Prediction of self-compacting concrete elastic modulus using two symbolic regression techniques

Emadaldin Mohammadi Golafshani[a*], Ashraf Ashour[b]

[a] Department of Civil Engineering, Tehran Science and Research Branch, Islamic Azad University, Tehran, Iran
[b] School of Engineering, University of Bradford, Bradford, BD7 1DP, UK

Abstract

This paper introduces a novel symbolic regression approach, namely biogeographical-based programming (BBP), for the prediction of elastic modulus of self-compacting concrete (SCC). The BBP model was constructed directly from a comprehensive dataset of experimental results of SCC available in the literature. For comparison purposes, another new symbolic regression model, namely artificial bee colony programming (ABCP), was also developed. Furthermore, several available formulas for predicting the elastic modulus of SCC were assessed using the collected database.

The results show that the proposed BBP model provides slightly closer results to experiments than ABCP model and existing available formulas. A sensitivity analysis of BBP parameters also shows that the prediction by BBP model improves with the increase of habitat size, colony size and maximum tree depth. In addition, among all considered empirical and design code equations, Leemann and Hoffmann and ACI 318-08's equations exhibit a reasonable performance but Persson and Felekoglu et al.'s equations are highly inaccurate for the prediction of SCC elastic modulus.

Keywords: Self- compacting concrete; Elastic modulus; Symbolic regression; Artificial bee colony programming; Biogeographical-based programming.

1. Introduction

Self-compacting concrete (SCC), initially proposed by Okamura in 1986, has gained a wide acceptance in the construction industry [1-3]. SCC is characterized by the ability to flow under its own weight to adequately fill the formwork without any internal or external mechanical vibration [4,5]. SCC also possesses enough viscosity to be handled without segregation or bleeding [6-9].

SCC mixtures are usually designed with limiting aggregate contents, high volumes of paste, a low water-powder ratio, large quantities of mineral fillers and high range water reducing admixtures [10]. Consequently, the fresh and hardened properties of SCC are different from normally vibrated concrete (NVC) [11]. Several researchers have investigated SCC mix design [12], fresh and hardened properties of SCC [13] and structural performance of SCC members [14]. Due to the rapid growth of the use of SCC, determination of its mechanical properties compared with conventional concrete is essential in order to fulfill design requirements and codes. Elastic modulus of concrete is a crucial mechanical property in design and analysis of concrete structures, for example member deflections for serviceability requirements, seismic analysis, drift calculations, elastic shortening of concrete in pressstressed concrete design and creep losses [11]. Various relationships were proposed for predicting the elastic modulus of SCC and NVC, mostly from concrete compressive strength [13,15-21].

Various researchers applied different branches of artificial intelligence for predicting the elastic modulus of different types of concrete. Demir [22,23] applied artificial neural network and fuzzy modeling for predicting the elastic modulus of both normal and high-strength concrete. Demir and Korkmas [24] presented a new approach for predicting the upper and lower bounds of elastic modulus of high-strength concrete using fuzzy models. Yan and Shi [25] investigated the use of support vector machine (SVM) to predict the elastic modulus of normal and high-strength concretes from their compressive strengths. Ahmadi-Nedushan [26] applied an adaptive network-based fuzzy inference system (ANFIS) to predict the elastic modulus of normal and high-strength concrete.

Symbolic regression, namely symbolic function identification, is a function discovery approach for analysis and modeling of numeric multivariate datasets. Unlike traditional linear and nonlinear regression methods that fit parameters to an equation of a given form, symbolic regression tries to form mathematical equations by searching the parameters and the form of equations [27; 28]. In other words, symbolic regression method searches nonlinear equation forms and its parameters simultaneously for an addressed modeling problem. It attempts to derive a mathematical function to describe the relationship between dependent and independent variables [28,29]. Different novel methods have been developed for symbolic function identification as briefly reviewed in the next paragraph.

In recent years, a variety of evolutionary algorithms (EA) have been developed as feasible and effective methods for optimization problems [30-35]. Many EAs have been proposed, including genetic algorithms (GA), evolution strategies (ES), ant colony optimization (ACO), particle swarm optimization (PSO), differential evolution (DE), estimation of distribution algorithms (EDA), immune system optimization, artificial bee colony optimization (ABCO), and many others [36]. Inspired by GA, Genetic programming (GP), developed by Koza [37], is the most popular technique used in symbolic regression. Afterwards, some researchers introduced different improved versions of genetic programming, for example, linear genetic programming [38], cartesian genetic programming [39] and gene expression programming [40], etc. However, few researches on using other evolutionary based algorithms in symbolic regression or automatic programming were also developed. Musilek et al. [41] described Immune Programming (IP), inspired from the vertebrate immune system, as a paradigm in the field of evolutionary computing. Based on ant colony optimization (ACO) and using dynamically changing pheromone table, Shirakawa et al. [42] proposed Dynamic Ant Programming (DAP) for automatic construction of programs. Gan et al. [43] developed clone selection programming (CSP) for symbolic regression and applied clone selection principle as a search strategy. Inspired by artificial bee colony optimization (ABCO) algorithm, Karaboga et al. [28] introduced artificial bee colony programming (ABCP) as a new symbolic regression method and compared its performance with genetic programming (GP) approach on a large set of symbolic regression benchmark problems. They concluded that ABCP is very feasible and robust on the considered test problems of symbolic regression.

Biogeography-based optimization (BBO), developed by Simon [44], is a relatively new evolutionary algorithm inspired by biogeography, which involves the study of the migration of biological species between habitats and successfully applied to different branches of engineering [44-64]. Motivated by BBO, biogeographical-based programming (BBP) as a novel symbolic regression model is proposed in this study for predicting the elastic modulus of self-compacting concrete. The results of the developed BBP model are also compared with artificial bee colony programming (ABCP) representing a recent symbolic regression model. The rest of this paper is organized as follows. Section 2 briefly introduces the mathematical model of ABCO and ABCP algorithms, while Section 3 presents the BBO algorithm and the proposed BBP model for the prediction of SCC elastic modulus. In Section 5, the properties of gathered data for modeling the elastic modulus of SCC are described, whereas section 5 presents the analysis and discussion of results, followed by the main conclusions in Section 6.

2. Artificial bee colony programming

2.1. Artificial bee colony optimization

Artificial Bee Colony optimization (ABCO) algorithm, proposed by Karaboga et al. [65], is one of the most recently introduced swarm-based optimization algorithms. The artificial bees are classified into three groups, namely employed, onlooker and scout bees. Employed bees search the food around the food source in their memory. Meanwhile, they pass their food information to onlooker bees via a woggle dance on the dancing area in the hive. Onlooker bees tend to select good food sources from those found by the employed bees and then further search the foods around the selected food source. When the food sources of an employed bee become abandoned, such employed bee becomes a scout and starts searching for a new food source. Similar to the other population-based algorithms, ABCO is an iterative technique, consisting of two main processes, namely, exploration and exploitation. Scout bees can be visualized as performing the job of exploration, whereas employed and onlooker bees are performing the job of exploitation [66,67].

In ABCO algorithm, each food source is a possible solution for the problem under consideration and the nectar amount of a food source represents the quality of the solution represented by the fitness value [68]. The number of food sources is the same as the number of employed bees and there is exactly one employed bee for every food source. For each ABC, a fitness value, fit_i, can be assigned to the solution. For a minimization problem, the following equation proposed by Akay and Karaboga [30] may be used as a fitness function:

$$fit_i = \begin{cases} \frac{1}{1+f_i} & if\ f_i \geq 0 \\ 1 + abs(f_i) & if\ f_i < 0 \end{cases} \tag{1}$$

where f_i is the cost value of the solution i. For maximization problems, the cost function can be directly used as a fitness function. An onlooker bee applies roulette wheel selection scheme and chooses a food source depending on the probability value p_i associated with that food source [30]:

$$p_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_i} \tag{2}$$

where fit_i is the fitness value of the solution i which is proportional to the nectar amount of food source i in the position i and SN is the number of food sources that is equal to the number of employed bees or onlooker bees. Obviously, the higher the fit_i is, the more probability that the ith food source is selected [30].

In order to create a candidate food position $\mathbf{V}_\mathrm{i}$ = [$v_{i,1}$, $v_{i,2}$, . . . , $v_{i,D}$] from the old one $\mathbf{X}$i = [$x_{i,1}$, $x_{i,2}$, . . . , $x_{i,D}$] in memory, the ABCO produces a modification on the position of the old food source and finds a neighboring food source using the following expression [68]:

$$v_{i,j} = x_{i,j} + \emptyset_{i,j}\left(x_{i,j} - x_{k,j}\right) \tag{3}$$

where $j \in \{1, 2, \ldots, D\}$ and $k \in \{1, 2, \ldots, SN\}$ are randomly chosen indexes, but k has to be different from i; D is the number of decision variables (problem dimension); $\Phi_{i,j}$ is a uniformly distributed real random number in the range $[-1, 1]$.

After each candidate source position is produced and evaluated by the artificial bee, a greedy selection is applied and the performance of newly produced source position is compared with that of the old one. If the new food source is superior to that of the old source in terms of profitability, the old one is replaced by the new one. Otherwise, the old one is retained. If a position cannot be improved further through a predetermined number of cycles, then the food source is assumed to be abandoned. The value of a predetermined number of cycles, known as the "*limit*", is an important control parameter of ABCO algorithm. Assume that the abandoned source is $\mathbf{X}_i$, then the scout bees discover a new food source to replace $\mathbf{X}_i$ as below [68]:

$$x_{i,j} = x_{min,j} + rand(0,1)(x_{max,j} - x_{min,j}) \tag{4}$$

where $x_{min,j}$ and $x_{max,j}$ are the minimum and maximum values of the jth decision variables, respectively and j is randomly selected between 1 and D [68].

2.2. Development of artificial bee colony programming

Artificial bee colony programming (ABCP) is a symbolic regression method that allows evolving expressions and constants in the same representation and automatically forming the mathematical functions. In ABCP, food sources' positions correspond to randomly generated computer programs that are represented by tree structures. Computer programs are composed of terminals and functions such as arithmetic operations, mathematical functions, programming operations, logical functions, or domain-specific functions. Depending on the problem under consideration and applied functions, terminals may vary in constants or different type variables [28].

