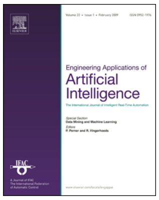
Engineering Applications of Artificial Intelligence 33 (2014) 58–68

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



An evolutionary computational approach for formulation of compression index of fine-grained soils

Danial Mohammadzadeh S. [a], Jafar Bolouri Bazaz [a], Amir H. Alavi [b,*]

[a] Civil Engineering Department, Ferdowsi University of Mashhad, Mashhad, Iran
[b] Department of Civil and Environmental Engineering, Michigan State University, East Lansing, MI 48824, USA

ARTICLE INFO

Article history:
Received 6 December 2013
Received in revised form
12 February 2014
Accepted 26 March 2014

Keywords:
Evolutionary computation
Multi-expression programming
Soil compression index
Soil engineering properties
Prediction

ABSTRACT

This study presents a robust evolutionary computational technique, called multi-expression programming (MEP), to derive a highly nonlinear model for the prediction of compression index of fine-grained soils. The proposed model relates the soil compression index to its liquid limit, plastic limit and void ratio. The experimental database used for developing the models was established upon 108 consolidation tests conducted on different soils sampled from different construction sites in Iran. The generalization capability of the model was verified via several statistical criteria. The parametric and sensitivity analyses were performed and discussed. The results indicate that the MEP approach accurately characterizes the soil compression index leading to a very good prediction performance. The correlation coefficients between the experimental and predicted soil compression index values are equal to 0.935 and 0.901 for the calibration and testing data sets, respectively. The developed model has a significantly better performance than the existing empirical equations for the soil compression index.

1. Introduction

A vital phase in geotechnical foundation design is to provide a reliable estimation of the compressibility characteristics of soils. The compressibility characteristics have a key role for the analysis of the settlement of the soil layers under the applied load (Tiwari and Ajmera, 2012; Singh and Noor, 2012). In general, the soil compressibility is defined as the volume reduction under pressure taking place due to the drainage of pore water. The rate of drainage of pore water is a time-dependent process because it is a function of the soil permeability. Consequently, analysis of the soil compressibility properties is mostly important for fine-grained, low permeable soils. Compression index (C_c), coefficient of compressibility (a_v), and coefficient of consolidation (C_v) are the main indicators of the soil compressibility. Among these parameters, C_c is often used for the direct calculation of settlement (Carter and Bentley, 1991; Gulhati and Datta, 2005; Singh and Noor, 2012). Typically, the settlement associated with load increments is obtained using the logarithm of the normal compressive stress (σ') against soil void ratio (e) curve. A schematic representation of the $e-\log\sigma$' curve is illustrated in Fig. 1. As can be seen in this figure, the characteristic curve has two distinct regions: (1) an elastic rebound curve and (2) a linear virgin compression curve at higher stresses. C_c is the modulus of the slope of the virgin compression curve which is usually obtained from a standard consolidation (oedometer) test on an undisturbed sample (Gregory et al., 2006). Subsequently, the total settlement (S_t) for a layer of normally consolidated clay can be determined using the following equation:

$$S_t = H\left(\frac{C_c}{1+e_0}\right)\left(\log\frac{\sigma_0'+\Delta\sigma}{\sigma_0'}\right) \tag{1}$$

where e_0 is the initial void ratio, σ' is the effective overburden pressure, $\Delta\sigma$ is the applied load, H is the thickness of the layer.

Determination of C_c from the oedometer test is a cumbersome, expensive and time consuming process, especially for the fine-grained soils. In order to avoid the labor of conducting consolidation tests, several studies have been focused on the prediction of the compressibility behavior of soils using its basic physical properties (e.g., Skempton, 1944; Nishida, 1956; Cozzolino, 1961; Terzaghi and Peck, 1967). Most of the existing models are developed based on traditional statistical analyses which notable modeling drawbacks (Alavi and Gandomi, 2011). Thus, more sophisticated methods are required to capture the complex behavior of C_c. In this context, computational intelligence (CI) techniques can be considered as efficient alternatives to traditional methods. They determine the structure of a prediction model by automatically learning from data. CI has different well-known branches such as artificial neural network (ANN), fuzzy inference system (FIS), adaptive neuro-fuzzy system (ANFIS), and support

[*] Corresponding author.
E-mail addresses: alavi@msu.edu, ah_alavi@hotmail.com (A.H. Alavi).
URL: http://www.egr.msu.edu/~alavi/ (A.H. Alavi).

http://dx.doi.org/10.1016/j.engappai.2014.03.012
0952-1976/© 2014 Elsevier Ltd. All rights reserved.

vector machines (SVM). These techniques have been successfully employed to solve problems in engineering field (e.g., Kerh and Chu, 2002; Kerh and Ting, 2005a,b; Kerh and Lee, 2006; Muttil and Chau, 2007; Traore et al., 2010; Azamathulla and Wu, 2011; Majid et al., 2011; Cheng et al., 2012; Azamathulla et al., 2012; Ismail et al., 2013; Masmoudi and Haït, 2013; Emamgholizadeh et al., 2013). Despite the good performance of ANNs, FIS, ANFIS, SVM and many of the other CI methods, they are considered black-box models. That is, they are not capable of generating practical prediction equations. This is a fundamental disadvantage that limits their practicability (Alavi and Gandomi, 2011). In order to cope with the limitations of the existing methods, a robust CI approach, namely genetic programming (GP) has been introduced (Koza, 1992). GP is an evolutionary computational (EC) approach. The EC methods use the principle of Darwinian natural selection to generate computer programs for solving a problem. GP has several advantages over the conventional and other similar techniques. A notable feature of GP and its variants is that they can produce highly nonlinear prediction equations without a need to pre-define the form of the existing relationship (Sette and Boullart, 2001; Javadi et al., 2006; Çiftçi et al., 2009; Guven, 2009; Guven et al., 2009; Gandomi et al., 2010; Gandomi and Alavi, 2011; Rezania et al., 2011; Tsai, 2011; Azamathulla et al., 2011; Alavi et al., 2011; Chen et al., 2012; Azamathulla, 2012; Mahmood et al., 2013).

Multi-expression programming (MEP) (Oltean and Dumitrescu, 2002) is new variant of GP. MEP has a special ability to encode multiple computer programs of a problem in a single program (Alavi et al., 2010). In contrast with traditional GP and its variants, and also other soft computing techniques, application of MEP in the field of civil engineering is totally new and original (Baykasoglu et al., 2008; Alavi et al., 2010; Alavi and Gandomi, 2011; Gandomi et al., 2011a; Gandomi and Alavi, 2013). This paper proposes the MEP technique to derive a precise predictive equation for the compression index of Iranian soils from basic soil parameters. A comprehensive and reliable set of data including 108 consolidation test results was established to develop the models. The robustness of the proposed model was verified through different validation phases.

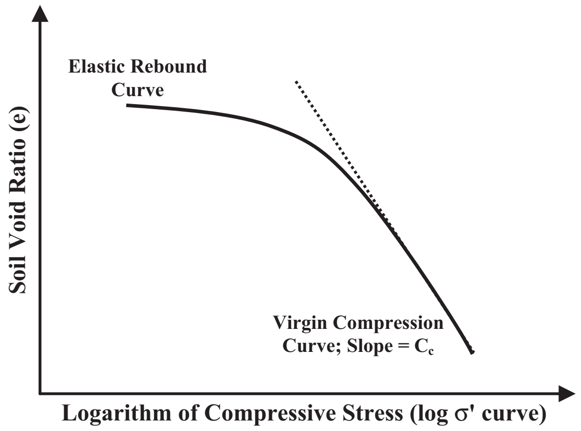

Soil Void Ratio (e)
Elastic Rebound Curve
Virgin Compression Curve; Slope = C_c
Logarithm of Compressive Stress (log σ' curve)

Fig. 1. The features of the soil compression characteristic (Gregory et al., 2006).

2. Review of previous studies

Over the decades, various empirical models have been developed to correlate C_c with various index properties of soils such as the liquid limit, natural water content, plasticity index, specific gravity, and void ratio (Skempton, 1944; Nishida, 1956; Cozzolino, 1961; Terzaghi and Peck, 1967; Sowers, 1970; Azzouz et al., 1976; Wroth and Wood, 1978; Mayne, 1980; Park and Lee, 2011). Table 1 presents some of the well-known empirical prediction equations in this field. Nearly all these relationships were derived by performing multiple linear regression analysis. In fact, the commonly used regression analyses can have large uncertainties. The classical regression analysis has major limitations due to the oversimplification of the complicated mechanism of the consolidation process. As can be seen in Table 1, this type of analysis assumes the structure of the model in advance by a limited number of linear or nonlinear equations. Thus, such models cannot efficiently consider the highly nonlinear interactions between the soil parameters and C_c.

