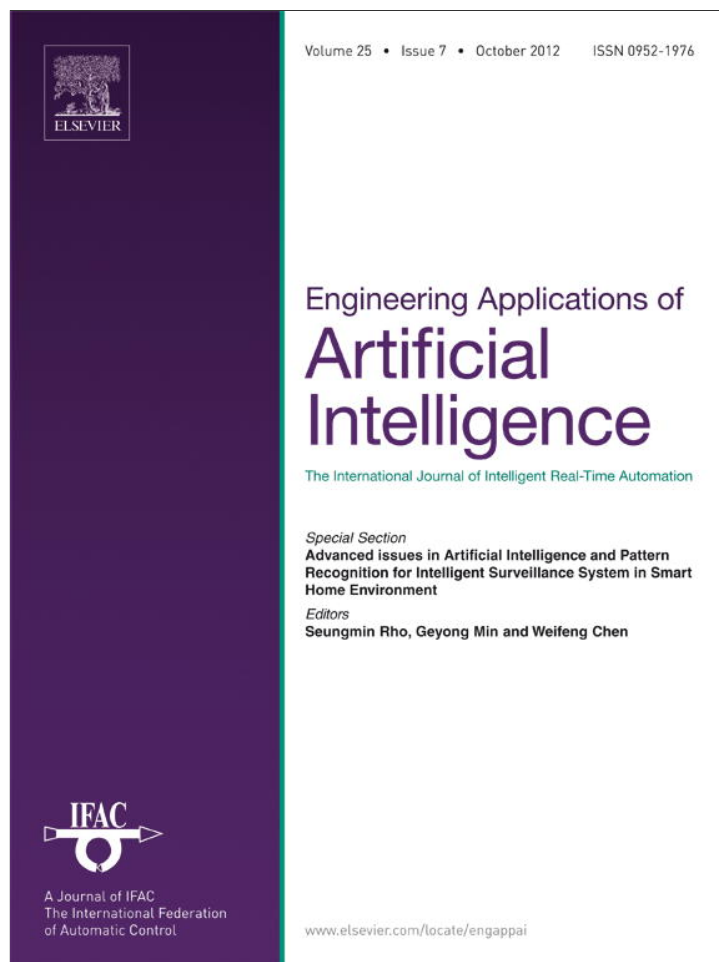


Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>

Contents lists available at [SciVerse ScienceDirect](http://SciVerse.Sciencedirect.com)

Engineering Applications of Artificial Intelligence

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

Nonlinear modeling of soil deformation modulus through LGP-based interpretation of pressuremeter test results

Azadeh Rashed^a, Jafar Bolouri Bazaz^{a,1}, Amir Hossein Alavi^{b,*}^a Civil Engineering Department, Ferdowsi University of Mashhad, Mashhad, Iran^b Young Researchers Club, Mashhad Branch, Islamic Azad University, Mashhad, Iran

ARTICLE INFO

Article history:

Received 13 June 2011

Received in revised form

13 November 2011

Accepted 23 November 2011

Available online 16 December 2011

Keywords:

Soil deformation modulus

Linear genetic programming

Pressuremeter test

Soil physical properties

ABSTRACT

Soil deformation modulus is an essential parameter for the analysis of behavior of substructures. Despite its importance, little attention is paid to developing empirical models for predicting the deformation moduli obtained from the field tests. To cope with this issue, this paper presents the development of a new prediction model for the pressuremeter soil deformation modulus utilizing a linear genetic programming (LGP) methodology. The LGP model relates the soil secant modulus obtained from the pressuremeter tests to the soil index properties. The best model was selected after developing and controlling several models with different combinations of the influencing parameters. The experimental database used for developing the models was established upon several pressuremeter tests conducted on different soil types at depths of 3–40 m. To verify the applicability of the derived model, it was employed to estimate the soil moduli of portions of test results that were not included in the analysis. Further, the generalization capability of the model was verified via several statistical criteria. The sensitivity of the soil deformation modulus to the influencing variables was examined and discussed. Moisture content and soil dry unit weight were found to be efficient representatives of the initial state and consolidation history of the soil for determining its deformation modulus. The results indicate that the LGP approach accurately characterizes the soil deformation modulus leading to a very good prediction performance. The correlation coefficients between the experimental and predicted soil modulus values are equal to 0.908 and 0.901 for the calibration and testing data sets, respectively.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

In general, soils undergo both elastic and plastic deformations when subjected to loading. Elastic and plastic behaviors are considered as soil recoverable and irrecoverable deformations, respectively (Karmakar et al., 2004). To analyze the elasto-plastic behavior of soils, stress–strain (load–settlement) curves of laboratory or field test results can be used (Briaud, 2001; Briaud et al., 2006; Mollahasani et al., 2011). Fig. 1 presents a typical stress–strain curve. Various soil moduli, including secant (E_p) modulus can be defined referring to this figure. E_p can be calculated from the secant slope (S_p) corresponding to the slope from the origin (O) to L_1 .

The soil deformation moduli are usually evaluated from laboratory or field methods. The field test results have been found to be more reliable than those of the laboratory methods

(Reznik, 1995; Murthy, 2008; Mollahasani et al., 2011). The field tests are very sensitive and costly. Furthermore, it is not always possible to conduct such tests. These limitations imply the necessity of conducting more research on the development of prediction models for the deformation moduli through the interpretation of the field test results. In this context, Reznik (1995) proposed prediction equations describing the dependence of the deformation moduli of collapsible soils obtained from plate load tests on void ratio and moisture content. Recently, Mollahasani et al. (2011) proposed empirical models to predict the plate load soil moduli using gene expression programming. Among the field tests, pressuremeter test is one of the well-known tests for determining the soil deformation modulus. Despite the high reliability of the pressuremeter test results (Murthy, 2008), investigations on deriving prediction models for the pressuremeter soil modulus are conspicuous by their absence.

The main purpose of this paper is to investigate the potential of linear genetic programming (LGP) technique to obtain new empirical relationships for determining the pressuremeter soil secant modulus. Various predictor variables employed for the analysis were coarse and fine-grained contents, grains size characteristics,

* Corresponding author.

E-mail addresses: azade.rashed@gmail.com (A. Rashed), bolouri@ferdowsi.um.ac.ir (J.B. Bazaz), ah_alavi@hotmail.com (A.H. Alavi).

¹ Associate Professor.

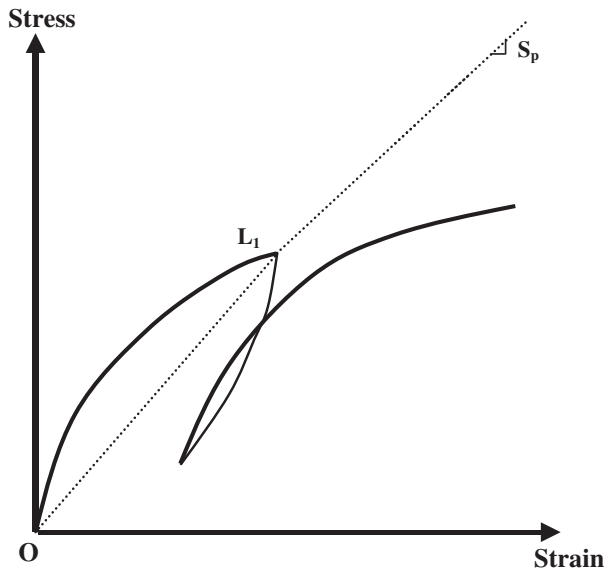


Fig. 1. Definitions of soil secant modulus (Briaud, 2001).

liquid limit, moisture content, and soil unit weight. A database of consisting of 106 pressuremeter test results was used for the analysis. The paper is organized as follows: Section 2 presents a brief literature review. Section 3 provides descriptions of the methodology used for the formulation of the soil modulus. Section 4 outlines the model development using LGP, experimental study and reviews the results. The detailed performance analysis of the proposed model is discussed in Section 5. The results of the sensitivity and parametric analyses are given in Sections 6 and 7, respectively. Finally, concluding remarks are outlined in Section 8.

2. Literature review

Numerous computer-aided modeling tools have been proposed by extending developments in computational software and hardware. Genetic programming (GP; Koza, 1992) is a new approach for behavioral modeling of engineering problems. GP is a specialization of genetic algorithms (GA) where the solutions are computer programs rather than binary strings. One of the main features of GP is its ability to generate greatly simplified prediction equations without assuming prior form of the existing relationship (Alavi et al., 2011). For the last decade, GA, GP, variants of GP, and other well-known soft computing techniques such as artificial neural networks (ANNs) and support vector machines (SVM) have been pronounced as alternative methods for simulating the behavior of engineering problems (Cheng et al., 2002; Xie et al., 2006; Lin et al., 2006; Chau, 2006; Javadi et al., 2006; Rezania and Javadi, 2007; Muttill and Chau, 2007; Jia et al., 2008; Ciftci et al., 2009; Javadi and Rezania, 2009; Pal et al., 2011; Rezania et al., 2011; Gandomi et al., 2011a; Gandomi and Alavi 2011). LGP (Brameier and Banzhaf, 2007) is a new subset of GP. LGP operates on programs that are represented as linear sequences of instructions of an imperative programming language (Brameier and Banzhaf, 2001, 2007). In contrast with traditional GP and other soft computing techniques, applications of LGP in the field of civil engineering are totally new and original (Guvén, 2009; Guvén et al., 2009; Gandomi et al., 2010; Alavi and Gandomi, 2011).