The quality of each food source is measured by evaluating the performance of each individual computer program, i.e. how the result of obtained function fits with the target one [28]. The root mean squared error (RMSE) can be used as a cost function which can be calculated as follows:

$$RMSE = \frac{1}{P}\sum_{i=1}^{P}(O_i - t_i)^2 \tag{5}$$

where P, O_i and t_i are the number of patterns, the output of each computer program and target value of ith pattern in the database, respectively.

In ABCP, the ramped half-and-half method, a combination of full and grow methods, is used for creating a random initial colony avoiding duplicate computer programs. The steps of ABCP are summarized below [28]:

- Initial colony generation and evaluation,
- Creation and evaluation of new computer programs for employed bees,
- Selection of computer programs depending on their quality,
- Creation and evaluation of new ones by each onlooker bee.
- After employed and onlooker bees phases, the unimproved computer programs are examined; if there is one of which the number of fail trials exceeds the limit value, a scout bee generates a new computer program to replace it with a grow method considering duplications.

These steps are iteratively performed until the termination criterion has been satisfied. The pseudo-code of the basic ABCP is shown in Fig. 1 [28].

```
1.  Generate initial computer programs (X_i) with Ramped half-and-half method
2.  Evaluate the computer programs
3.  Set cycle to 1
4.  Repeat
5.  For each employed bee {
            Produce new computer programs v_i by using information sharing mechanism
            Evaluate the computer programs by using Eqs. (1) and (5)
            Apply greedy selection process between x_i and v_i}
6.  Calculate the probability values p_i for computer programs by Eq. (2)
7.   For each onlooker bee {
            Select a computer program x_i depending on p_i probabilistically.
            Produce new computer program v_i by using information sharing mechanism
            Evaluate the computer programs by using Eqs. (1) and (5)
            Apply greedy selection process between x_i and v_i}
8.  If there is an abandoned computer program
            then replace it with a new computer program generated by grow method by a scout
9.  Memorize the best solution so far
10. cycle = cycle + 1
11. until cycle = maximum cycle number
```

Fig. 1. The pseudo-code of the basic ABCP [28].

The main modification of the ABCO algorithm in adaptation to ABCP is on the neighborhood mechanism while producing candidate solutions, which is called an information sharing mechanism. Since the solutions' structures are tree based on ABCP, the dimensions of the solutions' strings are not of a fixed-length. Therefore, Eq. (3) cannot be used directly in ABCP. In generation of a candidate solution (v_i) in ABCP, a tree node from a neighborhood solution (x_k) is randomly chosen by either a computer program in the probability of P_{ip} (set to 0.9) or a terminal in the probability of 1- P_{ip}. This chosen node from the neighborhood solution (x_k) determines which information and how much of it will be shared with the current solution. Then, a tree node in the current solution (x_i), which determines how neighborhood information will be used, is also randomly chosen in P_{ip} probability distribution. The candidate solution (v_i) is produced by replacing the node from the current solution (x_i) with the node from the neighborhood solution (x_k). This sharing mechanism is shown in Fig. 2, where Figs. 2(a) and (b) demonstrate the current solution (x_i) and the neighborhood solution (x_k) respectively, the received information from the neighborhood is shown in Fig. 2(c) and the produced candidate solution is given in Fig. 2(d). As in ABCO algorithm, after the candidate solution is produced, a greedy selection process is applied between the current solution (x_i) and the candidate solution (v_i) [28].

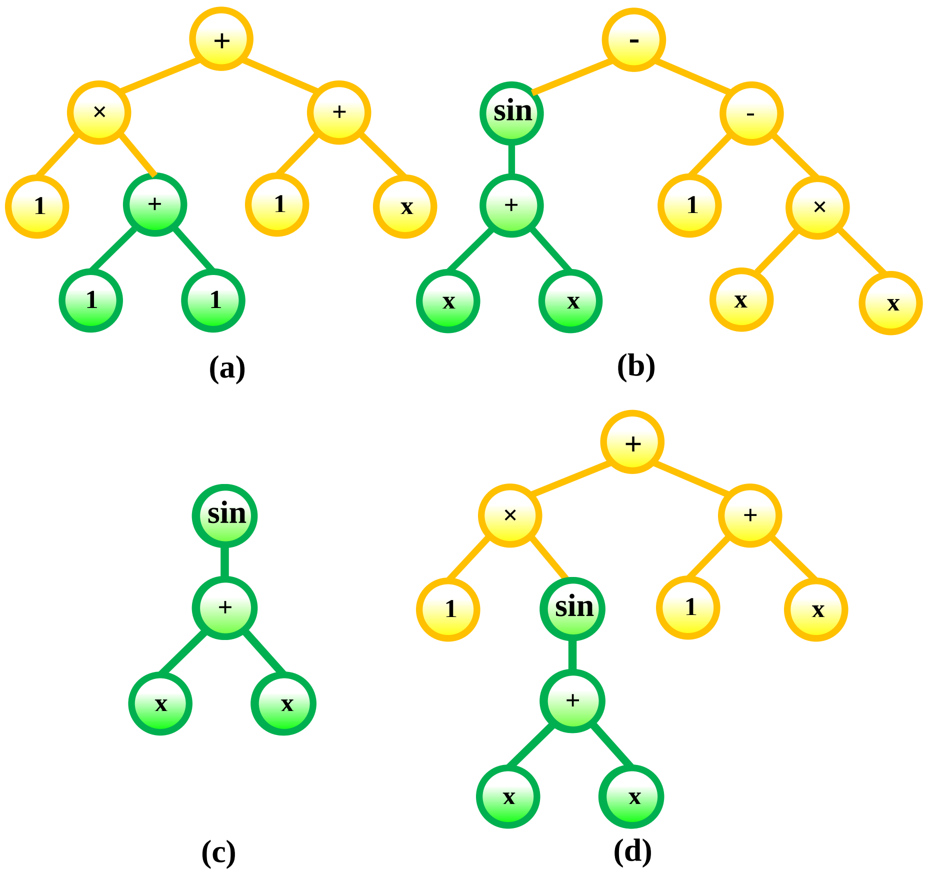


Fig. 2. The sharing example of ABCP [28].

3. Biogeographical-based programming (BBP)

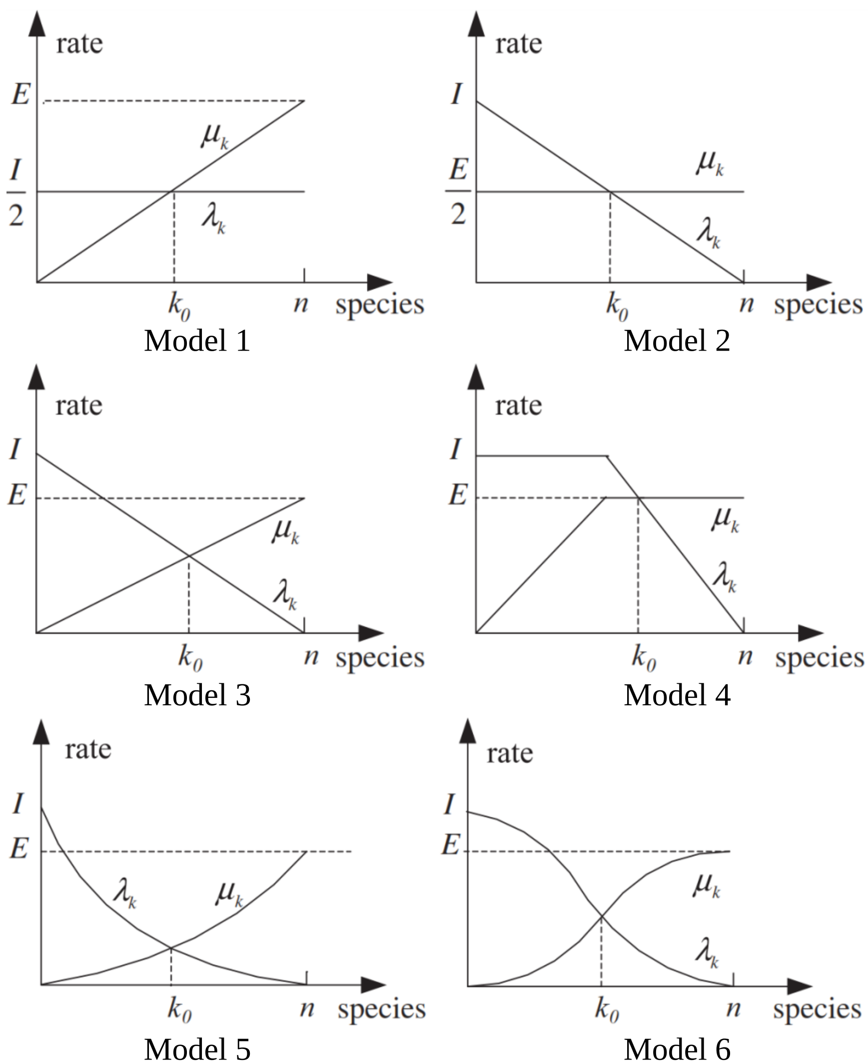
3.1. Biogeographical-based optimization (BBO)

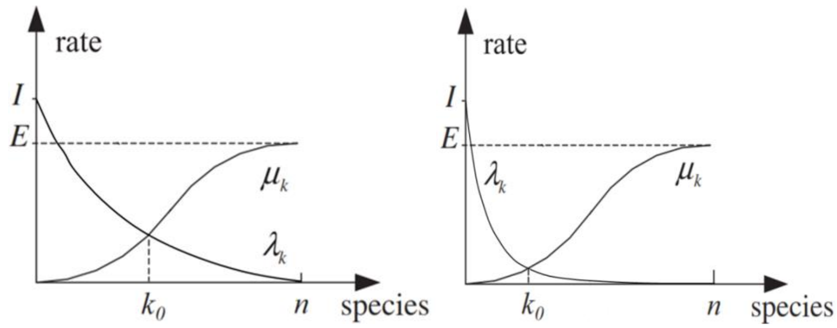
BBO, initially proposed by Simon [44], is a novel evolutionary optimization algorithm that is inspired by biogeography. This algorithm simulates the colonization and extinction of species between habitats in a multi-dimensional space [69]. In this algorithm, a set of biogeography habitats denotes a population of candidate solutions and habitat suitability index (HSI) corresponds to the goodness of each candidate solution. This corresponds to a geographical habitat that is well suited for hosting biological species. A good solution with high fitness value is analogous to a biological habitat with a high HSI. On the other hand, a poor solution is analogous to a habitat that is not well suited for hosting biological species. High fitness BBO solutions correspond to biological habitats with a large number of species, whereas low fitness BBO solutions indicate habitats with a small number of species [70]. Each candidate solution in BBO probabilistically shares decision variables with other candidate solutions to improve candidate solution fitness, similar to other evolutionary algorithms. This sharing process is analogous to migration in biogeography. That is, each candidate solution immigrates decision variables from other candidate solutions based on its immigration rate, and emigrates decision variables to other candidate solutions based on its emigration rate. BBO consists of two main steps: migration and mutation [71], as explained below.

