Artificial Intelligence Review (2021) 54:1863–1885
https://doi.org/10.1007/s10462-020-09894-7

Genetic programming in civil engineering: advent, applications and future trends

Qianyun Zhang[1] · Kaveh Barri[1] · Pengcheng Jiao[2,3] · Hadi Salehi[4] · Amir H. Alavi[1,5]

Published online: 29 September 2020

Abstract

Over the past two decades, machine learning has been gaining significant attention for solving complex engineering problems. Genetic programing (GP) is an advanced framework that can be used for a variety of machine learning tasks. GP searches a program space instead of a data space without a need to pre-defined models. This method generates transparent solutions that can be easily deployed for practical civil engineering applications. GP is establishing itself as a robust intelligent technique to solve complicated civil engineering problems. This paper provides a review of the GP technique and its applications in the civil engineering arena over the last decade. We discuss the features of GP and its variants followed by their potential for solving various civil engineering problems. We finally envision the potential research avenues and emerging trends for the application of GP in civil engineering.

Keywords Civil engineering · Prediction · Classification · Genetic programming · Machine learning · Deep learning

✉ Amir H. Alavi
alavi@pitt.edu
http://www.engineering.pitt.edu/AmirhosseinAlavi/

Pengcheng Jiao
pjiao@zju.edu.cn

[1] Department of Civil and Environmental Engineering, University of Pittsburgh, Pittsburgh, PA, USA

[2] Institute of Port, Coastal and Offshore Engineering, Ocean College, Zhejiang University, Zhoushan 316021, Zhejiang, China

[3] Engineering Research Center of Oceanic Sensing Technology and Equipment, Zhejiang University, Ministry of Education, Zhoushan 316021, China

[4] Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, USA

[5] Department of Computer Science and Information Engineering, Asia University, Taichung, Taiwan

1 Introduction

Artificial intelligence (AI) emulates the complex biological processes such as learning, reasoning and self-correction to explore solutions for engineering problems. Unlike traditional statistical methods, AI can provide solutions without prior knowledge of the nature of the relationship between the dependent and independent variables. Therefore, the AI techniques can provide alternative solutions to determine engineering design parameters. This can be particularly important when it is not possible to carry out laboratory or field testing. In general, AI can be classified into several sub-categories such as reasoning, programming, artificial life, belief revision, data mining, distributed AI, expert systems, evolutionary computation, systems, knowledge representation, machine learning, natural language understanding, neural networks, theorem proving, constraint satisfaction, and theory of computation (Kushchu 2002; Zhang and Rockett 2007). Figure 1 shows an illustration of the interrelation between AI and major data science techniques including machine learning (ML) and data mining (DM). These techniques have been extensively applied to the development of reliable empirical, analytical and numerical models for complex engineering problems (Salehi and Burgueño 2018; Araujo 2007; Peker 2016; Wang et al. 2019). ML deals with designing computer models that can learn from a set of data and identify patterns. Biological learning is the main source of inspiration for many of the ML methods (Salehi and Burgueño 2018). The main components of ML are presented in Fig. 2. Some of the widely-used ML methods are artificial neural network (ANN), fuzzy inference system (FIS), adaptive neuro-fuzzy systems (ANFIS), and support vector machines (SVM).

The reliability of the AI algorithms has been often criticized because they do not operate similar to the classical methods based on the laws of physics. In fact, the problem is the AI methods generate complex models that are not easy to understand for users (Ghiringhelli et al. 2015). AI-based characterization and design of new materials would be doubtful if the connection between features and prediction is not well-defined. Therefore, two key issues in interpreting the AI-based models are the transparency of the models and the knowledge extraction process. Figure 3 shows a pictorial representation of the classification

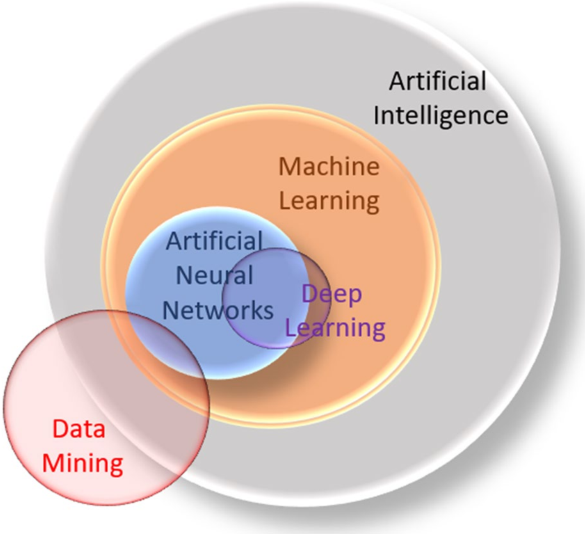


Fig. 1 Relationship between AI, ML and other data mining techniques

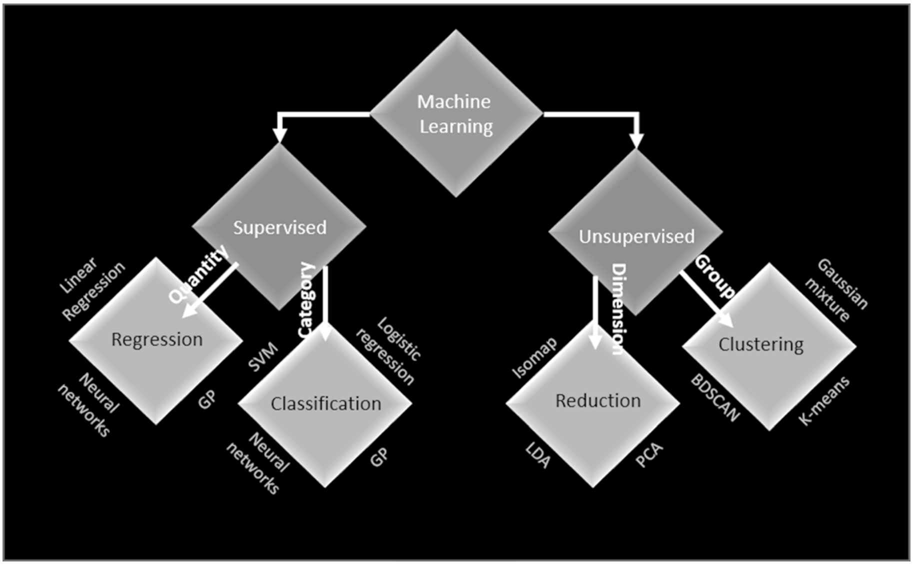


Fig. 2 ML and its main components

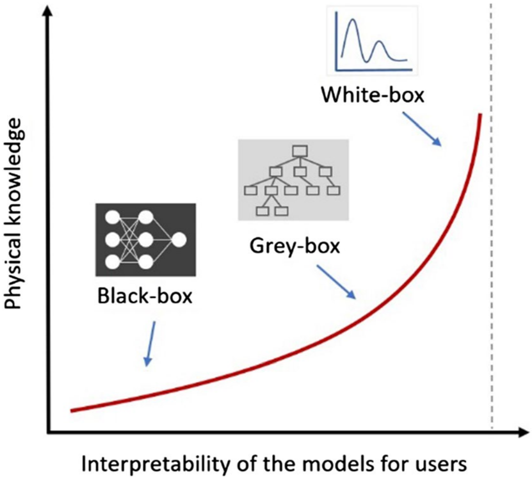


Fig. 3 Classification of the modelling techniques (Adapted from Giustolisi et al. 2007)

of modelling techniques (Giustolisi et al. 2007). Referring to this figure, white-, black- and grey-box models are three main categories for mathematical modelling. If the variables and parameters are known and the model is based on first principles (e.g. laws of physics), then it is possible to explain the underlying physical relationships of the system. Such models are classified as white-box models. Black-box methods explore the relationships between the input and output data without providing a feasible structure of the model. Conceptual methods that do not only identify the existing patterns between the data but also provide a mathematical structure of the model belong to the grey-box category. ANN is perhaps the most widely-used ML method in the engineering arena (Hamedi 2005; Yalpir 2016; Bui et al. 2018). It is a computing system inspired by biological neural networks. Despite its acceptable accuracy, ANN belongs to the class of black-box modelling techniques. That is, the ANN solution representation is in the form of weights and bias that do not provide details about the nature of the derived relationships. Genetic programming (GP) is a fairly new ML method developed based on Darwin's evolution theory (Koza 1992; Brameier and Banzhaf 2007). In contrast to the black-box methods, GP is classified as a grey-box technique with the ability to develop explicit prediction functions (Giustolisi et al. 2007). GP creates predictive models through a complicated evolutionary process. The model transparency is a notable advantage of GP over nearly all other ML methods, especially the black-box ANN and DL methods. Mathematical structures derived by GP can readily be used to gain important information about the system performance.