3. Linear genetic programming

GP creates computer programs to solve a problem using the principle of Darwinian natural selection (Koza, 1992). Most of the

genetic operators used in GA can also be implemented in GP with minor changes. The main difference between GP and GA is the representation of the solution. GA creates a string of numbers that represent the solution. The GP solutions are computer programs represented as tree structures and expressed in a functional programming language (such as LISP; Koza, 1992; Alavi et al., 2011). In other words, the programs (individuals) evolved by GP are parse trees that can vary in length throughout the run rather than fixed-length binary strings. GP gives the basic structure of the approximation model together with the values of its parameters (Javadi and Rezania, 2009). The fitness of each program in the population is evaluated using a fitness function. Thus, the fitness function is the objective function GP aims to optimize (Gandomi et al., 2011b).

In addition to classical tree-based GP, there are other types of GP where programs are represented in different ways. These are linear and graph-based GP (Banzhaf et al., 1998). Recently, several linear variants of GP have been developed such as linear genetic programming (LGP) and multi-expression programming (MEP). The linear variants of GP make a clear distinction between the genotype and phenotype of an individual. In these variants, individuals are represented as linear strings (Oltean and Grosan, 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 many runs to be conducted in realistic timeframes. This leads to derivation of consistent and high-precision models with little customization (Francone and Deschaine, 2004; Poli et al., 2007; Gandomi et al., 2011b).

LGP is a new subset of GP with a linear structure similar to the deoxyribonucleic acid (DNA) molecule in biological genomes. Similar to classical GP, the major difference between LGP and GA pertains to the representation of the solution. A comparison of classical GP with LGP reveals that LGP substitutes expressions of a functional programming language (such as LISP) in GP by programs of an imperative language (such as C/C++; Brameier and Banzhaf, 2001, 2007). Fig. 2 presents a comparison of structure of a program evolved by LGP and classical GP. Furthermore, this figure demonstrates a typical conversion of the evolved programs into a functional representation ($y=f[0]=(v[0]/1)/v[1]$). The LGP program is converted into a functional form by successive replacements of the variables starting with the last effective instruction (Oltean and Grosan, 2003). The translation of the program generated by classical GP into explicit form can be obtained by reading the tree structure from left to right. As shown in Fig. 2a, a linear genetic program can be seen as a data flow graph generated by multiple usage of register content. In

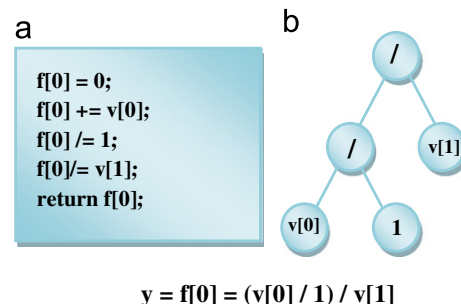


Fig. 2. A comparison of a GP program structure evolved by: (a) LGP (b) Classical tree-based GP (Gandomi et al., 2011b).

classical tree-based GP, the data flow is more rigidly determined by the tree structure of the program (Brameier and Banzhaf, 2001; Gandomi et al., 2011b).

In the LGP system described here, a program is interpreted as a variable-length sequence of simple C instructions. The instruction set or function set of LGP contains arithmetic operations, conditional branches, and function calls. The terminal set of the system is composed of variables and constants. The instructions are restricted to operations that accept a minimum number of constants or memory variables, called registers (r), and assign the result to a destination register, e.g., $r_0 := r_1 + 1$. A part of a linear genetic program in C code is represented in Fig. 3. In this figure, register $r[0]$ holds the final program output (Gandomi et al., 2010).

Here are the steps which the LGP system follows for a single run (Brameier and Banzhaf, 2007; Gandomi et al., 2010):

- I. Initializing a population of randomly generated programs and calculating their fitness values.
- II. Running a Tournament. In this step four programs are selected from the population randomly. They are compared based on their fitness. Two programs are then picked as the winners and two as the losers.
- III. Transforming the winner programs. After that, two winner programs are copied and transformed probabilistically into two new programs via crossover and mutation operators.
- IV. Replacing the loser programs in the tournament with the transformed winner programs. The winners of the tournament remain unchanged.
- V. Repeating steps two through four until termination or convergence conditions are satisfied.

Crossover occurs between instruction blocks. Fig. 4 demonstrates a two-point linear crossover used in LGP for recombining two tournament winners. As it is seen, a segment of random position and arbitrary length is selected in each of the two parents and exchanged. If one of the two children would exceed the maximum length, crossover is aborted and restarted with exchanging equally sized segments (Brameier and Banzhaf, 2001; Gandomi et al., 2011b). The mutation operation occurs on a single instruction. Two commonly used types of standard LGP mutations are micro- and macro-mutation. The micro-mutation changes an operand or an operator of an instruction. The macro-mutation operation inserts or deletes a random instruction (Brameier and Banzhaf, 2001; Gandomi et al., 2011b).

4. Numerical simulation of soil deformation modulus

The significant influence of the soil physical properties such as particle size distribution, dry unit weight, moisture content, and plasticity on its mechanical properties is well understood (Briaud, 2001; Reznik, 2007; Mollahasani et al., 2011). For instance, dry

```

r[0] = r[1] + 1;
r[2] = r[0] - 2;
if (r[2] > 2)
r[0] = r[2] × r[1];
r[2] = r[0] + 3;
r[0] = sin(r[2]);
    
```

Fig. 3. An excerpt of a linear genetic program.

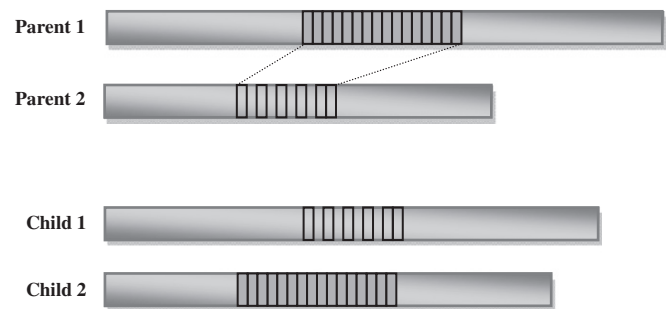


Fig. 4. Crossover in LGP (Brameier and Banzhaf, 2001).

unit weight is an indicator of compressibility of a soil. If the soil particles are closely packed, the modulus tends to be high. The moisture content also has a major influence on the soil modulus. At very low water contents, the compaction of coarse-grained soils is not as efficient as it is at higher moisture contents since the lubrication effect of water is not sufficient. In this case, very low moisture content results in low modulus (Briaud, 2001; Mollahasani et al., 2011). Herein, the LGP approach was employed to develop a new prediction equation for the soil secant modulus (E_p). The most important factors representing the behavior of the soil deformation modulus were detected on the basis of a literature review (Briaud, 2001; Briaud et al., 2006; Reznik, 1995, 2007; Mollahasani et al., 2011). E_p (kg/cm^2) was considered to be a function of the following parameters:

$$E_p = f(\text{FC}, \text{CC}, D_{10}, D_{30}, D_{60}, C_u, C_c, \text{LL}, \omega, \gamma, \gamma_d) \quad (1)$$

where

- FC (%): Fine-grained content,
- CC (%): Coarse-grained content,
- D_{10} (mm): Grain size for which 10 percentage of the sample is finer,
- D_{30} (mm): Grain size for which 30 percentage of the sample is finer,
- D_{60} (mm): Grain size for which 60 percentage of the sample is finer,
- C_u : Coefficient of uniformity (D_{60}/D_{10}),
- C_c : Coefficient of curvature ($(D_{30})^2/(D_{60} \times D_{10})$),
- LL (%): Liquid limit,
- ω (%): Moisture content,
- γ (kN/m^3): Soil unit weight,
- γ_d (kN/m^3): Soil dry unit weight.

FC, CC, D_{10} , D_{30} , D_{60} , C_u , C_c , and LL represent the intrinsic soil properties. ω , γ_d , and γ carry information on the state of the soil and its compressibility and previous history. Over-consolidation ratio (OCR) could have been included in the analysis. However, it was not included herein as it should be obtained from time-consuming laboratory tests. On the other hand, γ and γ_d can be easily calculated for a soil (Mollahasani et al., 2011).