Migration is a probabilistic operator that improves a habitat H_i. The migration rates of each habitat are used to probabilistically share features between habitats. For each habitat H_i, its immigration rate λ_i is used to probabilistically decide whether or not to immigrate. If migration is selected, then the emigrating habitat H_j is probabilistically selected based on emigration rate μ_j. Migration is defined by [71]:

$$H_i(SIV) \leftarrow H_j(SIV) \tag{6}$$

In biogeography, a suitability index variable (SIV) is a solution feature which determines the habitability of a habitat [44]. Immigration rates and emigration rates are based on migration curves. Several types of linear and non-linear immigration and emigration rates are shown in Fig. 3, where the maximum immigration and emigration rates are equal to I and E, respectively. The equations of different migration models are also given in Table 1 [36].





Model 7 Model 8

Fig. 3. Eight migration models [71,72].

Table 1
The formulas of migration models [71,72].

Models	Description	Immigration rate (λ_k)	Emigration rate (μ_k)
Model 1	Constant immigration and linear emigration model*	$\frac{I}{2}$	$E\frac{k}{n}$
Model 2	Linear immigration and constant emigration model	$I(1-\frac{k}{n})$	$\frac{E}{2}$
Model 3	Linear migration model	$I(1-\frac{k}{n})$	$E\frac{k}{n}$
Model 4	Trapezoidal migration model**	$\begin{cases} I & ,k \le i_0 \\ 2I\left(1-\frac{k}{n}\right) & ,i_0 < k \le n \end{cases}$	$\begin{cases} 2E\frac{k}{n} & ,k \le i_0 \\ E & ,i_0 < k \le n \end{cases}$
Model 5	Quadratic migration model	$I(\frac{k}{n}-1)^2$	$E(\frac{k}{n})^2$
Model 6	Sinusoidal migration model	$\frac{I}{2}(\cos\left(\frac{k\pi}{n}\right)+1)$	$\frac{E}{2}(-\cos\left(\frac{k\pi}{n}\right)+1)$
Model 7	Sinusoidal immigration and biquadratic emigration model	$\frac{I}{2}(\cos\left(\frac{k\pi}{n}\right)+1)$	$E(\frac{k}{n})^4$
Model 8	Sinusoidal immigration and sixteenth degree emigration model	$\frac{I}{2}(\cos\left(\frac{k\pi}{n}\right)+1)$	$E(\frac{k}{n})^{16}$

* k and n are the number and the maximum number of species in the habitat, respectively.

** i_0 is the smallest integer that is larger than or equal to $\frac{(n+1)}{2}$, namely, $i_o = \text{ceil}\ (\frac{(n+1)}{2})$.

Mutation is a probabilistic operator that randomly modifies a habitat's SIV based on the habitat's priori species count probability. The purpose of mutation is to increase the exploration of search space. Mutation gives low HSI solutions a chance of enhancing the quality of solutions, whereas improves high HSI solutions even more [70,72].

3.2. Development of biogeographical-based programming (BBP)

Similar to other automatic programming algorithms, the proposed BBP aims at reaching at an explicit mathematical expression between one or more inputs and an output using mathematical functions, variables and constants. The process of programming is a subset of symbolic function identification and differs from conventional regression in that it does not calculate the coefficients of functions. Indeed, BBP finds equations by performing an extensive and structured search in an evolving search space.

The proposed BBP continues the trend of dealing with the problem of representation in BBO by increasing the complexity of the structures undergoing adaptation. In particular, the structures undergoing adaptation in BBP are general, hierarchical computer programs of dynamically varying size and shape. The proposed BBP is based on mathematical models of biogeography describing natural ways of distributing species, i.e., how species migrate, how they arise and become extinct. The BBP serves to provide a platform with operators such as random generation, mutation and migration that produce, alter and select habitats in population. The flowchart of BBP is given in Fig. 4 and the main steps of the proposed metaheuristic programming algorithm are explained in more details below.

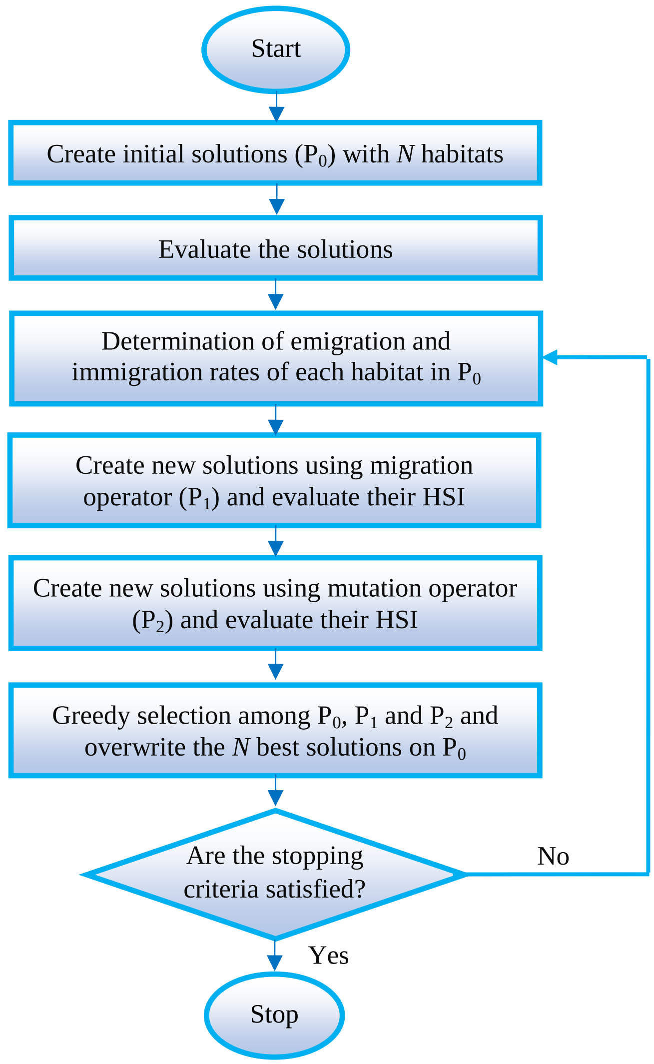


Fig. 4. The flowchart of BBP algorithm.

- Create initial population (P_0):
  It is composed of N initial habitats that are randomly generated. Each habitat presents a possible solution of the regression problem. It mathematically has an expression tree consisting of variables, functions and constants. From a population based programming viewpoint, the population of solutions refers to biogeography.
- Evaluate the solutions:
  The goal of BBP is to find a habitat that performs well for all patterns in the database with least error. In the BBP, habitat suitability index (HSI) is used for evaluation of each solution. In the proposed algorithm, the term HSI is related to the root mean squared error (RMSE), presented in Eq. (5). Habitat with a high HSI has a low RMSE and tends to

have a large number of species. RMSE minimization results in better expressions over the generations. The best expression with the largest HSI or least RMSE is chosen.

- Determination of emigration and immigration rates:

  The immigration rate λ is assigned to control habitat immigration. The maximum immigration rate in a single habitat (I) occurs when there are no species in this habitat. As the number of species in a habitat increases, it becomes crowded, fewer species would be able to survive in this habitat and the immigration subsequently decreases. Emigration rate μ is another attribute of a habitat in BBP that controls habitat emigration. If there are no species in a habitat, its emigration is null. As the number of species increases, species are capable to leave their habitat in order to explore other and may be better residences. Maximum emigration rate (E) occurs in a single habitat when containing the maximum number of species it can support.

- Create and evaluate new solutions using migration operator (P_1):

  The migration operator is used to migrate species between two habitats according to their immigration and emigration rates. Similar to ABCP, the migration from one habitat to another one is done by swapping a part of one habitat with a part of the other. The migration operator used for the BBP proceeds by the following steps:

  - Choose two habitats based on their immigration and emigration rates. The habitat with high HSI is more probable to be selected for emigration and the chance of habitat with low HSI being selected for immigration is high. Indeed, the migration from a habitat with high HSI to a habitat with low HSI is more probable. The candidate habitats for emigration and immigration are named emigrated and immigrated habitats, respectively.
  - Select a random subtree in emigrated and immigrated habitats.
  - Migrate the selected subtree of the emigrated habitat to the selected subtree of the immigrated habitat. The resulting tree is a new habitat where its depth should be checked.

  The process of migration operator is shown in Fig. 5, using two arbitrary simple expressions as emigrated and immigrated habitats. After the new habitat was created, its HSI is evaluated and saved in migrated population P_1.