It is worth mentioning that among different CI techniques, only ANNs have been used to predict the C_c of soil layers (Desai et al., 2009; Jianping et al., 2010; Farkhonde and Bolouri, 2010; Daryaee et al., 2010; Kumar and Rani, 2011; Park and Lee, 2011; Kumar et al., 2012; Rani et al., 2013). However, this powerful method does not provide practical predictions which limit its applicability for further analysis.

3. Evolutionary computation

Inspired by the natural evolution and the Darwinian concept of “Survival of the Fittest”, the EC-based methods generate computer models to solve complicated problems. Some of the well-known

Table 1
Some of the well-known empirical prediction equations for $C_{c.}$

Reference	Equation	Applicability
Skempton (1944)	$C_c = 0.007(LL-10)$	Remolded clays
Nishida (1956)	$C_c = 1.15(e_0-0.35)$	All clays
Cozzolino (1961)	$C_c = 0.43(e_0-0.11)$	Brazilian clays
Terzaghi and Peck (1967)	$C_c = 0.009(LL-10)$	Normally consolidated clays
Sowers (1970)	$C_c = 0.75(e_0-0.50)$	Soils of very low plasticity
Azzouz et al. (1976)	$C_c = 0.40(e_0-0.25)$	All natural soils
	$C_c = 0.01(\omega-5)$	
	$C_c = 0.006(LL–9)$	
Wroth and Wood (1978)	$C_c = 0.50 \times PI \times G_s$	All remolded normally consolidated clays
Mayne (1980)	$(LL-13)/109$	All clays
Koppula (1981)	0.01ω	Chicago and Alberta clays
Herrero (1983)	$0.01\omega-0.075$	Normally consolidated clays
Nagaraj and Murty (1985)	$C_c = 0.2343(LL/100)G_s$	All inorganic clays
Park and Lee (2011)	$C_c = 0.49(e_0-0.11)$	Korean natural soils
	$C_c = 0.014(LL-0.168)$	

e_0 is the initial void ratio, ω is the natural water content, LL is the plastic limit, PI is the plastic index, G_s is the specific gravity.

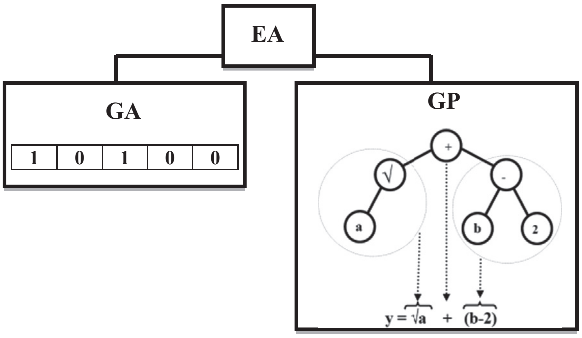


Fig. 2. A comparative illustration of encoded solutions by GA and GP.

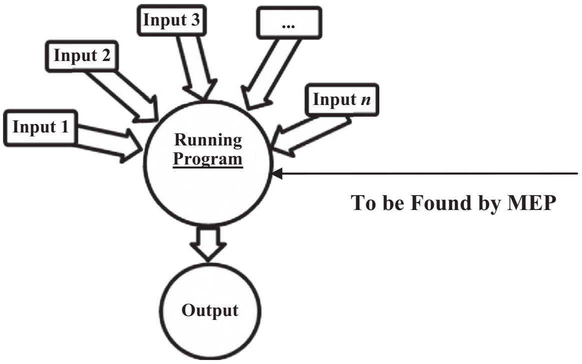


Fig. 3. Conceptual scheme of input–process–output (IPO) in GP.

branches of EC are genetic algorithms (GA) (Holland, 1975), evolutionary strategies (ESs) (Rechenberg, 1973), and evolutionary programming (EP) (Fogel et al., 1996). These techniques are collectively known as evolutionary algorithms (EAs). In general, an EA consists of an initial population of random individuals improved by a set of genetic operators (e.g., reproduction, mutation and recombination). The individuals are encoded solutions in the form of binary strings of numbers evaluated by some fitness functions (Coello et al., 2007). Improvement of the population is a process to reach the fittest solution with the maximum convergence. Typically in an EA, a population of individual is randomly created and then the members are ranked according to a fitness function. The members with the highest fitness ranking are given a higher chance to become parents for the next generation (offspring). The approach used to generate offspring from the parents is referred to as the reproduction heuristic. Then selected members are randomly transformed into new members via mutation, recombination or crossover. These steps are repeated until the convergence conditions are satisfied and the fittest member is selected (Fogel et al., 1996; Koza, 1992; Coello et al., 2007). The differences between EAs are in the way that they represent the individual structures, types of selection mechanism, forms of genetic operators, and measures of performance.

GA has been shown to be a suitably robust EA for dealing with a wide variety of complex civil engineering problems (e.g. Keedwell and Khu, 2005; Castilho et al., 2007; Sanchis et al., 2010). GP is a specialization of GA where the encoded solutions (individuals) are computer programs rather than binary strings (Banzhaf et al., 1998). Fig. 2 shows a comparison of the encoded solutions (individuals) by GA and GP. In GP, inputs and corresponding output data samples are known and the main goal is to generate predictive models relating them (see Fig. 3) (Weise, 2009).

There GP solutions are represented in different ways such tree-shaped, graph-shaped and linear encodings (Banzhaf et al., 1998; Alavi and Gandomi, 2011). Tree-shaped is the mostly widely used representation of the GP programs. However, the emphasis of the present study is placed on the linear-based GP techniques.

3.1. Expression programming

Recently, several linear variants of GP have been developed such as linear genetic programming (LGP) (Brameier and Banzhaf, 2007), gene expression programming (GEP) (Ferreira 2001), multi-expression programming (MEP) (Oltean and Dumitrescu, 2002), grammatical evolution (GE) (Ryan and O'Neill, 1998), and cartesian genetic programming (CGP) (Miller and Thomson, 2002). These variants make a clear distinction between the genotype and the phenotype of an individual. Thus, the individuals are represented as linear strings that are decoded and expressed like nonlinear entities (trees) (Oltean and Groşan, 2003). There are some main reasons for using linear GP. Computers do not naturally run tree-shaped programs. Therefore, slow interpreters have to be used as a part of classical tree-based GP. Conversely, by evolving the binary bit patterns, the use of an expensive interpreter is avoided. Consequently, a linear GP system can run several orders of magnitude faster than comparable interpreting systems. The enhanced speed of the linear variants of GP (e.g., LGP and MEP) permits conducting many runs in realistic timeframes. This leads to deriving consistent and high-precision models with little customization (Francone and Deschaine, 2004; Poli et al., 2007; Gandomi et al., 2011b).

EP techniques such as GEP and MEP are the most common linear-based GP methods. MEP was first introduced by Oltean and Dumitrescu (2002). Linear chromosomes are used by MEP for solution encoding. This technique encodes multiple computer programs in a single chromosome. A program with the best fitness represents the chromosome. The MEP decoding process is not more complicated than other GP variants storing a single program in a chromosome (Alavi et al., 2010). The steady-state algorithm of MEP starts by the creation of a random population of computer programs. MEP uses the following steps to evolve the best program until a termination condition is reached (Oltean and Groşan, 2003; Alavi et al., 2010; Alavi and Gandomi, 2011):

I. Selection of two parents using a binary tournament procedure (Koza, 1992) and recombination of them with a fixed crossover probability.
II. Obtaining two offspring by the recombination of two parents.
III. Mutation of the offspring and replacement of 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).