4.1. Experimental study and test results

The experimental program consisted of laboratory and field tests. The field activities included the pressuremeter tests. Various aspects of the pressuremeter test, including test setup and test procedure, are described below. For the laboratory testing purposes, several disturbed and undisturbed soil samples were taken from the sites. A soil sample is called “disturbed” if the tests

of its structural properties are not representatives of in-situ conditions. In other words, the structure of a disturbed soil is sufficiently changed during the sampling process. Otherwise, the soil sample is called “undisturbed”. The undisturbed soil samples are obtained by methods in which disturbance to the sample is minimized. After extracting, the cores were carefully taken to the geotechnical laboratory and maintained in a wet chamber to avoid loosing of water content.

4.1.1. Basic geotechnical characterization tests

Extensive geotechnical laboratory tests were carried out for determining the physical and plastic characteristics of soil. The tests included measuring the water (or moisture) content (ω), defined as the ratio between the mass of water and the mass of dry soil; natural unit weight (γ), which is obtained by measuring the relationship between the weight and volume of an undisturbed soil sample, Atterberg limits (plastic and liquid limit), specific gravity, and grain size distribution. Fig. 5 illustrates the range of grain size distribution of the samples tested. Different soil types tested were silty clay with sand (CL–ML), gravelly lean clay with sand (CL), silty, clayey sand (SC–SM), gravelly silt with sand (ML), and silty gravel with sand (GM), silty sand (SM), and silty, clayey gravel (GC–GM).

4.1.2. Pressuremeter tests

The pressuremeter test is a traditional in-situ method for estimating the stress–strain response of the soil. Within the scope of this study, 106 pressuremeter tests were performed to investigate the pressure–volume deformation characteristics of soils at some locations in Khorasan Province, Iran. These tests were conducted by Menard pressuremeter. This apparatus consists of three parts, namely probe, control unit and tubing. The pressuremeter tests conducted in the present study followed the procedures described in ASTM D4719-87 (1987). The following technique was employed to perform the pressuremeter tests. Before beginning the pressuremeter tests, the pressuremeter was calibrated for pressure loss and volume loss. The pressure loss, P_c , occurs due to the rigidity of the probe membranes. For determining the P_c , the probe was placed in the open air and inflated step by step with increasing the pressure. Volume loss, V_c , occurs due to the expansion of the tubing system and the compressibility of any part of the testing equipment. V_c was determined by putting the probe in a steel tube and inflating the tube by application of water pressure. Diameter of the probe used in this study was 58 mm and the diameters of the holes drilled for conducting the tests were 101, 116, and 125 mm. The probe was lowered to the hole soon after boring to the desired elevation. Then the test was started by opening the valves for admitting water and gas to the measuring cell and guard cells, respectively.

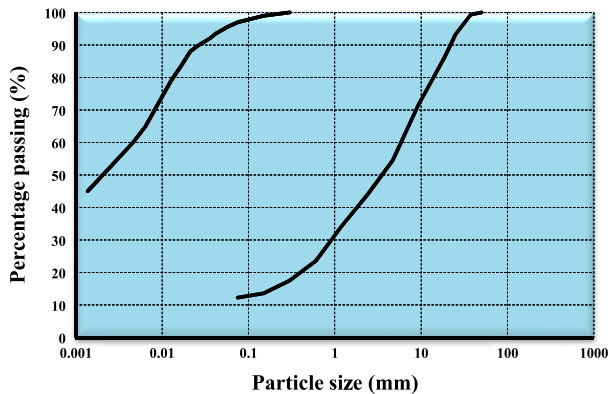


Fig. 5. Lower and upper limits of the grain size distribution of the soil samples.

The pressure within the measuring cell was held constant for approximately 60 s. Then, the increase in volume required to maintain the constant pressure was recorded. Steps were continued until yielding in the soil became disproportionate. Finally, a corrected pressure–volume (P – V) curve was plotted. The corrected pressure (P) and volume (V) were obtained using the following equations:

$$P = P_r + P_w - P_c \tag{2}$$

$$V = V_r - V_c \tag{3}$$

where P_r and V_r are the actual pressure and volume read from the control unit, respectively. P_c and V_c are the pressure and volume loss, respectively. $P_w = \gamma_w \times H_w$ in which H_w is the difference in head between the center of the measuring cell in the bore hole and the pressure gage in central unit, and γ_w is the unit weight of water. A typical corrected pressuremeter curve is shown in Fig. 6. The initial part of the curve (OA) represents the recompression of the relaxed soil around the borehole. At the end of this part, P_o is the pressure which corresponds to the volume V_o at the start of the straight line where the yielded wall of the hole is pushed to its original position. The second part (AB) indicates the elastic deformation of the soil. It is also called the pseudo-elastic phase of the test. P_r is known as the creep pressure. The curve BC marks the plastic phase that becomes eventually asymptotic at limit pressure P_l at which the soil is deemed to have failed. The formulation used for determining the pressuremeter modulus is as follow (Murthy, 2008):

$$E_p = 2(1 + \nu)(V_o + V_m) \frac{\Delta P}{\Delta V} \tag{4}$$

where

- E_p (kPa): Pressuremeter modulus,
- ν : Poisson ration,
- V_o : Volume of the uninflated probe at ground surface,
- ΔP : Corrected pressure increase in the center part of the straight line portion of the pressure–volume curve,
- ΔV : Corrected volume increase in the center part of the straight line portion of the pressure–volume curve, corresponding to ΔP pressure increase,
- V_m : Corrected volume reading in the center portion of the ΔV volume increase.

The complete list of the data is presented in Table 1. The information cited in this table includes FC, D_{10} , D_{30} , D_{60} , C_u , C_c , LL, ω , and γ_d . E_p is the measured soil deformation modulus. A major part of the database comprises the test results on fine-grained soil

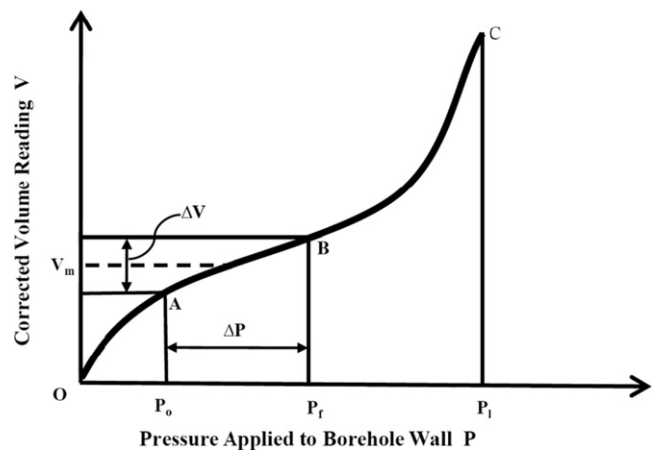


Fig. 6. A typical corrected pressuremeter curve.

Table 1
Geotechnical properties of soils used for the model development.

