Periodica Polytechnica Civil Engineering, 63(3), pp. 856–871, 2019

Meta-heuristic Optimization Algorithms for Predicting the Scouring Depth Around Bridge Piers

Yousef Hassanzadeh[1], Amin Jafari-Bavil-Olyaei[1]*, Mohammad Taghi-Aalami[1], Nazila Kardan[2]

[1] Department of Civil Engineering, University of Tabriz,
Bahman Blvd, P.O. Box: 5166616471, Tabriz, Iran
[2] Department of Civil Engineering, Azarbaijan Shahid Madani University,
Azarshahr Road P.O. Box: 5375171379, Tabriz, Iran
* Corresponding author, e-mail: Amin.Jafari@tabrizu.ac.ir

Received: 30 June 2018, Accepted: 16 June 2019, Published online: 14 August 2019

Abstract

An accurate estimation of bridge pier scour has been considered as one of the important parameters in designing of bridges. However, due to the numerous involved parameters and convolution of this phenomenon, many existing approaches cannot predict scour depth with an acceptable accuracy. Obtained results from the empirical relationships show that these relationships have low accuracy in determining the maximum scour depth and they need a high safety factor for many cases, which leads to uneconomic designs of bridges. To cover these disadvantages, three new models are provided to estimate the bridge pier scour using an adaptive network-based fuzzy inference system. The parameters of the system are optimized by using the colliding bodies optimization, enhanced colliding bodies optimization and vibrating particles system methods. To evaluate the efficiency of the proposed methods, their results were compared with those of simple adaptive network-based fuzzy inference system and its improved versions by using the particle swarm optimization and genetic algorithm as well as the empirical equations. Comparison of results showed that the new vibrating particles system based algorithm could find better results than other two ones. In addition, comparison of the results obtained by the proposed methods with those of the empirical relations confirmed the high performance of the new methods.

Keywords

adaptive neuro-fuzzy inference system, colliding bodies optimization, empirical equations, scour depth, vibrating particles system

1 Introduction

Nowadays in many countries due to the economic growth and development of societies, the traffic load increases, and therefore some structures such as bridges that have a significant influence on road and rail transportation have become more important. As a result, bridge failures should be studied in details. The reason is clear: every year, river floods destroy a large number of bridges. In addition to the issue of safety, bridge failures impose vast costs. Investigations carried out by many researchers have shown that the local scour is the main cause of bridge failures [1–3]. The local scour is a complex phenomenon caused by bridge piers that obstruct the natural flow of the water in rivers. The erosive action of this abrupt makes change in flow filed and carries away sediments from the bed around the piers. Prediction of the equilibrium depth of a local scour around bridge piers is an essential subject in the design of bridges. It is clear that using a low reliability factor may not guarantee the safety of the bridge, and vice versa, a design with a high degree of reliability leads to a non-economic project. Therefore, a precise method is required for determining the depth of scouring for designing efficient and economical bridges.

Until now, different methods have been proposed to estimate the depth of the bridge pier scour. Most of them are based on physical modeling and empirical relationships extracted from laboratory data obtained by using only limited parameters. As a result, many important factors of scour have been consumedly or incorrectly assessed [4]. Muller et al. [5] showed that the depth of scouring obtained from empirical equations in some cases was several times greater than actual values as reported by experimental observations. Shepard et al. [4] examined the performance of different empirical equations for scour depth estimation and showed that some new equations

Cite this article as: Hassanzadeh, Y., Jafari-Bavil-Olyaei, A., Taghi-Aalami, M., Kardan, N. "Meta-heuristic Optimization Algorithms for Predicting the Scouring Depth Around Bridge Piers", Periodica Polytechnica Civil Engineering, 63(3), pp. 856–871, 2019. https://doi.org/10.3311/PPci.12777

have better efficiency. However, they concluded that some of the previous equations estimate even negative values for scouring. This means that most of the empirical equations tend to work well only in specific experimental conditions which limits their usefulness [6].

