of the mix. Numerous studies have concentrated on assessing the performance characteristics of HPC mixes using computational approaches in the literature. Artificial neural networks (ANNs) [7, 8] have a noteworthy quality of learning the relationship between the input and output parameters as a result of training with previously recorded data. ANNs have been applied to predict the compressive strength and slump flow of HPC mixes several times [9-12]. There has been only limited research with the specific objective of opening ANN models adequately and introducing explicit formulations of compressive strength and slump flow of HPC mixes by means of them.

Rajasekaran and Amalraj [12] presented a sequential learning approach (SLA) for single hidden radial basis function (RBF) neuron neural networks proposed by Zhang and Morris [13]. Their developed sequential learning neural network (SLNN) model was utilized for the prediction of strength and workability of high performance concrete. The values for learning rate and gamma have been respectively chosen as 0.6 and 0.000001 for the architecture of their proposed SLNN (RBF) model. In that work two equations were introduced based on experimental results and by using the values of the weights obtained from neural network training to predict the compressive strength (σ) and slump flow (S) that are given as follows:

$$\sigma_{SLNN}(MPa) = 8.2628e^{-W_1} \quad (1)$$

$$S_{SLNN}(mm) = 12.2872e^{-W_2} \quad (2)$$

where,

$$W_1 = \frac{1}{(2.6808)^2}[(\frac{x_1}{2}+2.3357)^2+(\frac{x_2}{4}+1.7148)^2+ (2x_3-1.5897)^2+(\frac{x_4}{30}-1.2737)^2+(\frac{x_5}{5}-1.9704)^2] \quad (3)$$

$$W_2 = \frac{1}{(1.41657)^2}[(\frac{x_1}{2}-1.2086)^2+(\frac{x_2}{2}-3.0106)^2+ (2x_3-0.23047)^2+(\frac{x_4}{30}-0.84016)^2+(\frac{x_5}{5}-0.71905)^2] \quad (4)$$

where,
x_1 sand cement ratio
x_2 coarse aggregate cement ratio
x_3 water cement ratio
x_4 silica fume (%)
x_5 superplasticizer (%)

and x_1, x_2,..., x_5 are the five input parameters to the model. It should be noted that the required data used for the training and testing of the SLNN model described above were taken from [14] and have also been utilized in the present study. In this paper, a novel approach for the formulation of compressive strength and slump flow of HPC mixes using MEP is proposed.

Genetic Programming (GP): Genetic programming (GP) is one of the branches of evolutionary methods that creates computer programs to solve a problem using the principle of Darwinian natural selection.GP was introduced by Koza as an extension of the genetic algorithms, in which programs are represented as tree structures and expressed in the functional programming language LISP [2]. A comprehensive description of GP is beyond the scope of this paper and can be found in [2, 3]. GP has been successfully applied to some of the civil engineering problems [15-18].

Multi Expression Programming (MEP): Multi expression programming (MEP) is a subarea of genetic programming (GP) that was developed by [4Oltean and Dumitrescu (2002)]. MEP uses linear chromosomes for solution encoding and has a special ability to encode multiple solutions (computer programs) of a problem in a single chromosome. According to the fitness values of the individuals, the best of the encoded solutions is chosen to represent the chromosome. Comparing MEP to other GP variants that store a single solution in a chromosome, there are not increases in the complexity of the decoding process except on the situations where the set of training data is not a priori known [5]. The evolutionary steady-

state MEP algorithm starts by the creation of a random population of individuals. In order to evolve the best expression from a data file of inputs and outputs along a specified number of generations, MEP uses the following steps until a termination condition is reached [19]:

- Selecting two parents by using a binary tournament procedure [2] and recombining them with a fixed crossover probability.
- Obtaining two offspring by the recombination of two parents.
- Mutating the offspring and replacing the worst individual in the current population with the best of them (if the offspring is better than the worst individual in the current population).