No.	FC (%)	D ₁₀ (mm)	D ₃₀ (mm)	D ₆₀ (mm)	C _u	C _c	LL (%)	ω (%)	γ _d (kN/m ³)	E _p (kg/cm ²)
1	93.6	0.0007	0.0043	0.027	38.571	0.978	42	16	15.5	133.0
2*	71.0	0.0001	0.016	0.051	510.000	50.196	18	21	15.5	339.0
3	97.0	0.0001	0.0003	0.007	70.000	0.129	40	22	15.5	193.0
4	76.9	0.0008	0.0065	0.03	37.500	1.760	30	27	16.0	202.0
5	94.0	0.0004	0.0011	0.024	60.000	0.126	28	20	15.5	129.0
6	96.0	0.0001	0.007	0.014	140.000	35.000	35	22	15.5	143.0
7	90.5	0.0001	0.0017	0.017	170.000	1.700	28	18	16.0	179.0
8	98.0	0.0001	0.00017	0.012	120.000	0.024	17	18	16.0	106.0
9	47.8	0.0001	0.025	0.28	2800.000	22.321	17	26	16.0	442.0
10	17.6	0.013	0.48	1.5	115.385	11.815	17	26	16.0	629.0
11	86.0	0.0001	0.003	0.02	200.000	4.500	17	26	16.0	221.0
12	99.4	0.0001	0.011	0.031	310.000	39.032	27	18	16.5	103.2
13	93.2	0.002	0.0084	0.025	12.500	1.411	34	17	17.8	225.2
14	61.0	0.0017	0.024	0.07	41.176	4.840	33	22	15.3	383.1
15	96.9	0.0007	0.007	0.022	31.429	3.182	34	10	17.5	137.7
16	94.9	0.006	0.02	0.034	5.667	1.961	24	19	17.0	121.9
17	86.6	0.0013	0.0073	0.035	26.923	1.171	31	15	17.0	281.3
18	63.4	0.0029	0.021	0.064	22.069	2.376	23	12	19.8	295.7
19	87.8	0.0012	0.017	0.03	25.000	8.028	25	17	16.9	238.9
20	66.7	0.0029	0.0165	0.06	20.690	1.565	22	23	16.7	313.7
21	87.8	0.0028	0.0115	0.035	12.500	1.349	26	17	17.8	295.2
22	86.75	0.0018	0.0081	0.031	17.222	1.176	27	13	17.0	231.0
23	92.75	0.0015	0.0085	0.025	16.667	1.927	34	13	17.0	83.0
24	66.06	0.003	0.041	0.063	21.000	8.894	23	13	17.0	567.0
25	93.66	0.0032	0.022	0.045	14.063	3.361	29	16	17.2	81.0
26	83.85	0.0061	0.042	0.064	10.492	4.518	22	16	17.2	148.0
27	80.44	0.0044	0.023	0.059	13.409	2.038	26	16	17.2	512.0
28	91.39	0.0044	0.031	0.06	13.636	3.640	26	16	17.2	207.0
29	97.0	0.0001	0.003	0.015	150.000	6.000	26	17	16.1	89.1
30	52.0	0.0004	0.005	0.84	2100.000	0.074	21	20	16.8	449.8
31	96.8	0.0011	0.004	0.021	19.091	0.693	20	19	16.1	166.4
32	90.0	0.0006	0.001	0.017	28.333	0.098	32	19	16.3	199.8
33	97.5	0.0007	0.0067	0.033	47.143	1.943	29	20	16.5	85.4
34	71.7	0.0006	0.004	0.03	50.000	0.889	32	26	16.8	321.7
35	86.5	0.0044	0.02	0.033	7.500	2.755	23	16	16.4	203.7
36	80.0	0.0007	0.009	0.029	41.429	3.990	17	9	18.0	288.7
37	81.4	0.0002	0.008	0.034	170.000	9.412	17	26	15.0	299.7
38	59.0	0.0022	0.013	0.074	33.636	1.038	17	10	19.0	353.9
39	32.5	0.0069	0.05	1.4	202.899	0.259	23	19	18.0	920.0
40	47.8	0.0001	0.018	0.6	6000.000	5.400	23	14	16.2	421.4
41	47.5	0.0003	0.018	0.95	3166.667	1.137	26	14	16.5	612.9
42	94.6	0.0001	0.006	0.025	250.000	31.000	22	18	17.5	156.0
43	12.3	0.007	0.9	6	857.143	19.286	16	6	14.6	1077.7
44	47.8	0.0001	0.013	0.7	7000.000	2.414	28	15	15.5	626.5
45	56.7	0.0003	0.008	0.15	500.000	1.422	28	15	15.5	568.7
46	92.0	0.0009	0.0071	0.02	22.222	2.801	27	15	15.5	121.1
47	87.5	0.0005	0.0035	0.025	50.000	0.980	30	16	17.0	252.7
48	52.0	0.0005	0.014	0.17	340.000	2.306	23	16	17.0	321.6
49	84.5	0.0006	0.0015	0.018	30.000	0.208	36	14	14.5	285.0
50	73.4	0.0014	0.0114	0.04	28.571	2.321	31	18	14.3	338.6
51	40.0	0.0007	0.02	2.1	3000.000	0.272	28	14	17.5	462.0
52	89.6	0.0007	0.003	0.018	25.714	0.714	29	18	14.5	253.7
53	91.8	0.0006	0.0027	0.021	35.000	0.579	29	18	14.5	101.9
54	90.8	0.0012	0.009	0.024	20.000	2.813	35	18	14.5	168.2
55	86.0	0.0023	0.01	0.03	13.043	1.449	38	4	19.0	283.0

Table 1 (continued)

No.	FC (%)	D ₁₀ (mm)	D ₃₀ (mm)	D ₆₀ (mm)	C _u	C _c	LL (%)	ω (%)	γ _d (kN/m ³)	F _p (kg/cm ²)
56	87.5	0.0004	0.0014	0.014	35,000	0.350	31	17	17.0	241.0
57	92.0	0.0004	0.0027	0.028	70,000	0.651	34	17	17.0	68.6
58	91.0	0.0002	0.005	0.028	140,000	4.464	17	18	17.5	133.3
59	97.0	0.0001	0.0012	0.02	200,000	0.720	31	18	17.5	91.3
60	60.0	0.0032	0.0215	0.073	22,813	1.979	18	15	17.5	358.3
61	89.6	0.0055	0.019	0.039	7,091	1.683	24	11	17.0	311.1
62	99.0	0.0012	0.011	0.029	24,167	3.477	28	17	17.0	101.3
63	71.0	0.0013	0.013	0.063	63,000	2.683	30	17	17.0	259.1
64	88.4	0.00155	0.014	0.044	28,387	2.874	30	10	18.9	266.7
65	97.7	0.0015	0.0084	0.021	14,000	2.240	29	17	15.0	150.2
66	64.8	0.0021	0.014	0.06	28,571	1.556	23	16	17.5	288.2
67	56.0	0.0012	0.015	0.1	83,333	1.875	18	16	17.5	331.0
68	78.8	0.001	0.011	0.045	45,000	2.689	28	15	18.1	240.9
69	81.0	0.0006	0.0013	0.025	41,667	0.113	29	16	18.3	211.6
70	94.5	0.0005	0.004	0.021	42,000	1.524	28	17	18.5	24.2
71	33.0	0.0005	0.04	1.1	2200,000	2.909	17	17	18.5	465.3
72	49.2	0.013	0.051	1.1	8,462	1.819	20	14	18.9	373.0
73	48.0	0.004	0.02	0.25	62,500	0.400	17	12	18.0	337.7
74	20.0	0.02	0.21	0.595	29,750	3.706	17	12	18.0	620.3
75	62.5	0.0006	0.012	0.05	83,333	4.800	28	17	16.8	377.7
76	28.0	0.008	0.1	1.2	150,000	1.042	21	17	16.8	898.4
77	26.5	0.0056	0.15	1.3	232,143	3.091	21	17	18.0	611.7
78	37.5	0.0053	0.031	0.71	133,962	0.255	21	10	19.0	585.9
79	26.7	0.009	0.15	1.44	160,000	1.736	20	10	19.0	652.8
80	22.0	0.011	0.595	6	545,455	5.364	19	12	17.5	703.5
81	35.5	0.0009	0.04	1	1111,111	1.778	24	11	19.5	643.7
82	44.0	0.001	0.023	1.1	1100,000	0.481	17	7	19.0	474.94
83	74.4	0.0003	0.0095	0.03	100,028	10.028	24	15.94	19.5	305.46
84	30.0	0.002	0.083	2.2	1100,000	1.566	23	11	19.5	575.22
85	36.0	0.0011	0.04	2.1	1909,091	0.693	23	11	19.5	453.91
86	93.0	0.0003	0.0042	0.012	40,000	4.900	39	4	19.0	152.8
87	45.0	0.0033	0.013	0.8	242,424	0.064	23	4	19.0	523.6
88	38.0	0.0041	0.022	2	487,805	0.059	17	4	19.0	409.0
89	41.0	0.0041	0.019	1.3	317,073	0.068	17	4	19.0	473.2
90	87.0	0.0065	0.00477	0.0165	25,385	2.121	30	4	18.7	249.0
91	34.0	0.0044	0.05	9.7	2204,545	0.059	17	4	19.0	447.0
92	34.0	0.0045	0.05	9.6	2133,333	0.058	17	4	19.7	477.8
93	94.0	0.0007	0.0084	0.0215	30,714	4.688	35	3	19.0	193.0
94	70.0	0.0065	0.007	0.035	53,846	2.154	33	3	19.6	291.0
95	51.0	0.0006	0.011	0.6	1000,000	0.336	38	4	19.0	446.0
96	96.7	0.00006	0.0005	0.018	300,000	0.231	30	22	16.0	43.0
97	90.0	0.00005	0.0007	0.0138	276,000	0.710	32	22	16.0	102.0
98	99.9	0.0001	0.0008	0.014	140,000	0.457	32	22	16.0	70.0
99	100.0	0.00008	0.00143	0.021	262,500	1.217	28	22	16.0	227.0
100	97.5	0.00007	0.0005	0.015	214,286	0.238	29	20	16.5	197.0
101	97.6	0.00007	0.00005	0.0115	164,286	0.003	34	20	16.5	190.0
102	88.8	0.0001	0.0022	0.023	230,000	2.104	27	20	16.5	281.0
103	86.8	0.0003	0.0035	0.022	73,333	1.856	29	20.85	16.0	221.0
104	56.5	0.00065	0.0021	2	3076,923	0.003	31	24	16.0	354.0
105	98.7	0.0007	0.0053	0.014	20,000	2.866	29	18	15.5	47.0
106	98.0	0.0004	0.014	0.03	75,000	16.333	23	17	16.5	66.0

* Bold sets are testing data sets.

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

Parameter	Minimum	Maximum	Mean	Standard deviation
FC (%)	12.28	100	71.35	25.3
D_{10} (mm)	0.0001	0.02	0.0021	0.0032
D_{30} (mm)	0.0001	0.9	0.04	0.11
D_{60} (mm)	0.01	9.7	0.59	1.58
C_u	5.67	70000	473.61	1102/38
C_c	0.003	50.2	4.06	7.93
LL (%)	16	41.9	26.02	6.28
ω (%)	5.9	27	16.07	4.26
γ_d (kN/m ³)	14	20.5	17.05	1.46
γ (kN/m ³)	15.4614	22.6083	19.72715	1.193521
E_p (kg/cm ²)	24.15	1077.65	305.66	201.8

samples. The descriptive statistics of the test results is given in Table 2. To visualize the distribution of the samples, the data are presented by frequency histograms (Fig. 7).