The representation of the MEP solutions is similar to the procedure followed by C and Pascal to convert expressions into a machine code. Functions and terminals are a part of a population member created by MEP (Alavi et al., 2010; Alavi and Gandomi, 2011). The terminal and function symbols are elements in the terminal and function sets, respectively. A function set can contain the basic arithmetic operations or any other mathematical functions. The terminal set can contain numerical constants, logical constants and variables. Each gene encodes a terminal or a function symbol. The first symbol in a chromosome is a terminal symbol. An example of a MEP chromosome is as given below:

1: A
2: B
3: * 1, 2
4: -2, 3

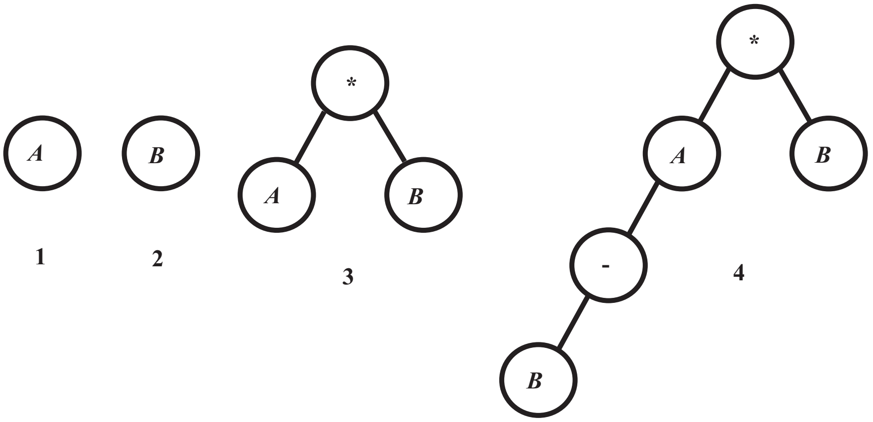


Fig. 4. Expressions encoded by an MEP chromosome and represented as trees.

The function set for the above example includes “ – " and “ – ”. A and B are the elements of the terminal set. The MEP individuals are converted into programs by reading the chromosome top-down starting with the first position. In this example, genes 1 and 2 encode simple expressions which are $E_1=A$ and $E_2=B$. Gene 3 indicates the operation “*” on the operands located at positions 1 and 2. Therefore, gene 3 encodes the expression: $E_3=A*B$. Gene 4 indicates the operation “–’ on the operands located at positions 2 and 3. Therefore, gene 4 encodes the expression: $E_4=B-(A*B)$. Each of the above expressions can be considered as a possible solution. The MEP chromosomes can be illustrated as a forest of trees rather than a single tree because of their multi-expression representation (see Fig. 4) (Alavi et al., 2010; Alavi and Gandomi, 2011). The best expression is selected after controlling the fitness of all expression in an MEP chromosome using the following equation (Oltean and Groşan, 2003):

$$f=\min_{i=l,m}\left\{\sum_{j=1}^{n}|E_j-O_j^i|\right\} \tag{2}$$

in which 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 (Alavi et al., 2010; Alavi and Gandomi, 2011).

4. Formulation for compression index of fine-grained soils

In order to reach reliable estimations of the C_c of soils, the impact of several parameters should be incorporated into the model development. The general forms of the existing prediction equations, represented in Table 1, indicate that C_c mainly depends on the soil physical properties. Referring to the form of the existing models (Skempton, 1944; Nishida, 1956; Cozzolino, 1961; Terzaghi and Peck, 1967; Sowers, 1970; Azzouz et al., 1976; Mayne, 1980; Park and Lee, 2011), the proposed model for the prediction of the C_c of fine-grained soils was considered to be a function of the following parameters:

$$C_c=f(LL,\ PL,\ e_0) \tag{3}$$

where LL (%) is the liquid limit, PL (%) is the plastic limit, e_0 is the initial void ratio.

LL, PL and e_0 represent the intrinsic soil properties. It is worth mentioning that the main purpose of this research was to prove the possibility of providing good estimations of the C_c of soil by using only its basic physical properties. That is why only LL, PL and e_0 were considered as the predictor variables. Determining LL, PL and e_0 does not require complicated laboratory tests compared to the consolidation tests. Therefore, using these basic soil properties to make precise predictions of C_c would result in a significant cost savings for many geotechnical investigations. Besides, the LL, PL and void ratio are rationally correlated to the natural water content for saturated soils (Bartlett and Lee, 2004).

The steps of the proposed methodology for developing a precise model are as follows:

1. Determination of input and output variables of the model.
2. Collect data set S containing intrinsic soil properties (LL, PL and e_0) and corresponding C_c values.
3. Divide S into three subsets: learning (S_{Learning}), validation ($S_{\text{Validation}}$) and testing (S_{Testing}) data.
   The details of the data division procedure are described in Section 4.2.
4. Run the MEP algorithm to estimate relation between input and output variables:
   (a) Select optimal parameters for the MEP algorithm.
   (b) Train the model using S_{Learning}.
   (c) Evaluate the model using $S_{\text{Validation}}$.
5. Pick the best MEP model based on the following criteria (Alavi et al., 2011):
   (a) The simplicity of the model.
   (b) Best fitness on S_{Learning}.
   (c) Best fitness on $S_{\text{Validation}}$.
6. Run MEP for S_{Testing}.
7. Calculate the parameters for evaluating the performance of the models on S_{Learning}, $S_{\text{Validation}}$ and S_{Testing}.

Steps of the proposed methodology are shown in Fig. 5.

In order to evaluate the performance of the derived models, the correlation coefficient (R), root mean squared error (RMSE) and mean percent error (MAE) were used:

$$R=\frac{\sum_{i=1}^{n}(h_i-\overline{h}_i)(t_i-\overline{t}_i)}{\sqrt{\sum_{i=1}^{n}(h_i-\overline{h}_i)^2\sum_{i=1}^{n}(t_i-\overline{t}_i)^2}} \tag{4}$$

$$RMSE=\sqrt{\frac{\sum_{i=1}^{n}(h_i-t_i)^2}{n}} \tag{5}$$

$$MAE=\frac{1}{n}\sum_{i=1}^{n}|h_i-t_i| \tag{6}$$

where h_i and t_i are, respectively, the actual and predicted output values for the ith output, $\overline{h_i}$ and $\overline{t_i}$ are, respectively, the average of the actual and predicted outputs, and n is the number of samples.

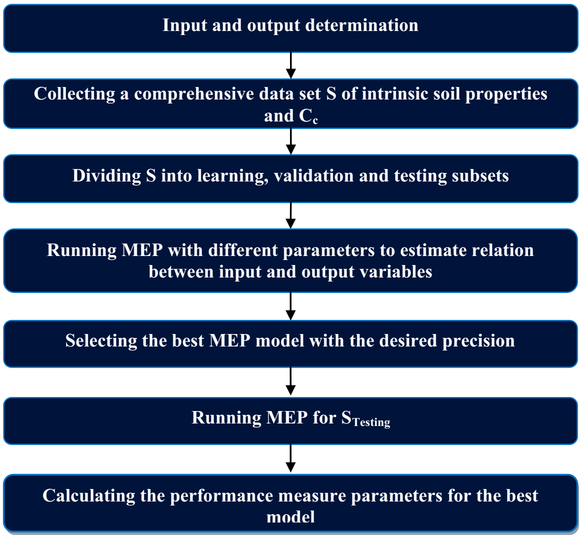

Input and output determination
Collecting a comprehensive data set S of intrinsic soil properties and C_c
Dividing S into learning, validation and testing subsets
Running MEP with different parameters to estimate relation between input and output variables
Selecting the best MEP model with the desired precision
Running MEP for $S_{Testing}$
Calculating the performance measure parameters for the best model

Fig. 5. Steps of the proposed methodology for developing a prediction model for C_c.

4.1. Experimental database

The data used for the model development consist of consolidation and index property test results for samples obtained from different locations in Khorasan Razavi Province, Iran. All the tests were performed under similar conditions and using the same technique. The soil samples were picked up from test pits at 0.5–1.0 m depth. The database includes a fairly wide range of soil index properties. The physical and plastic characteristics of soil were determined through extensive geotechnical laboratory tests. Conventional oedometer tests were performed to determine C_c. The testing procedure was as described in Abbasim et al. (2012). A conventional oedometer apparatus having brass ring, 75 mm in diameter and 20 mm height was used to perform the tests. The top and bottom of the specimens were covered by saturated porous stones and filter papers. The cell containing ring and specimen was submerged and allowed to saturate for 24 h. Then, vertical dead load was applied using a loading device until there was no change in dial gauge reading for two consecutive hours. Other details of the test were performed in general accordance with ASTM D2435 procedure for one-dimensional consolidation properties of soils (ASTM D2435, 2000). At the end of each test, the variation of void ratio versus pressure was plotted for each specimen on a semi-logarithmic scale to obtain the C_c values (Abbasim et al., 2012). It should be noted that the one-dimensional consolidation and basic geotechnical characterization tests are very well-known tests. Thus, for brevity, it was not within the scope of this study to provide all the details about them. The information cited in this table includes *LL*, *PL*, and e_0. C_s is the measured compression index. The database comprises 101 test results on fine-grained soil samples. To develop a generalized correlation, a previously published database of seven consolidation tests was further added to the available experimental results. Different soil types used in this study were silty clay with sand (CL–ML), gravelly lean clay with sand (CL), and silty, clayey sand (SC–SM). The descriptive statistics of the test results are given in Table 2.