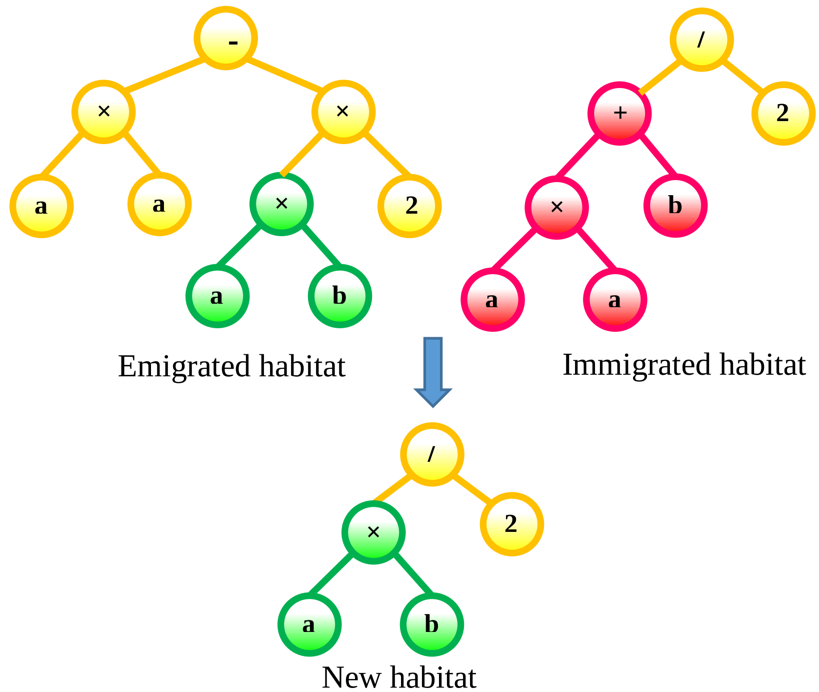


Fig. 5. Example of migration operator.

- Create and evaluate new solutions using mutation operator (P_2):
  The mutation operator introduces random changes in the structure of the habitats in the population P_0. Mutation begins by selecting a random point within the tree structure of a random habitat, that is then replaced by a randomly generated subtree at that point. This operator is controlled by a parameter that specifies the maximum size of tree depth for the newly inserted subtree. This process is shown in Fig. 6, where an arbitrary simple habitat is mutated by replacing one of its subtrees with a random tree. After applying the mutation operator, the performance of the created habitat is evaluated and saved in mutated population P_2.

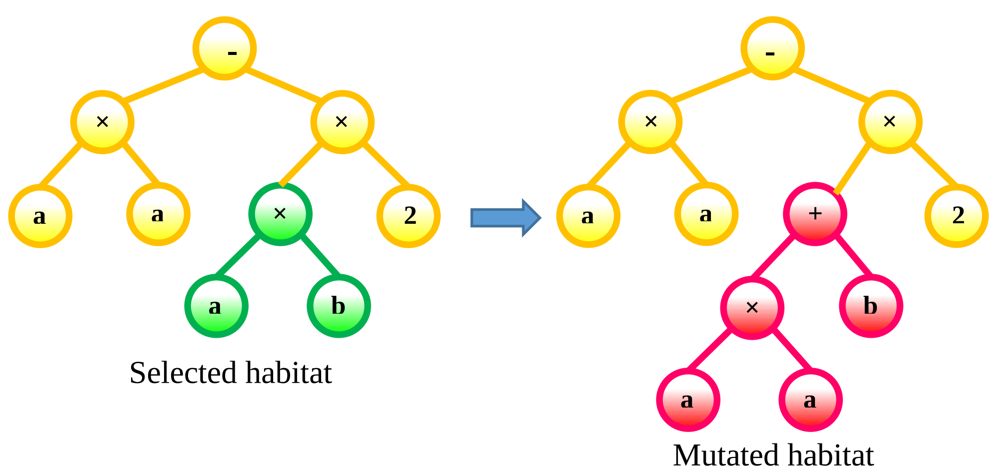


Fig. 6. Example of mutation operator.

- Greedy selection:
  In this step, the populations P_0, P_1 and P_2 are merged. The N best habitats are selected based on greedy selection of habitats in the merged population. The old population P_0 is then replaced by the new population.

- Stopping criteria:
  Termination is the criterion by which BBP decides whether to continue or stop the searching process. Different types of stopping criteria may be considered, for example generation number, time period, fitness threshold, the number of node evaluations, etc. For fair comparison of the two developed BBP and ABCP models, the number of node evaluations considering all nodes in the tree structures of all solutions in the generated populations were considered as stopping criteria in this study.

4. Experimental database

4.1. Data description and preparation

The efficiency of the prediction models depends on the quality of the gathered dataset. A final dataset of 413 samples was obtained from a comprehensive search of various sources in the literature. To test the reliability of the developed models, 82 samples were randomly selected as the test set, while the remaining 331 samples were used to train the models. The SCC database covers a wide range of pozzolanic materials including silica fume, fly ash and slag. . The frequency histograms of input and output records for training and testing data are shown in Fig. 7, indicating that both the training and testing datasets cover the range of compressive strength and elastic modulus of SCC.

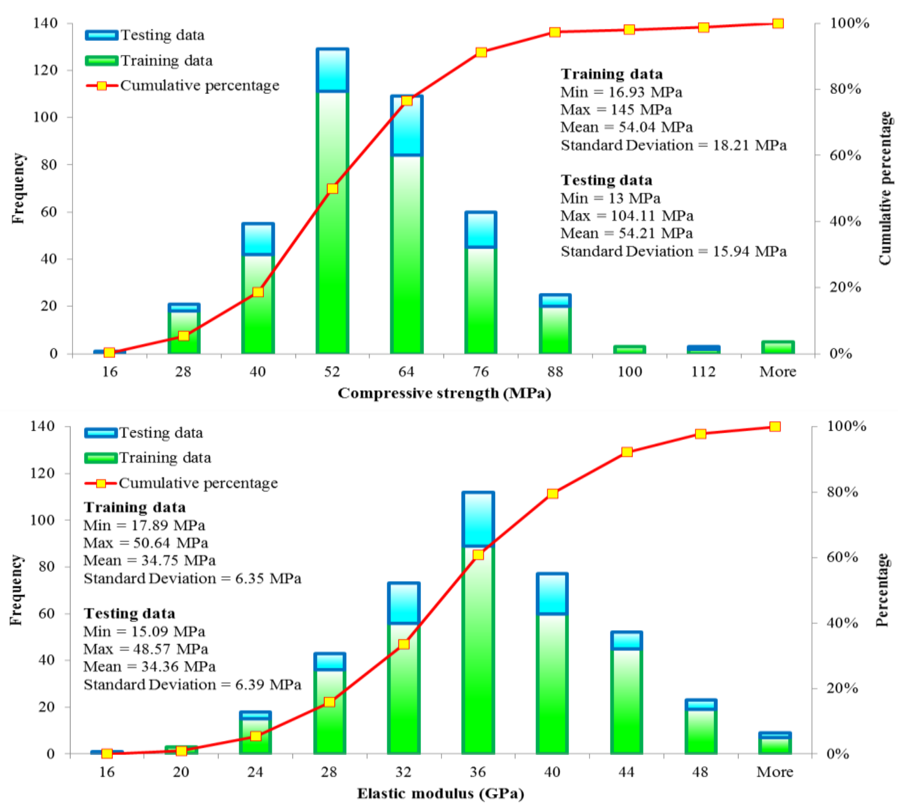


Fig. 7. Frequency histograms of input and output records for training and testing data.

4.2. Development of ABCP and BBP models

In this section, the development stages of the proposed BBP and ABCP algorithms for predicting the elastic modulus of SCC are explained. Because most of empirical models and design codes have related the elastic modulus of concrete to its compressive strength, it is also proposed in the current models to relate the elastic modulus of SCC with its compressive strength. Both of the developed models were coded in Matlab environment. In order to have a reliable and fair comparison, the same parameter values are chosen for both BBP and ABCP models. Problem specific parameters of ABCP and BBP algorithms are given in Table 2. As shown in this Table, a measure based on the number of function node evaluations to estimate the computational cost of both BBP and ABCP runs was adopted. By using the number of node evaluations, the additional computational effort of calculations used in both BBP and ABCP algorithms can be readily estimated. According to previous research investigations [28, 73], the number of node evaluations was set to 15×10^6. The arithmetic operations (+, -, /, ×) and simple mathematical functions (x^2,x^3,$\sqrt{x}$, $\sqrt[3]{x}$) were also used to form the functions and the real input variable f_c' and the constant between [-10,10] with selection probability of 0.2 were used as terminals.

Table 2
Parameters of BBP and ABCP models.

Parameters	Value
Functions	+, -, ×, /, x^2, x^3, $\sqrt[2]{\ }$, $\sqrt[3]{\ }$
Variables	f_c'
Number of node evaluations	15×10^6
Constants	A random number between -10 and 10
Probability of constant selection	0.2
Cost function	Root mean squared error
Number of independent runs	10 independent runs

The control parameters used for the two developed models are given in Table 3. Three levels were considered for population size, namely 100, 300 and 500 habitats for BBP and 100, 300 and 500 employed and onlooker bees for ABCP. These three levels are to be adjusted before running of each experiment. The selection of a high depth tree may complicate the produced formula whereas a tree with low depth may result in an inefficient consequence. In this regard, three levels for the maximum depth of trees including 6, 8 and 10 were considered. In addition, eight migration curves, presented in Table 1, were investigated for BBP. Furthermore, three levels of limit values i.e. (0.5, 1, 2)×colony size were considered for ABCP. In addition, all experiments were independently conducted 10 times, and the average, best and standard deviation values of repetitions were recorded for comparison purposes. In total, 720 and 270 experiments were designed and implemented in Matlab environment for BBP and ABCP, respectively.

Table 3
Control parameters of BBP and ABCP models

BBP		ABCP	
Parameter	Value	Parameter	Value
Habitat size	100, 300, 500	Colony size	100, 300, 500
The maximum tree depth	6, 8, 10	The maximum tree depth	6, 8, 10
Migration curves	Curves 1, ..., Curves 8	Limit	(0.5,1,2) × Colony size

5. Results and discussion

The average of the best RMSE, mean RMSE and standard deviation values of different BBP and ABCP models with various parameters are given in Figs. 8 and 9, respectively. As illustrated in these figures, with increasing the habitat size, the average of the best RMSE, mean RMSEs and standard deviation values were decreased for training data. Therefore, increasing the habitat size could improve the learning capability of BBP. The average of mean RMSE and standard deviation values were also increased with increasing the habitat size for testing data. However, increasing the habitat size does not significantly affect the average of the best RMSE values for testing data.

