Neural Comput & Applic
DOI 10.1007/s00521-016-2345-1

ORIGINAL ARTICLE

Performance evaluation of nature-inspired algorithms for the design of bored pile foundation by artificial neural networks

Gurdeepak Singh[1] · **B. S. Walia**[1]

Received: 4 September 2015 / Accepted: 9 May 2016

Abstract The bored pile foundations are gaining popularity in construction industry because of ease in construction, low noise and vibrations. The load-carrying capacity of bored pile foundations is dependent upon soil–structure interaction. This being a three-dimensional problem is further complicated due to large variations in soil properties. Also, modeling of soil is difficult because of its nonlinear and anisotropic nature. For such cases, the artificial neural network (ANN) and nature-inspired optimization techniques have been found to be highly suitable to attain acceptable levels of accuracy. In the present study, two ANNs have been trained for determination of unit skin friction and unit end bearing capacity from soil properties. The training data for ANNs have been obtained from finite element analysis of pile foundations for 4809 different soil types. A dataset of 50 field pile loading test results is used to check the performance of the developed artificial neural networks. To enhance the accuracy of the developed ANNs, two correlation factors have been determined by applying four popular nature-inspired optimization algorithms: particle swarm optimization (PSO), fire flies, cuckoo search and bacterial foraging. In order to rank these optimization algorithms, parametric and non-parametric statistical analysis has been carried out. The results of optimization algorithms have been compared to find the most suitable solution for this multi-dimensional problem which has a large number of nonlinear equality constraints. The effectiveness and suitability of the nature-inspired algorithms for the presented problem have been demonstrated by computing correlation coefficients with field pile loading test results and then with the total execution time taken by each algorithm. The results of comparison show that PSO is the best performer for such constrained problems.

Keywords Particle swarm optimization · Firefly algorithm · Cuckoo search · Bacterial foraging · Bored pile foundation

1 Introduction

Solving real-life optimization problems using conventional methods requires substantial effort and time as the solution lies in the precise quantities associated with the variables which have nonlinear relationships. The variables also have equality and non-equality constraints. The presence of large number of variables and constraints generates an exceptionally wide search space. Classical optimization techniques such as exhaustive search have been found to be inefficient for solving such type of problems. Approximate algorithms based upon the concept of evolution and swarm intelligence have been found to be more efficient in solving high-dimensional problems characterized by the presence of large number of variables and constraints. These algorithms have been found to provide high speedups, easier to implement and provide near optimal solutions by augmenting the current best solution using the principle of randomization. The approximate algorithms are being widely used in multifarious fields of science including computational intelligence, artificial intelligence and soft computing. Many of approximate algorithms are inspired by various phenomena which occur in nature. genetic algorithm (GA), particle swarm optimization (PSO),

✉ Gurdeepak Singh
klergs@yahoo.com

1 Guru Nanak Dev Engineering College, I.K.G. Punjab Technical University (Kapurthala), Ludhiana 141006, India

Published online: 20 May 2016

artificial bee colony (ABC), firefly (FF) algorithm, ant colony optimization (ACO), cuckoo search (CS), gravitational search algorithm (GSA), bacterial foraging (BF), etc. are popular nature-inspired optimization algorithms. These algorithms are being used in the area of civil engineering. Cui and Sheng [1] used genetic algorithm and finite element displacement method to compute reliability index. They found that genetic algorithm relatively took less computation time. Liu et al. [2] used automatic grouping genetic algorithm for optimizing the pile group design. Liu et al. [3] proposed genetic algorithm for determining load-carrying capacity of composite foundation. Elbeltagia et al. [4] compared PSO with genetic algorithm and other evolutionary techniques and found the PSO algorithm to be more efficient and easy to implement. ANNs have been used to compute the load-carrying capacity of driven piles by a number of researchers [4–6]. Ismail et al. [7] used ANN for load deformation analysis of pile foundation. The ANN was trained first by PSO and then by back propagation by [8, 9]. Ismail et al. [10] analyzed load deformation of pile foundation by PSO-BP hybrid.

The input parameters considered by various researchers for the design of pile foundation are given in Table 1. The input parameters used for the study were effective angle of shearing resistance, effective cohesion intercept, modulus of deformation, Poisson's ratio, unit weight and depth of soil layer. These input parameters were the same as required by the finite element method for the determination of unit skin friction resistance and unit tip bearing capacity of the pile foundation. The ANNs were developed using the input and target data obtained from the finite element method. These input/output data are free from noise which generally characterizes the field observations due to heterogeneous and anisotropic behavior of the soil. The manual and instrumental errors also add to the noise in the field observations. The ANNs so developed in the study reported a high value of coefficient of determination averaging 0.99996. This study differs from the earlier

Table 1 Methodology followed by various researchers for the design of pile foundation