MEP is represented similar to the way in which *C* and *Pascal* compilers translate mathematical expressions into machine code [20]. The number of MEP genes per chromosome is constant and specifies the length of the chromosome. A terminal (an element in the terminal set T) or a function symbol (an element in the function set F) are encoded by each gene. A gene that encodes a function includes pointers towards the function arguments. Function parameters always have indices of lower values than the position of that function itself in the chromosome. The first symbol in a chromosome must be a terminal symbol as stated by the proposed representation scheme.

An example of a MEP chromosome can be seen below. It should be noted that numbers to the left stand for gene labels that do not belong to the chromosome. Using the set of functions $F = \{+, *, /\}$ and the set of terminals $T = \{x_1, x_2, x_3, x_4\}$, the example is given as follows:

0: x_1
1: x_2
2: * 0, 1
3: x_3
4: + 2, 3
5: x_4
6: / 4, 5

The translation of MEP individuals into computer programs can be obtained by reading the chromosome top-down starting with the first position. A terminal symbol defines a simple expression and each of function symbols specifies a complex expression obtained by connecting the operands specified by the argument positions with the current function symbol [19]. In the present example, genes 0, 1, 3 and 5 encode simple expressions formed by a single terminal symbol. These expressions are:

$E_0 = x_1$,
$E_1 = x_2$,
$E_3 = x_3$,
$E_5 = x_4$,

Gene 2 indicates the operation * on the operands located at positions 0 and 1 of the chromosome. Therefore gene 2 encodes the expression:

$E_2 = x_1 * x_2$.

Gene 4 indicates the operation + on the operands located at positions 2 and 3. Therefore gene 4 encodes the expression:

$E_4 = (x_1 * x_2) + x_3$.

Gene 6 indicates the operation / on the operands located at positions 4 and 5. Therefore gene 6 encodes the expression:

$E_6 = ((x_1 * x_2) + x_3) / x_4$.

In order to choose one of these expressions ($E_1,...,E_6$) as the chromosome representer, multiple solutions in a single chromosome are encoded. Each of MEP chromosomes encodes a number of expressions equal to the chromosome length (the number of genes). Each of these expressions can be considered as a possible solution of a problem. The fitness of each expression encoded in a MEP chromosome is defined as the fitness of the best expression encoded by that chromosome. For solving symbolic regression problems the fitness of a MEP chromosome may be computed by using the formula [5]:

$$f = \min_{i=1,m} \left\{ \sum_{j=1}^{n} \left| E_j - O_j^i \right| \right\}, \tag{5}$$

where n is the number of fitness cases, E_j is the expected value for the fitness case j, O_j^i is the value returned for the jth fitness case by the ith expression encoded in the current chromosome and m is the number of chromosome genes.

Table 1: Database used in developing the models

Mix No.	FA/C	CA/C	W/C	SF (%)	SP (%)	σtest (MPa)	Stest (mm)
Training							
1	1.88	2.84	0.45	9.99	2	80	110
2	1.6	2.4	0.4	9.99	2	90.08	90
3	1.8	2.68	0.45	19.98	3	110.08	120
4	1.3	2.12	0.35	9.99	2	110.08	80
5	0.96	1.72	0.3	5.01	2	130.08	50
6	1.02	1.8	0.3	15.03	2.5	120	60
7	1.08	1.92	0.3	15.03	2.5	130.08	70
8	0.78	1.52	0.25	9.99	2	140	40
9	1.64	2.44	0.35	12	2	80	100
10	1.7	2.56	0.35	15.99	2.5	90.08	120
11	1.8	2.68	0.35	20.01	3	90.08	130
12	1.9	2.84	0.35	24	3.5	90.08	150
13	1.98	3	0.35	27.99	4	90.08	170
14	1.38	2.28	0.35	7.5	2	90.08	80
Testing							
15	1.64	2.48	0.4	9.99	2	80	100
16	1.76	2.64	0.43	15	2.5	100	110
17	1.92	2.888	0.45	9.99	2	80	120
18	1.36	2.2	0.36	9.99	2	100	80
19	1.44	2.36	0.38	15	2.5	100	90
20	1.54	2.52	0.46	20.01	3	140	110
21	1.16	2.08	0.32	9.99	2	110.08	70
22	1.1	1.96	0.3	5.01	2	100	60
23	1.24	2.2	0.34	15	2.5	120	90