4.2. Data preprocessing

For the analysis, the available data sets were randomly divided into learning, validation and testing subsets. The learning data were used for the training process (genetic evolution). The validation data were used to specify the generalization capability of the evolved programs on data they did not train on (model selection). In other words, the learning and validation data sets were used to select the best evolved programs and included in the training process. Thus, they were categorized into one group referred to as “training data”. The testing data were finally used to measure the performance of the models obtained by LGP on data that played no role in building the models (Gandomi et al., 2011b). A trial study was conducted to find a consistent data division. The selection was such that the statistical properties (e.g. mean and standard deviation) of the training and testing subsets were similar. Out of the 106 data, 79 data were used as the training data (70 sets as the learning data and 9 sets as the validation data). The remaining 27 data sets were taken for the testing of the generalization capability of the models.

4.3. Parameters for measuring performance

The parameters used to evaluate the performance of the models were correlation coefficient (R), root mean squared error ($RMSE$) and mean absolute error (MAE). These parameters were calculated using the following equations:

$$R = \frac{\sum_{i=1}^n (h_i - \bar{h}_i)(t_i - \bar{t}_i)}{\sqrt{\sum_{i=1}^n (h_i - \bar{h}_i)^2 \sum_{i=1}^n (t_i - \bar{t}_i)^2}} \quad (5)$$

$$RMSE = \frac{\sqrt{\sum_{i=1}^n (h_i - t_i)^2}}{n} \times 100 \quad (6)$$

$$MAE = \left[\frac{1}{n} \sum_{i=1}^n |h_i - t_i| \right] \quad (7)$$

where h_i and t_i are the actual and predicted output values for the i th output, respectively. \bar{h}_i and \bar{t}_i are, respectively, the average of the actual and predicted outputs, and n is the number of samples.

4.4. LGP-based prediction model for soil deformation modulus

The available database was used for establishing the LGP prediction model relating E_p to FC, CC, D_{10} , D_{30} , D_{60} , C_u , C_c , LL, ω , γ , and γ_d . GP and its variants such as LGP are able to find the global

solution without difficulties using the crossover and mutation operations (Galvan-Lopez et al., 2008). This ability to find the very best, or global solution, cannot be guaranteed mathematically as nature does not guarantee perfection either. In this study, several runs were conducted to obtain a parameterization of LGP that provided enough robustness and generalization to solve the problem. The LGP parameters were changed for different runs to find the global solution. The parameters were selected on the basis of both previously suggested values (Francone, 2001; Baykasoglu et al., 2008; Gandomi et al., 2010; Alavi and Gandomi, 2011) and making several preliminary runs and observing the performance behavior. Three optimal levels were set for the population size (500, 1500, and 3000) and two optimal levels were considered for the crossover rate (50% and 95%) and mutation rate (50% and 90%). The success of the LGP algorithm usually increases with increasing the initial and maximum program size parameters. In this case, the complexity of the evolved functions increases and the speed of the algorithm decreases. The initial and maximum program sizes were, respectively, set to optimal values of 80 and 256 bytes as tradeoffs between the running time and the complexity of the evolved solutions. The number of demes was set to 20. This parameter is related to the way that the population of programs is divided. Note that demes are semi-isolated subpopulations that evolution proceeds faster in them in comparison to a single population of equal size (Brameier and Banzhaf, 2007). In this study, four basic arithmetic operators (+, −, ×, /) and basic mathematical functions (√, sin, cos) were utilized to get the optimum LGP models. There are 3 (optimal population size levels) × 2 (optimal crossover rate levels) × 2 (optimal mutation rate levels) = 12 different combinations of the parameters of the LGP algorithm. All of these parameter combinations were tested and 5 replications for each were carried out. Therefore, the overall number of runs was equal to 12 × 5 = 60 for each of the input combinations. A fairly large number of tournaments (900,000) were tested on each run to find models 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 run terminated automatically. Each run was observed while in progress for overfitting. For this aim, situations were checked in which the fitness of the samples for the learning of LGP was negatively correlated with the fitness on the validation data sets. To evaluate the fitness of the evolved program, the average of the squared raw errors was used. For the runs showing signs of overfitting, the LGP parameters were progressively changed so as to reduce the computational power available to the LGP algorithm until the observed overfitting was minimized. The resulting run was then accepted as the production run. For the LGP-based analysis, the Discipulus software (Conrads et al., 2004) was used.

An extensive trial study was performed to select the most relevant input parameters for the LGP models. Several LGP models were developed using different combinations of the input parameters. The best model was chosen on the basis of a multi-objective strategy as follows:

- i. Finding the simplest model, although this was not a predominant factor. This item was considered for the cases where the models evolved by LGP had the same input variables and a similar performance. In such cases, the structurally simpler model was kept as the optimal one.
- ii. Providing the best fitness value on the training (learning and validation) set of data.

The best LGP model for predicting E_p was built using FC, C_u , ω , and γ_d . The best LGP program obtained at the end of training in C is given in Appendix A. This program can be run in C++ environment. The resulting code may be linked to the optimizer and compiled or it may be called from the optimization

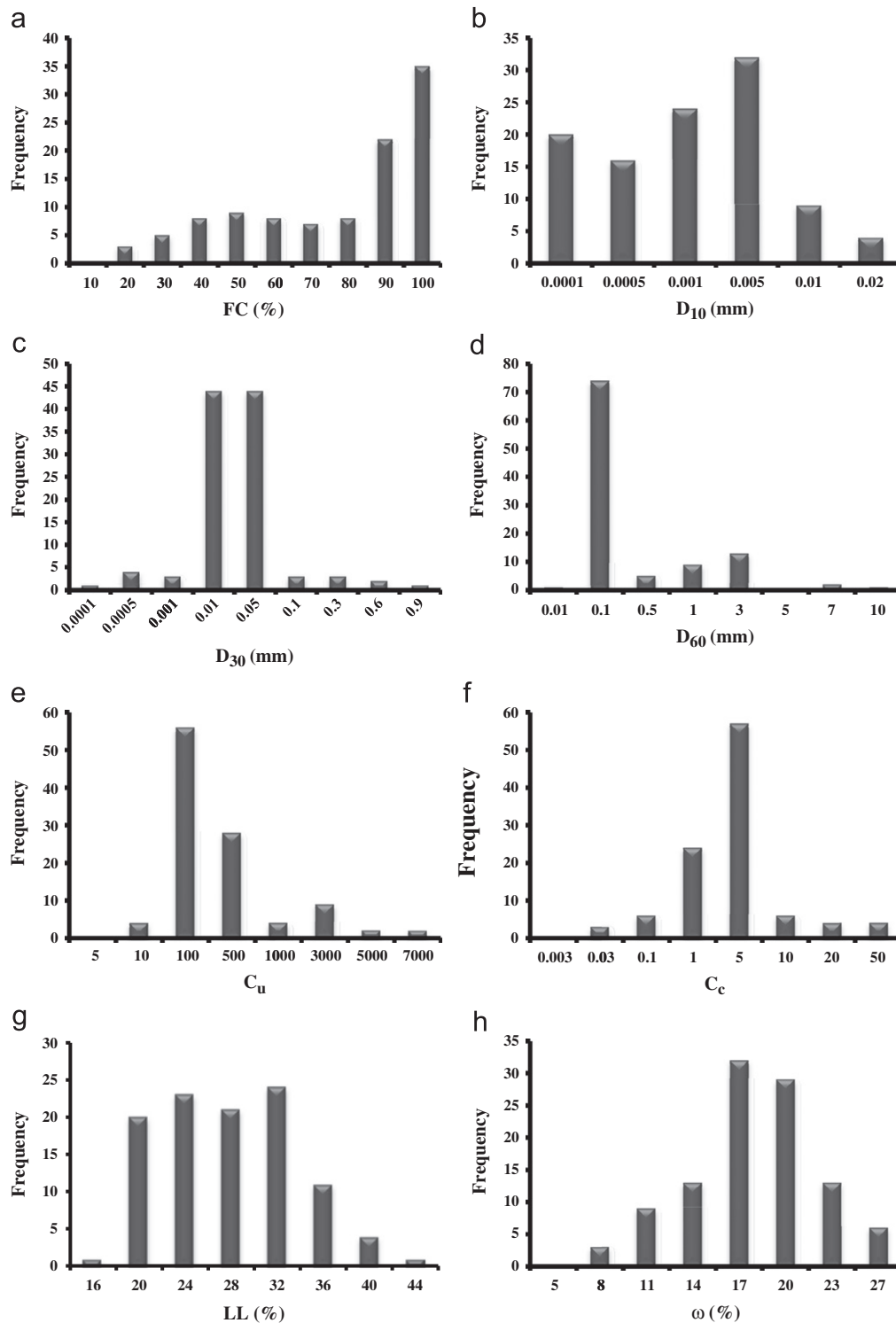


Fig. 7. Histograms of the variables used for the model development.

routines (Deschaine, 2000). To facilitate the use of the derived code, it was converted into a functional representation following a typical procedure shown in Fig. 2. The optimal LGP-based formulation of E_p is as follows:

$$E_p \text{ (kg/cm}^2\text{)} = 28 - 4FC + 4\gamma_d \left(7 + \frac{\frac{288}{\omega \times \gamma_d^3} \left(\frac{12\sqrt{C_u} + 12FC}{\gamma_d} + 7 \right)^2 + 60}{FC} \right) \quad (8)$$

The population size, crossover rate, and maximum number of tournaments for the optimal run were equal to 1500, 95% and 780,000, respectively. This run took 6 min and 33 s on a Pentium 4 personal computer with 3.00 GHz of processor speed and 1 Gb of memory. The above model was selected among a total of 504,138,427 programs evolved and evaluated by the LGP method during the conducted runs. Fig. 8 shows a comparison between the experimental and predicted E_p values.