Sr. No.	Researcher	Techniques	Input parameters	Output parameter	Performance
1.	Goh [11, 12]	Artificial neural network	Pile length, mean effective stress, pile diameter and undrained shear strength	Skin friction resistance in clays	Correlation coefficient: In training: 0.985 In testing: 0.956
2.	Goh [13, 14]	Artificial neural network	Hammer weight, Hammer type, Hammer drop and, and pile length, cross-sectional area, weight, modulus of elasticity and pile set	Load capacity of driven piles in cohesionless soils	Correlation coefficient: In training: 0.96 In testing: 0.97
3.	Chan et al. [5]	Artificial neural network	Elastic compression of pile, energy delivered to the pile elastic compression of soil and pile set	Pile load capacity	RMSE: In training: 13.5 % In testing: 12.0 %
4.	Lee and Lee [15]	Artificial neural network (model pile load test results)	Penetration depth, mean normal stress of the calibration chamber, number of blows and penetration ratio	Ultimate bearing capacity of pile	Average summed square error < 15 %
5.	Abu-Kiefa [16]	Three artificial neural network models	Angle of shear resistance along the pile shaft, effective overburden pressure at the pile tip, angle of shear resistance at the pile tip, equivalent pile cross-sectional area and pile length	Pile shaft capacity, tip capacity and load capacity of driven piles in cohesionless soils	Coefficients of determination: 0.95
6.	Shahin and Jaksa [17, 18]	Multi-layer perceptrons artificial neural network and B-spline neurofuzzy network	Anchor diameter, average cone sleeve friction along the anchor length, anchor embedment length, installation technique, average cone tip resistance along the anchor length	Pullout capacity in case of micro-piles	Correlation coefficient: In MLP model: 0.83 In B-spline neurofuzzy model: 0.84
7.	Nejad et al. [19]	Artificial neural network	Applied load, soil properties, embedded length of pile and SPT values	Pile settlement	Correlation coefficient: 0.972 for settlement $\leq$ 185 mm
8.	Alkroosh and Nikraz [20]	Gene expression programming correlation model	SPT data and dynamic input	Pile load capacity	Coefficient of determination: In training: 0.94 In testing: 0.96

studies in that it considers soil–structure interaction also (given by finite element method) along with the field observations.

2 Analysis of pile foundation

The loading of pile foundation results in corresponding settlement. The settlement of a pile foundation is resisted by soil. This resistance is due to development of skin friction all around pile shaft and tip bearing capacity. When the settlement of pile is in the range of 5–10 mm relative to soil, the skin friction gets fully mobilized [21]. The tip bearing capacity varies almost linearly with settlement of pile. The tip bearing capacity reaches a maximum value when the settlement of pile is 10–25 % of the pile diameter [21]. The total settlement of pile foundation is sum of time-independent immediate settlement and time-dependent compaction of soil. Poulos and Davis [22] proved that immediate settlement is the maximum contributing factor in the total settlement of pile foundation. The immediate settlement can be calculated by the methods proposed by Meyerhof [23] and Vesic [24] which are based on semi-empirical approach. The most commonly used methods for pile foundation analysis are load transfer method, elastic method (using equations by Mindlin [25]) and finite element method.

The finite element method (FEM) was developed by [26, 27]. It is widely used in the analysis of soil–structure interaction problems. In this method, the soil mass is divided into finite number of elements. All these elements are connected to each other at their common points which are called nodal points. A set of simultaneous algebraic equations at node is developed using this procedure [28]. The soil is represented as a mesh with soil elements and nodes. The displacements of each soil elements are determined from the displacements of its nodal points. The strain of soil elements is determined from nodal displacements by a displacement function [29]. The level of stress is calculated from strain in the soil element and elastic properties of soil. Once the constitutive relationship of soil is known, the equilibrium equations of soil elements can be solved for the elastic/plastic state and the stress level at each nodal point. Depending upon the magnitude of externally applied load and weight of soil, some portion of the shear strength is mobilized. The maximum value of applied shear stress should be less than the shear strength of soil to prevent failure [30]. The shear strength of soil is the governing factor for load-carrying capacity of pile foundation. In the analysis of pile foundation by FEM, Mohr–Coulomb material model was used for soil which has elastic-perfectly plastic behavior and the pile was modeled as linear elastic material. The value of soil stiffness was assumed as constant. The finite element analysis requires the computation of the following soil parameters.

1. Effective angle of shearing resistance (φ' in °)
2. Effective cohesion intercept (c' in kN/m^2)
3. Modulus of deformation (E in kN/m^2)
4. Poisson's ratio (μ dimensionless)
5. Unit weight (γ in kN/m^3)

The above said soil properties can be determined in laboratory by conducting unconsolidated undrained triaxial test on soil samples collected from the site. The failure criterion of Mohr–Coulomb model agrees well with failure in soil samples when tested with triaxial test [31]. The soil properties can also be estimated by conducting field sounding tests, namely standard penetration test, cone penetration test, etc.

3 Expert system for the design of pile foundation

In an effort to develop an expert system for the design of pile foundation, 4809 simulations of pile foundations were carried out using FEM for soils with wide range of unit weight, poisson ratio, modulus of deformation and effective shear parameters. The corresponding unit skin frictions and unit tip bearing capacities were calculated. These input and output data of FEM were used for the development of two artificial neural networks (one ANN for unit skin friction and second ANN for unit tip bearing capacity) with 20 hidden neurons in each case. A remarkable value of coefficient of determination (R^2) averaging 0.99996 was achieved for both the ANNs during training, validation and testing.

Pile loading test is considered to be the most reliable method for the determination of pile load capacity. The axial pile load test on a single pile is justified for pile design load [32]. In the present study, 50 case histories of bored cast-in-place piles reported by Alsamman [33] and Eslami [34] were analyzed using ANNs. The ultimate load-carrying capacities were calculated and then compared with the actual load taken by piles through field pile loading tests. It was observed that the ultimate load-carrying capacities of most of pile case histories computed by ANNs did not match with values computed by pile loading tests. Hence, the correlation factors were calculated for the correct interpretation of the pile load capacities computed by the ANNs. Two correlation factors (skin friction correlation factor C_{sf} and tip bearing capacity correlation factor C_{tbc}) were calculated by comparing the pile load capacities obtained using ANNs and field pile loading test results. The relationship between correlation factors and soil parameters was found using nature-inspired techniques, namely particle swarm optimization (PSO), firefly (FF) algorithm, cuckoo search (CS) and bacterial foraging (BF).