To visualize the distribution of the samples, the data are presented by frequency histograms (Fig. 6). As we can observe from Fig. 6, the distributions of the predictor variables are not uniform. The derived model provides better predictions for the cases where the densities of the variables are higher.

4.2. Data classification

Overfitting is one of the essential problems in generalization of the CI techniques. Overfitting is a case in which the error on the learning set is driven to a very small value, but when new data presented to the model, the error becomes very large. An approach to avoid overfitting is to test individuals from the run on a validation set to find a better generalization. Then, another data set should be used at the end of the data analysis to verify the generalization performance of the model (Banzhaf et al., 1998; Gandomi et al., 2011b). Accordingly, in the present study, the available data sets were randomly classified into three subsets: (1) learning, (2) validation, and (3) test subsets. The learning set was used to fit the models and the validation set was used to estimate the prediction error for model selection. Thus, both of the learning and validation data were involved in the modeling process and were categorized into one group referred to as training data (Alavi et al., 2011). Finally, the test set was employed for the evaluation of the generalization ability of the final chosen model. The training, validation and test data are usually taken as 50–70%, 15–25% and 15–25% of all data, respectively (Shahin and Jaksa, 2005; Alavi et al., 2011). In the present study, about 80% of the data sets were taken for the training and validation processes (71 data vectors for the training process and 16 data sets as the validation data). The remaining 20% of the data sets were used for the testing of the obtained models.

Table 2
Descriptive statistics of the variables used for the model development.

Parameter	e_0	*LL* (%)	*PL* (%)	C_c
Mean	0.745	36.156	22.605	0.171
Standard deviation	0.123	12.786	5.639	0.047
Sample variance	0.015	163.481	31.794	0.002
Kurtosis	−0.462	−0.028	1.994	−1.033
Skewness	0.538	1.044	1.422	0.484
Range	0.516	52.600	29.200	0.176
Minimum	0.514	19.400	14.800	0.077
Maximum	1.030	72.000	44.000	0.253

4.3. Development of the MEP-based model

An extensive trial study was performed to select the most relevant input parameters for the MEP model. Table 3 presents various parameters involved in the MEP algorithm. There are eight parameters for MEP to be tuned. Several runs were conducted to obtain a parameterization of MEP with enough robustness and generalization. The MEP parameters were changed for different runs to find the global solution. The parameters were selected on the basis of both previously suggested values (Baykasoglu et al., 2008; Alavi et al., 2010; Alavi and Gandomi, 2011; Gandomi et al., 2011a; Gandomi and Alavi, 2013) and making several preliminary runs and observing the performance behavior. As shown in Table 3, the number of generations was set to 100, 300 and 500. A fairly large number of generations were tested on each run to find a model with minimum error. For each case, the program was run until there was no longer significant improvement in the performance of the models or the runs was terminated automatically. Three different values were set for the population size. Large populations were used with the runs to guarantee sufficient diversity (Alavi et al., 2010; Gandomi and Alavi, 2013). Two different values were considered for the crossover and mutation rates. The success of the algorithms usually increases with increasing the chromosome length in MEP. In this case, the complexity of the evolved functions increases and the speed of the algorithm decreases. Different optimal levels were considered for this parameter as tradeoffs between the running time and the complexity of the evolved solutions (Alavi et al., 2010; Gandomi and Alavi, 2013). Basic arithmetic operators and mathematical functions were utilized to get the optimum model. There are $3 \times 4 \times 2 \times 2 \times 3 = 144$

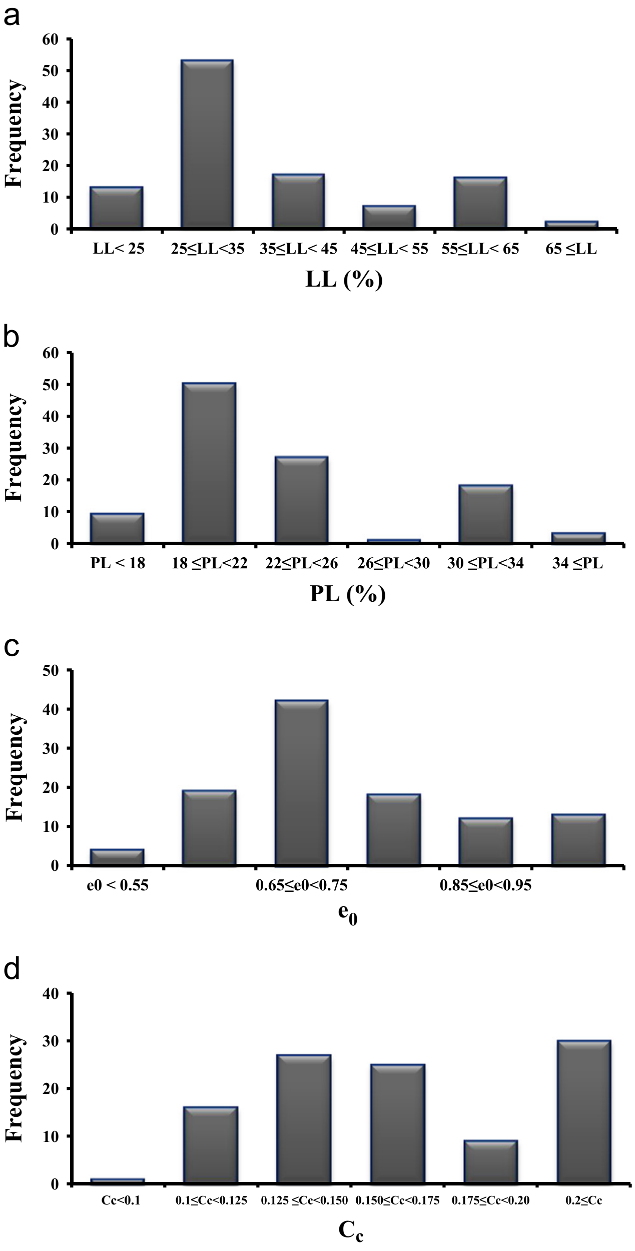


Fig. 6. Histograms of the variables used for the model development.

Table 3
Parameter settings for the MEP algorithm.

Parameters	Setting
Number of generation	100, 300, 500
Population size	250, 500, 1500, 3000
Function set	$+$, $-$, $\times$, /, $\sqrt{}$, exp, ln
Mutation rate (%)	10, 90
Fitness function	Linear error function
Crossover rate (%)	50, 95
Crossover type	Uniform
Chromosome length	25, 50, 80 genes

different combinations of the parameters. All these parameter combinations were tested and three replications for each were carried out. Therefore, the overall number of optimal individual runs is equal to $144 \times 3 = 432$. The source code of MEP (Oltean, 2004) in C++ was modified by the authors to be utilizable for the available problems.

Finally, the best MEP model for predicting the C_c of fine-grained soils was in the form of following equation:

$$C_c = \frac{7}{\exp(e_0{}^2 - e_0 PL + PL) + \frac{LL((8/PL) + e_0 LL)}{LL - PL} + 7} + \frac{LL}{LL - e_0(-e_0{}^2 LL + LL + 2PL - 9PL(2LL - PL))} + \frac{e_0}{2PL(LL^2 + 403)(e_0 + \exp(e_0)\exp(6 - PL - LL) - 5) + 5} \quad (7)$$

The prediction results provided by the best solution found by MEP are illustrated in Fig. 7. The number of generation, population size, mutation rate, crossover rate, and chromosome length for the optimal run were equal to 500, 3000, 10, 95, and 80, respectively.

5. Performance analysis and validation

In order to evaluate the performance of a model, Smith (1986) proposed the following statistical criteria:

- For $|R| > 0.8$, a strong correlation exists and the model is suitable.
- For $0.2 < |R| < 0.8$ a correlation exists.
- For $|R| < 0.2$, a weak correlation exists and is not suitable to use.