In the case of ABCP, increasing the colony size decreases the average of mean RMSE and standard deviation values for both training and testing data. In addition, the average of the best RMSE values increases, when the colony size increases for both training and testing data.

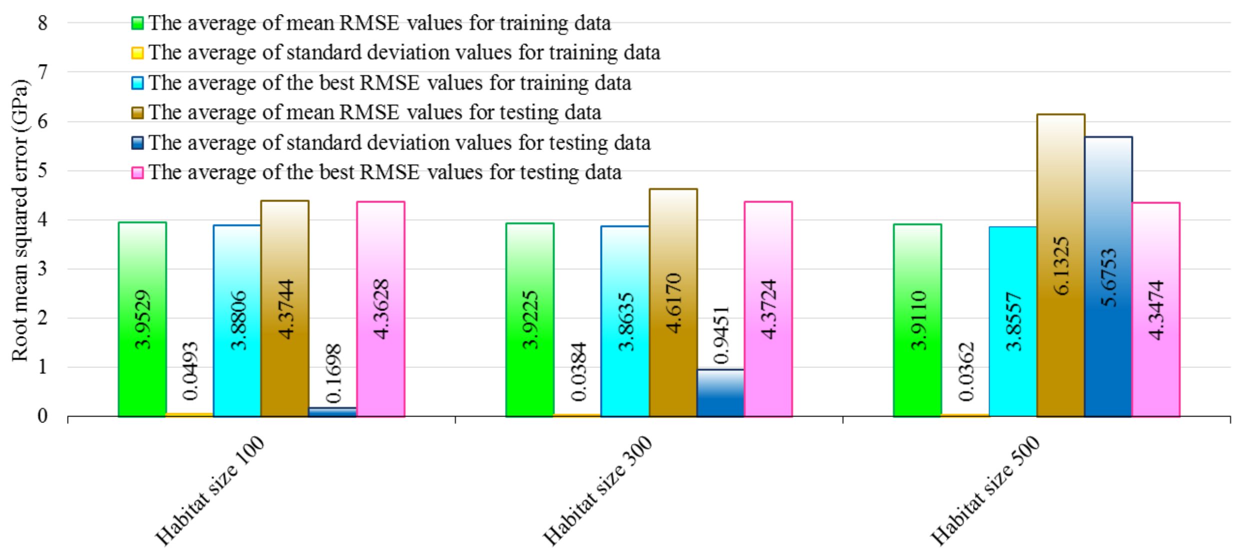


Fig. 8. The results of developed BBP models with various habitat sizes.

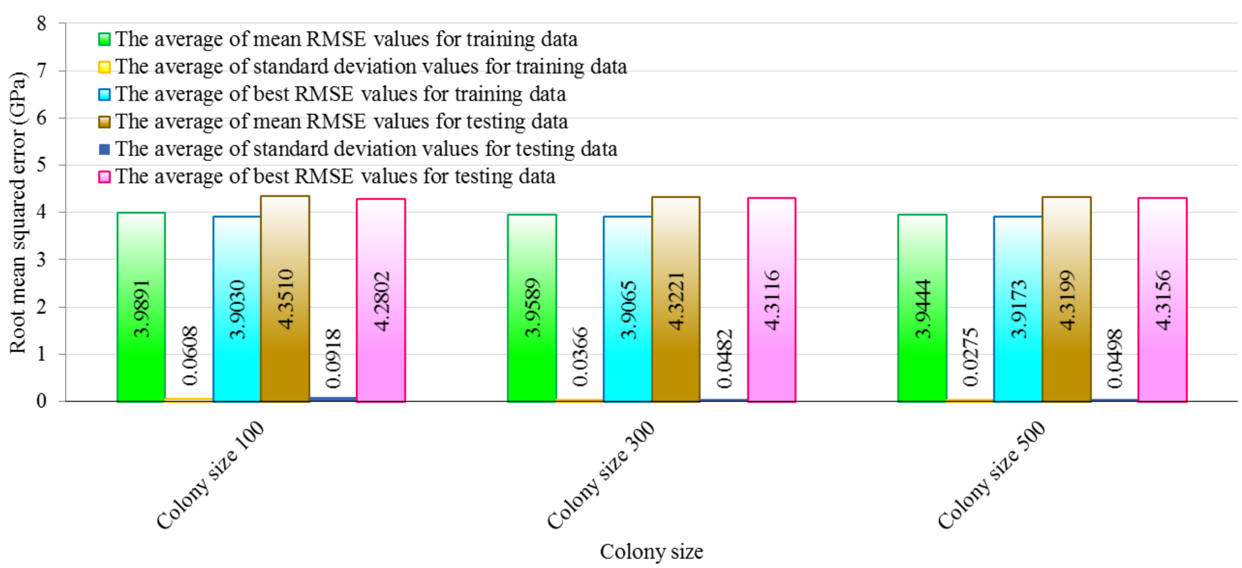


Fig. 9. The results of developed ABCP models with various colony sizes.

The effect of maximum tree depth on the results of BBP and ABCP models are shown in Figs. 10 and 11, respectively. For BBP and ABCP models, the results of high maximum tree depth are more accurate for both training and testing datasets. The diversity of results for high tree depth models is also less than low tree depth model for both BBP and ABCP models.

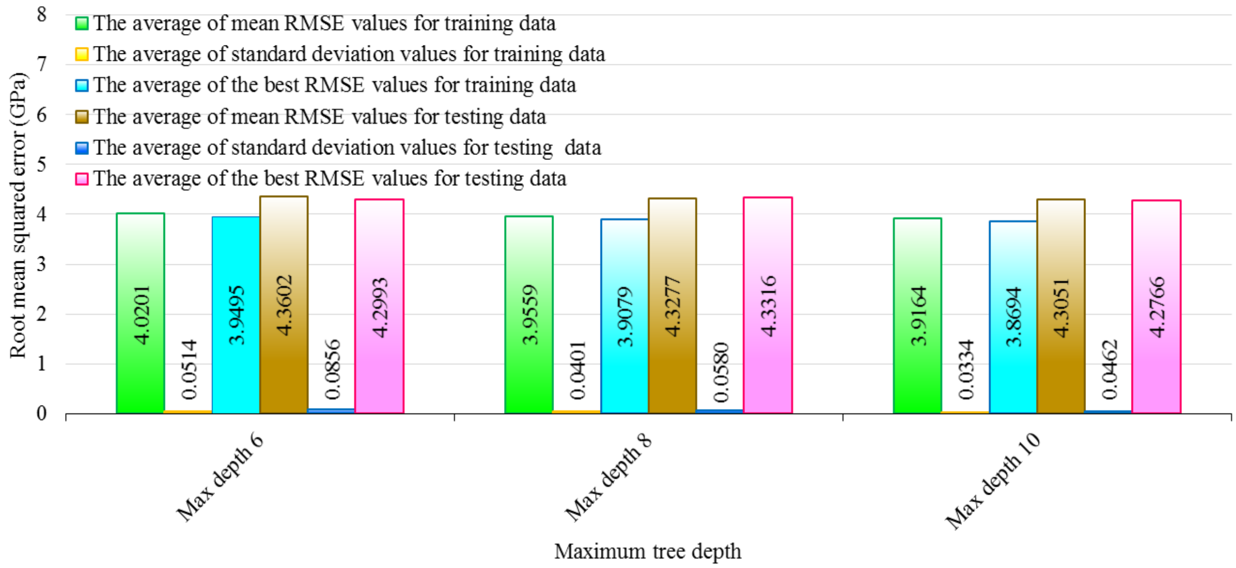


Fig. 10. The results of developed BBP models with various maximum tree depths.

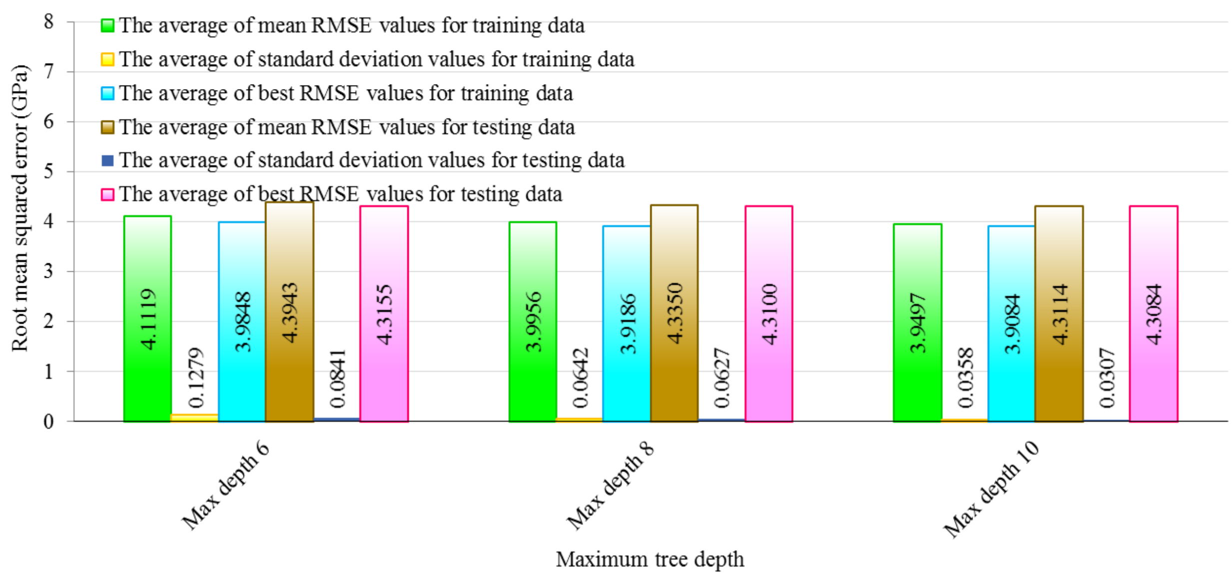


Fig. 11. The results of developed ABCP models with various maximum tree depths.