4 Nature-inspired algorithms

These algorithms are very suitable for finding solutions of continuous nonlinear optimization problems. When randomness is used in nature-inspired algorithms, they are known as metaheuristic algorithms [35–37]. These algorithms have remarkable convergence, better learning mechanism and consume less computation time when compared with other optimization algorithms. They also have lesser number of controlling parameters and can be programmed easily. The derivative of objective function is also not required in these techniques. The nature of objective function does not have direct influence on their performance. The initial values of variables can be very approximate at the start of iterations. These algorithms do not get entrapped in local minima of multi-modal objective functions and give a good performance. Most of the nature-inspired algorithms can be applied easily to real-world optimization problems because of their simplicity and efficiency. This paper elaborates a method to find optimal solution to multivariable nonlinear systems through nature-inspired algorithms. For the purpose of comparison of various algorithms, the parameters are regulated as given in Table 2.

4.1 Particle swarm optimization

Particle swarm optimization is widely used as a search algorithm [38]. This method performs better than evolutionary algorithms [38–42]. It is a nature-inspired algorithm based on the behavior of birds in a flock. The flying birds keep distance from each other and stay in optimal region of multi-dimensional search space. In simplified terms, each particle attains position from its own experience and that of the neighboring particles [43]. Let, at any time t, the ith particle is at position x_i^t; then, at time $(t + 1)$, the new position x_i^{t+1} is calculated from velocity v_i^{t+1} as follows:

$$x_I^{T+1} = x_I^t + v_i^{t+1} \quad \text{and} \quad x_i^0 \sim U(x_{\min}, x_{\max}) \tag{1}$$

where $U(x_{\min}, x_{\max})$ denotes uniform distribution with $x_{\min}$ as minimum value and $x_{\max}$ as maximum values of x.

In PSO, all particles are assigned random positions at the time of initiation. Their fitness values are calculated from objective function. The personal best (P_{best}) value is calculated from best fitness value of each particle. The global best (G_{best}) is calculated from the best fit particle having highest fitness value in the entire swarm. This method is also called global best PSO as new positions of the particle are governed by position of best fit particle [43].

In this method, the position of a particle is updated by successive iterations, and for every particle, the new values of personal best positions and its velocities are calculated.

Table 2 Parameter regulation of various algorithms selected for comparison

Parameters of various algorithms	PSO	FF	CS	BF
No. of iterations (n)	7000	7000	7000	
population size (p)	25	25		
Self-confidence factor (c_1)	2			
Swarm confidence factor (c_2)	2			
Velocity weight at the beginning (w_{in})	0.9			
Velocity weight at the end (w_f)	0.4			
Randomness (α)		0.5		
Step size (s)		0.2		
Absorption coefficient (γ)		1.0		
Number of nests (n')			25	
Discovery rate of alien eggs (p_a)			0.25	
Attractiveness (β)			1.5	
Dimension of search space (p)				8
The number of bacteria (s')				25
The number of chemotactic steps (N_c)				1750
Limits the length of a swim (N_s)				4
The number of reproduction steps (N_{re})				4
The number of elimination–dispersal events (N_{ed})				2
The number of bacteria reproductions per generation (S_r)				s/2
The probability that each bacteria will be eliminated/dispersed (P_{ed})				0.25

The new position of each particle is calculated from personal best, global best and current velocity. The stopping criterion for the iterations is decided. In minimization problems, the personal best $P_{\text{best},i}$ at time $(t + 1)$ is computed as follows:

$$P_{\text{best},i}^{t+1} = \begin{cases} P_{\text{best},i}^{t+1} & \text{if} \quad f\left(x_i^{t+1}\right) > f(P_{\text{best},i}^{t}) \\ x_i^{t+1} & \text{if} \quad f\left(x_i^{t+1}\right) \le f(P_{\text{best},i}^{t}) \end{cases} \tag{2}$$

The global best position G_{best} at any time (t) is calculated as follows:

$$G_{\text{best}} = \min\left\{P_{\text{best},i}^{t}\right\}, \quad \text{where} \quad i \in [1, \ldots, n] \quad \text{and} \quad n > 1 \tag{3}$$

The velocity of ith particle is calculated as follows:

$$v_{ij}^{t+1} = v_{ij}^{t} + c_1 r_{1j}^{t}\left[P_{\text{best},i}^{t} - x_{ij}^{t}\right] + c_2 r_{2j}^{t}\left[G_{\text{best}} - x_{ij}^{t}\right] \tag{4}$$

In the above equation, c_1 and c_2 are the acceleration constants, r_{1j}^t and r_{2j}^t are random numbers which are uniformly distributed in $U(0,1)$ at any time t.

4.2 Cuckoo search

It is a nature-inspired optimization algorithm. It was proposed by Yang and Deb [44–47]. This algorithm was developed from the behavior of some cuckoo species, namely Ani and Guira, which lay eggs in the nests of other species. Sometimes, these cuckoo species remove the eggs of host birds to increase chances of hatching of their own eggs. It is assumed that one cuckoo lays one egg. There is some probability associated with the identification of egg by the host bird. The egg may be either hatched or rejected by host bird. In the case of rejection, the host bird may abandon the nest and build a new nest. The rejected eggs are represented by the fraction p_a of n number of host nests and this is used to arrive at new random solutions. It is also assumed that high quality eggs are hatched by the best nest. It is represented in the algorithm by fitness value of high quality eggs or solution found by the objective function. The replacement of egg by cuckoo in the nest of the other species represents as substitution of a solution with better solution. The algorithm can be made more complex when each nest has multiple eggs corresponding to a set of solutions. The CS algorithm uses random walk as local search and Levy flights as global search. The local search is represented as follows:

$$x_i^{t+1} = x_i^t + \alpha s H(p_a - \in).\left(x_j^t - x_k^t\right) \tag{5}$$

The global search is represented as follows:

$$x_i^{t+1} = x_i^t + \alpha L(s, \lambda) \tag{6}$$

$$L(s, \lambda) = \frac{\lambda \Gamma(\lambda) \sin\left(\frac{\pi\lambda}{2}\right)}{\pi} \frac{1}{s^{1+\lambda}} \quad (s \gg s_o > 0) \tag{7}$$

α is a scaling factor which decide the step size. The value of α lies between $L/10$ and $L/100$, where L is a characteristic scale of the problem. In Levy flight, the next solution is calculated from current solution by Markov chain. It uses far field randomization order to avoid local optima [45–47].