In addition, the error values should be considered in all cases (Alavi et al., 2011). It can be observed from Fig. 7 that the MEP model, with $R > 0.8$ and low *RMSE* and *MAE* values, is able to predict the target values with an acceptable degree of accuracy. The performance of the model on the training and testing data suggests that it has both good predictive ability and generalization performance. The reliability of the models created by MEP is notably dependant on the amount of data used for the training process (Alavi et al., 2011). In this context, Frank and Todeschini (1994) argue that the minimum ratio of the number of data sets in the database over the number of predictor variables (inputs) for model acceptability is 3. Also, they suggest that considering a higher ratio equal to 5 is safer. In the present study, this ratio is higher and is equal to $108/3 = 36$.

Furthermore, new criteria recommended by Golbraikh and Tropsha (2002) were checked for external validation of the models on the testing data sets. It is suggested that at least one slope of regression lines (k or k') through the origin should be close to 1. k is the slope of the regression line in plot of actual (h_i) against predicted (t_i) values. k' is the slope of the regression line in plot of t_i against h_i values (Golbraikh and Tropsha, 2002). Recently, Roy and Roy (2008) introduced a confirm indicator of the external predictability of models (R_m). For $R_m > 0.5$, the condition is satisfied. Either the squared correlation coefficient (through the origin) between predicted and experimental values (Ro^2), or the coefficient between experimental and predicted values (Ro'^2) should be close to 1. The considered validation criteria and the relevant results obtained by the models are presented in Table 4. As it is seen, the derived model satisfies the required conditions. For the R_m criterion, it slightly violates the condition. The validation phase ensures the derived MEP model is strongly valid and it is not established by chance. Note that the proposed model was developed using the basic soil physical properties (LL, PL, and e_0) and, therefore, it can easily be used for prediction purposes via hand calculations.

Furthermore, the predictions made by the MEP model were compared with those provided by an ANN model developed in this study. The ANN model was established upon the same data used for the development of the MEP-based model. Various training algorithms are implemented for the training of the ANN network such as gradient descent (traingd), Levenberg–Marquardt

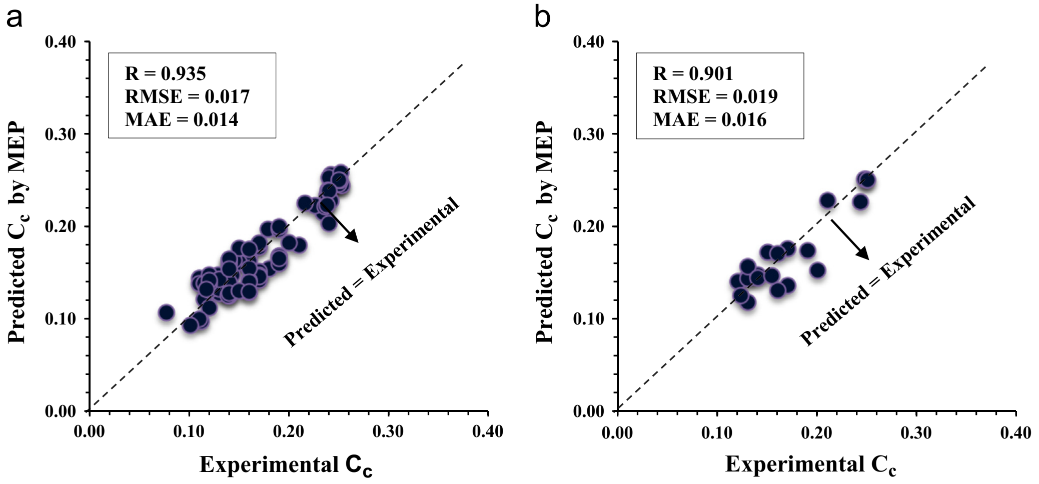


Fig. 7. Experimental versus predicted C_c values using the MEP model: (a) training data and (b) testing data.

(trainlm), Quasi-Newton back-propagation (trainbfg), and resilient (trainrp) back propagation algorithms. The best results were obtained by the Quasi-Newton back-propagation method. Also, the transfer function between the input and the hidden layer was log-sigmoid of form $1/(1+e^{-x})$. A linear transfer function (purelin) was adopted between the hidden layer and the output layer. The ANN architecture that gave the best results for the prediction of C_c was found to contain:

- One invariant input layer, with three (*LL*, *PL* and e_0) arguments.
- One invariant output layer with 1 node providing the value of C_c.
- One hidden layer having 7 ($m=7$) nodes.

The *R*, *RMSE* and *MAE* values of the ANN model on the training data were equal to 0.933, 0.017 and 0.012, respectively. The performance of ANN on the testing data was also good ($R=0.927$; $RMSE=0.017$; $MAE=0.014$). As it is, the ANN model slightly outperforms the MEP model on the training and testing data. However, this insignificant performance difference would not question the capabilities of MEP. In fact, MEP possesses a notable advantage over ANN. MEP has a great capability in generating a transparent and structured representation of the system being studied. Due to the large complexity of the network structure, ANN does not give a transparent function relating the inputs to the corresponding outputs.

Besides, Fig. 8 presents a comparative study between the results obtained by proposed MEP and ANN models and those provided by the well-known models of Skempton (1944), Nishida (1956), Cozzolino (1961), Terzaghi and Peck (1967), Sowers (1970), Azzouz et al. (1976), Mayne (1980), and Park and Lee (2011). The performance of the models was evaluated on the entire database. As can be observed from Fig. 8, the ANN model has the best performance followed by the MEP model. The proposed MEP formula notably outperforms the existing regression-based models. In particular, the prediction error values (*RMSE* and *MAE*) for the existing models are much higher than those for the MEP model. It is worth mentioning that most of the existing models are derived based on the traditional statistical analyses (e.g. regression analysis). The major limitation of this type of analysis is that the structures of the models are designated after controlling only few equations established in advance. Thus, such models cannot efficiently consider the interactions between the dependent and independent variables (Alavi et al., 2011). On the other hand, MEP introduces completely new features. Conversely from the empirical and analytical methods, a major distinction of MEP for determining C_c lies in its powerful ability to model the mechanical behavior without requesting a prior form of the existing relationships or any assumptions. The best equations generated by the MEP technique are determined after controlling numerous linear and nonlinear preliminary models (Alavi et al., 2011; Alavi and Gandomi, 2011). It is worth mentioning that the MEP algorithm is parameter sensitive. The performance of MEP can be improved by using any form of optimally controlling the parameters of the run (Dimopoulos and Zalzala, 2001). In this context, further research can be focused on hybridizing MEP with other optimization algorithms such as GAs, simulated annealing, ant colony, or tabu search.

Table 4
Statistical parameters of the MEP model for the external validation.

Item	Formula	Condition	MEP
1	Eq. (4)	$0.8 < R$	0.901
2	$K=\frac{\Sigma_{i=1}^{n}(h_i \times t_i)}{h_i^2}$	$0.85 < k < 1.15$	1.000
3	$K'=\frac{\Sigma_{i=1}^{n}(h_i \times t_i)}{t_i^2}$	$0.85 < k' < 1.15$	0.988
4	$R_m=R^2 \times (1-\sqrt{\lvert R^2-Ro^2\rvert}$	$0.8 < R_{\mathrm{m}}$	0.46
where	$Ro^2=1-\frac{\Sigma_{i=1}^{n}(t_i-h_i^o)}{\Sigma_{i=1}^{n}(t_i-\overline{t}_i)^2}, \quad h_i^o=k \times t_i$		1.000
	$Ro'^2=1-\frac{\Sigma_{i=1}^{n}(h_i-t_i^o)}{\Sigma_{i=1}^{n}(h_i-\overline{h}_i)^2}, \quad t_i^o=k' \times h_i$		0.997

6. Parametric and sensitivity analyses

In order to ensure the validity of MEP-based model, a comparative parametric analysis was performed. The parametric analysis represents the response of the C_c in the MEP-based model to the variation of the input variables. The methodology is based on changing only one predictor variable at a time while the other variables are kept constant at the average values of their entire data set. This procedure is repeated using another variable until the model response is obtained for all the predictor variables (Alavi et al., 2011). The robustness of the design equations is determined by examining how well the predicted values agree with the underlying physical behavior of the investigated system (Kuo et al., 2009). Fig. 9 shows the tendency of the C_c predictions to the variations of the influencing parameters, i.e., *LL*, *PL* and e_0.