In order to improve the performance of standard GP in various aspects, several GP branches have been developed such as linear-based GP, cartesian GP, grammatical GP, stack GP, etc. Gene expression programming (GEP), linear genetic programming (LGP), multi-expression programming (MEP), and multigene genetic programming (MGGP) are the most widely-used GP variants in engineering domain (Alavi and Gandomi 2011; Li et al. 2019; Hadianfard and Jafari 2016; Mansouri and Farzampour 2018; Arruda et al. 2014; Aval et al. 2017; Jafari and Mahini 2017). Although GP and its variants are not as popular as ANNs, a number of researchers discovered their significant potential in solving civil engineering problems. This paper intends to provide a comprehensive survey of applications of GP and its variants in civil engineering. Future avenues for applications of GP and remaining challenges are further discussed. To perform the literature review of genetic programming applications in civil engineering, a content analysis approach developed by Krippendorff (Krippendorff (2018)) is used. Samples are collected through a sophisticated search within major academic databases such as Scopus, Web of Science, Google Scholar, Wiley Online Library, Science Direct, Sage, ASCE Library, IEEE Explore, ACM, and Emerald. Various keywords, e.g., "genetic programming", "civil engineering", "LGP", "GEP", "MGGP", "evolutionary algorithm", "concrete", "structural", "geotechnical", etc., are used to ensure all the related studies are included in the survey. The time period under review is from 2010–2019, leading to study approximately 327 candidate articles. The articles are filtered using a two-round article selection technique: 1) checking titles, abstract, and keywords of the articles, and 2) filtering out the papers closely related to the objective of this review paper. Finally, 31 papers are selected for the survey. This review paper is organized as follows. Section 2 presents the introduction of GP and its variants. The most relevant works addressing the applications of GP for solving civil engineering problems are discussed in Sect. 3. Section 4 reviews the development and improvement of GP and highlights the trends for future applications of GP. Finally, conclusion is provided in Sect. 5.

2 Genetic programming and its variants

2.1 Tree-based genetic programming

GP is a symbolic regression technique for evolving computer programs following the natural selection principles. GP was first introduced by Koza (1992). GP generated candidate solutions are in abstract syntax trees representation expressed by a functional programming language. That is why the classical GP is called tree-based GP (TGP) or canonical GP. The TGP algorithm starts with creating a random initial population and applying genetic

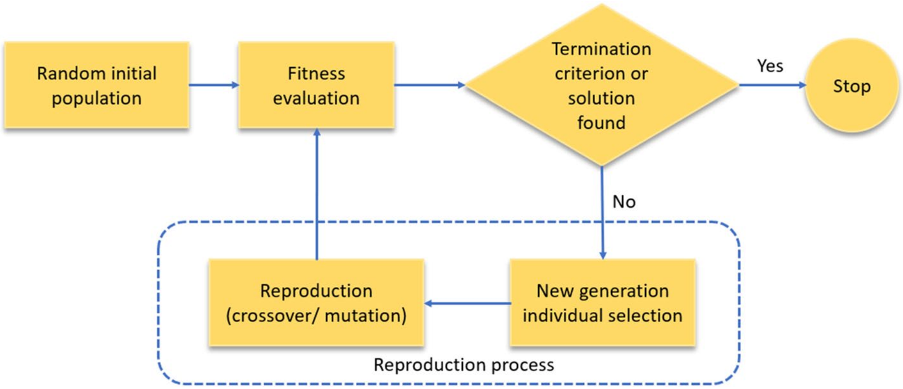
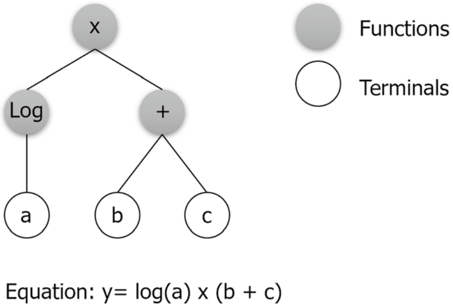

Random initial population
Fitness evaluation
Termination criterion or solution found
Yes
Stop
No
Reproduction (crossover/ mutation)
New generation individual selection
Reproduction process

Fig. 4 Flowchart of the GP approach

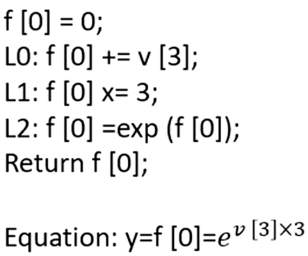
Fig. 5 Example of a tree GP program



operations (e.g. crossover, mutate and reproduce, etc.) until an optimal or at least a suitable model is found. A simplified flow chart of the GP process is illustrated in Fig. 4. Population members in TGP consist of functions and terminals (see Fig. 5). Functions can be basic arithmetic operations, Boolean logical and any other mathematical functions. Terminals contain arguments for functions, which can be variables, numerical and logical contents.

2.2 Linear genetic programming

LGP is a linear variant of classical GP developed by Brameier and Banzhaf (2007). It uses a specific linear representation of individuals or computer programs. In LGP, classical tree-like expressions are substituted by programs of an imperative language (like C or C++) (Gandomi et al. 2014; Guven 2009). Compared to the rigid data flow determined by the tree structure programs in classical GP, LGP uses register content multiple times to generate a data flow graph. A sequence of instructions in C represent an LGP program. The instruction sets consists of arithmetic operations, logical functions and other mathematical functions. Terminal sets include variables and constants (Gandomi et al. 2014). As a result, LGP can run faster than tree-based GP and is easier to interpret. An example of LGP is shown in Fig. 6. As illustrated in Fig. 7, the evolution process in LGP is similar to TGP. Further details about LGP can be found in Brameier and Banzhaf (2007).



```
f [0] = 0;
L0: f [0] += v [3];
L1: f [0] x= 3;
L2: f [0] =exp (f [0]);
Return f [0];

Equation: y=f [0]=e^{v [3]×3}
```

Fig. 6 Example of LGP

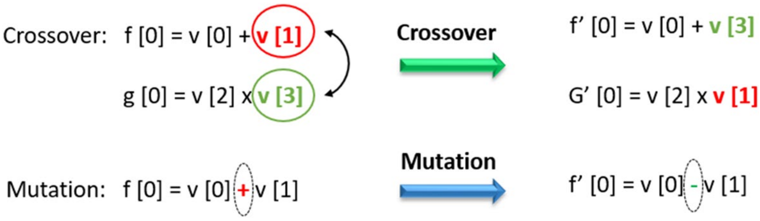


Fig. 7 Typical operations in LGP

2.3 Gene expression programming

GEP is a natural extension of GP introduced by Ferreira (2001) and Özcan (2012). A schematic representation of the GEP algorithm is presented in Fig. 8. The main difference between GEP and classical GP is that GEP generated candidate solutions are represented by fixed-length string characters, which are later expressed as parse-trees with different sizes and shapes when fitness is evaluated. Chromosomes and expression tress (ETs) are two major parameters of GEP. The translation process decodes information from chromosomes to ETs based on a set of rules. The genetic code and rules are very simple and presented in two languages (language of gene and language of ETs). A typical GEP expression tree is shown in Fig. 9. Further details about GEP can be found in Ferreira (2001).

2.4 Multi expression programming

MEP is another variant of GP introduced by Oltean (2004). MEP is capable of storing multiple solutions of a problem in a single chromosome. This advantage enables MEP to explore larger zones of the search space without running-time penalty. The best solution is determined based on the fitness agreement. The representation of MEP is not specified, and it depends on the problem type. In a simplest situation, the MEP chromosomes are linear strings of instructions (Oltean 2004; Oltean and Groşan 2003; Oltean and Grosan 2004). Figure 10 shows a typical representation of the MEP solutions. Further details about this method can be found in Oltean (2004).