4.3 Firefly algorithm

It was proposed by Yang [35, 36, 46, 48]. It was inspired by the behavior of fireflies. The fireflies are unisexual organisms. They get attracted to each other by the brightness of their flashings. The less bright fireflies are attracted toward more bright fireflies. In this algorithm, the brightness of fireflies is calculated from the fitness value of the objective function. Attractiveness is directly proportional to brightness. As brightness reduces with distance, the attractiveness β also follows the same relationship as per the following relation:

$$\beta = \beta_0 e^{-\gamma r^2} \tag{8}$$

where β_0 is the maximum value of attractiveness at distance $r = 0$. The new location of ith firefly under attraction of brighter firefly j can be calculated as follows:

$$x_i^{t+1} = x_i^t + \beta_o e_{ij}^{-\gamma r^2}\left(x_j^t - x_i^t\right) + \alpha_t \in_i^t \tag{9}$$

In the above equation, the first term is the previous location of ith firefly. The second term denotes attraction of the less bright firefly toward brighter firefly. The third term depicts random walk of ith firefly governed by $\in_i^t$, which is a set of random numbers obtained from uniform/Gaussian distribution at any time t. If $\beta_o = 0$, the movement of firefly is completely governed by random walk. When $\gamma = 0$, the movement of firefly is similar to that given by variant of particle swarm optimization. α_t is the parameter which tunes the randomness. The value of α_t is changed with successive iterations using the following equations:

$$\alpha_t = \alpha_o \delta^t, \quad 0 < \delta < 1 \tag{10}$$

In the above equation, α_o and δ are initial randomness scaling factors and cooling factors, respectively. Usually the value of δ varies between 0.95 and 0.97 [36]. α_o is taken as $0.01L$ where L is the average scale of the problem and factor 0.01 is used to reduce the step size for proper exploitation [36, 46]. γ is equal to $1/\sqrt{L}$. The best range for population size is 25–40 [36, 48].

4.4 Bacterial foraging optimization

This algorithm is based on the foraging strategy followed by *E. coli* bacteria [49]. The bacteria try to get maximum energy intake (E) in unit time (T). They follow four key processes in lifetime, namely chemotaxis, swarming, reproduction and elimination–dispersal [49–52]. These key processes are described as follows.

4.4.1 Chemotaxis

The change in position of *E. coli* bacteria from previous position in its search for food takes place by chemotactic movement. If this movement occurs along the same direction as the previous direction, then it is called swimming; otherwise, it is called tumbling. The chemotactic movement is represented mathematically by the following equation:

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\cdot\Delta(i)}} \tag{11}$$

In the above equation, ith bacterium is following gth step of chemotactic, kth step of reproduction and lth step of elimination–dispersal. $C(i)$ is the step size of randomization. $\Delta(i)$ is a vector of random numbers in the range $(-1,1)$ which indicates random direction for the movement of bacterium.

4.4.2 Swarming

The motile species of bacteria release a substance called aspartate which acts as an attractant for a group of bacterium. This takes place under the high concentration of a substance called succinate. This group of bacterium moves in the form of concentric patterns toward nutrient rich locations [53] which is modeled in [54]. A bacterium swarm is formed which is under the influence of an attractant and a repellant from the different members of same swarm. This process can be represented mathematically as follows:

$$\begin{aligned} J_{cc}(\theta(j+1,k,l)) &= \sum_{i=1}^{S} J_{cc}(\theta,\theta^i(j,k,l)) \\ &= \sum_{i=1}^{S}\left[-d_{\text{att}}\exp\left(-w_{att}\sum_{m=1}^{P}(\theta_m-\theta_m^i)^2\right)\right] \\ &\quad + \sum_{i=1}^{S}\left[-h_{\text{rep}}\exp\left(-w_{rep}\sum_{m=1}^{P}(\theta_m-\theta_m^i)^2\right)\right] \end{aligned} \tag{12}$$

In the above equation, S is the number of bacterium in a swarm. $\theta = (\theta_1, \theta_2,\ldots, \theta_p)^T$ is any random point in the search space with p as its dimension. The coefficients d_{att}, w_{att}, h_{rep}, w_{rep} represents quantity and diffusion rate of the attractant and the repellant, respectively [55].

4.4.3 Reproduction

Health of ith bacteria is calculated from the fitness value, which is further calculated from the following equation:

$$J^i_{\text{health}} = \sum_{j=1}^{N_c+1} J^i(j,k,l) \tag{13}$$

where N_c is the number of chemotactic steps travelled by the bacteria under consideration. The bacterium population is divided into two groups depending upon their fitness values. The group with lesser fitness values is subjected to elimination–dispersal, whereas the group with higher fitness is allowed to double by asexual reproduction [56]. Hence, the total population count is maintained [57].

4.4.4 Elimination–dispersal

After passing through some fixed numbers of reproduction cycles selected as stopping criterion, the bacteria with lower fitness are eliminated. This is done to ensure that the bacterium do not entrap in local minima.