Due to the imperfections of empirical relationships, researchers have begun to use artificial intelligence (AI) methods to predict the scour depth. Using AI methods is a good alternative for physical models and empirical relationships [6]. Several studies have been carried out using AI techniques to estimate the maximum scour depth around bridge piers. Bateni et al. [6] modeled the bridge pier scour using laboratory data and neural network. Their results showed that the developed neural network predicted scour depth better than empirical relationships. Kermani et al. [7] predicted the depth of scouring for pier groups using the back propagation algorithm and the radial base neural network. Their results showed that these methods had sufficient accuracy to estimate the scour depth around the bridge pier. Akib et al. [8] used the Adaptive Network-based Fuzzy Inference System (ANFIS) as a modeling tool to predict the scour depth in bridges. Comparison of results between ANFIS and classical linear regression (LR) showed that its results had a high and satisfactory accuracy. Ebtehaj et al. [9] predicted the scour depth around bridge piers by extreme learning machine (ELM). To predict scour depth, the effective dimensionless parameters were determined through dimensional analysis. The results showed that the most important parameters affecting relative scour depth (d_s/y) included ratio of pier width to flow depth (D/y) and ratio of pier length to flow depth (L/y). In addition, other researchers have performed a number of studies in this field as well, such as references [10–17].

Swarm intelligence-based optimization algorithms use casual elements to increase solution diversity during the search process. According to this, a number of researchers have used optimization methods in order to improve predictive accuracy of the classical AI models. Chou et al. [18] investigated the potential use of Genetic Algorithm (GA)-based support vector regression (SVR) model to predict the bridge scour depth near piers and abutments. Results showed that their model performed more accurate than classic methods. In another study, Chou et al. [19] developed a hybrid artificial firefly colony algorithm based on SVR model for predicting bridge pier scour. Their model achieved better results compared to other predictive methods.

In this research, a new hybrid method is developed for estimation of the bridge pier scour by hybridizing the ANFIS with three new optimization algorithms. The used optimization methods are: Colliding Bodies Optimization (CBO) developed by Kaveh and Mahdavi [20], the Enhanced Colliding Bodies Optimization (ECBO) expanded by Kaveh and Ilchi Ghazaan [21], as well as Vibrating Particles System (VPS) presented by Kaveh and Ilchi Ghazaan [22]. These methods are new and effective algorithms in solving different optimization problems [23, 24]. Different applications of these methods can be found in Kaveh [25] Results these application have shown that CBO and ECBO provided a good balance between the exploration and the exploitation paradigms of the algorithm. Kaveh and Ilchi Ghazaan [26] employed VPS for structural optimization with frequency constraints. Results demonstrated the efficiency and robustness of their method and its competitive performance to other algorithms for structural optimization problems. Rabiei et al. [27] evaluated the capability of CBO, ECBO and VPS methods for solving reservoir operation issues. Their results showed that the best solutions were obtained by the VPS, which could solve optimization problems with less computational efforts and a fast convergence rate.

In the new models proposed in this paper, the CBO, ECBO and VPS optimization methods optimized the parameters of the ANFIS and increased its efficiency and robustness. In order to develop the model, the US Federal Highway Administration (FHWA) Data was used to train and validate models. Finally, the performance of the models was compared to the standard ANFIS and empirical relationships reported by technical literature.

2 Utilized method

Despite the greater efficiency of Artificial Neural Network (ANN) methods than existing empirical relationships, it is still possible to achieve better results by optimizing the performance of the ANN-based methods. ANFIS, which is considered as the basic method in this research, is composed of neural network and fuzzy system. To optimize the ANFIS model, the CBO, ECBO and VPS algorithms have been used. CBO algorithm is based on one-dimensional collisions between bodies [20]. ECBO method consists of a multi-agent algorithm inspired by a collision between two objects in one-dimension.. VPS algorithm is based on free vibration of single degree of freedom system with viscous damping. The following subsections describe the utilized methods in details.

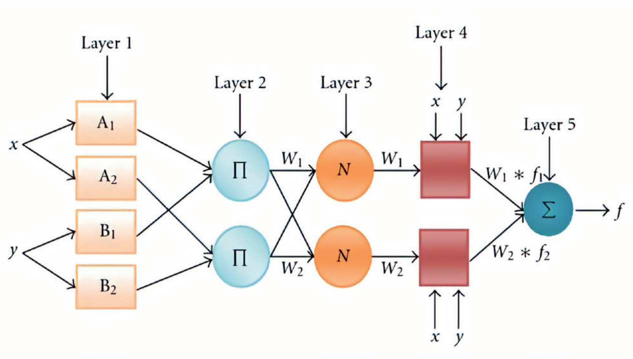


Fig. 1 General framework of the ANFIS

2.1 Adaptive neuro-fuzzy inference system

ANFIS is an artificial neural network developed based on the Takagi-Sugeno fuzzy inference system in 1990 [28]. In this method, the neural network and the fuzzy system are combined in a coordinated structure. This model can be introduced as neural network with a fuzzy parameter or a distributed learning fuzzy system. Fuzzy systems can reduce the dimensionality of the search space by distributing input information over the network and also find the optimum values of the control parameter for non-linear problems by using the back propagation behaviors of neural networks [29]. The architecture of this algorithm is generally composed of five layers as shown in Fig. 1.