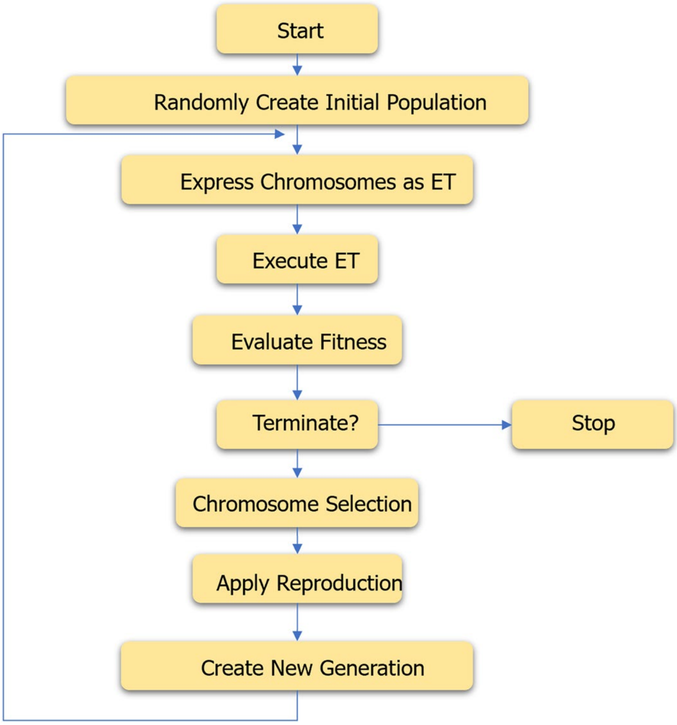


Fig. 8 Flowchart of the GEP algorithm

Fig. 9 Example of a GEP program

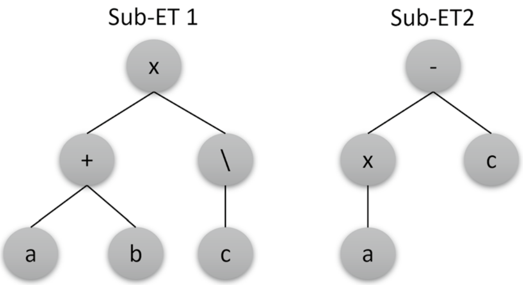
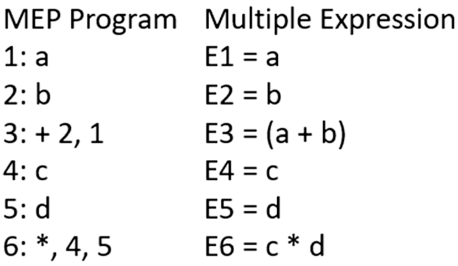
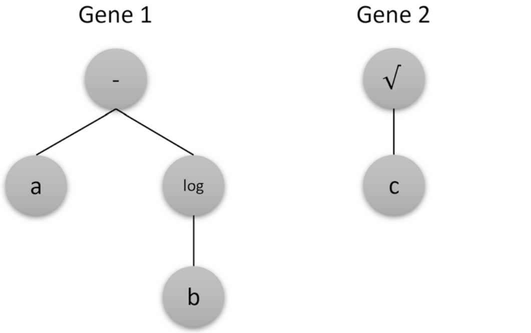


Fig. 10 A simple MEP program



MEP Program	Multiple Expression
1: a	E1 = a
2: b	E2 = b
3: + 2, 1	E3 = (a + b)
4: c	E4 = c
5: d	E5 = d
6: *, 4, 5	E6 = c * d



Equation: $y = \alpha_0 + \alpha_1(a - \log(b)) + \alpha_2\sqrt{c}$
(α_0, α_1, α_2 is gene weights for each gene)

Fig. 11 Example of an MGGP program

2.5 Multigene genetic programming

MGGP is a fairly new variant of GP developed by Searson et al. (2011). In MGGP, a single GP individual is derived from a number of genes, where each GP is a tree expression (Searson et al. 2011; Gandomi and Alavi 2012). MGGP combines multiple GP programs via a weighted linear approach. An example of MGGP program is illustrated in Fig. 11. Further details about MGGP can be found in Searson et al. (2011).

3 Review of application of GP and its variants in civil engineering

3.1 Tree-based GP applications

Various studies have shown the potential of TGP in solving civil engineering related problems. According to the literature, GP has been used to develop structural behavior property prediction models (Ferreira 2001; Özcan 2012; Oltean 2004; Oltean and Groşan 2003; Oltean and Grosan 2004; Searson et al. 2011), local scour of bridge vertical abutment prediction model (Begum 2013), and fundamental period of vibration (Joshi et al. (2014). Tsai and Lin (2011) proposed a new GP approach by introducing weight coefficients to each GP linkage in a tree called weighted genetic programming (WGP). The WGP method evolves models with a fully weighted connection. This method has been applied to predict high-strength concrete parameters such as compressive strength, cost and slump. Gandomi et al. (2013) proposed an empirical prediction model for shear strength of reinforced concrete (RC) deep beams using a hybrid algorithm combining GP and simulated annealing called genetic simulated annealing (GSA). The authors used 214 RC deep beams test results collected from literature to develop and evaluate the GSA model. The proposed model was compared with design codes and a parameter sensitivity analysis was conducted to further validate the proposed model. It revealed that the model had a higher accuracy than existing models. Jalal et al. (2013) utilized GP and adaptive neuro-fuzzy inference system (ANFIS) to develop models for strength enhancement of concrete cylinders retrofitted by carbon-fiber-reinforced polymer (CFRP) composites. The developed models were compared with ANNs, multiple regression and empirical models. It was shown that the ANFIS model was the most accurate and the accuracy of GP was higher than regression, ANN and other empirical models. Castelli et al. (2013) proposed a new intelligent system based on general GP and geometric semantic genetic operators, called Geometric Semantic Genetic Programming. The proposed method was used to develop prediction models for predicting the strength of high-performance concrete, where the results were compared to general GP and other machine learning methods (e.g. SVM, ANN). It was demonstrated that the proposed method is capable of providing a significantly better result. More, Gandomi et al. (2016) proposed an algorithm for complex civil engineering problems called multi-objective genetic programming (MOGP). MOGP could find model structure selection like standard GP, parameter estimation like classical regression and optimization for complexity and goodness-of-fit in a system. The proposed algorithm was validated by creating a time-dependent concrete creep model based on a big data set. Data selected from the NU-ITI data bank was used to develop the creep model, and the obtained model was compared to some existing creep models, which has a much higher accuracy. Accordingly, it was shown that the proposed MOGP algorithm is efficient to solve complex civil engineering problems.

Further, Moradi et al. (2019) proposed a model to estimate tensile stress–strain diagrams of steel fiber reinforced concrete (SFRC) using GP. To develop the model, 41 experimental data were collected from literature. The proposed model was compared with other models using 11 specimens collected from literatures. Results of the study revealed that the proposed model had better accuracy than other models. Begum (2013) also proposed alternative equations for prediction of local scour at vertical bridge abutment with GP. The developed model was compared with empirical models from literature, where it was shown that the GP model had a higher accuracy. In addition, Joshi et al. (2014) applied genetic programming to develop models for fundamental period of vibration using 206 analyzed buildings. Models in four categories were generated with various inputs numbers, where the obtained equations for four categories were compared with equations recommended by other researchers. The developed models had reasonable and acceptable accuracy, but experimental investigation was needed as suggested by the author. However, reviewing the existing limited applications of TGP in civil engineering shows its acceptable performance. Table 1 presents the summary of the best performances of each study utilizing GP, from which it can be seen that the correlation coefficient (R) between model prediction and actual experimental data is above 0.9, further denoting the capability and potentials of TGP in civil engineering domain.

3.2 Linear genetic programming applications

This sub-section summarizes studies related to the application of LGP for solving civil engineering problems. Gandomi et al. (2010) proposed a new formulation of elastic modulus for normal-strength and high-strength concrete by applying LGP approach. The models were developed with experimental data collected from literature. The results revealed that LGP models were more accurate that the building codes and some other models from literature. More, Gandomi et al. (2014) proposed a prediction model for shear strength of reinforced concrete beams without stirrups by using LGP. Accordingly, 1938 experimental test results were used to develop the model. The proposed model was compared with existing design codes, where the accuracy was shown to be higher than the design codes. Mollahasani et al. (2011) developed a new prediction model for secant soil deformation modulus using LGP, where the model was developed based on set of plate load

Table 1 Summary of the TGP model performance indexes in the reviewed studies

Literature	Tsai and Lin (2011)	Gandomi et al. (2013)	Jalal et al (2013)	Gandomi et al (2016)	Moradi et al (2019)	Begum (2013)	Joshi et al. (2014)
R	0.97 (TR*) 0.98 (TE**)	0.93 (Tr) 0.94 (Te)	0.99 (O***)	0.91(O)	0.95 (O)	0.96 (Tr) 0.95 (Te)	0.99 (O)

*Training data

**Testing data

***Overall

tests conducted on different types of soil. The best model was selected among the several models developed with various input parameters, and a sensitivity analysis was conducted. It was shown that the proposed model provided precise prediction. Rashed et al. (2012) developed a prediction model for pressure meter soil deformation modulus utilizing LGP. Babanajad et al. (2013) developed two prediction models for predicting the strength of concrete under multiaxial compression. 370 test data were used to develop training and test models. The first model was developed for prediction of tensile-compression strength ratio, for which uniaxial compressive strength and confining pressure were selected as the effective input parameters. The second model was proposed to predict triaxial strength, for which uniaxial compressive strength and confining pressure were selected as the effective input parameters. Results confirmed that both developed LGP models had a high accuracy performance. Besides, Rostami et al. (2018) proposed a new predictive model using LGP for predicting confined compressive strength and strain at confined peak stress of circular-reinforced concrete columns. Comparing results of the proposed model with models recommended by other researchers revealed that the proposed model had a significantly better performance. Sadrossadat et al. (2017) proposed a new equation for compressive strength of high strength concrete (HSC) specimens 28 days in age by using LGP. The LGP model shows a good degree of accuracy and can be used for manual calculation. Table 2 list the best performance of each method in the reviewed studies.