5 Implementation and results

This section shows the closeness of predicted pile load capacities by the different algorithms under study and measured pile load capacities. The aim is to evaluate the suitability of the given optimization techniques for determining correlation factors and comparing them. Negative of the correlation coefficient is taken as the fitness value. The stopping criteria and initial population are fixed to same values for all optimization algorithms. The fitness values obtained by various algorithms in 30 independent runs are as given in Table 3. Parametric and nonparametric statistical tests were performed for proper comparison and analysis of the results obtained.

5.1 Parametric tests

These tests can be performed only if the results obtained by various algorithms are independent and follow a normal distribution. As all the algorithms have different mechanism to arrive at a solution, the condition of independence was fairly satisfied. To check the normality of data, Kolmogorov–Smirnova (K–S) test and Shapiro–Wilk (S–W) test [58] were conducted and results obtained are presented in Table 4. The P values were calculated by K–S test and S–W test. For data to follow normal distribution, the P value should be more than 0.1. The results of all algorithms exhibited normal distribution as P values obtained were more than 0.1. Hence, with the both conditions

Table 3 Fitness values obtained by various algorithms in 30 independent runs

Sr. No.	FF	PSO	CS	BF
1	−0.9603	−0.8928	−0.8171	−0.9004
2	−0.9356	−0.9246	−0.8907	−0.8033
3	−0.9001	−0.9435	−0.9331	−0.7674
4	−0.9419	−0.9688	−0.8979	−0.7891
5	−0.91	−0.8632	−0.9523	−0.9261
6	−0.8637	−0.9035	−0.9482	−0.9034
7	−0.8943	−0.9154	−0.8784	−0.9337
8	−0.9565	−0.9402	−0.8528	−0.9219
9	−0.8638	−0.9822	−0.8154	−0.915
10	−0.9606	−0.8886	−0.8231	−0.9115
11	−0.8727	−0.9355	−0.8097	−0.776
12	−0.8648	−0.9537	−0.8458	−0.9348
13	−0.9549	−0.9438	−0.8574	−0.9224
14	−0.8931	−0.9762	−0.9027	−0.8056
15	−0.8791	−0.9411	−0.8262	−0.87
16	−0.9447	−0.9812	−0.8137	−0.9395
17	−0.8984	−0.9782	−0.8162	−0.9194
18	−0.9581	−0.8912	−0.8625	−0.7764
19	−0.9588	−0.916	−0.9114	−0.9027
20	−0.9538	−0.9261	−0.9502	−0.8071
21	−0.9569	−0.8843	−0.8807	−0.9281
22	−0.9396	−0.9114	−0.8303	−0.89
23	−0.9558	−0.8887	−0.8841	−0.7738
24	−0.9461	−0.9305	−0.8364	−0.8474
25	−0.9457	−0.8835	−0.8636	−0.8736
26	−0.903	−0.8589	−0.8276	−0.8301
27	−0.9188	−0.9541	−0.9008	−0.7987
28	−0.8901	−0.889	−0.9192	−0.9002
29	−0.9511	−0.878	−0.9513	−0.8152
30	−0.9403	−0.9812	−0.8514	−0.7868

Table 4 *P* values of results of algorithms as calculated by K–S test and S–W test

Tests	FF	PSO	CS	BF
Kolmogorov–Smirnova	0.2002	0.1761	0.1352	0.1892
Shapiro–Wilk	0.1435	0.1063	0.1022	0.1386

satisfied, the parametric as well as nonparametric tests were conducted.

5.2 Nonparametric tests

These tests are used to rank the various algorithms based on their performance. The level of significance α was selected as 5 % for all the nonparametric tests. The objective function values obtained using various algorithms were compared to find the most suitable technique for problem under study and related problems. The algorithms were also compared with each other for the purpose of ranking. The average ranks of FF, PSO, CS and BF as calculated by Friedman Test [59] and Kendall's W Test [60] was 3.800, 4.687, 2.450 and 1.512, respectively. Thus, PSO achieved the best rank among all the algorithms under study. Friedman test with post hoc test was also used in order to compare the algorithms, and the results obtained are shown in Fig. 1.

The Tukey HSD test [61], Scheffé test [62], Bonferroni test [63] and Holm test [64] were conducted for multiple comparisons between experimental observations and algorithm outputs and as well as comparison of the considered algorithms. These post hoc tests help in determining whether pairs of treatments are significantly different. If the *P* values of these tests are less than 0.05, the pair of treatments is considered significantly different; on the other hand, high *P* values from these tests indicate greater similarity. *P* value of ANOVA, Tukey HSD *P* value, Scheffé *P* value, Bonferroni *P* value and Holm *P* value in Table 5 showed maximum similarity of the PSO results to experimental results. The results generated by CS showed similarity to results given by BF. Hence, these algorithms exhibit similar capabilities in dealing with these types of problems. In Table 6, the contrast estimation values are given which depict the differences between the average values of different algorithms. The results given by PSO algorithm show the maximum closeness to the experimental observations. Hence, PSO, FF, CS and BF algorithms can be ranked in the order from first to fourth, respectively.