To simplify the explanations, the fuzzy inference system under consideration is assumed to have two inputs (*x* and *y*) and one output (*f*). For a first order of Sugeno fuzzy model.

A typical rule set with base fuzzy if-then rules can be expressed as following.

If *x* is equal to A_1 and y is B_1, the first rule is Eq. (1).

$$f = p_1 x + q_1 y + r_1. \tag{1}$$

In addition, if *x* is equal to A_2 and y is equal to B_2, then the second rule is Eq. (2).

$$f = p_2 x + q_2 y + r_2. \tag{2}$$

In these two rules, *x* and y are input variables, A_i and B_i are fuzzy conditions, *f* is output, and *p*, *r*, *q* are design parameters that should be determined during the learning process of the ANFIS. The application of different layers of this algorithm is as follows:

In the first layer, each node *i* is represented by a membership function as Eqs. (4–5).

$$O_{1,i} = \sigma_{A_i(x)}, \tag{3}$$

$$O_{1,i} = \sigma_{B_i(x)}. \tag{4}$$

In Eq. (3), A_i is the linguistic variable, *x* is the input to node *i* and $O_{1,i}$ is the membership function of A_i. In this study, Gaussian membership functions are used as Eq. (5).

$$\sigma_{A_i(x)} = exp\left(-\frac{(x-c)^2}{2\sigma^2}\right), \tag{5}$$

Where σ is the standard deviation and *c* is the center of the above Gaussian membership function. In the second layer, each node refers to the rules generated by the fuzzy logic inference system and the firing strength of a rule is determined by Eq. (6).

$$\omega_i = \sigma_{A_i(x)} \times \sigma_{B_i(x)} \quad i = 1,2\ . \tag{6}$$

In the third one, the firing strength of each rule is normalized. This normalization is the ratio of the firing strength of the *i*th rule to the total of all firing strengths by using the Eq. (7).

$$\bar{\omega} = \frac{\omega_i}{\omega_1 + \omega_2} \quad i = 1,2\ . \tag{7}$$

The set of fuzzy rules is calculated as follows in the fourth layer by Eq. (8).

$$\bar{\omega}_i f_i = \bar{\omega}_i \left(p_i x + q_i y + r_i \right) \quad i = 1,2\ . \tag{8}$$

Finally, in the fifth layer, all the outputs of the fourth layer are combined Eq. (9).

$$\sum \bar{\omega}_i f_i = \frac{\sum_i \omega_i f_i}{\sum_i \omega_i} \quad i = 1,2\ . \tag{9}$$

2.2 Colliding Bodies Optimization

Colliding Bodies Optimization (CBO), developed by Kaveh and Mahdavi [20], is based on one-dimensional collisions between bodies with each agent solution being considered as the massed object. After a collision of two moving bodies with determined masses and velocities, these bodies are separated with new speeds. This collision pushes the agents toward better positions in the inquiry space. The laws of momentum and energy govern collisions between bodies. When in an isolated system collision happens, the total momentum of objects is protected as shown in Fig. 2.

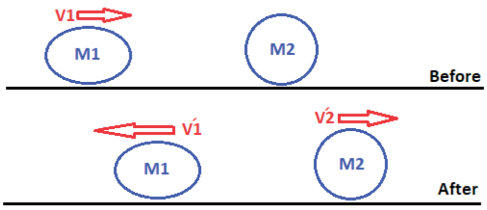


Fig. 2 Collision of bodies

In this algorithm, there are no external forces on the bodies and total momentum of the system before and after the collision is constant. This process can be expressed by the following equation:

$$m_1 v_1 + m_2 v_2 = m_1 v_1 + m_2 v_2 . \tag{10}$$

Similar to the momentum energy, the kinetic energy is constant and can be expressed as:

$$\frac{1}{2} m_1 {v_1}^2 + \frac{1}{2} m_2 {v_2}^2 = \frac{1}{2} m_1 v_1^2 + \frac{1}{2} m_2 v_2^2 + Q. \tag{11}$$