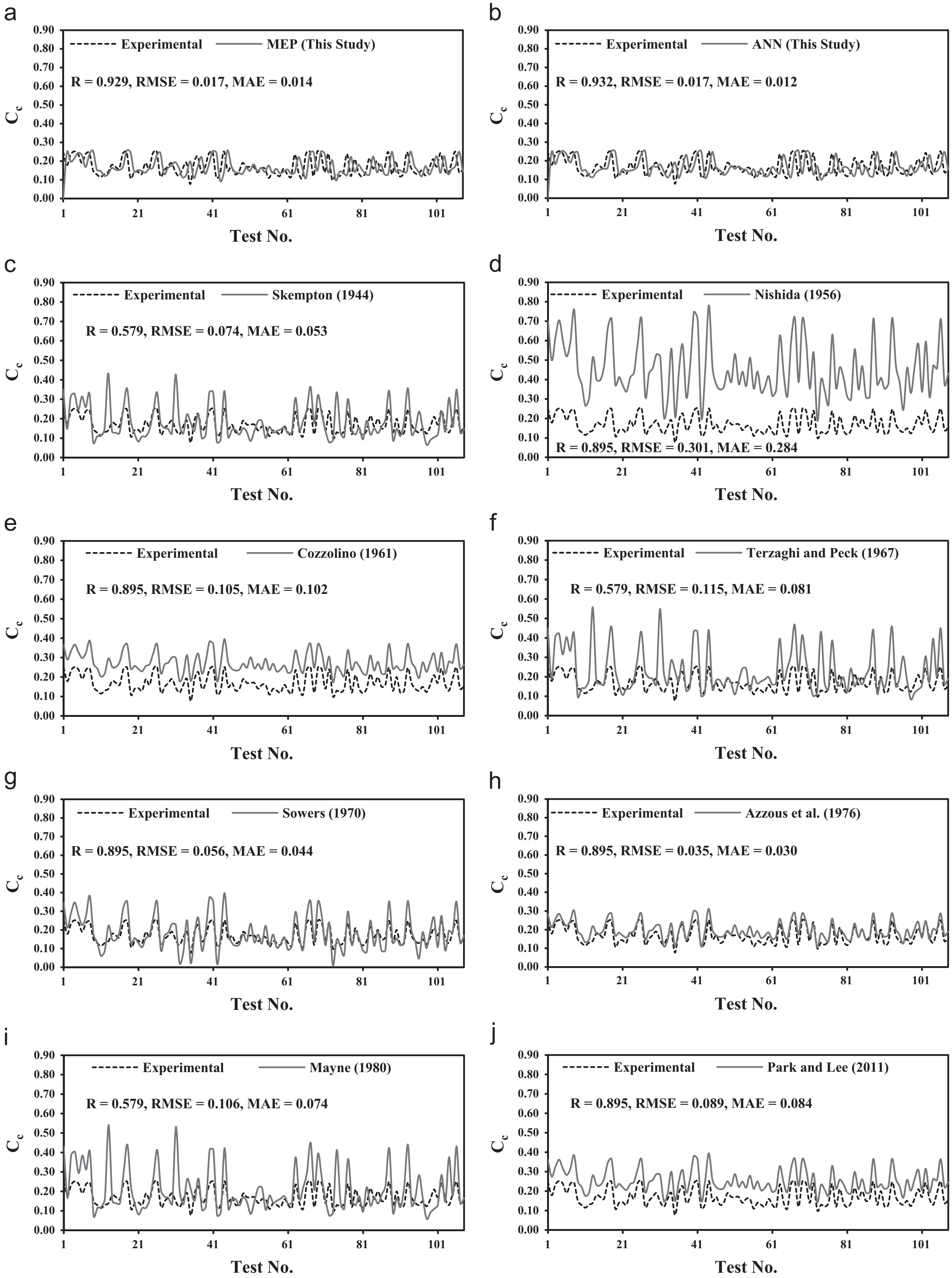


Fig. 8. A comparison of the predictions provided by different models.

Fig. 9 indicates that C_c notably increases with increasing e_0. The results of the parametric analysis for *LL* and *PL* are more complicated than those for e_0. Depending on the ranges of e_0, the behavior of the model differs. For e_0 between 0.5 and 0.7, it can be observed from Fig. 9(a) that the model is not very sensitive to the changes of *LL*. For e_0 higher than 0.9, which represents the

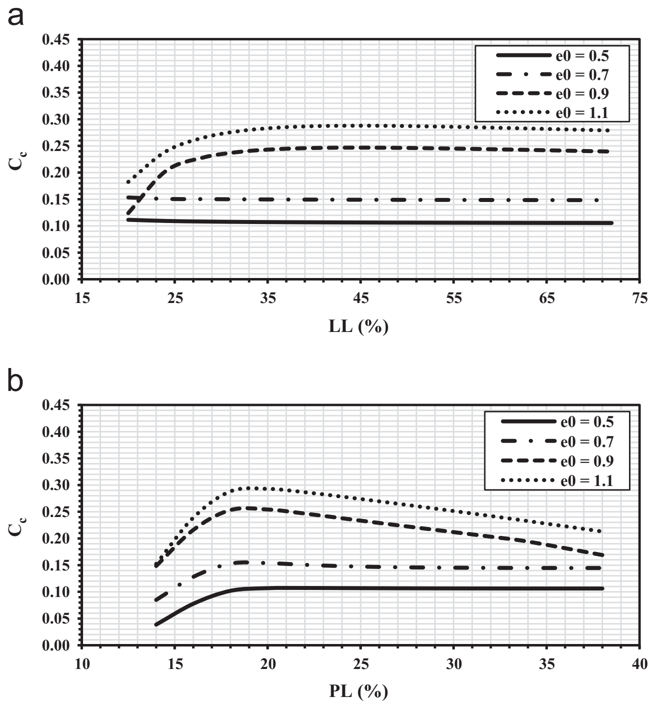


Fig. 9. Parametric analysis of C_c in the MEP model.

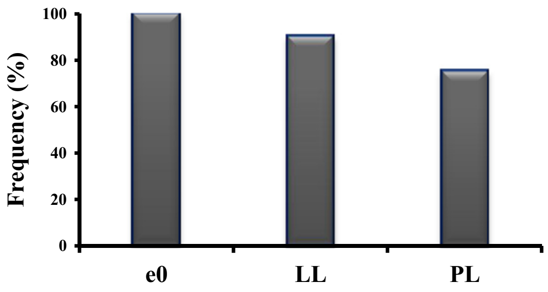


Fig. 10. Contributions of the predictor variables in the MEP analysis.

range for soft clay, C_c increases with increasing *LL* up to about 36% and thereafter the increment declines for higher *LL* values. As shown in Fig. 9(b), for e_0 between 0.5 and 0.7, C_c increases with increasing *PL* up to about 18% and thereafter the increment declines for higher *PL* values. For e_0 higher than 0.9, the results are different. In this case, C_c increases with increasing *PL* up to 18% and afterward it starts decreasing.

Providing an estimation of relative importance of each parameter is an important concern for the aim of model developments or field investigations. As discussed before, the effect of all the considered parameters (i.e., *LL*, *PL* and e_0) on C_c is well understood. Ignoring any of these three parameters for the model development resulted in a model with poor performance. Herein, a sensitivity analysis was conducted to provide a more in depth understanding of the contribution of these important parameters to the prediction of C_c. A common approach for the sensitivity analysis in the GP-based modeling is to obtain the frequency values of the input parameters (Francone, 1998-2004; Alavi et al., 2011; Gandomi et al., 2011a,b). A frequency value equal to 100% for an input indicates that this input variable has been appeared in 100% of the best 30 programs evolved by MEP. The sensitivity analysis results are summarized in Fig. 10. This figure indicates that MEP-based model is more sensitive to e_0, *LL* and *PL*. There is a good agreement between the results of the MEP sensitivity analysis and those reported by other researchers (Daryaee et al., 2010).

7. Conclusion

This paper aimed at developing a new nonlinear MEP-based model for the estimation of the C_c of fine-grained soils using *LL*, *PL* and e_0. A comprehensive database was used for the development of the proposed model. The optimal MEP-based model was selected after several assessment procedures. The validation of the model was verified with different criteria. The results indicate that the proposed model provides precise estimations of C_c. The derived model has a notably better performance than the existing traditional models. Although the existing linear regression-based models may yield accurate results for their relevant databases, their success for other data sets cannot be trusted. This is due to high nonlinearity in the soil compressibility behavior. Despite the slightly better performance of ANN for the investigated problem, a major advantage of MEP over ANN is that it provides simplified equations that can be readily used for the design purposed via hand calculating.