6 Application of optimization algorithms

Nature-inspired optimization algorithms have been applied for the determining the skin friction correlation factor (C_{sf}) and tip bearing capacity correlation factor (C_{tbc}). For

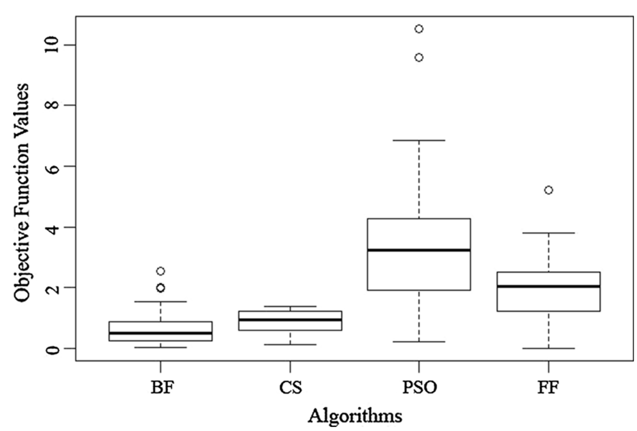

Fig. 1 Comparison of various algorithms from results of Friedman test with post hoc test

Table 5 Results of parametric and nonparametric tests conducted for multiple comparisons

Algorithms	P	Tukey HSD P value	Scheffé P value	Bonferroni P value	Holm P value
EXP versus FF	0.9468	0.8457	0.9667	4.5233	1.1094
EXP versus PSO	0.9682	0.9	0.9863	5.5472	1.357
EXP versus CS	0.9315	0.8245	0.8654	3.8622	0.9035
EXP versus BF	0.9125	0.7766	0.717	3.4814	0.8852
FF versus PSO	0	0.6496	0.8088	1.8382	1.1073
FF versus CS	0	0.8706	1.0133	3.4083	1.4201
FF versus BF	0	0.7702	0.8547	2.2834	1.1263
PSO versus CS	0	0.1439	0.2953	0.2186	0.2893
PSO versus BF	0	0.1255	0.1791	0.1104	0.1809
CS versus BF	0	0.9239	1.0316	7.9385	0.8825

Table 6 Contrast estimation as given by average values of results of various algorithms

	EXP	FF	PSO	CS	BF
EXP	0	−0.622	−1.041	−1.83	−2.11
FF	0.622	0	−1.425	−1.025	−1.303
PSO	1.041	1.425	0	−2.457	−2.74
CS	1.83	1.025	2.457	0	−0.275
BF	2.11	1.303	2.74	0.275	0

determining the values of both these factors, five variables as given in Sect. 2 along with the depth values of centre of soil layer (D) and length of pile (L) were used. The results are analyzed for carrying out a comparative study of nature-inspired algorithms. For determining correlation factor (C_{sf}), the number of nonlinear equality constraints are 170 and that for correlation factor (C_{tbc}) are 50. The constraints used were found from field pile loading test results. The objective function used for determining C_{sf} and C_{tbc} are represented as follows:

$$C_{sf} = 0.3319 - 0.0242\left(\frac{\gamma^{4.3821}\cdot\varphi'^{17.5485}\cdot D^{0.1394}}{c'^{2.1430}\cdot\mu^{4.8695}\cdot E^{2.7243}}\right)^{5.8412}$$
$$C_{tbc} = 0.1285 + 0.1646\left(\frac{\varphi'^{0.4405}\cdot c'^{0.3972}\cdot E^{0.0516}\cdot L^{3.2236}}{\gamma^{4.7047}\cdot\mu^{3.7618}}\right)^{0.6404} \quad (14)$$

The objective functions with 18 dimensions are considered. For each iteration, the fitness value is calculated from the correlation coefficient found between the results of a particular algorithm and those obtained from the pile loading tests. The ultimate load-carrying capacity of 40 bored pile foundations are calculated using all the algorithms under study. The values obtained in each case are compared with the field pile loading test results as shown in Fig. 2.

The PSO, FF, CS and BF algorithms have achieved maximum value of correlation coefficient as 0.9821, 0.9606, 0.9523 and 0.9348, respectively, with the field pile loading test results in 30 trials, whereas ANNs (without optimization) have given value of correlation coefficient as 0.8768 with the field pile loading test results. The values computed by PSO algorithm are found to be closest to the field test values. Thus, the PSO algorithm was found to perform significantly better than the other algorithms. Comparing the computational efficiency of the algorithms under consideration in Table 7, the figures clearly indicate that PSO requires minimum computational time (when run on a computer with 2.2 GHz Intel Core 2 Duo processor having 2 GB RAM). Further, with a few parameters to be tuned, PSO has been found to be relatively easy to implement. Figure 3 illustrates the convergence history of the algorithms under study. All the algorithms are found to provide optimal solution at 7000 iterations. PSO outperforms the other algorithms for this particular problem as it gives the best fitness vales.

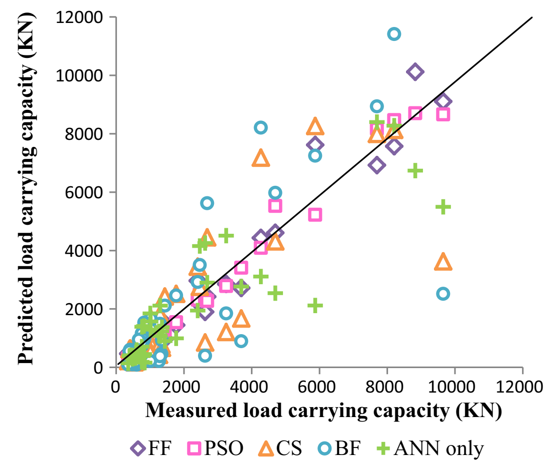

Fig. 2 Comparison of pile load capacities as predicted by algorithms and measured in field pile loading tests

Table 7 Computational efficiency of all various algorithms

Sr. No.	Algorithm	Seconds	Remarks
1	FF	52	1000 Iterations
2	PSO	46	1000 Iterations
3	CS	48	1000 Iterations
4	BF	516	250 Chemotactic steps