Table 2: The variables used in model development

Parameters	Range	Normalization value	Code
Inputs			
Sand cement ratio *FA/C*	0.78–1.98	2	x_1
Coarse aggregate cement ratio *CA/C*	1.52–3	4	x_2
Water cement ratio *W/C*	0.25–0.46	0.5	x_3
Silica fume *SF* (%)	5.01–27.99	30	x_4
Superplasticizer (%)	2–4	5	x_5
Outputs			
Compressive strength σ (MPa)	80–140	160	–
Slump flow *S* (mm)	40–170	200	–

Table 3: Parameter settings for MEP

Parameter	Settings s1 , S1	Settings s1 , S1
Function set	+, -, *, /, exp, sin, cos	+, -, *, /
Population size	250-500	250-500
Chromosome length	50 genes	50 genes
Number of generations	250	250
Crossover probability	0.5,0.9	0.5,0.9
Crossover type	Uniform	Uniform
Mutation probability	0.01	0.01
Terminal set	Problem inputs	Problem inputs

Database: The database contains 23 compressive strength and slump flow of HPC test results managed by Rajaseraran *et al*. [14]. Table 1 shows the experimental database used for proposed models. The other cited information in Table 1 consists of sand (FA)/cement (C), coarse aggregate (CA)/cement (C), water (W)/cement (C), silica fume content (%SF) and superplasticizer content (%SP) as the five input parameters to the models to predict the compressive strength(σ)and slump flow (*S*) of HPC mixes as the outputs. It is noteworthy that the input and output parameters entering the models have been normalized between 0 and 1 before the learning process. The range of the samples, normalization values and the format of the input data used in this study are given in Table 2.

Model Development Using MEP Model: The main goal is to obtain the explicit formulations of the compressive strength and slump flow of HPC mixes as functions of variables given as follows:

$$\sigma, s = f\left(\text{FA/C, CA/C, W/C, SF, SP}\right)$$

The five parameters are used for the MEP models as the input variables. Two MEP models for single output have been separately used, one for strength and the other for the slump flow. In order to evaluate the capabilities of the MEP model, the correlation coefficient (R), mean squared error (MSE) and mean absolute error (MAE) are used as the criteria between the actual and predicted values. Various parameters are involved in MEP predictive algorithm such as population size, chromosome length, number of generations, tournament size and other parameters that are shown in Table 3. The parameter selection will affect the model generalization capability of MEP. They were selected based on some previously suggested values [6] and also after trial and error approach. Four formulations of compressive strength and slump flow, two formulas for each of them, have been considered using two different function sets for runs. The first function set consists of nearly all functions and the latter includes just addition, subtraction, division and multiplication in order to obtain short and very simple formulas. Subsequently, the results obtained from these equations were compared with each other, as well as the results obtained by other researcher. For the analysis, source code of MEP [21] in C++ was modified by the authors to be utilizable for the available problems. For the prediction of compressive strength and slump flow of HPC the first 14 values of Table 1 are taken for training the

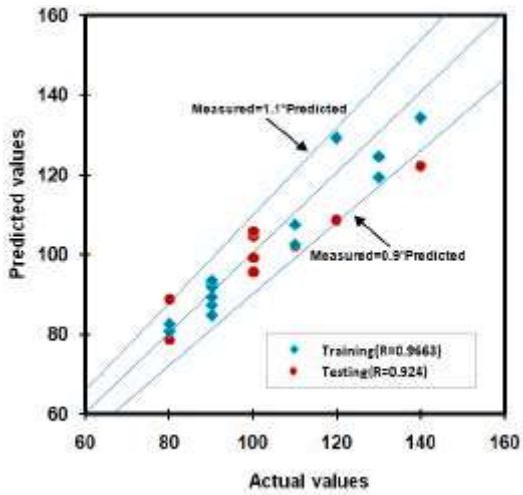

Fig. 1: Results of compressive strength prediction for Eq. (6).