A general criticism about the GP-based models is that they are only randomly formed functions which are not based on the physical processes. This ambiguity was illuminated by the parametric and sensitivity analyses. The consistency between the parametric and sensitivity analysis results and the known behavior of C_c indicates that the derived model is a meaningful combination of the predictor variables. However, MEP uses only the experimental data to specify the model structure. Thus, the derived model mainly has a predictive capability within the data range used for its calibration. This model can be improved to make more accurate predictions for a wider range by including the data for other soil types and test conditions. In general, the models derived using this method are suggested to be used for pre-planning and pre-design purposes or to check the general validity of the laboratory or field test results. Moreover, these models are good alternatives to determine C_c when testing is not possible.

References

Abbasim, N., Javadi, A.A., Bahramloo, R., 2012. Prediction of compression behaviour of normally consolidated fine-grained soils. World Appl. Sci. J. 18 (1), 6–14.

Alavi, A.H., Gandomi, A.H., 2011. A robust data mining approach for formulation of geotechnical engineering systems. Int. J. Comput. Aided Methods Eng.—Eng. Comput. 28 (3), 242–274.

Alavi, A.H., Ameri, M., Gandomi, A.H., Mirzahosseini, M.R., 2011. Formulation of flow number of asphalt mixes using a hybrid computational method. Constr. Build. Mater. 25 (3), 1338–1355.

Alavi, A.H., Gandomi, A.H., Sahab, M.G., Gandomi, M., 2010. Multi expression programming: a new approach to formulation of soil classification. Eng. Comput. 26 (2), 111–118.

ASTM D2435, 2000. Standard Test Methods for One-Dimensional Consolidation Properties of Soils Using Incremental Loading. vol. 04.08, American Society for Testing Materials, USA.

Azamathulla, H.M., 2012. Gene expression programming for prediction of scour depth downstream of sills. J. Hydrol. 460–461, 156–159.

Azamathulla, H.M., Ghani, A.A., Fei, S.Y., 2012. ANFIS-based approach for predicting sediment transport in clean sewer. Appl. Soft Comput. 12 (3), 1227–1230.

Azamathulla, H.M., Guven, A., Demir, Y.K., 2011. Linear genetic programming to scour below submerged pipeline. Ocean Eng. 38 (8–9), 995–1000.

Azamathulla, H.M., Wu, F.C., 2011. Support vector machine approach for longitudinal dispersion coefficients in natural streams. Appl. Soft Comput. 11 (2), 2902–2905.

Azzouz, A.S., Krizek, R.J., Corotis, R.B., 1976. Regression analysis of soil compressibility. Soils Found 16 (2), 19–29.

Banzhaf, W., Nordin, P., Keller, R., Francone, F., 1998. Genetic Programming—An Introduction on the Automatic Evolution of Computer Programs and its Application. dpunkt/Morgan Kaufmann, San Francisco, CA, USA.

Bartlett, S.F., Lee, H.S., 2004. Estimation of Compression Properties of Clayey Soils, Salt Lake Valley, Utah, University of Utah. Report No. UT-04.28. Utah Department of Transportation.

Baykasoglu, A., Gullub, H., Canakcı, H., Ozbakır, L., 2008. Prediction of compressive and tensile strength of limestone via genetic programming. Expert Syst. Appl. 35 (1 and 2), 111–123.

Brameier, M., Banzhaf, W., 2007. Linear Genetic Programming. Springer Science+Business Media, New York.

Carter, M., Bentley, S.P., 1991. Correlations of Soil Properties. Pentech Press, London.

Castilho, V.C., El Debs, M.K., Nicoletti, M.C., 2007. Using a modified genetic algorithm to minimize the production costs for slabs of precast prestressed concrete joists. Eng. Appl. Artif. Intell. 20 (4), 519–530.

Chen, H.M., Kao, W.K., Tsai, H.C., 2012. Genetic programming for predicting a seismic abilities of school buildings. Eng. Appl. Artif. Intell. 25 (6), 1103–1113.

Cheng, M.Y., Tsai, H.C., Sudjono, E., 2012. Evolutionary fuzzy hybrid neural network for dynamic project success assessment in construction industry. Autom. Constr. 21, 46–51.

Çiftçi, O.N., Fadıloğlu, S., Goğüş, F., Guven, A., 2009. Genetic programming approach to predict a model acidolysis system. Eng. Appl. Artif. Intell. 22 (4 and 5), 759–766 (June).

Coello, C.C.A., Lamont, G.B., Van Veldhuizen, D.A., 2007. Evolutionary Algorithms for Solving Multi-Objective Problems. Genetic and Evolutionary Computation, 2nd ed. Springer, New York.

Cozzolino, V.M., 1961. Statistical forecasting of compression index. In: Proceedings of the Fifth International Conference on Soil Mechanics and Foundation Engineering, Paris, vol. 1, pp. 51–53.

Daryaee, M., Kashefipour, M., Ahadiyan, J., Ghobadiyan, R., 2010. Modeling the compression index of fine soils using artificial neural network and comparison with the other empirical equations. J. Water Soil 24 (4), 659–667.

Desai, V.G.M., Desai, V., Rao, D.H. 2009. Prediction of compression index using artificial neural networks. In: Indian Geotechnical Conference (IGC-2009), Guntur, India, pp. 614–617.

Dimopoulos, C., Zalzala, A.M.S., 2001. Investigating the use of genetic programming for a classic one-machine scheduling problem. Adv. Eng. Softw. 32 (6), 489–498.

Emamgholizadeh, S., Bateni, S.M., Jeng, D.S., 2013. Artificial intelligence-based estimation of flushing half-cone geometry. Eng. Appl. Artif. Intell. 26 (10), 2551–2558.

Farkhonde, S., Bolouri, J., 2010. Estimation of compression index of clayey soils using artificial neural network. In: Fifth National Conference on Civil Engineering, Mashhad, Iran, Paper ID 1151.

Ferreira, C., 2001. Gene expression programming: a new adaptive algorithm for solving problems. Complex Syst. 13 (2), 87–129.

Fogel, L.J., Owens, A.J., Walsh, M.J., 1996. Artificial Intelligence Through Simulated Evolution. Wiley, New York.

Francone, F., 1998–2004. Discipulus Owner's Manual. Register Machine Learning Technologies, Littleton, CO, USA.

Francone, F.D., Deschaine, L.M., 2004. Extending the boundaries of design optimization by integrating fast optimization techniques with machine–code–based, linear genetic programming. Inf. Sci. (NY) 161, 99–120.

Frank, I.E., Todeschini, R., 1994. The data analysis handbook. ElsevierAmsterdam, The Netherlands.

Gandomi, A.H., Alavi, A.H., 2011. Multi-stage genetic programming: a new strategy to nonlinear system modeling. Inf. Sci. (NY) 181 (23), 5227–5239.

Gandomi, A.H., Alavi, A.H., Sahab, M.G., 2010. New formulation for compressive strength of CFRP confined concrete cylinders using linear genetic programming. Mater. Struct, Springer, 43; , pp. 963–983.

Gandomi, A.H., Alavi, A.H., Yun, G.J., 2011a. Formulation of uplift capacity of suction Caissons using multi expression programming. KSCE J. Civil Eng. 15 (2), 363–373.

Gandomi, A.H., Alavi, A.H., 2013. Expression programming techniques for formulation of structural engineering systems. In: Gandomi, A.H., et al. (Eds.), Metaheuristic Appl. Struct. Infrastruct.. Elsevier, pp. 437–454 (Chapter 18).

Gandomi, A.H., Alavi, A.H., Yun, G.J., 2011b. Nonlinear modeling of shear strength of SFRCB beams using linear genetic programming. Struct. Eng. Mech. 38 (1), 1–25.

Golbraikh, A., Tropsha, A., 2002. Beware of q2. J. Mol. Graph. Model. 20, 269–276.

Gregory, A.S., Whalley, W.R., Watts, C.W., Hallett, P.D., Whitmore, A.P., Bird, N.R.A., 2006. Calculation of the compression index and precompression stress from soil compression test data. Soil Tillage Res. 89, 45–57.