Fig. 12 shows the effect of migration curves on the results of BBP. As illustrated in this figure, migration models 7 and 8 have the best performance among all other considered migration models for training and testing data, respectively. In addition, the migration models 2 and 1 are not suitable for training and testing data, respectively.

Fig. 12. The effect of migration curves on BBP results.

Another parameter considered in this section is the effect of “limit” parameter value on the performance of ABCP model. The results of investigation of this parameter are presented in Fig. 13. As shown in this figure, varying the "limit" parameter value does not significantly affect the results of ABCP model in predicting the elastic modulus of SCC. However, considering the "limit" value equal to (2×colony size) is slightly better than the two other considered "limit" values.

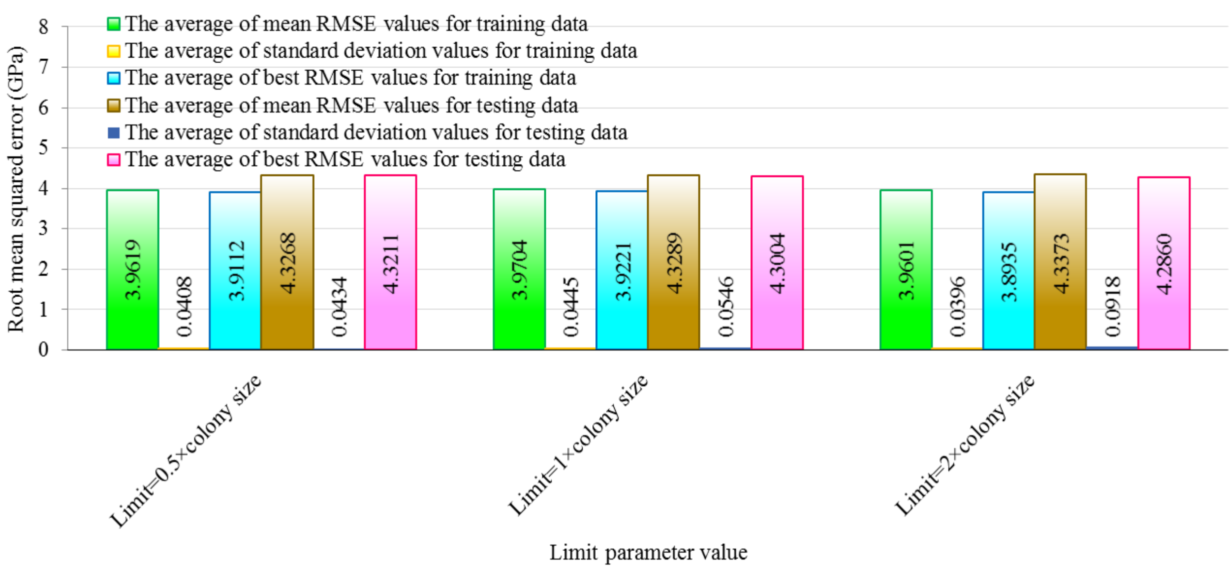


Fig. 13. The effect of "limit" parameter value on ABCP results.

Regardless of the mentioned average values, the formulations of the best BBP and ABCP models among 990 total runs are shown in Table 4, indicating that the proposed BBP model has a simpler formula than the best proposed ABCP model. Furthermore, both developed models have mainly related the elastic modulus of SCC to the square root of compressive strength. The expression tree of the proposed BBP model is shown in Appendix A.

Table 4
The final BBP and ABCP equations for predicting the elastic modulus of SCC (SI units, MPa).

BBP model
$E_c = 4906.45\sqrt{\acute{f}_c} - 0.004\acute{f}_c^{\,3} + \dfrac{366.68\acute{f}_c}{(125.89 - 2\acute{f}_c)(130.94 - \acute{f}_c - 4.91\sqrt{\acute{f}_c})}$
ABCP model
$E_c = 4758.36\sqrt{\acute{f}_c} + 128.78\sqrt{\lvert 2\acute{f}_c - 108.02\rvert(\acute{f}_c + \dfrac{\acute{f}_c + 7.92}{-5.15 + \sqrt[3]{\acute{f}_c}})} - \dfrac{(9.95\sqrt{\acute{f}_c + 8.05} - 77.23)(3.27\acute{f}_c + \sqrt{\acute{f}_c} - 28.44)}{2.16 + \dfrac{\sqrt[3]{\acute{f}_c}}{92.78 - \acute{f}_c}}$

For comparison purposes, the results of the best BBP and ABCP models were compared with several empirical and design codes' equations given in Table 5. In this regard, statistical parameters including root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE) and absolute fraction of variance (R^2) were used. In order to consider the simultaneous effect of different statistical parameters, another parameter, namely OBJ, were also calculated [74]. The above mentioned parameters are defined below:

$$\mathrm{MAE} = \frac{1}{P}\sum_{i=1}^{P}|O_i - t_i| \tag{7}$$

$$\mathrm{MAPE} = \frac{1}{P}\sum_{i=1}^{P}\frac{|O_i - t_i|}{t_i} \times 100 \tag{8}$$

$$R^2 = 1 - \left(\frac{\sum_{i=1}^{P}(O_i - t_i)^2}{\sum_{i=1}^{P}(O_i)^2}\right) \tag{9}$$

$$\mathrm{OBJ} = \left(\frac{No._{Train} - No._{Test}}{No._{Train} + No._{Test}}\right)\frac{RMSE_{Train} + MAE_{Train}}{R_{Train} + 1} + \frac{2No._{Test}}{No._{Train} + No._{Test}} \times \frac{RMSE_{Test} + MAE_{Test}}{R_{Test} + 1} \tag{10}$$

where P, O_i, t_i, $No._{Train}$ and $No._{Test}$ are the total number of data points in each set of data, the predicted elastic modulus obtained from each model, the experimental elastic modulus of SCC of *i*th pattern in database, the number of training and testing data, respectively. The statistical parameters of different models for predicting the elastic modulus of SCC are given in Table 6. As shown in this table, the two proposed symbolic regression models, namely BBP and ABCP, have the best predictions among all other models considered. Among the empirical equations and design codes' relationships, Leemann and Hoffmann, ACI 318-08 and EC-2 equations show very close predictions to experimental results. Moreover, Persson and Felekoglu et al.'s equations give the worst predictions for the elastic modulus of SCC.

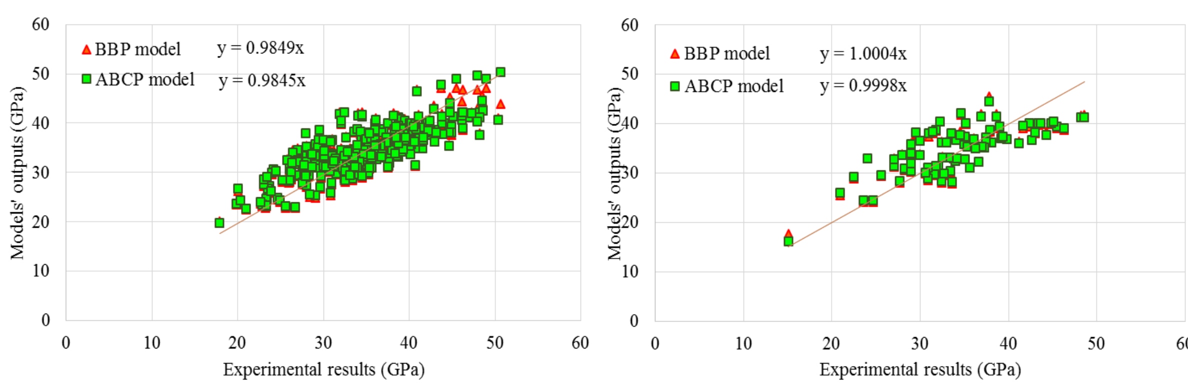
Table 5
Empirical equations and design codes' relationships for estimating the elastic modulus of SCC.

References	Estimating models of E_c for SCC (SI units, MPa)
ACI 318-08 [15]	$E_c = 4700(\acute{f}_c)^{0.5}$
EC-2 [16]	$E_c = 22000(\acute{f}_c/10)^{0.3}$
NZS 3101 [17]	$E_c = 3320(\acute{f}_c)^{0.5} + 6900$
CSA A23.3-04 [18]	$E_c = 4500(\acute{f}_c)^{0.5}$
Persson [13]	$E_c = 3750(\acute{f}_c)^{0.5}$
Leemann and Hoffmann [19]	$E_c = 4740(\acute{f}_c)^{0.5}$
Felekoglu et al. [20]	$E_c = 1570(\acute{f}_c)^{0.8}$
Dinakar et al. [21]	$E_c = 4180(\acute{f}_c)^{0.5}$

Table 6
The statistical parameters values of different models.