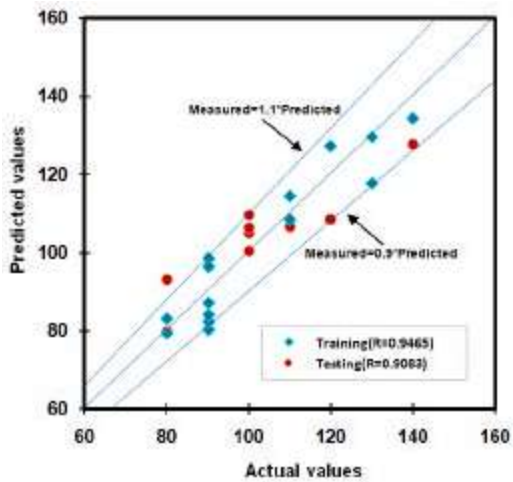

Fig. 2: Results of compressive strength prediction for Eq. (7).

MEP algorithm and the next 9 values were used for testing the generalization capability of MEP based models. The details of the compressive strength and slump flow predictive models are highlighted in next sections.

Explicit Formulation of Compressive Strength and Analysis Using MEP Model: Formulation of compressive strength in functional form in terms of the independent variables, sand cement ratio (FA/C = x1), coarse aggregate cement ratio (CA/C = x2), water cement ratio (W/C = x3), percentage of silica fume (%SF = x4) and percentage of superplasticizer (%SP = x5) as presented in Table 1, for the best result by the MEP algorithm are given in Eq. (6) and Eq. (7) for two different function sets.

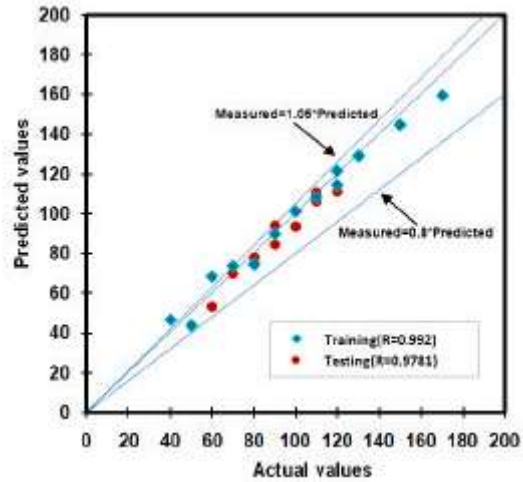

Fig. 3: Results of slump flow prediction for Eq. (8).

$$\sigma_1(MPa) = \frac{320}{x_2}\cos(\cos\left(\frac{x_1}{4} - \frac{x_5}{5} - 2x_3\right)^2) \quad (6)$$

$$\sigma_2(MPa) = 80\ (x_5\left(\frac{x_1 - x_2 + 2}{10}\right) - x_1 - x_2/2 + 4\,x_3 + 2\Big) \quad (7)$$

The comparison of MEP prediction and actual compressive strength of HPC for Eq. (6) is shown in Fig. 1. It can be seen from Fig. 1 that Eq. (6) generated by MEP model yielded high R values for training and testing data equal to 0.9663 and 0.924, respectively. Fig. 2 shows the relevant results obtained by Eq. (7). It can be observed from this figure that Eq. (7) yielded R values for training and testing data equal to 0.9465 and 0.9083, respectively. It can be concluded from these figures that Eq. (6) has better performance than Eq. (7) for both of training and testing sets.