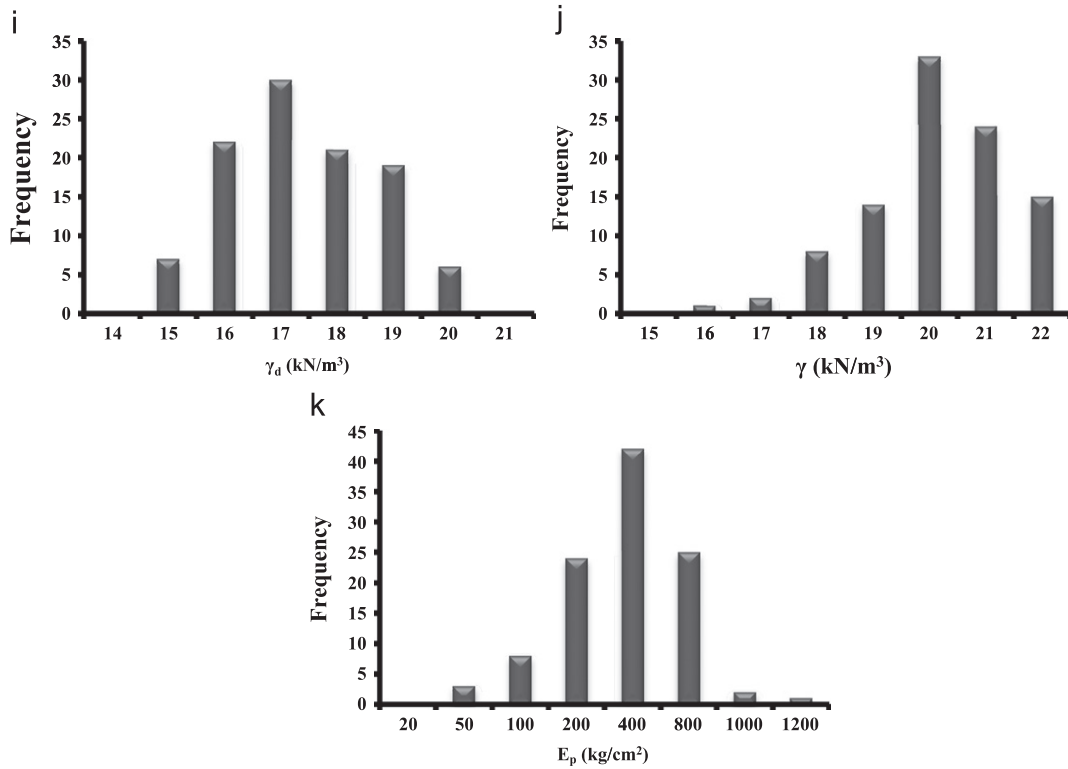


Fig. 7. (continued)

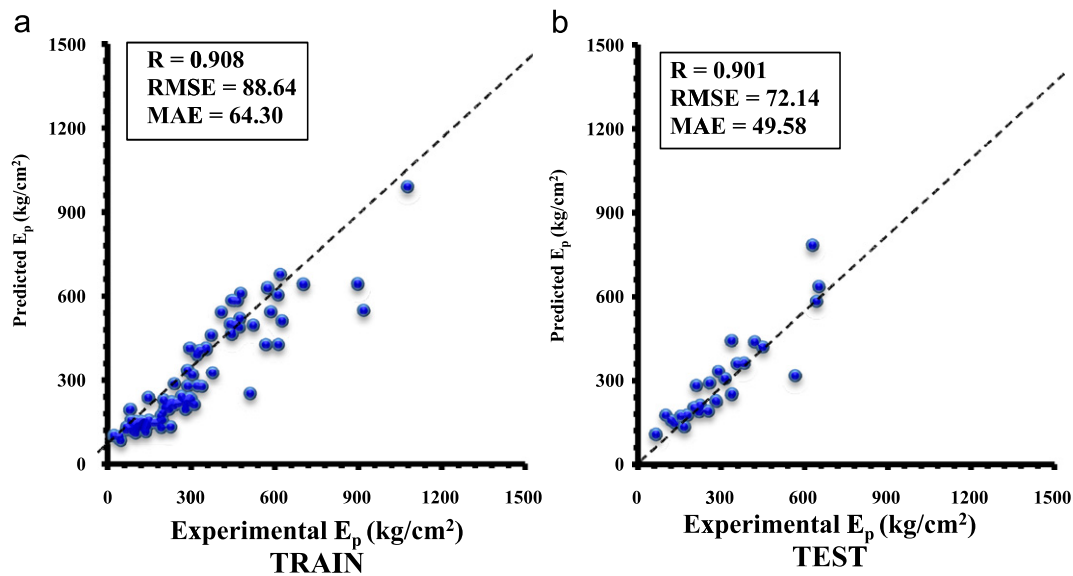


Fig. 8. Experimental versus predicted soil modulus values using the LGP model. (Note: 1.0 kg/cm²=0.1 MPa.)

5. Discussion of model validity

Based on a rational hypothesis, Smith (1986) suggested the following criteria for judging performance of a model:

- if a model gives $|R| > 0.8$, a strong correlation exists between the predicted and measured values.
- if a model gives $0.2 < |R| < 0.8$ a correlation exists between the predicted and measured values.

- if a model gives $|R| < 0.2$, a weak correlation exists between the predicted and measured values.

In all cases, the error values (e.g., RMSE and MAE) should be at the minimum. It can be observed from Fig. 8 that the LGP model with high R 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 LGP 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 objects over the number of selected variables 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 $106/4=26.5$.

No rational model to predict soil secant modulus has been developed yet that would encompass the influencing variables considered in this study. Therefore, it was not possible to conduct a comparative study between the results obtained herein and those provided by any existing models. Instead, 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). Also, the performance indexes of m and n should be lower than 0.1. 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 3. As it is seen, the derived model satisfies the required conditions. The validation phase ensures the derived LGP model is strongly valid and it is not established by chance. Note that the proposed model was developed using the basic soil physical properties (FC, C_u , ω , γ_d) and, therefore, can easily be used for prediction purposes via hand calculations.

Furthermore, the predictions made by the LGP model were compared with those provided by an ANN model developed in this study. The ANN model was established using the training and testing data considered for the LGP-based modeling process. The ANN architecture that gave the best results for the prediction of the pressuremeter soil modulus was found to contain:

- One invariant input layer, with 4 (FC, C_u , ω , and γ_d) arguments;
- One invariant output layer with 1 node providing the value of E_p .
- One hidden layer having 4 ($m=4$) nodes.

The results obtained by the LGP and ANN models are presented in Table 4. It can be observed from this table that the ANN model outperforms the LGP model. GP and its branches (e.g., LGP) directly learn from raw experimental data presented to them in order to extract the subtle functional relationships among the data. This is the same task followed by ANNs and other soft computing techniques. Although LGP does not provide better results than ANN for the investigated problem, it possesses a notable advantage over ANN. LGP 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, ANNs do not give a transparent function

Table 3
Statistical parameters of the LGP model for the external validation.

Item	Formula	Condition	The proposed model
1	R	$0.8 < R$	0.901
2	$k = \frac{\sum_{i=1}^n (h_i \times t_i)}{h_i^2}$	$0.85 < k < 1.15$	0.993
3	$k' = \frac{\sum_{i=1}^n (h_i \times t_i)}{t_i^2}$	$0.85 < k' < 1.15$	0.963
where	$Ro^2 = 1 - \frac{\sum_{i=1}^n (t_i - h_i^0)^2}{\sum_{i=1}^n (t_i - \bar{t})^2}$	$h_i^0 = k \times t_i$, should be close to 1	1.000
	$Ro'^2 = 1 - \frac{\sum_{i=1}^n (h_i - t_i^0)^2}{\sum_{i=1}^n (h_i - \bar{h})^2}$	$t_i^0 = k' \times h_i$, should be close to 1	0.994

Table 4
The prediction performance of the LGP and ANN models.