Models	MAE		MAPE		RMSE		R^2		OBJ
	Training data	Testing data	Training data	Testing data	Training data	Testing data	Training data	Testing data	All data
BBP	3.1308	3.4988	9.3646	10.5636	3.8246	4.2391	0.9881	0.9855	3.6450
ABCP	3.1183	3.4904	9.3867	10.6160	3.8393	4.2564	0.9882	0.9856	3.6473
ACI 318-08 [15]	3.3592	3.6548	9.7186	10.7992	4.1326	4.3518	0.9857	0.9842	3.8626
EC-2 [16]	3.6003	3.9533	11.3966	12.7131	4.4262	4.8226	0.9851	0.9824	4.1788
NZS 3101 [17]	4.6347	4.5177	12.5711	12.4962	5.5940	5.5132	0.9679	0.9689	5.1159
CSA A23.3-04 [18]	3.7862	3.9728	10.6059	11.3253	4.5622	4.6146	0.9810	0.9806	4.2421
Persson [13]	7.5965	7.1252	21.0185	19.6997	8.5630	8.2919	0.9035	0.9098	8.1282
Leemann & Hoffmann [19]	3.3246	3.6234	9.6979	10.7870	4.1049	4.3574	0.9861	0.9844	3.8383
Felekoglu et al. [20]	5.1237	5.2942	14.8223	16.0640	7.3701	7.2321	0.9646	0.9658	6.3089
Dinakar et al. [21]	5.0518	4.8957	13.8384	13.4714	5.9818	5.8383	0.9621	0.9640	5.5090

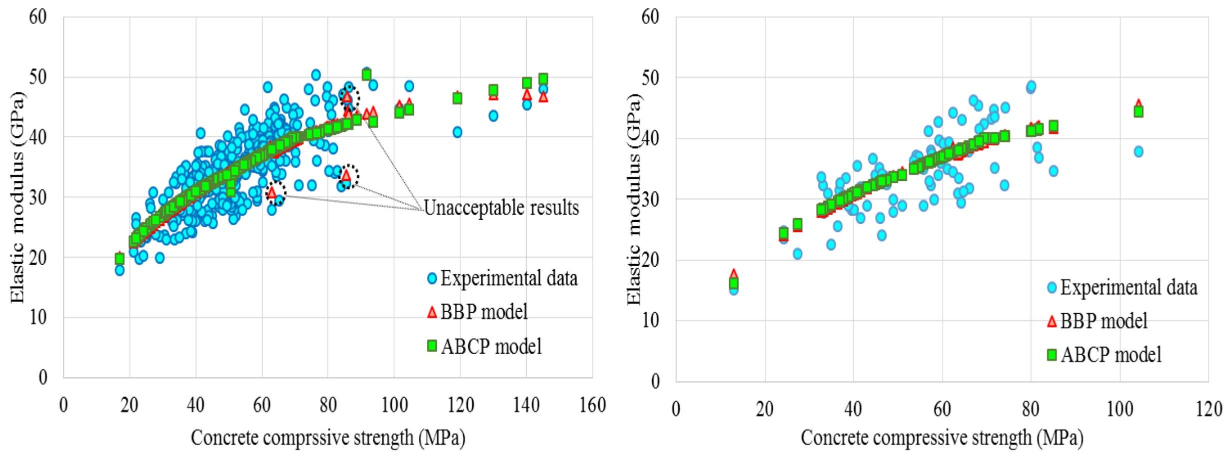
The comparisons between the experimental and predicted elastic modulus of SCC using the proposed BBP and ABCP are illustrated in Fig. 14. As shown in this figure, the training set results prove that the proposed models have clearly learned the non-linear relationship between the input and output variables with good correlation. Comparing the BBP and ABCP models' predictions with the experimental results for the testing dataset demonstrates a high generalization capability of the proposed models.



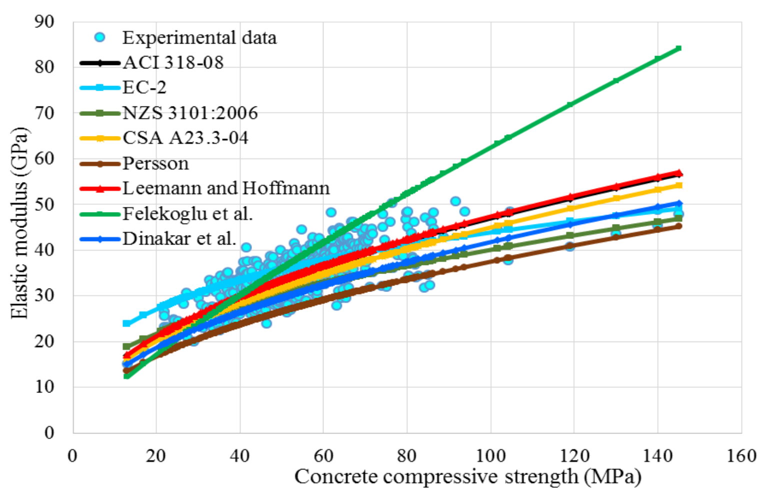
(a) Training dataset (b) Testing dataset

Fig. 14. Comparison of actual and predicted elastic modulus of SCC for training and testing data.

In Fig. 15, the trend of the elastic modulus of SCC predicted by BBP and ABCP, and other existing models is plotted against the compressive strength. The figure also includes the experimental results for all data in the database. Figure 15 indicates that the predictions of BBP and ABCP models are compliance with each other in most cases. Furthermore, the trend of SCC elastic modulus obtained by BBP and ABCP is similar to that by most existing formula but closer to the experimental results. Moreover, the prediction of Felekoglu et al.'s equation is significantly different from the rest of the models and considerably larger than experiments, especially for compressive strength of SCC greater than 80MPa. Furthermore, most existing formulas, especially Persson's equation, underestimate the elastic modulus of SCC.



(a) Training dataset (b) Testing dataset



(c) All dataset

Fig. 15. Relationship between the elastic modulus and compressive strength of SCC for different models.

6. Conclusions

In this paper, biogeographical-based programming (BBP) was employed to predict the elastic modulus of SCC. In order to compare the results of the proposed BBP model with other symbolic regression programming models, a relatively new method, namely artificial bee colony programming (ABCP), was chosen. These two models were coded in Matlab environment and applied for predicting the elastic modulus of SCC. The results showed that the proposed BBP model has a simplified formula and slightly better performance compared to ABCP model and all other empirical and design codes' equations. The results of the BBP parametric study indicated that increasing the habitat size and the maximum tree depth could improve the performance of the BBP model. Furthermore, sinusoidal immigration and biquadratic emigration model and sinusoidal immigration and sixteenth degree migration model have the best performance whereas constant immigration and linear emigration model and linear immigration and constant emigration model have the worst performance among all considered migration curves. On the other hand, the results of the ABCP parametric study show that increasing the colony size and maximum tree depth decreases the error value of ABCP. In addition, varying the value of “limit” parameter from half of colony size to twice of colony size does not significantly affect on the results of ABCP model. However, considering the "limit" value equal to (2×colony size) is slightly better. In addition, among all considered empirical and design codes’ equations, Leemann and Hoffmann and ACI 318-08’s equations have suitable performances and Persson and Felekoglu et al.’s equations are not recommended for predicting the elastic modulus of SCC.

The following are recommended for future study. Firstly, the overall dataset may be divided into three subsets with the additional dataset used for validation or n-fold cross validation is used to minimize chances of model overfitting. Secondly, inspired by evolutionary computations, variants of symbolic regression and hybrid models can be developed and their results can be compared with the developed BBP and ABCP models. Thirdly, other engineering applications can be modeled using the proposed symbolic regression BBP model.

7. References

[1] K. Ozawa, K. Maekawa, M. Kunishima, H. Okamura, Development of high performance concrete based on the durability design of concrete structures, in: Proceedings of the 2nd East-Asia and Pacific conference on structural engineering and construction, EASEC-2, 1989, pp. 445-450.

[2] H. Okamura, K. Maekawa, K. Ozawa, High performance concrete, Gihido, Tokyo, 1993.

[3] K. Maekawa, K. Ozawa, Development of SCC’s prototype, Self-compacting high performance concrete, Social System Institute, 1999, pp. 20-32.

[4] T.K. Erdem, K.H. Khayat, A. Yahia, Correlating rheology of self-consolidating concrete to corresponding concrete-equivalent mortar, ACI. Mater. J. 106(2) (2009) 154-160.

[5] R. Gaimster, N. Dixon, Self-compacting concrete, in: Newman, J., & Choo, B.S. editors. Advanced concrete technology set, Oxford, Butterworth-Heinemann, 2003, pp. 1-23.

[6] H.S. Najm, Specialized construction applications, in: Nawy, E.G. editor. Concrete construction engineering handbook, Boca Raton, CRC Press, 2008, pp. 1-26.

[7] A. Skarendal, Changing concrete construction through use of self-compacting concrete. In: Proceedings PRO 42-1st international symposium on design, performance and use of self-consolidating concrete SCC’2005, China, 2005, pp. 17-24.

[8] A. Beycioglu, H.Y. Aruntas, Workability and mechanical properties of self-compacting concretes containing LLFA, GBFS and MC, Constr. Build. Mater. 73 (2014) 626-635.

[9] W.R.L. Da-Silva, P. Stemberk, Expert system applied for classifying self-compacting concrete surface finish, Adv. Eng. Softw. 64 (2013) 47-61.

[10] H. Okamura, M. Ouchi, Self-compacting concrete, J. Adv. Concr. Technol. 1(1) (2003) 5-15.

[11] P. Gardoni, D. Trejo, M. Vannucci, C. Bhattacharjee, Probabilistic Models for Modulus of Elasticity Probabilistic Models for Modulus of Elasticity, J. Eng. Mech. 135 (2009) 295-306.

[12] N. Su, K.C. Hsu, H.W. Chai, A simple mix design method for self-compacting concrete, Cem. Concr. Res. 31(12) (2001) 1799-1807.