3.3 Gene expression programming applications

Gene expression is a powerful linear-based GP method that has been widely used to solve civil engineering problems (Sadrossadat et al. 2020). Cevik et al. (2010) utilized GEP to develop prediction model for torsional strength of reinforced concrete. Accordingly, 76 sets of rectangular RC beams experimental data obtained from existing databased were used to develop the model. The proposed model was compared with building codes, where it was concluded that the GEP model had a high accuracy than the building codes. Ozcan (2012) developed two GEP models to predict splitting tensile strength of cylinder specimens with 150 mm diameter and 300 mm height. 190 experimental results were used to develop the models. The first model GEP-I was a function of concrete compressive strength, while the second model GEP-II was a function of concrete compressive strength as well as water cement ratio. The proposed models were compared with some existing models, where it was shown that the proposed GEP-based models can be used as promising tools for splitting tensile strength prediction. Further, Mousavi et al. (2012) GEP to develop a prediction model for predicting the compressive strength of high performance concrete (HPC) mixes. Linear and nonlinear least square regression models were developed and compared with the GEP model. For this purpose, 133 data collected from literature were used. Results of GEP, LLSR and NLSR models were compared, and it was revealed that the GEP model had a higher accuracy than two other regression models. Ozbek et al. (2013) developed five different GEP models to estimate the uniaxial compressive strength (UCS) of the five types of rock with different characteristics. The agreement between experimental data and predicted results was achieved. In addition, Gandomi et al. (2014) proposed a new design equation for shear strength of slender reinforced concrete beams without stirrups, where 1942 experiment test results obtained from published database were used to develop the model. During the development of GEP model, several different sets of training and test combinations were considered to obtain the best GEP model with good simplicity and high accuracy.

Table 2 Summary of the LGP performance indexes in the reviewed studies

Literature	Gandomi et al. (2010)	Gandomi et al 2014)	Mollahasani et al. (2011)	Rashed et al. (2012)	Babanajad et al. (2013)	Rostami et al. (2018)	Sadrossadat et al. (2017)

R	0.95 (O)	0.92 (O)	0.97 (Tr)
			0.96 (Te)
				0.91 (Tr)
				0.90 (Te)
					0.99 (Tr)
					0.99 (Te)
						0.97 (Tr)
						0.96 (Te)
							0.88(Tr)
							0.88(Te)

The obtained model was compared with some existing models, where it was shown that the GEP model had a higher accuracy compared to existing models. Also, a sensitivity and parametric analysis were conducted to show the effects of each parameter. Gandomi et al. (2017) also used the similar approach noted above to design equations for shear strength of slender reinforced concrete beams with stirrups, and results confirmed the good performance of the model with high accuracy. Danish (2012) used gene expression programming technique to develop a scour depth prediction model at bridge abutments in cohesive sediments. The model was developed based on laboratory data collected from literature. The results showed that GEP model provided a better prediction than nonlinear regression model, which revealed that GEP had a big potential in scour depth prediction. Hadianfard and Jafari (2016) conducted a study to estimate the compressive strength of lightweight aggregate concrete by generating suitable mathematical functions. Three type of lightweight concrete mixtures were assessed by ultrasonic pulse velocity test. The results were used to generate compressive strength prediction functions by exponential function fitting and GEP. Three prediction functions for single type of lightweight concrete and one correlation function were generated, where it was revealed that GEP functions had high accuracy for all three types, but exponential function only had high accuracy for one type material. Further, Gholampour et al. (2017) proposed a new empirical model using GEP for recycled aggregate concrete (RAC) mechanical properties prediction. 650 compressive strength, 421 elastic modulus and 346 splitting tensile strength and 152 flexural strength testing results from literature were used to develop the GEP models (i.e., compressive strength, elastic modulus, flexural strength and splitting tensile strength). The developed models were compared with some existing models. Results confirmed that proposed GEP models had higher accuracy than the existing models.

Aval et al. (2017) conducted a study to develop more accurate prediction model for shear strength of SR-RCCs due to the limitations of existing simple design models. Thirty sets of lab testing data were collected, where one third of the collected data were used to calibrate a FE model to be accurate to represent the behaviors of SR-RCCs. The calibrated FE model was used to generate 53 simulations, and results were used to develop the prediction models utilizing GEP and nonlinear regression (NR) analysis. The rest of lab testing data were used to evaluate the accuracy of obtained GEP model and NR model through comparison with ACI and EC2 design codes. It was revealed that GEP had much higher accuracy for shear strength prediction of SR-RCCs than NR model, ACI code and EC2 code. Similarly, due to the limitation of ACI proposed compressive strength and density standards for lightweight concrete, Jafari and Mahini (2017) utilized GEP to generate prediction functions for compressive strength of a specific mixture. A hundred of specimens made with three types of concrete mixes were tested in the lab. Test results were compared to other models. Three prediction functions with parameters (the amount of water, sand, cement and the lightweight aggregate) for compressive strength were developed for three types of concrete mixtures. Results showed that the prediction accuracy was high. This study proved that GEP could be used to generate prediction functions for compressive strength when there is not a specific standard for lightweight concrete aggregate selection and mix proportions. In addition, Mansouri and Farzampour (2018) proposed a method to predict the buckling load of imperfect cylindrical shells. Because the classical buckling theory is insufficient for some specific applications, researches related to thin-shell structures are needed. The proposed method was applied to an existing experimental data set. The performance of the proposed method was excellent on training and test data set. This study showed that GEP is an efficient and accurate approach to estimate the buckling load of imperfect cylindrical shells. Sadrossadat and Basarir (2019) developed two prediction models to predict confined compressive strength and strain of RC columns with circular cross section by using GEP. The developed model performs notably better than traditional models.

As an another example highlighting the applicability of GEP, Jumaa and Yousif (Arruda et al. 2014) proposed three prediction models for shear capacity prediction of FRP-reinforced concrete beams without stirrups, which utilized back-propagation neural networks, GEP and nonlinear regression analysis. Accordingly, 269 test results collected from 42 different studies were used to derive the proposed models. For the ANN model training, the trial and error procedure was used to obtain the best fitting model, which consisted of thirteen neurons in a single hidden layer. To develop the best GEP model, various models were evolved throughout a set of genetic operators. For NLR, the proposed equation was obtained based on one existing model proposed by Zsutty (1971). The predicting results of the proposed three models were compared to the common design provisions and seven equations proposed by other researchers. The comparison revealed that all of the proposed models had a sufficient accuracy, while ANN and GEP models had a higher accuracy than NLR model, but NLR model was computationally efficient than other two models. Ipek and Güneyisi (2019) deployed GEP to assess the axial strength of concrete-filled composite columns. GEP was utilized to generate the proposed model based on 103 sets of available data from some literature. The proposed model had a higher accuracy value and lower error value than other models (İpek and Güneyisi 2019). The best performance of each GEP method is summarized in Table 3.

3.4 Multi expression programming applications

Although MEP has not been widely used in civil engineering area, some studies discovered the potential of MEP. Alavi and Gandomi (2011) and his group presented the capabilities of LGP, GEP and MEP in dealing with geotechnical engineering problems. Models were developed from reliable databased gathered from literature. Compared to simple conventional models, LGP, GEP and MEP models had more comprehensive parameters and higher prediction accuracy. Further, Alavi et al. (2010) proposed two soil classification formulations developed by MEP, where results indicated that MEP models achieved a high accuracy. The best performance of each study is summarized in Table 4.