Item	Training data			Testing data		
	R	RMSE	MAE	R	RMSE	MAE
ANN	0.947	67.80	45.25	0.928	63.48	45.32
LGP	0.908	88.64	64.30	0.901	72.14	49.58

relating the inputs to the corresponding outputs. Contrary to the LGP and ANN methods, most conventional methods (like regression and finite element method) need prior knowledge about the nature of the relationships among the data. Classical constitutive models rely on assuming the structure of the model in advance, which may be suboptimal. On the other hand, the best solutions generated by the GP-based techniques are determined after controlling numerous preliminary models, even millions of linear and nonlinear models. Thus, they can efficiently consider the interactions between the dependent and independent variables (Alavi et al., 2011). It is worth mentioning that the LGP algorithm is parameter sensitive, especially when difficult experimental training data sets like the one used in this paper are employed. The performance of LGP can be improved by using any form of optimally controlling the parameters of the run (Dimopoulos and Zalzal, 2001). In this context, further research can be focused on hybridizing LGP with other optimization algorithms such as GAs, Simulated Annealing, Ant Colony, or Tabu Search.

However, one of the main aims of introducing the GP-based approaches into the design processes is better handling of the information in the pre-design phase (Alavi et al., 2011). It is idealistic to have some initial estimates of the outcome before performing any extensive laboratory or field work. The LGP approach employed in this research is based on the data alone to determine the structure and parameters of the model. Thus, the derived model is considered to be mostly valid for use in preliminary design stages and should cautiously be used for final decision-making. The proposed LGP model is suggested to be used to check the general validity of the laboratory test results. Further, this solution is a good alternative to determine the soil modulus when testing is not possible.

6. Sensitivity analysis

The contributions of the final predictor variables (FC, C_u , ω , γ_d) in the best LGP model were evaluated through a sensitivity analysis. These variables were identified after developing and controlling several models with different combinations of the soil physical properties. To perform the sensitivity analysis, frequency values of the input parameters were obtained. A frequency value equal to 100% for an input indicates that this variable has been appeared in 100% of the best thirty programs evolved by LGP. This is a common approach in the GP-based analyses (Gandomi et al., 2011c). The frequency values of the predictor variables are presented in Fig. 9. In addition to frequencies, this figure presents the average impact of removing all instances of each input from the best thirty programs of the project. A value of 100% represents the largest impact value possible. The greater the value, the more impact removal had. It should be noted that the frequency and average impact values do not necessarily reflect the contributions of the final predictor variables in the best model presented herein (Eq. (8)). It can be observed from Fig. 9 that the contributions of all of the final predictor variables to the prediction of E_p are high. According to this figure, E_p is less sensitive to C_u than other soil properties. As it is seen, the contributions of FC, ω and γ_d to the prediction of E_p are fairly similar. However, the average impact of removing FC is higher than other predictor variables.

7. Parametric analysis

For further verification of the LGP-based prediction model, a parametric analysis was performed in this study. The parametric analysis investigates the response of the predicted soil deformation modulus from the LGP model to a set of predictor variables. The robustness of a design equation is determined by examining how well the predicted target values agree with the underlying physical behavior of the investigated system (Kuo et al., 2009). Fig. 10 presents the tendency of the E_p predictions to the variations of the soil physical properties, i.e., FC, C_u , ω , and γ_d .

The results of the parametric analysis indicate that E_p decreases with increasing FC and ω . It can also be observed that E_p continuously increases with increasing C_u and γ_d . The parametric

analysis results for CC, LL, and γ in the proposed models are generally expected cases from a geotechnical engineering viewpoint (Murthy, 2008). The results confirm that the proposed design equation is capable of capturing the important characteristics of the soil deformation modulus.

8. Conclusion

This paper presents a new approach to derive new prediction equations for the soil deformation moduli utilizing the LGP paradigm. The proposed relationship was developed based on several pressuremeter tests performed in this research. The developed model gives precise estimations of the E_p values. The validation phases confirm the accuracy of the model for its general application to the soil moduli estimation. The developed model is mostly suitable for fine-grained soils with physical properties similar to the soil samples used in this study. Although LGP does not outperform ANN, it provides a simple and straightforward equation that can readily be employed for pre-design purposes or may be used as a quick check on solutions developed by more in-depth deterministic analyses. The optimal model takes into account the role of several important parameters representing the soil moduli behavior. The results indicate that ω and γ_d efficiently represent the initial state and consolidation history of the soil for determining the soil moduli. The predictive capabilities of the derived model are limited to the range of the data used in the training process. To deal with this limitation, the LGP model can be easily retrained and improved to make more accurate predictions for a wider range by including the data for other soil types and test

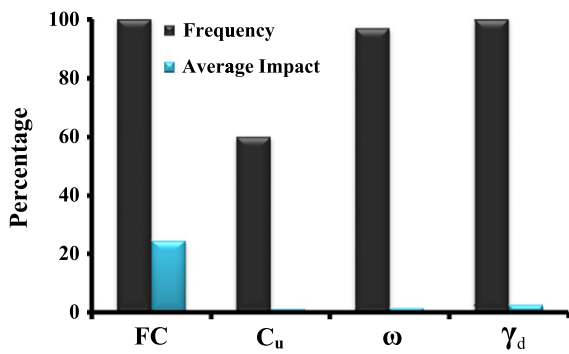


Fig. 9. Contributions of the predictor variables in the LGP analysis.

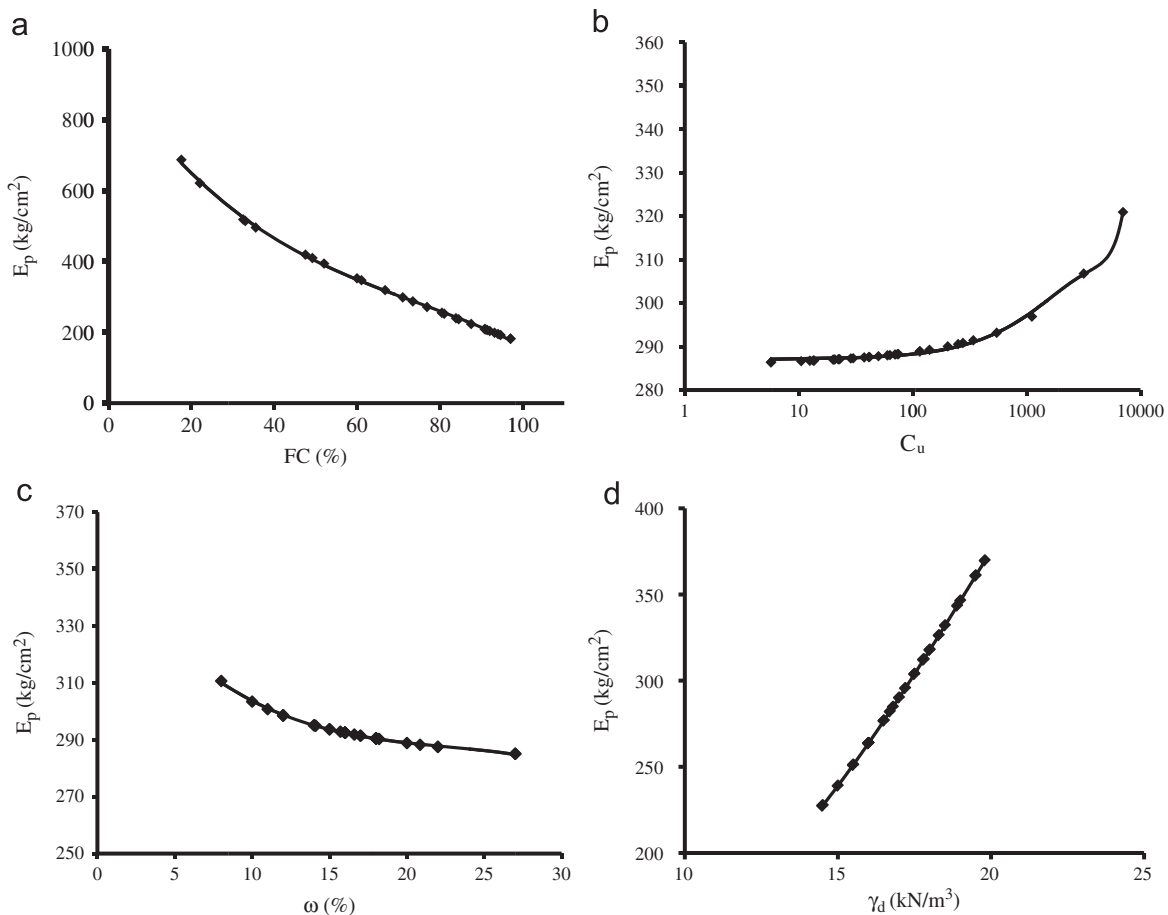


Fig. 10. Parametric analysis of E_p in the LGP model.

conditions. The constitutive model derived using LGP is basically different from the conventional constitutive models based on first principles (e.g., elasticity and plasticity theories). One of the distinctive features of the LGP-based model is that it is based on the experimental data rather than on assumptions made in developing the conventional models. The present study shows that LGP is very practical for nonlinear system modeling. However, the underlying assumption that the input parameters are reliable is not always the case. Fuzzy logic can provide a systematic method to deal with imprecise and incomplete information. Hence, the process of developing hybrid fuzzy-LGP prediction models can be a suitable topic for further studies.