Explicit Formulation of Slump Flow and Analysis Using MEP Model: Similar to compressive strength, for the slump flow five parameters are considered in the formulation process, namely FA/C (x1), CA/C (x2), W/C (x3), %SF (x4) and %SP (x5). These values have been chosen from Table 1 as inputs for the MEP model. The prediction equations for the best result by the MEP algorithm are given in Eq. (8) and Eq. (9) for the aforementioned function sets.

$$S_1(mm) = \frac{50x_2(x_4 + 15x_1)}{15e^{1/4x_2} + 30x_3 + x_5} \quad (8)$$

$$S_2(mm) = \frac{500x_1x_2}{40x_3 - 8x_3x_5 + 5x_2} \quad (9)$$

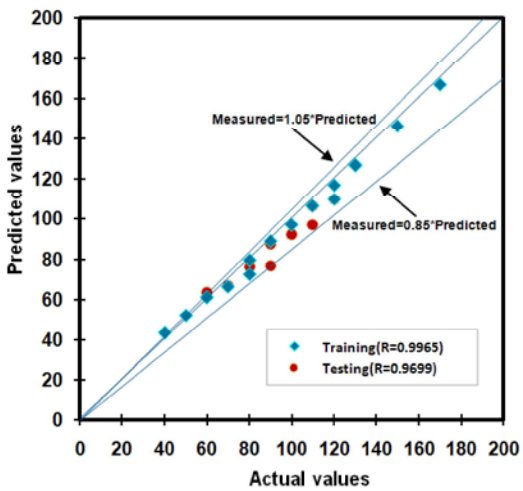

Fig. 4: Results of slump flow prediction for Eq. (9).

The comparison of MEP prediction and actual slump flow of HPC for Eq. (8) is shown in Fig. 3. It can be seen from Fig. 3 that Eq. (8) yielded high R values for training and testing data equal to 0.992 and 0.9761, respectively. Fig. 4 demonstrates that Eq. (9) has produced results with very high R values for training and testing data equal to 0.9965 and 0.9699, respectively. It can be seen from these figures that while Eq. (8) outperforms Eq. (9) on the testing data, better results are obtained by Eq. (9) for the training set.

DISCUSSION

Four formulations of compressive strength and slump flow of HPC in functional form in terms of FA/C, CA/C, W/C, %SF and %SP were obtained by using MEP and given in Eqs. (6)-(9). As mentioned previously, R, MSE and MAE are selected as the target statistical parameters to evaluate the performance of the models. Figures 5 and 6 represent the results for all element test data for compressive strength and slump flow, respectively. Statistical performance of MEP based formulations, as well as SLNN model [12], in terms of their prediction capabilities are summarized in Tables 4 and 5. The results for compressive strength, presented in

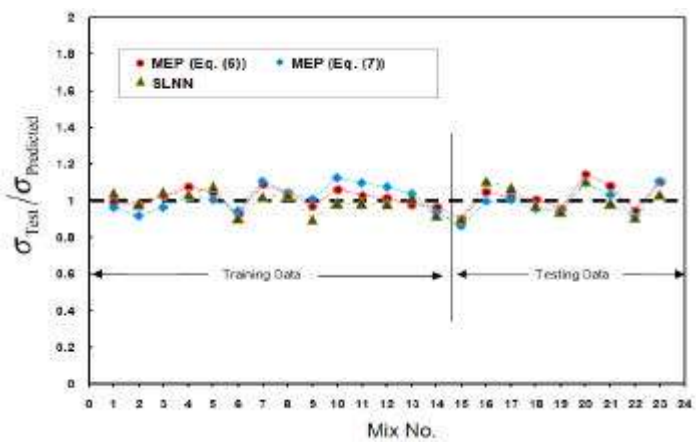

Fig. 5: Compressive strength relative comparison for all element tests data.

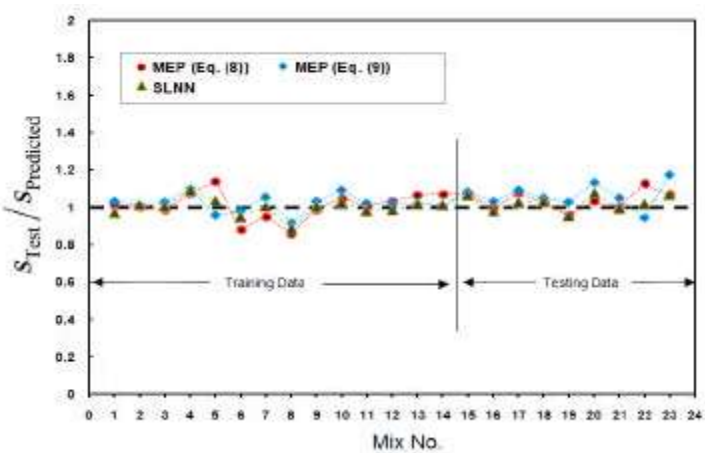

Fig. 6. Slump flow relative comparison for all element tests data.