3.5 Multigene genetic programming applications

Various studies have been conducted to assess the performance of MGGP to solve engineering problems. Gandomi and Alavi (2012) presented a structural engineering system behavior modeling method using MGGP applied to four structural engineering problems. For this purpose, 1133 test results were used to develop the MGGP model to formulate the compressive strength of high-performance concrete. Also, 55 pure bending test results were utilized to develop the MGGP model for ultimate pure bending of steel circular tubes. Further, 156 test results were used to develop MGGP model for surface roughness in end-milling. Finally, 251 test results were used to develop model for failure modes of beams subjected to patch loads. Additionally, Bayazidi et al. (2014) proposed a MGGP-based prediction model for elastic modulus of both normal and high strength concrete. 70 groups of NSC and 89 groups of HSC experimental data extracted from literature were used to develop three MGGP models, where compressive strength of concrete was selected as an effective input for MGGP models. Three models were developed with different training parameters (e.g., generation number,

Table 3 Summary of the GEP performance indexes in the reviewed studies

Literature	Ozcan (2012)	Mousavi et al. (2012)	Ozbek et al. (2013)	Gandomi et al. (2014)	Danish (2012)	Hadianfard and Jafari (2016)	Aval et al. (2017)	Jafari and Mahini (2017)	Mansouri and Farzampour (2018)	Jumaa and Yousif (2014)	Ipek and Güneyisi (2019)	Sadrossadat et al. (2019)
R	0.93(Tr) 0.94(Te)	0.91(Tr) 0.91(Te)	0.99	0.92	0.87	0.98(Tr) 0.98(Te)	0.90	0.99	1.00 (Tr) 0.97(Te)	0.98	0.99(Tr) 0.99(Te)	0.97(Tr) 0.97(Te)

Table 4 Summary of the MEP performance indexes in reviewed studies

Literature	Alavi et al. (2011)	Alavi et al. (2010)
R	0.91 (Tr) 0.90 (Te)	0.99 (Tr) 0.98 (Te)

Table 5 Summary of the MGGP performance indexes in reviewed studies

Literature	Gandomi and Alavi (2012)	Bayazidi et al. (2014)	Li et al. (2019)
R	1.00	0.96	0.92

crossover ratio, etc.). Based on the performance of the proposed models on test data, it was revealed that they provided reliable estimates of elastic modulus of both HSC and NSC. As an another example, Li et al. (2019) proposed a method to estimate elastic modulus of reservoir rocks. An experiment was conducted to validate the proposed method, where results predicted by the proposed method were compared with empirical method and the measured results. The proposed method had a higher accuracy than traditional empirical method and required less input parameters (Li et al. 2019). The overall performance of each method is summarized in Table 5.

4 Discussion and future trends

This survey provides evidence that TGP, LGP, GEP, MEP, and MGGP can be efficiently used for solving various civil engineering problems that are challenging to solve with traditional methods. Their application is not limited to classical civil engineering tasks but also to other relevant arenas (Xu et al. 2013; Nourani et al. 2014; Garg et al. 2014; Pandey et al. 2015; Tran et al. 2016a; Suganuma et al. 2017). In most cases, the GP variants provide a comparable prediction performance. However, it has been observed that LGP, GEP and MEP can run several orders of magnitude faster than TGP and MGGP because the individuals in these variants are represented as linear strings (Alavi and Gandomi 2011). On the other hand, TGP and MGGP evolve tree-shaped programs. Since computers do not naturally run tree-shaped programs, slow interpreters have to be used as part of the TGP and MGGP simulation process. The enhanced speed of LGP, GEP and MEP permits conducting many runs in realistic timeframes, which leads to deriving consistent, high-precision models with little customization (Brameier and Banzhaf 2007).

Comparing GP with other ML methods, it can be inferred that the major advantage of GP is the transparency of the solutions. Precisely developed GP-based prediction equations can replace the design equations in various civil engineering codes (e.g. American Concrete Institute (ACI) Code) that are extensively used by civil engineers. These explicit models can be readily incorporated into numerical analysis tools such as finite element codes to replace the conventional (analytical) constitutive models. Another unique characteristic of GP is its inherent built-in feature selection mechanism which is known to avoid over-fitting (Vanneschi et al. 2011). In case of function fitting and regression, the overall performance of GP and other ML methods heavily depends on the complexity of the problem as well as other factors such as quality and completeness of the training data, error measures, etc. An ML method may depict better performance on a specific problem while in some other problems, it is outperformed by similar methods. However, there is more to ML than merely function approximation. Researchers are interested in methods that provide both good accuracy and model interpretability. Besides, there are other issues that needs to be considered when comparing the ML methods such as the training time, computational complexity, number of hyperparameters, scaling of the complexity with the number of variables, etc. A limitation of the GP-based methods is that they are parameter sensitive. The performance of these algorithms can be improved by using any form of optimally controlling the parameters of the run such as advanced optimization methods. In addition, the GP methods are computationally intensive techniques compared to ANNs, especially when applied to regression or classification tasks with large datasets. Exploiting highly parallel architectures in the form of Graphics Processing Units (GPUs) and Central Processing Unit (CPU) can improve their computational speed.

However, applications of GP and its variants in civil engineering area are fairly new. Most of the studies reviewed in this paper mainly focus on using the original versions of the GP methods. Although the obtained results sound promising, improving the existing GP algorithms can efficiently enable their ability in solving more complex civil engineering problems and achieving more accurate results. Several attempts made to improve the existing GP methods for application in other engineering areas can also be deployed to enhance the GP practice in the civil engineering domain. This section provides an insight into some of these GP developments that can be considered for tackling more complicated civil engineering problems. As an example, extensive research has been done to improve the GP evolutionary process, i.e. initialization, selection and reproduction. For traditional GP, the initial population are randomly selected, which may not be well-scattered enough. This issue can affect the convergence speed of the evolution. To tackle this challenge, Chen et al. (2013) proposed a novel evolutionary algorithm called uniform design-aided gene expression programming (UGEP) based on traditional GEP. In UGEP, a mixed-level uniform table is used to search uniformly scattered points in the whole searching space to generate a well-distributed initial population. During the selection procedure, the diversity of the population decreases, while the similarity among individuals in population increases. Therefore, it is important to maintain the diversity of population to avoid premature convergence. Premature convergence would result in convergence even the fitness is not the best or ideal. Several studies have been conducted to address such issue. Gan et al. (2007) proposed a clonal selection based gene expression programming (CS-GEP), merging Clonal Selection Algorithm (CSA) with general GEP to avoid the premature convergence issue. Similarly, Litvinenko et al. (2005) combined CSA with GP to overcome the premature convergence drawback by improving the diversity of population. Guo et al. (2014) proposed an enhanced GEP inspired by the principle of minimal free energy in thermodynamics, called CTSGEP. The authors created a formula to calculate the free energy component of combination of newly created offspring and their parents. Individuals with smaller free energy components were selected for the next generation. In addition, Wang et al. (2015) proposed a similarity weight tournament selection procedure to increase the diversity of population and improve the performance of existing GEP. In this selection procedure each individual is assigned with probabilities to be selected. Individuals that are the historical best would have a lower probability to be selected, but individuals with a better fitness in the tournament would be still selected to the next generation. The proposed method is applied to discover the dynamic relationship between the microstructure and physical properties of cement from the image.

Reproduction procedure of GP is highly dependent on genetic operators. To improve the searching accuracy and efficiency of GP, new algorithms with novel operators have been developed. Jiang et al. (2005) proposed a new algorithm that combines GEP and Simulated Annealing (SA), called GEPSA. The proposed algorithm decreases the dependence on genetic operators without affecting the GEP performance. To improve the accuracy and efficiency of GEP, Zhong et al. (2016) proposed a new self-learning GEP based on the existing GEP, named SL-GEP, in which each chromosome is assigned with a subfunction which are self-learned or self-evolved during the evolutionary search. With these subfunctions, SL-GEP is able to improve the accuracy and efficient of the search. Zeng et al. (2007) proposed an Immune Gene Expression Programming (IGEP) based on the Artificial Immune System (AIS) to address the importance of rule mining issue. Since population similarity can be described by edit distance between individuals, crossover is a way to create and maintain the distance. Also, Qu et al. (2015) proposed two edit distance-based crossover operators for GEP to preserve the population diversity. Further, Chen et al. (2017a,b,2019) improved the geometric semantic GP framework by introducing geometric search operators.