Appendix A

The optimum LGP program for the prediction of E_p .

The following LGP programs can be run in the Discipulus interactive evaluator mode or can be compiled in C++ environment. (Note: $v[0]$, ..., $v[3]$ represent FC, C_u , ω , and γ_d , respectively. $f[0]$ holds the output.)

```
float DiscipulusCFunction(float v[0], v[1], v[2], v[3])
{long double f[8];
long double tmp=0;
int cflag=0;
f[0]=f[1]=f[2]=f[3]=f[4]=f[5]=f[6]=f[7]=0;

L0:  f0+=v[1];
L1:  f0=sqrt(f0);
L2:  f0+=v[0];
L3:  f0*=6;
L4:  f0+=f0;
L5:  f0/=v3;
L6:  f0--=-7;
L7:  f0/=v[3];
L8:  f0*=f0;
L9:  f0+=f0;
L10: f[0]=v[2];
L11: f[0]*=6;
L12: f[0]+=f[0];
L13: f[0]=v[3];
L14: f[0]-=2;
L15: f0--=-7;
L16: f0/=v[0];
L17: f0*=6;
L18: f0+=f0;
L19: f0--=-7;
L20: f0*=v3;
L21: f0--=-7;
L22: f0-=v[0];
L23: f0+=f0;
L24: f0+=f0;

L25:
return f[0];}
```

References

- Alavi, A.H., Gandomi, A.H., 2011. A robust data mining approach for formulation of geotechnical engineering systems. *Int. J. Comput. Aided Methods 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. *Construct. Build. Mater.* 25 (3), 1338–1355.
- ASTM D4719-87, 1987. Standard Test Method for Pressuremeter Testing in Soils. STM International.
- 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.
- Baykasoglu, A., Gullub, H., Canakci, H., Ozbakir, L., 2008. Prediction of compressive and tensile strength of limestone via genetic programming. *Expert Syst. Appl.* 35 (1–2), 111–123.
- Brameier, M., Banzhaf, W., 2001. A comparison of linear genetic programming and neural networks in medical data mining. *IEEE Trans. Evolut. Comput.* 5 (1), 17–26.
- Brameier, M., Banzhaf, W., 2007. *Linear Genetic Programming*. Springer Science+Business Media, New York.
- Briaud, J.L., 2001. Introduction to Soil Moduli, Geotech. News. BiTech Publishers, Richmond, B. C., Canada.
- Briaud, J.L., Li, Y., Rhee, K., 2006. BCD: a soil modulus device for compaction control. *J. Geotech. Geoenviron. Eng. (ASCE)* 132 (1), 108–115.
- Chau, K.W., 2006. Particle swarm optimization training algorithm for ANNs in stage prediction of Shing Mun River. *J. Hydrol.* 329 (3–4), 363–367.
- Cheng, C.T., Ou, C.P., Chau, K.W., 2002. Combining a fuzzy optimal model with a genetic algorithm to solve multiobjective rainfall-runoff model calibration. *J. Hydrol.* 268 (1–4), 72–86.
- Ciftci, O.N., Fadiloglu, S., Gogus, F., Guven, A., 2009. Genetic programming approach to predict a model acidolysis system. *Eng. Appl. Artif. Intell.* 22 (4–5), 759–766.
- Conrads, M., Dolezal, O., Francone, F.D., Nordin, P., 2004. Discipulus—Fast Genetic Programming based on AIM Learning Technology. Register Machine Learning Technologies Inc., Littleton, CO.
- Deschaine, L.M. 2000. Using Genetic Programming to Develop a C/C++ Simulation Model of a Waste Incinerator Science Applications. International Corp, Draft Technical Report.
- Dimopoulos, C., Zalzal, A.M.S., 2001. Investigating the use of genetic programming for a classic one-machine scheduling problem. *Adv. Eng. Software* 32 (6), 489–498.
- Francone, F.D., 2001. Discipulus Pro™ Software Owner's Manual. Register Machine Learning Technologies Inc., Littleton, CO.
- 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.* 161, 99–120.
- Frank, I.E., Todeschini, R., 1994. *The Data Analysis Handbook*. Elsevier, Amsterdam.
- Galvan-Lopez, E., Dignum, S., Poli, R., 2008. The effects of constant neutrality on performance and problem hardness in GP. In: *Proceedings of the Eleventh European Conference on Genetic Programming, EuroGP 2008*. Napoli, Italy, pp. 312–324.
- Gandomi, A.H., Alavi, A.H., Mousavi, M., Tabatabaei, S.M., 2011c. A hybrid computational approach to derive new ground-motion attenuation models. *Eng. Appl. Artif. Intell.* 24 (4), 717–732.
- Gandomi, A.H., Alavi, A.H., Mirzahosseini, R., Moghadas Nejad, F., 2011a. Nonlinear genetic-based models for prediction of flow number of asphalt mixtures. *J. Mater. Civil Eng. ASCE* 23 (3), 1–18.
- 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.* 43 (7), 963–983.
- Gandomi, A.H., Alavi, A.H., 2011. Multi-stage genetic programming: a new strategy to nonlinear system modeling. *Inf. Sci.* 181 (23), 5227–5239.
- Gandomi, A.H., Alavi, A.H., Yun, G.J., 2011b. Nonlinear modeling of shear strength of srfcb beams using linear genetic programming. *Struct. Eng. Mech.* 38 (1), 1–25.
- Golbraikh, A., Tropsha, A., 2002. Beware of q^2 !. *J. Mol. Graphics Model.* 20 (4), 269–276.
- 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 (12–13), 985–991.
- Javadi, A.A., Rezaia, M., 2009. Applications of artificial intelligence and data mining techniques in soil modeling. *Geomech. Eng.* 1 (1), 53–74.
- Javadi, A.A., Rezaia, M., Mousavi Nezhad, M., 2006. Evaluation of liquefaction induced lateral displacements using genetic programming. *Comput. Geotech.* 33 (4–5), 222–233.
- Jia, W., Ling, B., Chau, K.W., Heutte, L., 2008. Palmprint identification using restricted fusion. *Appl. Math. Comput.* 205 (2), 927–934.
- Karmakar, S., Sharma, J., Kushwaha, R.L., 2004. Critical state elasto-plastic constitutive models for soil failure in tillage—a review. *Can. Biosyst. Eng.* 46, 19–23.
- Koza, J., 1992. *Genetic Programming, on the Programming of Computers by means of Natural Selection*. MIT Press, Cambridge (MA).
- Kuo, Y.L., Jaks, 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.
- Lin, J.Y., Cheng, C.T., Chau, K.W., 2006. Using support vector machines for long-term discharge prediction. *Hydrol. Sci. J.* 51 (4), 599–612.
- Mollahasani, A., Alavi, A.H., Gandomi, A.H., 2011. Empirical modeling of plate load test moduli of soil via gene expression programming. *Comput. Geotech.* 38 (2), 281–286.
- Murthy, S., 2008. *Geotechnical Engineering: Principles and Practices of Soil Mechanics*, 2nd edition CRC Press, Taylor & Francis, UK.
- Muttill, N., Chau, K.W., 2007. Machine learning paradigms for selecting ecologically significant input variables. *Eng. Appl. Artif. Intell.* 20 (6), 735–744.

- Oltean, M., Grosan, C., 2003. A comparison of several linear genetic programming techniques. *Complex Syst.* 14 (4), 1–29.
- Pal, M., Singh, N.K., Tiwaria, N.K., 2011. Support vector regression based modeling of pier scour using field data. *Eng. Appl. Artif. Intell.* 24 (5), 911–916.
- 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.
- 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.
- Rezania, M., Javadi, A.A., 2007. A new genetic programming model for predicting settlement of shallow foundations. *Can. Geotech. J.* 44 (12), 1462–1473.
- Reznik, Y.M., 1995. Comparison of results of oedometer and plate load tests performed on collapsible soils. *Eng. Geol.* 39, 17–30.
- Reznik, Y.M., 2007. Influence of physical properties on deformation characteristics of collapsible soils. *Eng. Geol.* 92, 27–37.
- Smith, G.N., 1986. *Probability and Statistics in Civil Engineering*. Collins, London.
- Xie, J.X., Cheng, C.T., Chau, K.W., Pei, Y.Z., 2006. A hybrid adaptive time-delay neural network model for multi-step-ahead prediction of sunspot activity. *Int. J. Environ. Pollut.* 28 (3–4), 364–381.