Table 4: Statistical performance of models for compressive strength prediction

	Training			Testing			All elements		
Models	R	MSE	MAE	R	MSE	MAE	R	MSE	MAE
MEP (Eq. (6))	0.9663	28.273	4.403	0.924	74.545	7.038	0.9441	46.379	5.434
MEP (Eq. (7))	0.9465	41.886	5.521	0.9083	69.180	6.861	0.9229	52.566	6.046
SLNN	0.9546	34.126	4.651	0.8977	61.212	6.898	0.9351	44.725	5.53

Table 5: Statistical performance of models for slump flow prediction

	Training			Testing			All element tests data		
Models	R	MSE	MAE	R	MSE	MAE	R	MSE	MAE
MEP (Eq. (8))	0.992	28.311	4.427	0.9781	24.974	4.162	0.9893	26.998	4.323
MEP (Eq. (9))	0.9965	17.780	3.455	0.9699	61.920	6.692	0.9903	35.052	4.722
SLNN	0.9965	9.129	2.471	0.9795	18.882	3.653	0.9936	12.945	2.934

Table 6: Comparative analysis of proposed MEP based formulae with experimental and SLNN results

Mix No.	σ_{Test} (Mpa)	σ_1 Eq.(6) (Mpa)	σ_{Test}/σ_1	σ_2 Eq.(7) (MPa)	σ_{Test}/σ_2	σ_{SLNN} (MPa)	$\sigma_{Test}/\sigma_{SLNN}$	S_{TESAT} (mm)	S_1 Eq.(8) (mm)	S_{TESAT}/S_1	S_1 Eq.(9) (mm)	S_{TESAT}/S_2	S_{SLNN} (mm)	S_{Test}/S_{SLNN}
Training														
1	80	80.74	0.991	83.11	0.963	76.96	1.040	110	108.44	1.014	106.78	1.030	113.40	0.970
2	90.08	92.69	0.972	98.56	0.914	91.84	0.981	90	89.98	1.000	88.89	1.013	88.80	1.014
3	110.08	107.44	1.025	114.24	0.964	105.44	1.044	120	121.50	0.988	117.09	1.025	119.40	1.005
4	110.08	102.43	1.075	108.32	1.016	106.56	1.033	80	74.46	1.074	72.53	1.103	73.40	1.090
5	130.08	124.49	1.045	129.63	1.004	120.64	1.078	50	43.86	1.140	52.25	0.957	48.20	1.037
6	120	129.38	0.927	127.21	0.943	132.64	0.905	60	68.20	0.880	61.20	0.980	63.20	0.949
7	130.08	119.43	1.089	117.73	1.105	127.04	1.024	70	73.59	0.951	66.46	1.053	69.60	1.006
8	140	134.30	1.042	134.21	1.043	135.68	1.032	40	46.52	0.860	43.59	0.918	45.00	0.889
9	80	82.67	0.968	79.19	1.010	88.96	0.899	100	101.24	0.988	97.13	1.030	99.60	1.004
10	90.08	84.82	1.062	80.20	1.123	91.36	0.986	120	114.34	1.050	109.90	1.092	116.80	1.027
11	90.08	87.35	1.031	82.24	1.095	91.46	0.985	130	129.05	1.007	126.95	1.024	132.80	0.979
12	90.08	89.31	1.009	83.96	1.073	91.65	0.983	150	144.73	1.036	146.63	1.023	152.00	0.987
13	90.08	91.93	0.980	87.05	1.035	88.72	1.015	170	159.50	1.066	166.85	1.019	167.00	1.018
14	90.08	93.48	0.964	96.39	0.935	98.24	0.917	80	74.72	1.071	79.45	1.007	79.00	1.013
Testing														
15	80	88.83	0.901	92.95	0.861	88.96	0.899	100	93.48	1.070	92.44	1.082	93.80	1.066
16	100	95.56	1.046	100.35	0.996	90.40	1.106	110	110.57	0.995	106.57	1.032	112.20	0.980
17	80	78.59	1.018	79.65	1.004	74.88	1.068	120	111.19	1.079	109.84	1.092	117.00	1.026
18	100	99.17	1.008	104.79	0.954	103.04	0.970	80	78.11	1.024	76.17	1.050	77.60	1.031
19	100	104.62	0.956	106.27	0.941	106.40	0.940	90	93.97	0.958	87.59	1.028	94.20	0.955
20	140	122.08	1.147	127.64	1.097	126.40	1.108	110	106.58	1.032	97.21	1.132	102.20	1.076
21	110.08	102.01	1.079	106.65	1.032	111.84	0.984	70	69.77	1.003	66.73	1.049	70.40	0.994
22	100	105.90	0.944	109.60	0.912	110.08	0.908	60	53.39	1.124	63.41	0.946	59.00	1.017
23	120	108.68	1.104	108.35	1.108	116.48	1.030	90	84.58	1.064	76.63	1.174	84.32	1.067