[13] B. Persson, A comparison between mechanical properties of self-compacting concrete and the corresponding properties of normal concrete, Cem. Concr. Res. 31(2) (2001) 193-198.
[14] A.A.A. Hassan, K.M.A. Hossain, M. Lachemi, Behavior of full-scale self-consolidating concrete beams in shear, Cem. Concr. Res. 30(7) (2008) 588-596.
[15] American Concrete Institute, ACI building code requirements for reinforced concrete, ACI 318-08 (2008).
[16] Eurocode 2, Design of Concrete Structures-Part 1-1: General Rules and Rules for Buildings. EC2-04 (2004).
[17] NZS 3101, The Design of Concrete Structures, New Zealand Standard, Wellington, 2006.
[18] CSA 2004, Design of Concrete Structures A23-2004, Canadian Standards Association, Rexdale, Ontario, 2004.
[19] A. Leemann, C. Hoffmann, Properties of self-compacting and conventional concrete differences and similarities, Mag. Concr. Res. 57(6) (2005) 315-319.
[20] B. Felekoglu, S. Turkel, B. Baradan, Effect of water/cement ratio on the fresh and hardened properties of self-compacting concrete, Build. Environ. 42(4) (2007) 1795-1802.
[21] P. Dinakar, K.G. Babu, M. Santhanam, Mechanical properties of high-volume fly ash self-compacting concrete mixtures, Struct. Concr. 9(2) (2008) 109-116.
[22] F. Demir, A new way of prediction elastic modulus of normal and high strength concrete-fuzzy logic, Cem. Concr. Res. 35 (2005) 1531-1538.
[23] F. Demir, Prediction of elastic modulus of normal and high strength concrete by artificial neural networks, Constr. Build. Mater. 22 (2008) 1428-1435.
[24] F. Demir, K.A. Korkmaz, Prediction of lower and upper bounds of elastic modulus of high strength concrete, Constr. Build. Mater. 22 (2008) 1385-1393.
[25] K. Yan, C. Shi, Prediction of elastic modulus of normal and high strength concrete by support vector machine, Constr. Build. Mater. 24 (2010) 1479-1485.
[26] B. Ahmadi-Nedushan, Prediction of elastic modulus of normal and high strength concrete using ANFIS and optimal nonlinear regression models, Constr. Build. Mater. 36 (2012) 665-673.
[27] M.D. Schmidt, H. Lipson, Co-evolving fitness predictors for accelerating and reducing evaluations, in: Genetic Programming Theory and Practice IV, Genetic and Evolutionary Computation, vol. 5, Springer, 2006.
[28] D. Karaboga, C. Ozturk, N. Karaboga, B. Gorkemli, Artificial bee colony programming for symbolic regression, Inf. Sci. 209 (2012) 1-15.
[29] J. Kleijnen, R. Sargent, A methodology for fitting and validating meta models in simulation, Eur. J. Oper. Res. 120 (1) (2000) 14-29.
[30] B. Akay, D. Karaboga, A modified Artificial Bee Colony algorithm for real-parameter optimization, Inf. Sci. 192 (1) (2012) 120-142.
[31] A. Auger, N. Hansen, A restart CMA evolution strategy with increasing population size, in: The 2005 IEEE Congress on Evolutionary Computation, Edinburgh, UK, September, 2005, pp. 1769-1776.
[32] S. Gustafson, E. Burke, The speciating island model: an alternative parallel evolutionary algorithm, J. Parallel. Distr. Com. 66 (8) (2006) 1025-1036.
[33] F. Neri, V. Tirronen, Scale factor local search in differential evolution, Memetic. Computation. 1(2) (2009) 153-171.
[34] M.A.M. Oca, T. Stutzle, M. Birattari, M. Dorigo, Frankenstein's PSO: a composite particle swarm optimization algorithm, IEEE Trans. Evol. Compu. 13(5) (2009) 1120-1132.
[35] X. Yang, S. Deb, Engineering optimisation by cuckoo search, Int. J. Math. Model. Num. Optim. 1(4) (2010) 330-343.
[36] H. Ma, D. Simon, M. Fei, X. Shu, Z. Chen, Hybrid biogeography-based evolutionary algorithms, Eng. Appl. Artif. Intell. 30 (2014) 213-224.
[37] J.R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, Cambridge, MA: MIT Press, 1992.
[38] P. Holmes, P.J. Barclay, Functional languages on linear chromosomes, In: Genetic Programming 1996: Proceedings of the First Annual Conference, 1996, p. 427.
[39] J.F. Miller, P. Thomson, Cartesian genetic programming, in: Proceedings of the European Conference on Genetic Programming, Springer-Verlag, London, UK, 2000, pp. 121-132.
[40] C. Ferreira, Gene expression programming: A new adaptive algorithm for solving problems, Complex. Syst. 13 (2001) 87-129.
[41] P. Musilek, A. Lau, M. Reformat, L. Wyard-Scott, Immune programming, Inf. Sci. 176 (2006) 972-1002.
[42] S. Shirakawa, S. Ogino, T. Nagao, Dynamic ant programming for automatic construction of programs, IEEJ. T. Electr. Electr. 3(5) (2008) 540-548.

[43] Z. Gan, T.W.S. Chow, W.N. Chau, Clone selection programming and its application to symbolic regression, Expert. Syst. Appl. 36(2-2) (2009) 3996-4005.
[44] D. Simon, Biogeography-based optimization, IEEE Trans. Evol. Comput. 12 (2008) 702-713.
[45] A. Bhattacharya, P. Chattopadhyay, Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch, IEEE Trans. Power. Syst. 25(4) (2010) 1955-1964.
[46] A. Bhattacharya, P. Chattopadhyay, Solving complex economic load dispatch problems using biogeography-based optimization, Expert. Syst. Appl. 37(5) (2010) 3605-3615.
[47] I. Boussaid, A. Chatterjee, P. Siarry, M. Ahmed-Nacer, Biogeography-based optimization for constrained optimization problems, Comput. Oper. Res. 39 (2012) 3293-3304.
[48] I. Boussaïd, A. Chatterjee, P. Siarry, M. Ahmed-Nacer, A comparative study of modified BBO variants and other metaheuristics for optimal power allocation in wireless sensor networks, in: A. Chatterjee, H. Nobahari, P. Siarry (Eds.), Advances in Heuristic Signal Processing and Applications, 2013, pp. 79-110.
[49] A. Chatterjee, P. Siarry, A. Nakib, R. Blanc, An improved biogeography based optimization approach for segmentation of human head CT-scanimages employing fuzzy entropy, Eng. Appl. Artif. Intel. 25 (2012) 1698-1709.
[50] B. Gardner, D. Simon, Evolutionary algorithm sandbox: A web-based graphical user interface for evolutionary algorithms, in: Proceedings of the IEEE Conference on Systems, Man, and Cybernetics, San Antonio, Texas, 2009, pp. 583-588.
[51] A. Hadidi, A robust approach for optimal design of plate fin heat exchangers using biogeography based optimization (BBO) algorithm, Appl. Energy. 150 (2015) 196-210.
[52] K. Jamuna, K. Swarup, Multi-objective biogeography based optimization for optimal PMU placement, Appl. Soft. Comput. 12(5) (2012)1503-1510.
[53] S.S. Kim, J.H. Byeon, H. Yu, H. Liu, Biogeography-based optimization for optimal job scheduling in cloud computing, Appl. Math. Comput. 247 (2014) 266-280.
[54] H. Kundra, A. Kaur, V. Panchal, An integrated approach to biogeography based optimization with case based reasoning for retrieving groundwater possibility, in: Proceedings of the 8th Annual Asian Conference and Exhibition on Geospatial Information, Technology and Applications, Singapore, August, 2009.
[55] M. Ovreiu, D. Simon, Biogeography-based optimization of neuro-fuzzy system parameters for diagnosis of cardiac disease, in: Proceedings of GECCO- 2010, Portland, Oregon, July, 2010.
[56] V. Panchal, P. Singh, N. Kaur, H. Kundra, Biogeography based satellite image classification, Int. J. Comput. Sci. Inf. Secyr. 6 (2009) 269-274.
[57] S. Rahmati, M. Zandieh, A new biogeography-based optimization (BBO) algorithm for the flexible job shop scheduling problem, Int. J. Adv. Manuf. Tech. 58(9) (2012) 1115-1129.
[58] A. Ramesh-Kumar, L. Premalatha, Optimal power flow for a deregulated power system using adaptive real coded biogeography-based optimization, Int. J. Elec. Power 73 (2015) 393-399.
[59] R. Rarick, D. Simon, F.E. Villaseca, B. Vyakaranam, Biogeography-based optimization and the solution of the power flow problem, in: Proceedings of the IEEE Conference on Systems, Man, and Cybernetics, San Antonio, Texas, 2009, pp. 1029-1034.
[60] P. Roy, S. Ghoshal, S. Thakur, Biogeography-based optimization for economic load dispatch problems, Elec. Pow. Compos. Sys. 38 (2010) 166-181.
[61] V. Savsani, R. Rao, D. Vakharia, Discrete optimisation of a gear train using biogeography based optimisation technique, Int. J. Des. Eng. 2 (2009) 205-223.
[62] U. Singh, H. Singla, T. Kamal, Design of Yagi-Uda antenna using biogeography based optimization, IEEE Trans. Ant. Propag. 58(10) (2010) 3375-3379.
[63] X. Wang, H. Duan, A hybrid biogeography-based optimization algorithm for job shop scheduling problem, Comput. Ind. Eng. 73 (2014) 96-114.
[64] L. Wang, Y. Xu, An effective hybrid biogeography-based optimization algorithm for parameter estimation of chaotic systems, Expert. Syst. Appl. 38(12) (2011) 15103-15109.
[65] D. Karaboga, An idea based on honey bee swarm for numerical optimization, Technical report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
[66] W.F. Gao, S.Y. Liu, A modified artificial bee colony algorithm, Comput. Oper. Res. 39 (2012) 687-697.
[67] D. Karaboga, B. Gorkemli, A quick artificial bee colony (qABC) algorithm and its performance on optimization problems, Appl. Soft. Comput. 23 (2014) 227-238.

[68] W. Gao, S. Liu, L. Huang, A global best artificial bee colony algorithm for global optimization, J. Comput. Appl. Math. 236 (2012) 2741-2753.
[69] I. Boussaid, A. Chatterjee, P. Siarry, M. Ahmed-Nacer, Two-stage update biogeography based optimization using differential evolution algorithm (DBBO), Comput. Oper. Res. 38(8) (2011) 1188-1198.
[70] H. Ma, D. Simon, M. Fei, Z. Xie, Variations of biogeography-based optimization and Markov analysis. Inf. Sci. 220 (2013) 492-506.
[71] H. Ma, An analysis of the equilibrium of migration models for biogeography-based optimization, Inf. Sci. 176(8) (2010) 3444-3464.
[72] W. Gua, L. Wang, Q. Wu, An analysis of the migration rates for biogeography-based optimization, Inf. Sci. 254 (2014) 111-140.
[73] Uy, N.Q., Hoai, N.X., O'Neill, M., McKay, R.I., Galvan-Lopez, E., 2011. Semantically-based crossover in genetic programming: application to real-valued symbolic regression. Genet. Program. Evol. M. 12, 91-119.
[74] Gandomi, A.H., Yun, G.J., Alavi, A.H., 2013. An evolutionary approach for modeling of shear strength of RC deep beams. Mater. Struct. 46(12), 2109-2119.

Appendix A. Expression tree of the proposed BBP model.