Another approach to improve of GP is to combine it with other methods such as optimization algorithms, ANNs, feature extraction algorithms, and deep learning. Most of the ML method are black-box techniques. To address this issue, studies have been performed to combine ANNs and deep neural networks (DNNs) with GP to provide more transparent models. Tran et al. (2016b) proposed a credit scoring framework by merging DNNs and the comprehensive GP. The framework shown in Fig. 12 was aimed to extract rules to build a more robust and reliable DL models. The framework was validated with experimental data (Tran et al. 2016b). Further, Garg et al. (2014), Suganuma et al. (2019), Evans et al. (2018) and Assunção et al. (2019) addressed the noted issue using similar approaches. Additionally, Aslam et al. (2012) combined K-nearest neighbor (KNN) and GP to conduct modulation classification. KNN was used to evaluate the fitness of GP during training process and to reduce the performance of the best individual in the testing process. The validation

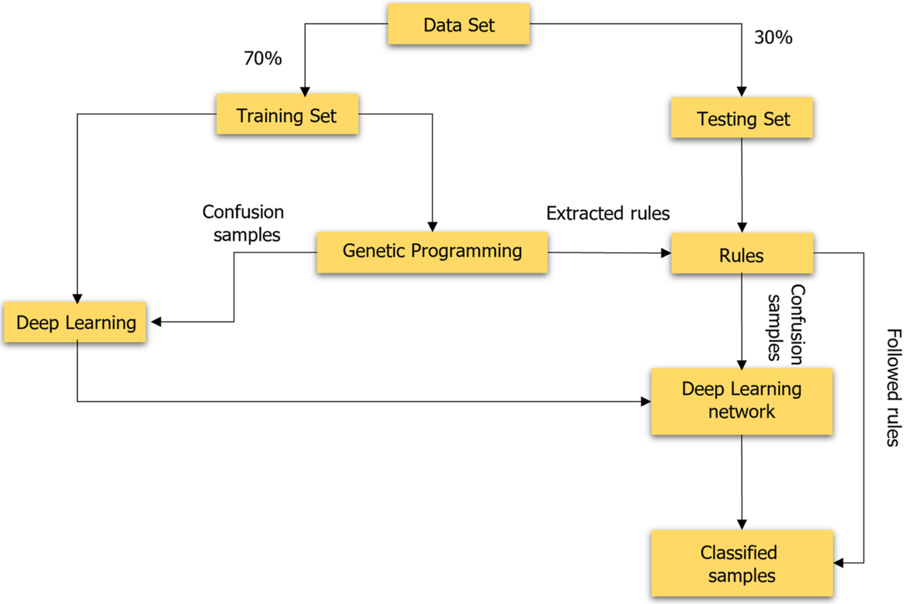


Fig. 12 Credit scoring framework in Tran et al. (2016b)

results for the proposed methods were better than other methods mentioned in the literature. Feature extraction algorithms are gaining more attention in recent years. Price et al. (2019) proposed a novel GP approach to automatically generate discriminative-rich features for image classification.

As discussed in this survey, applications of GP in civil engineering are significantly promising. Yet, such intelligent method could be further improved from various aspects to enhance its performance. Based on the review of the recent works on improving GP, two main approaches can be suggested for future research in the area of GP for civil engineering applications: improving the GP evolutionary process or hybridizing GP with other powerful methods. Figure 13 illustrates the development trend of GP for each approach.

5 Conclusion

Compared to other ML techniques, GP has been rarely used in civil engineering domain. This comprehensive review aims at presenting the importance of GP for civil engineering applications over the last decade. The most recent works concerning the applications of GP in civil engineering domain are discussed. It is shown that different GP branches such as TGP, LGP, GEP, MEP, and MGGP can be effectively adapted for various civil engineering applications. The survey reveals that GP and its variants are able to outperform existing models with high accuracy. GP allows solving complex problems which are difficult to be solved using conventional methods. Compared to other ANNs or other black-box ML methods, GP allows inclusion of experts' knowledge on the problem domain resulting in a powerful mix of ML and human insight (Rodriguez-Coayahuitl et al. 2019). However, based on the review, improvements of GP and its variants is vital given rapid advancement of the ML methods. The survey suggests that GP can be enhanced through either improvement of GP itself or a combination with other ML methods. According to review, potential future research trends for GP is discussed. It is highlighted that GP and its variants can

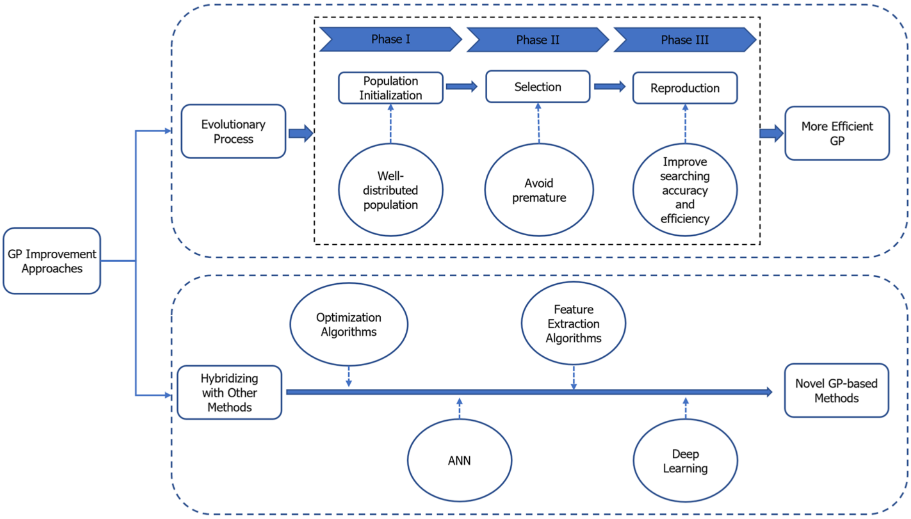

Phase I
Phase II
Phase III
Population Initialization
Selection
Reproduction
Evolutionary Process
More Efficient GP
Well-distributed population
Avoid premature
Improve searching accuracy and efficiency
GP Improvement Approaches
Optimization Algorithms
Feature Extraction Algorithms
Hybridizing with Other Methods
Novel GP-based Methods
ANN
Deep Learning

Fig. 13 The development trend of GP for civil engineering applications

be enhanced and effectively used as pioneering methods for variety of civil engineering applications. In particular, discovering novel materials is a trending topic which has been gaining significant attentions in recent years. There has been an explosion in the number of works deploying AI/ML in materials science domain (Liu et al. 2017). This computational paradigm change for material characterization and discovery is being further promoted by many researchers via more data-intensive approaches. While researchers are now heavily focused to revolutionize the traditional material science field with ML, the entire concept of ML for characterizing and designing new construction materials is still in its infancy let alone GP. Therefore, harnessing the power of GP for material discovery and design should be considered as a major future direction.

Acknowledgements Amir H. Alavi acknowledges the startup fund from the Swanson School of Engineering at the University of Pittsburgh. This study is supported in part by the Fundamental Research Funds for the Central Universities, China (2020-KYY-529112-0002). Pengcheng Jiao acknowledges the Startup Fund of the Hundred Talent Program at the Zhejiang University, China.

References

Alavi AH, Gandomi AH (2011) A robust data mining approach for formulation of geotechnical engineering systems. Eng Comput 28(3):242–274. https://doi.org/10.1108/02644401111118132

Alavi AH, Gandomi AH, Sahab MG, Gandomi M (2010) Multi expression programming: a new approach to formulation of soil classification. Eng Comput 26(2):111–118. https://doi.org/10.1007/s00366-009-0140-7

Araujo L (2007) How evolutionary algorithms are applied to statistical natural language processing. Artif Intell Rev 28(4):275–303

Aslam MW, Zhu Z, Nandi AK (2012) Autmomatic modulation classification using combination of genetic programming and KNN. IEEE Trans Wirel Commun. https://doi.org/10.1109/TWC.2012.060412.110460

Assunção F, Lourenço N, Machado P, Ribeiro B (2019) DENSER: deep evolutionary network structured representation. Genet Program Evol Mach 20(1):5–35. https://doi.org/10.1007/s10710-018-9339-y

Aval SB, Ketabdari H, Asil Gharebaghi S (2017) Estimating shear strength of short rectangular reinforced concrete columns using nonlinear regression and gene expression programming. Structures 12:13–23. https://doi.org/10.1016/j.istruc.2017.07.002

Babanajad SK, Gandomi AH, Mohammadzadeh D, Alavi AH (2013) Numerical modeling of concrete strength under multiaxial confinement pressures using linear genetic programming. Autom Constr 36:136–144. https://doi.org/10.1016/j.autcon.2013.08.016