Gulhati, S.K., Datta, M., 2005. Geotechnical Engineering. Tata Mc Graw Hill Publishing Company Limited, New Delhi.

Guven, A., 2009. Linear genetic programming for time-series modelling of daily flow rate. J. Earth Syst. Sci. 118 (2), 137–146.

Guven, A., Azamathulla, H.M., Zakaria, N.A., 2009. Linear genetic programming for prediction of circular pile scour. Ocean Eng. 36 (12 and 13), 985–991.

Herrero, O.R., 1983. Universal compression index equation; closure. J Geotech Eng Div ASCE 109 (5), 755–761.

Holland, J.H., 1975. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. The University of Michigan Press, Ann Arbor.

Ismail, A., Jeng, D.S., Zhang, L.L., Zhang, J.S., 2013. Predictions of bridge scour: application of a feed-forward neural network with an adaptive activation function. Eng. Appl. Artif. Intell. 26 (5 and 6), 1540–1549.

Javadi, A.A., Rezania, M., Mousavi Nezhad, M., 2006. Evaluation of liquefaction induced lateral displacements using genetic programming. Comput. Geotech. 33 (4 and 5), 222–233.

Jianping, I., Yangsong, Z., Changhong, Y., Guangyun, G., 2010. Application of BP neural network in prediction of compression index of soil. J. Central South Univ. (Sci. Technol.) 41 (2), 722–727.

Keedwell, E., Khu, S.T., 2005. A hybrid genetic algorithm for the design of water distribution networks. Eng. Appl. Artif. Intell. 18 (4), 461–472.

Kerh, T., Lee, C.S., 2006. Neural networks forecasting of flood discharge at an unmeasured station using river upstream information. Adv. Eng. Softw. 37 (8), 533–543.

Kerh, T., Ting, S.B., 2005a. Neural network estimation of ground peak acceleration at stations along Taiwan high-speed rail system. Eng. Appl. Artif. Intell. 18 (7), 857–866.

Kerh, T., Chu, D., 2002. Neural networks approach and microtremor measurements in estimating peak ground acceleration due to strong motion. Adv. Eng. Softw. 33 (11 and 12), 733–742.

Kerh, T., Ting, S.B., 2005b. Neural network estimation of ground peak acceleration at stations along Taiwan high-speed rail system. Eng. Appl. Artif. Intell. 18 (7), 857–866.

Koppula, S.D., 1981. Statistical estimation of compression index. Geotech. Test. J. 4 (2), 68–73.

Koza, J., 1992. Genetic Programming, on the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge, MA.

Kumar, V.P., Rani, C.H.S., 2011. Prediction of compression index of soils using artificial neural networks (ANNs). Int. J. Eng. Res. Appl. (IJERA) 1 (4), 1554–1558.

Kumar, Y., Venkatesh, K., Kumar, V. 2012. Prediction of compression index of cohesive soil using neural network approach. In: International Conference on Advances in Architecture and Civil Engineering, pp. 363–366.

Kuo, Y.L., Jaksa, M.B., Lyamin, A.V., Kaggwa, W.S., 2009. ANN-based model for predicting the bearing capacity of strip footing on multi-layered cohesive soil. Comput. Geotech. 36, 503–516.

Mahmood, M.T., Majid, A., Han, J., Choi, Y.K., 2013. Genetic programming based blind image deconvolution for surveillance systems. Eng. Appl. Artif. Intell. 26 (3), 1115–1123.

Majid, A., Khan, A., Choi, T.S., 2011. Predicting lattice constant of complex cubic perovskites using computational intelligence. Comput. Mater. Sci. 50 (6), 1879–1888.

Masmoudi, M., Haït, A., 2013. Project scheduling under uncertainty using fuzzy modelling and solving techniques. Eng. Appl. Artif. Intell. 26 (1), 135–149.

Mayne, P.W., 1980. Cam-clay predictions of undrained strength. J. Geotech. Eng. Div. ASCE 106 (11), 1219–1242.

Miller, J., Thomson, P., 2002. Cartesian genetic programming. In: Poli, R., Banzhaf, W., Langdon, B., Miller, J., Nordin, P., Fogarty, T.C. (Eds.), Genetic Programming. Springer-Verlag, Berlin.

Muttil, N., Chau, K.W., 2007. Machine learning paradigms for selecting ecologically significant input variables. Eng. Appl. Artif. Intell. 20 (6), 735–744.

Nagaraj, T.S., Murty, B.R.S., 1985. Prediction of the preconsolidation pressure and recompression index of soils. Geotech. Test. J. 8 (4), 199–202.

Nishida, Y., 1956. A brief note on compression index of soils. J. Soil Mech. Found. Div., ASCE 82 (SM3) (1027-1-1027-14).

Oltean, M., 2004. Multi Expression Programming source code. Available at: ⟨http://www.mep.cs.ubbcluj.ro/⟩.

Oltean, M., Dumitrescu, D., 2002. Multi Expression Programming. Babeş-Bolyai University. Technical Report, UBB-01-2002. Cluj-Napoca, Romania.

Oltean, M., Grosşan, C., 2003. A comparison of several linear genetic programming techniques. Adv. Complex Syst. 14 (4), 1–29.

Park, H., Lee, S.R., 2011. Evaluation of the compression index of soils using an artificial neural network. Comput. Geotech. 38, 472–481.

Poli, R., Langdon, W.B., McPhee, N.F., Koza, J.R., 2007. Genetic Programming: An Introductory Tutorial and a Survey of Techniques and Applications. Technical Report [CES-475]. University of Essex, UK.

Rani, C.H.S., Kumar, V.P., Togati, V.K., 2013. Artificial neural networks (ANNS) for prediction of engineering properties of soils. Int. J. Innov. Technol. Explor. Eng. (IJITEE) 3 (1), 123–130.

Rechenberg, I., 1973. Evolutions strategie-OptimierungtechnischerSystemenachPrinzipien der biologischen Evolution. Frommann-Holzboog, Stuttgart.

Rezania, M., Faramarzi, A., Javadi, A.A., 2011. An evolutionary based approach for assessment of earthquake-induced soil liquefaction and lateral displacement. Eng. Appl. Artif. Intell. 24 (1), 142–153.

Roy, P.P., Roy, K., 2008. On some aspects of variable selection for partial least squares regression models. QSAR Comb. Sci. 27, 302–313.

Ryan, C., O'Neill, M., 1998. Grammatical evolution: a steady state approach. In: Koza, J.R. (Ed.), Genetic Programming. University of Wisconsin, Madison, WI.

Sanchis, J., Martínez, M.A., Blasco, X., Reynoso-Meza, G., 2010. Modelling preferences in multi-objective engineering design. Eng. Appl. Artif. Intell. 23 (8), 1255–1264.

Sette, S., Boullart, L., 2001. Genetic programming: principles and applications. Eng. Appl. Artif. Intell. 14 (6), 727–736.

Shahin, M., Jaksa, M., 2005. Neural networks prediction of pullout capacity of marquee ground anchors. Comput. Geotech. 32 (3), 153–163.

Singh, A., Noor, S., 2012. Soil compression index prediction model for fine grained soils. Int. J. Innov. Eng. Technol. (IJIET) 1 (4), 34–37.

Skempton, A.W., 1944. Notes on the compressibility of clays. Quart. J. Geol. Soc. Lond. 100, 119–135.

Smith, G.N., 1986. Probability and Statistics in Civil Engineering. Collins, London.

Sower, G.B., 1970. Introductory Soil Mechanics and Foundation, 3rd ed. The Macmillan company of Collier-Macmillan Ltd, London.

Terzaghi, K., Peck, R.B., 1967. Soil Mechanics in Engineering Practice. John Wiley & Sons Inc., New York.

Tiwari, B.T., Ajmera, B., 2012. New correlation equations for compression index of remoulded clays. J. Geotech. Geoenviron. Eng. 138 (6), 757–762.

Traore, S., Wang, Y.M., Kerh, T., 2010. Artificial neural network for modeling reference evapotranspiration complex process in Sudano–Sahelian zone. Agric. Water Manag. 97 (5), 707–714.

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.

Weise, T., 2009. Global Optimization Algorithms – Theory and Application. Germany: it-weise.de (self-published). [Online]. Available: 〈http://www.it-weise.de〉.

Wroth, C.P., Wood, D.M., 1978. The correlation of index properties with some basic engineering properties of soils. Can. Geotech. J. 15, 137–145.