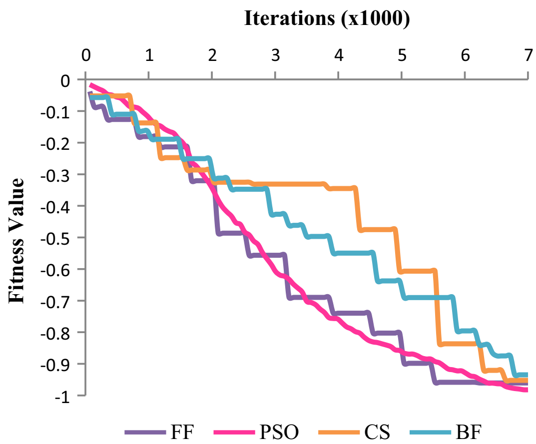

Fig. 3 Convergence history of various algorithms under study

7 Conclusion

The study portrays that the use of nature-inspired algorithms for finding the correlation factors substantially improves the accuracy of ANNs for predicting the pile load capacities. The dataset comprising of 50 full-scale pile loading tests has been used for calculating the correlation factors. The ANNs have been developed using the data acquired from the finite element analysis of axially loaded piles of various geometrical properties and from the data pertaining to a wide range of soil parameters. The pile foundation was analyzed 4809 times to ascertain the exact soil–structure interaction and for determining the accurate values of unit skin friction and unit end bearing capacity. The skin friction correlation factor was taken as objective function. A comparative study of four nature-inspired algorithms for designing bored pile foundation was carried out. Data pertaining to field pile loading tests were used for the evaluating the performance of these algorithms. The correlation coefficient between the predicted values and field pile loading test data was found to be the highest (0.982) in the case of PSO. The PSO was thus found to be the best performing algorithm. PSO was also found to exhibit maximum computational efficiency as it took least computational time due to less number of calculation steps. Parametric and nonparametric statistical analysis tests were carried out to further substantiate the results obtained. The *P* value obtained using ANOVA test showed maximum similarity between the PSO predicted values and field test values. The *P* values of Tukey HSD, Scheffé and Bonferroni also depicted similar results. The PSO has also been found to be more efficient in terms of time taken and is less complex as compared to conventional methods like load transfer method and elastic method for finding the pile load capacity. Hence, it can be concluded that PSO algorithm is the most suitable algorithm for the optimization of pile foundation design in a constrained environment.

References

1. Cui L, Sheng D (2005) Genetic algorithms in probabilistic finite element analysis of geotechnical problems. Comput Geotech 32:555–563
2. Liu M, Qie Z, Wu X, Dong W, Zheng H (2008) Model optimization of load-bearing capacity of macadam pile composite foundation based on genetic algorithms. In: IEEE control and decision conference Yantai, pp 3903–3907
3. Liu X, Cheng G, Wang B, Lin S (2012) Optimum design of pile foundation by automatic grouping genetic algorithms. ISRN Civil Engineering
4. Elbeltagi E, Hegazy T, Grierson D (2005) Comparison among five evolutionary based optimization algorithms. Adv Eng Inf 19:43–53
5. Chan WT, Chow YK, Liu LF (1995) Neural network: an alternative to pile driving formulas. Comput Geotech 17:135–156
6. Nawari NO, Liang R, Nusairat J (1999) Artificial intelligence techniques for the design and analysis of deep foundations. Electron J Geotech Eng 4. http://geotech.civeng.okstate.edu/ejge/ppr9909/index.html
7. Ismail A, Jeng D-S (2011) Modelling load settlement behaviour of piles using high order neural network (HON-PILE model). Eng Appl Artif Intell 24(5):813–821
8. Zhang JR, Zhang J, Lok TM, Lyu MR (2007) A hybrid particle swarm optimization back-propagation algorithm for feedforward neural network training. Appl Math Comput 185:1026–1037
9. Cui Y, Ju SG, Han F, Gu TY (2009) An improved approach combining random PSO with BP for feedforward neural networks. Artif Intell Comput Intell 5855:361–368
10. Ismail A, Jeng D-S, Zhang LL (2013) An optimised product-unit neural network with a novel PSO–BP hybrid training algorithm: applications to load–deformation analysis of axially loaded piles. Eng Appl Artif Intell 26(10):2305–2314
11. Goh ATC (1994) Nonlinear modelling in geotechnical engineering using neural networks. Aust Civil Eng Trans 36(4):293–297
12. Goh ATC (1995) Empirical design in geotechnics using neural networks. Geotechnique 45(4):709–714
13. Goh ATC (1995) Back-propagation neural networks for modeling complex systems. Artif Intell Eng 9:143–151
14. Goh ATC (1996) Pile driving records reanalyzed using neural networks. J Geotech Eng ASCE 122(6):492–495
15. Lee IM, Lee JH (1996) Prediction of pile bearing capacity using artificial neural networks. Comput Geotech 18(3):189–200
16. Abu-Kiefa MA (1998) General regression neural networks for driven piles in cohesionless soils. J Geotech Geoenviron Eng 124(12):1177–1185
17. Shahin MA, Jaksa MB (2005) Neural network prediction of pullout capacity of marquee ground anchors. Comput Geotech 32(3):153–163