Table 4, show that the best performance is achieved by Eq. (6) for both of the training (R = 0.9663, MSE=28.273, MAE = 4.403) and testing data (R = 0.924, MSE=74.545, MAE = 7.038). Comparing the results of the SLNN based formula and Eq. (7) for the training set, it can be seen that the former performs superior than the latter. It can be observed from Table 4 that both of the formulae obtained by MEP approach outperform the SLNN formulation on the testing data set. The results for all element tests data demonstrate that Eq. (6) has better performance followed by SLNN and Eq. (7).

Considering the slump flow, it can be concluded from Table 5 that while Eq. (9) and SLNN formula yielded R values equal to 0.9965 for the training data set, SLNN

slightly outperforms the other regarding its lower MSE and MAE values. In this case, Eq. (8) has performed poorer than the other models. As can be observed in Table 5, SLNN based formula with R, MSE and MAE values equal to 0.9795, 18.882 and 3.6533 has produced better results on the testing data set followed by Eq. (8) and Eq. (9). Considering the all element tests data, it can be seen that the SLNN model has better performance followed by Eq. (9) and Eq. (8). It should be noted that in spite of the better performance of SLNN models in some of the aforementioned situations, the MEP based prediction equations are really short, very simple and can be used facilitatory.Table 6 shows a comparative analysis of results of the proposed MEP formulations and the results obtained by SLNN model including compressive strength and slump flow actual experimental values.

CONCLUSIONS

This paper proposes a novel approach for the prediction of compressive strength and slump flow of HPC mixes using a variant of GP namely, MEP. Four formulations of compressive strength and slump flow, two formulas for each of them, have been obtained by means of MEP and considering two different function sets. A reliable database including previously published compressive strength and slump flow of HPC test results was used for training and testing the prediction models. The MEP based formulations results were compared with the experimental results and the existing model proposed in the literature namely, SLNN (RBF).

Based on the values of performance measures for the models it can be observed that the proposed MEP models are able to predict the target values to an acceptable degree of accuracy. The results of testing data demonstrated that for the prediction of compressive strength both of the formulae evolved by MEP outperform the proposed formulation result of SLNN. Considering the relevant results for the explicit formulation of slump flow it can be observed that SLNN has produced slightly better results than the MEP based formulas. When the performance of the MEP based prediction equations is taken into consideration it can be seen that in addition to their considerable accuracy they are quite short and very simple and seem to be more practical for use compared to the prediction equations produced by SLNN. However, this investigation revealed that MEP is very promising approach that can be utilized in order to produce explicit formulations to be able to capture the underlying relationship between the different interrelated input and output data for many of civil engineering tasks.