Bayazidi A, Wang G-G, Bolandi H, Alavi AH, Gandomi AH (2014) Multigene genetic programming for estimation of elastic modulus of concrete. Math Probl Eng 2014:1–10. https://doi.org/10.1155/2014/474289

Begum SAB (2013) Genetic programming for prediction of local scour at vertical bridge abutment. Int J Res Eng Technol 02(14):74–77. https://doi.org/10.15623/ijret.2013.0214015

Brameier M, Banzhaf W (2007) Linear genetic programming. Springer, New York

Bui D-K, Nguyen T, Chou J-S, Nguyen-Xuan H, Ngo TD (2018) A modified firefly algorithm-artificial neural network expert system for predicting compressive and tensile strength of high-performance concrete. Constr Build Mater 180:320–333. https://doi.org/10.1016/j.conbuildmat.2018.05.201

Castelli M, Vanneschi L, Silva S (2013) Prediction of high performance concrete strength using Genetic Programming with geometric semantic genetic operators. Expert Syst Appl 40(17):6856–6862. https://doi.org/10.1016/j.eswa.2013.06.037

Cevik A, Arslan MH, Köroğlu MA (2010) Genetic-programming-based modeling of RC beam torsional strength. KSCE J Civ Eng 14(3):371–384. https://doi.org/10.1007/s12205-010-0371-6

Chen Y, Chen D, Khan SU, Huang J, Xie C (2013) Solving symbolic regression problems with uniform design-aided gene expression programming. J Supercomput 66(3):1553–1575. https://doi.org/10.1007/s11227-013-0943-6

Chen Q, Zhang M, Xue B (2017) New geometric semantic operators in genetic programming: perpendicular crossover and random segment mutation. In: Proceedings of the genetic and evolutionary computation conference companion on - GECCO ’17, Berlin, Germany, pp 223–224. https://doi.org/10.1145/3067695.3076008

Chen Q, Zhang M, Xue B (2017) Feature selection to improve generalization of genetic programming for high-dimensional symbolic regression. IEEE Trans Evol Comput 21(5):792–806. https://doi.org/10.1109/TEVC.2017.2683489

Chen Q, Xue B, Zhang M (2019) Improving generalization of genetic programming for symbolic regression with angle-driven geometric semantic operators. IEEE Trans Evol Comput 23(3):488–502. https://doi.org/10.1109/TEVC.2018.2869621

Danish M (2012) Prediction of scour depth at bridge abutments in cohesive bed using gene expression programming, vol 5, no 11, p 8 (2012)

de Arruda PM, Davis Júnior CA, Gontijo Carrano E, de Vasconcelos JA (2014) A niching genetic programming-based multi-objective algorithm for hybrid data classification. Neurocomputing 133:342–357

Evans B, Al-Sahaf H, Xue B, Zhang M (2018) Evolutionary deep learning: a genetic programming approach to image classification. In: 2018 IEEE Congress on evolutionary computation (CEC), Rio de Janeiro, pp 1–6. https://doi.org/10.1109/CEC.2018.8477933

Ferreira C (2001) Gene expression programming: a new adaptive algorithm for solving problems. ArXiv Prepr. Cs0102027

Gan Z, Yang Z, Li G, Jiang M (2007) Automatic modeling of complex functions with clonal selection-based gene expression programming. In: Third international conference on natural computation (ICNC 2007), Haikou, China, 2007, pp 228–232. https://doi.org/10.1109/ICNC.2007.278.

Gandomi AH, Alavi AH (2012) A new multi-gene genetic programming approach to nonlinear system modeling. Part I: materials and structural engineering problems. Neural Comput Appl 21(1):171–187. https://doi.org/10.1007/s00521-011-0734-z

Gandomi AH, Alavi AH, Sahab MG, Arjmandi P (2010) Formulation of elastic modulus of concrete using linear genetic programming. J Mech Sci Technol 24(6):1273–1278. https://doi.org/10.1007/s12206-010-0330-7

Gandomi AH, Alavi AH, Shadmehri DM, Sahab MG (2013) An empirical model for shear capacity of RC deep beams using genetic-simulated annealing. Arch Civ Mech Eng 13(3):354–369. https://doi.org/10.1016/j.acme.2013.02.007

Gandomi AH, Alavi AH, Kazemi S, Gandomi M (2014) Formulation of shear strength of slender RC beams using gene expression programming, part I: Without shear reinforcement. Autom Constr 42:112–121. https://doi.org/10.1016/j.autcon.2014.02.007

Gandomi AH, Mohammadzadeh D, Pérez-Ordóñez JL, Alavi AH (2014) Linear genetic programming for shear strength prediction of reinforced concrete beams without stirrups. Appl Soft Comput 19:112–120. https://doi.org/10.1016/j.asoc.2014.02.007

Gandomi AH, Sajedi S, Kiani B, Huang Q (2016) Genetic programming for experimental big data mining: a case study on concrete creep formulation. Autom Constr 70:89–97. https://doi.org/10.1016/j.autcon.2016.06.010

Gandomi AH, Alavi AH, Gandomi M, Kazemi S (2017) Formulation of shear strength of slender RC beams using gene expression programming, part II: With shear reinforcement. Measurement 95:367–376. https://doi.org/10.1016/j.measurement.2016.10.024

Garg A, Garg A, Tai K (2014) A multi-gene genetic programming model for estimating stress-dependent soil water retention curves. Comput Geosci 18(1):45–56. https://doi.org/10.1007/s10596-013-9381-z

Ghiringhelli LM, Vybiral J, Levchenko SV, Draxl C, Scheffler M (2015) Big Data of materials science: critical role of the descriptor. Phys Rev Lett 114(10):105503. https://doi.org/10.1103/PhysRevLett.114.105503

Gholampour A, Gandomi AH, Ozbakkaloglu T (2017) New formulations for mechanical properties of recycled aggregate concrete using gene expression programming. Constr Build Mater 130:122–145. https://doi.org/10.1016/j.conbuildmat.2016.10.114

Giustolisi O, Doglioni A, Savic DA, Webb BW (2007) A multi-model approach to analysis of environmental phenomena. Environ Model Softw 22(5):674–682. https://doi.org/10.1016/j.envsoft.2005.12.026

Guo Z, Wu Z, Dong X, Zhang K, Wang S, Li Y (2014) Component thermodynamical selection based gene expression programming for function finding. Math Probl Eng 2014:1–16. https://doi.org/10.1155/2014/915058

Guven A (2009) Linear genetic programming for time-series modelling of daily flow rate. J Earth Syst Sci 118(2):137–146. https://doi.org/10.1007/s12040-009-0022-9

Hadianfard MA, Jafari S (2016) Prediction of lightweight aggregate concrete compressive strength using ultrasonic pulse velocity test through gene expression programming. Sci Iran 23(6):2506–2513. https://doi.org/10.24200/sci.2016.2309

Hamedi M (2005) Intelligent fixture design through a hybrid system of artificial neural network and genetic algorithm. Artif Intell Rev 23(3):295–311

İpek S, Güneyisi EM (2019) Ultimate axial strength of concrete-filled double skin steel tubular column sections. Adv Civ Eng 2019:1–19. https://doi.org/10.1155/2019/6493037

Jafari S, Mahini SS (2017) Lightweight concrete design using gene expression programing. Constr Build Mater 139:93–100. https://doi.org/10.1016/j.conbuildmat.2017.01.120

Jalal M, Ramezanianpour AA, Pouladkhan AR, Tedro P (2013) Application of genetic programming (GP) and ANFIS for strength enhancement modeling of CFRP-retrofitted concrete cylinders. Neural Comput Appl 23(2):455–470. https://doi.org/10.1007/s00521-012-0941-2

Jiang S, Cai Z, Zeng D, Liu Y, Li Q (2005) Gene expression programming based on simulated annealing*. In: Proceedings. 2005 international conference on wireless communications, networking and mobile computing, 2005, Wuhan, China, vol 2, pp 1218–1221. https://doi.org/10.1109/WCNM.2005.1544273.