18. Shahin MA, Jaksa MB (2006) Pullout capacity of small ground anchors by direct cone penetration test methods and neural networks. Can Geotech J 43(6):626–637
19. Nejad PF, Jaksa MB, Kakhi M, McCabe BA (2009) Prediction of pile settlement using artificial neural networks based on standard penetration test data. Comput Geotech 36:1125–1133
20. Alkroosh I, Nikraz H (2014) Predicting pile dynamic capacity via application of an evolutionary algorithm. Soils Found 54(2):233–242
21. Das BM (1999) Principles of foundation engineering, 4th edn. Brooks/Cole Publishing Co., Pacific Grove California
22. Poulos HG, Davis E (1980) Pile foundation analysis and design. Wiley, New York
23. Meyerhof GG (1976) Bearing capacity and settlement of pile foundations. J Geotech Eng Div ASCE 102(3):195–228
24. Vesic AS (1977) Design of pile foundations. Nat Cooperative Hwy Res Program Synthesis of Pract No 42 Transportation Research Board, Washington
25. Mindlin RD (1936) Force at a point in the interior of a semi-infinite soil. Physics 7:195–202
26. Alexander H (1941) Solution of problems of elasticity by the framework method. J Appl Mech 84:169–175
27. Courant R (1943) Variational methods for the solution of problems of equilibrium and vibrations. Bull Am Math Soc 49:1–23
28. Cook R, Malkus D, Plesha M (1989) Concepts and applications of finite element analysis. Wiley, New York
29. Cook R (1995) Finite element modeling for stress analysis. Wiley, New York
30. Coduto D (2001) Foundation design: principals and practices, 2nd edn. Prentice-Hall Inc, New Jersey
31. Goldscheider M (1984) True triaxial tests on dense sands. Constitutive relations for soils. Balkema, Rotterdam
32. Abdelrahman GE, Shaarawi EM, Abouzaid KS (2003) Interpretation of axial pile load test results for continuous flight auger piles. In: Emerging technologies in structural engineering proceedings of the 9th Arab structural engineering conference, Abu Dhabi, pp 791–802
33. Alsamman O (1995) The use of CPT for calculating axial capacity of drilled shaft. PhD Thesis University of Illinois Urbana
34. Eslami A (1996) Bearing capacity of piles from cone penetration data. PhD Thesis University of Ottawa, Ottawa
35. Yang XS (2010) Engineering optimisation: an introduction with metaheuristic applications. Wiley, New York
36. Yang XS (2008) Nature-inspired metaheuristic algorithms. Luniver Press, UK
37. Blum C, Aguilera MJB, Roli A, Sampels M (2008) Hybrid metaheuristics: an emerging approach to optimization. Studies in computational intelligence. Springer, Berlin
38. Kennedy J, Eberhart R (1995) Particle swarm optimization. Proc IEEE Int Conf Neural Netw 4:1942–1948
39. Angeline P (1998) Using selection to improve particle swarm optimization. In: Proceedings of the IEEE international conference on evolutionary computation. IEEE world congress on computational intelligence, pp 84–89
40. Eberhart RC, Shi Y (1998) Comparison between genetic algorithms and particle swarm optimization. In: Proceedings of the 7th international conference on evolutionary programming VII ser EP'98 London, UK. Springer, Berlin, pp 611–616
41. Shi Y, Eberhart R (1998) A modified particle swarm optimizer. In: Proceedings on evolutionary computation IEEE world congress on computational intelligence, pp 69–73
42. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the 6th IEEE international symposium on micro machine and human science (MHS'95), Nagoya, Japan, pp 39–43
43. Chen D, Zhao C, Zhang H (2011) An improved cooperative particle swarm optimization and its application. Neural Comput Appl 20:171–182
44. Yang XS, Deb S (2014) Cuckoo search: recent advances and applications. Neural Comput Appl 24(1):169–174
45. Yang XS, Deb S (2009) Cuckoo search via Ĺevy flights. In: Proceedings of world congress on nature & biologically inspired computing (NaBIC 2009). IEEE Publications USA, pp 210–214
46. Yang XS, Deb S (2010) Engineering optimization by cuckoo search. Int J Math Model Num Opt 1(4):330–343
47. Yang XS, Deb S (2012) Multiobjective cuckoo search for design optimization. Comput Oper Res 40(6):1616–1624
48. Yang XS (2009) Firefly algorithms for multimodal optimization. Proceedings of the 5th symposium on stochastic algorithms foundations and applications. Lect Notes Comput Sci 5792:169–178
49. Passino KM (2002) Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Syst Mag 22(3):52–67
50. Gazi V, Passino KM (2010) Swarm stability and optimization. Springer, New York, pp 233–249
51. Brownlee J (2011) Clever algorithms: nature inspired programming recipes, pp 257–264
52. Atasagun Y, Kara Y (2014) Bacterial foraging optimization algorithm for assembly line balancing. Neural Comput Appl 25:237–250
53. Coelho LS, Silveira CC, Sierakowski CA, Alotto P (2010) Improved bacterial foraging strategy applied to TEAM workshop benchmark problem. IEEE Trans Magn 46(8):2903–2906
54. Woodward DE, Tyson R, Myerscough MR, Murray JD, Budrene EO, Berg HC (1995) Spatio-temporal patterns generated by Salmonella typhimurium. Biophys J 68:2181–2189
55. Das S, Dasgupta S, Biswas A, Abraham A (2009) On stability of the chemotactic dynamics in bacterial-foraging optimization algorithm. IEEE Trans Syst Man Cybern 39(3):670–679
56. Okaeme NA, Zanchetta P (2013) Hybrid bacterial foraging optimization strategy for automated experimental control design in electrical drives. IEEE Trans Ind Informat 9(2):668–678
57. Dasgupta S, Das S, Abraham A, Biswas A (2009) Adaptive computational chemotaxis in bacterial foraging optimization: an analysis. IEEE Trans Evol Comput 13(4):919–941
58. Shapiro SS, Wilk MB (1965) An analysis of variance test for normality (complete samples). Biometrika 52(3/4):591–611
59. Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J Am Stat Assoc 32(200):675–701
60. Kendall MG, Babington SB (1939) The problem of m rankings. Ann Math Stat 10(3):275–287
61. Tukey J (1949) Comparing Individual Means in the Analysis of Variance. Biometrics 5(2):99–114
62. Scheffé H (1999) The analysis of variance. Wiley, New York. ISBN 0-471-34505-9
63. Bonferroni CE (1936) Teoria statistica delle classi e calcolo delle probabilità. Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze
64. Holm S (1979) A simple sequentially rejective multiple test procedure. Scand J Stat 6(2):65–70