REFERENCES

1. Malier, Y., 1992. High Performance concrete, from Material to Structure. London: E & FN Spon.
2. Koza, J.R., 1992. Genetic programming: On the programming of computers by means of natural selection. Cambridge (MA): MIT Press.
3. Banzhaf, W., P. Nordin, R. Keller and F. Francone, 1998. Genetic Programming – An Introduction: On the Automatic Evolution of Computer Programs and Its Application. Heidelberg/San Francisco: Morgan Kaufmann.
4. Oltean, M., 2002. Dumitrescu D. Multi expression programming, technical report. Babeş-Bolyai University, [UBB-01-2002].
5. Oltean, M. and C. Grosşan, 2003. A comparison of several linear genetic programming techniques. Complex Sys,14(4): 1-29.
6. Baykasoğlu, A., H. Güllü, H. Çanakçı and L. Özbakır, 2007. Prediction of compressive and tensile strength of limestone via genetic programming. Expert Systems with Applications, Article in press.
7. Hayati, M. and Z. Mohebi, 2007. Temperature Forecasting Based on Neural Network Approach. World Appl. Sci. J., 2(6): 613-620.
8. Shayanfar, M.A., S.R. Massah and H. Rahami, 2007. An Inverse Reliability Method Using Neural Networks and Genetic Algorithms. World Appl. Sci. J., 2(6): 594-601.
9. Yeh, C., 2006. Exploring concrete slump model using artificial neural networks. J. Comput. Civil Eng., 20(3): 217-221.
10. Yeh, I., 1998. Modeling of strength of high-performance concrete using artificial neural networks. Cem. Conc. Res., 28(12): 1797-808.
11. Kasperkiewicz, J., J. Racz and A. Dubrawski, 1995. HPC strength prediction using artificial neural network. J. Comput Civil Eng., 9(4): 279-84.
12. Rajasekaran, S. and R. Amalraj, 2002. Predictions of design parameters in civil engineering problems using SLNN with a single hidden RBF neuron. Comput & Struc, 80(31): 2495-2505.
13. Zhang, J. and AJ. Morris, 1988. A sequential learning approach for single hidden layer neural networks. Neural Networks, 11(1): 65-80.
14. Rajasekaran, S., R. Amalraj and S. Anandakumar, 2001. Optimization of mix proportions for high performance concrete using cellular genetic algorithms. Proceedings of National Seminar on Concrete Technology for 21st Century. Annamalainagar: Annamalai University.

15. Ashour, A.F., L.F. Alvarez and V.V. Toropov, 2003. Empirical modelling of shear strength of RC deep beams by genetic programming. Comput & Struc, 81(55): 331-338.
16. Javadi, A.A., M. Rezani and M. Mousavi Nezhad, 2006. Evaluation of liquefaction induced lateral displacements using genetic programming. Comput Geotech, 33(4): 222-233.
17. Baykasoğlu, A., T. Dereli and S. Tanıs, 2004. Prediction of cement strength using soft computing techniques. Cem. Con. Res., 34(11): 2083-2090.
18. Alavi, A.H., A.A. Heshmati, A.H. Gandomi, A. Askarinejad and M. Mirjalili, 2008. Utilisation of Computational Intelligence Techniques for Stabilised Soil. In: M. Papadrakakis and B.H.V. Topping editors. Proceedings of the 6th International Conference on Engineering Computational Technology, Civil-Comp Press, Scotland.
19. Oltean, M. and C. Grosşan, 2003. Evolving evolutionary algorithms using multi expression programming. In: W. Banzhaf *et al*. editors. Proceedings of the 7th european conference on artificial life. Dortmund: LNAI. pp: 651-658.
20. Aho, A., R. Sethi and J.D. Ullman, 1986. Compilers: Principles, Techniques and Tools. Reading (MA): Addison Wesley.
21. Oltean, M., 2004. Multi Expression Programming source code.