Joshi SG, Londhe SN, Kwatra N (2014) Determination of natural periods of vibration using genetic programming. Earthq Struct 6(2):201–216. https://doi.org/10.12989/EAS.2014.6.2.201

Koza JR (1992) Genetic programming, on the programming of computers by means of natural selection. MIT Press, Cambridge

Krippendorff K (2018) Content analysis: an introduction to its methodology. SAGE Publications, New York

Kushchu I (2002) An evaluation of evolutionary generalization in genetic programming. Artif Intell Rev 18(1):3–14

Li N, Chen H,Han J (2019) Application of multigene genetic programming for estimating elastic modulus of reservoir rocks. In: 2019 Symposium on piezoelectrcity, acoustic waves and device applications (SPAWDA), Harbin, China, pp 1–4. https://doi.org/10.1109/SPAWDA.2019.8681879

Litvinenko VI, Bidyuk PI, Bardachov JN, Sherstjuk VG, Fefelov AA (2005) Combining clonal selection algorithm and gene expression programming for time series prediction. In: 2005 IEEE intelligent data acquisition and advanced computing systems: technology and applications, Sofia, Bulgaria, 2005, pp 133–138. https://doi.org/10.1109/IDAACS.2005.282956

Liu Y, Zhao T, Ju W, Shi S (2017) Materials discovery and design using machine learning. J Materiomics 3:159–177

Mansouri I, Farzampour A (2018) Buckling assessment of imperfect cylindrical shells under axial loads using a gep technique. Elektron Časopis Građev Fak Osije. https://doi.org/10.13167/2018.17.9

Mollahasani A, Alavi AH, Gandomi AH, Bazaz B (2011) A new prediction model for soil deformation modulus based on PLT results, p 10 (2011)

Moradi M, Bagherieh AR, Esfahani MR (2019) Tensile modeling of steel fiber reinforced concrete. Asian J Civ Eng 20(2):269–280. https://doi.org/10.1007/s42107-018-00104-y

Mousavi SM, Aminian P, Gandomi AH, Alavi AH, Bolandi H (2012) A new predictive model for compressive strength of HPC using gene expression programming. Adv Eng Softw 45(1):105–114. https://doi.org/10.1016/j.advengsoft.2011.09.014

Nourani V, Pradhan B, Ghaffari H, Sharifi SS (2014) Landslide susceptibility mapping at Zonouz Plain, Iran using genetic programming and comparison with frequency ratio, logistic regression, and artificial neural network models. Nat Hazards 71(1):523–547. https://doi.org/10.1007/s11069-013-0932-3

Oltean M (2004) Multi expression programming source code. Available at: https://mep.cs.ubbcluj.ro/

Oltean M, Groşan C (2003) Evolving evolutionary algorithms using multi expression programming. In: European conference on artificial life, pp 651–658

Oltean M, Grosan C (2004) Evolving digital circuits using multi expression programming. In: Proceedings. 2004 NASA/DoD conference on evolvable hardware, pp 87–94

Ozbek A, Unsal M, Dikec A (2013) Estimating uniaxial compressive strength of rocks using genetic expression programming. J Rock Mech Geotech Eng 5(4):325–329. https://doi.org/10.1016/j.jrmge.2013.05.006

Özcan F (2012) Gene expression programming based formulations for splitting tensile strength of concrete. Constr Build Mater 26(1):404–410. https://doi.org/10.1016/j.conbuildmat.2011.06.039

Pandey DS, Pan I, Das S, Leahy JJ, Kwapinski W (2015) Multi-gene genetic programming based predictive models for municipal solid waste gasification in a fluidized bed gasifier. Bioresour Technol 179:524–533. https://doi.org/10.1016/j.biortech.2014.12.048

Peker M (2016) An efficient sleep scoring system based on EEG signal using complex-valued machine learning algorithms. Neurocomputing 207:165–177

Price SR, Anderson DT, Price SR (2019) GOOFeD: extracting advanced features for image classification via improved genetic programming. In: 2019 IEEE congress on evolutionary computation (CEC), Wellington, New Zealand, pp 1596–1603. https:doi.org/10.1109/CEC.2019.8790347

Qu L, Hongbing C, Lin HX (2015) Edit distance based crossover operator in gene expression programming. In: 2015 8th international conference on biomedical engineering and informatics (BMEI), Shenyang, China, pp 468–472. https://doi.org/10.1109/BMEI.2015.7401550

Rashed A, Bazaz JB, Alavi AH (2012) Nonlinear modeling of soil deformation modulus through LGP-based interpretation of pressuremeter test results. Eng Appl Artif Intell 25(7):1437–1449. https://doi.org/10.1016/j.engappai.2011.11.008

Rodriguez-Coayahuitl L, Morales-Reyes A, Escalante HJ (2019) A comparison among different levels of abstraction in genetic programming. In: 2019 IEEE international autumn meeting on power, electronics and computing (ROPEC), pp 1–6

Rostami MF, Sadrossadat E, Ghorbani B, Kazemi SM (2018) New empirical formulations for indirect estimation of peak-confined compressive strength and strain of circular RC columns using LGP method. Eng Comput 34(4):865–880. https://doi.org/10.1007/s00366-018-0577-7

Sadrossadat E, Basarir H (2019) An evolutionary-based prediction model of the 28-day compressive strength of high-performance concrete containing cementitious materials. Adv Civ Eng Mater 8(3)

Sadrossadat E, Ghorbani B, Hamooni M (2017) Numerical formulation of confined compressive strength and strain of circular reinforced concrete columns using gene expression programming approach. Struct Concrete

Sadrossadat E, Basarir H, Karrech A, Durham R, Fourie A, Bin H (2020) The optimization of cemented hydraulic backfill mixture design parameters for different strength conditions using artificial intelligence algorithms. In: Topal E (eds) Proceedings of the 28th international symposium on mine planning and equipment selection - MPES 2019. Springer Series in Geomechanics and Geoengineering. Springer, Cham

Salehi H, Burgueño R (2018) Emerging artificial intelligence methods in structural engineering. Eng Struct 171:170–189

Searson DP, Leahy DE, Willis MJ (2011) Predicting the toxicity of chemical compounds using GPTIPS: a free genetic programming toolbox for MATLAB. In: Ao S-I, Castillo O, Huang X (eds) Intelligent control and computer engineering, vol 70. Springer, Dordrecht, pp 83–93

Suganuma M, Shirakawa S, Nagao T (2017) A genetic programming approach to designing convolutional neural network architectures. In: Proceedings of the genetic and evolutionary computation conference on - GECCO ’17, Berlin, Germany, 2017, pp 497–504. https://doi.org/10.1145/3071178.3071229

Suganuma M, Kobayashi M, Shirakawa S, Nagao T (2019) Evolution of deep convolutional neural networks using Cartesian genetic programming. Evol Comput. https://doi.org/10.1162/evco_a_00253

Tran B, Xue B, Zhang M (2016a) Genetic programming for feature construction and selection in classification on high-dimensional data. Memetic Comput 8(1):3–15. https://doi.org/10.1007/s12293-015-0173-y

Tran K, Duong T, Ho Q (2016) Credit scoring model: a combination of genetic programming and deep learning. In: 2016 Future technologies conference (FTC), San Francisco, CA, USA, pp 145–149. https://doi.org/10.1109/FTC.2016.7821603

Tsai HC, Lin Y-H (2011) Predicting high-strength concrete parameters using weighted genetic programming. Eng Comput 27(4):347–355. https://doi.org/10.1007/s00366-011-0208-z

Vanneschi L, Farinaccio A, Mauri G, Antoniotti M, Provero P, Giacobini M (2011) A comparison of machine learning techniques for survival prediction in breast cancer. BioData Min 4(1):12

Wang L, Yang B, Wang S, Liang Z (2015) Building image feature kinetics for cement hydration using gene expression programming with similarity weight tournament selection. IEEE Trans Evol Comput 19(5):679–693. https://doi.org/10.1109/TEVC.2014.2367111

Wang Y, Yu Y, Cao S, Zhang X, Gao S (2019) ‘A review of applications of artificial intelligent algorithms in wind farms. Artif Intell Rev

Xu C, Wang W, Liu P (2013) A genetic programming model for real-time crash prediction on freeways. IEEE Trans Intell Transp Syst 14(2):574–586. https://doi.org/10.1109/TITS.2012.2226240

Yalpir S (2016) Enhancement of parcel valuation with adaptive artificial neural network modeling. J Artif Intell Rev 1–13

Zeng T, Tang C, Xiang Y, Chen P, Liu Y (2007) A model of immune gene expression programming for rule mining, p 14

Zhang Y, Rockett P (2007) A comparison of three evolutionary strategies for multi-objective genetic programming. Artif Intell Rev 27:149–163

Zhong J, Ong Y-S, Cai W (2016) Self-learning gene expression programming. IEEE Trans Evol Comput 20(1):65–80. https://doi.org/10.1109/TEVC.2015.2424410

Zsutty T (1971) Shear strength prediction for separate catagories of simple beam tests. J Proc 68:138–143