Where m_1 is the mass of the first object, v_1 is the initial velocity of the first object before impact and v'_1 is the final velocity of the first object after impact. In addition, m_v is the mass of the second object, v_2 is the initial velocity of the second object before impact and v'_2 is the final velocity of the second object after impact and Q is the loss of the kinetic energy due to the impact. The equations for the velocities after a one-dimensional collision are:

$$v_1 = \frac{(m_1 - \varepsilon m_2) v_1 + (m_2 + \varepsilon m_2) v_2}{m_1 + m_2},$$
$$v_2 = \frac{(m_2 - \varepsilon m_1) v_2 + (m_1 + \varepsilon m_1) v_1}{m_1 + m_2}. \tag{12}$$

Where ε is the Coefficient of Restitution (COR) of the two colliding bodies defined as the ratio of relative velocity of separation to relative velocity of approach:

$$\varepsilon = \frac{|v_2 - v_1|}{|v_2 - v_1|} = \frac{v}{v}. \tag{13}$$

Based on the coefficient of restitution value in each collision, two modes are possible. If there is no kinetic energy dissipation when encountering two bodies ($Q = 0$ and $\varepsilon = 1$), it can be considered as an elastic collision. In this case, after collision the velocity of separation is high. However if some part of the kinetic energy is converted to other types of energy ($Q \neq 0$ and $\varepsilon \leq 1$) during encounter, this kind of collision is called non-elastic, which has less separation velocity than the elastic one. For the most real objects, the value of ε is between zero and one.

In this algorithm, each solution candidate X_i containing a number of variables is considered as collision body (CB). The massed bodies are composed of two main equal groups, i.e., stationary and moving ones. When moving bodies move to follow stationary bodies, a collision occurs between pairs of bodies, which will improve the moving bodies positions and push stationary bodies toward better positions. After collision of bodies, their new positions and velocities are updated based on Eqs. (10–13).

The pseudo-code for CBO algorithm can be expressed as follows:

1. The initial positions of CBs are determined with random initialization of a population of individuals in the search area as Eq. (14).

$$x_i^0 = x_{min} + rand\left(x_{max} - x_{min}\right),\ i = 1, 2, 3, \ldots, n\ , \tag{14}$$

where, $x_i{}^0$ determines the initial value vector of the *i*th CB. x_{max} and x_{min} are the maximum and minimum allowable value vectors of variables; *rand* is a random number in the interval (0, 1); and *n* is the number of CBs.

2. The mass of each CB is defined by Eq. (15).

$$m_k = \frac{\dfrac{1}{fit(k)}}{\sum_{i=1}^{n} \dfrac{1}{fit(i)}},\ k = 1, 2, 3, \ldots, n\ , \tag{15}$$

where *fit*(*i*) represents the objective function value of the agent *i* and *n* is the population size. As can be seen, CBs with higher values are more massive than others.

3. The arrangement of the CBs objective function values is performed in ascending order (Fig. 3). The sorted CBs are equally divided into two groups. The lower half of CBs are good agents which are stationary and the velocity of these bodies before collision is zero and The upper half of CBs move toward the lower half. Then the better and worse CBs, i.e. agents with upper objective function value, of each group will crash together. The change of the bodies position represents the velocity of these bodies before collision as:

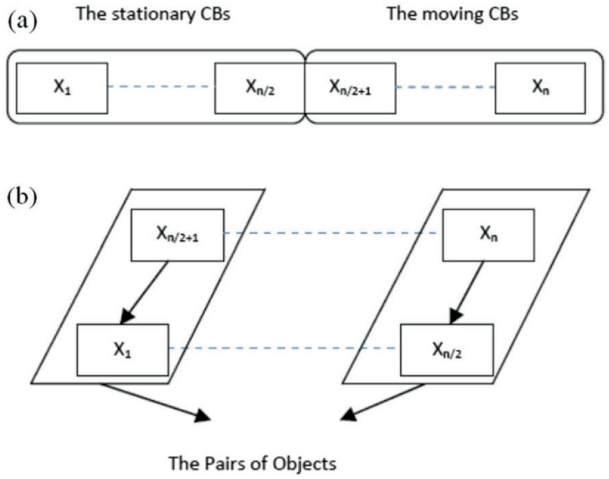


Fig. 3 Arrangement of the CBs objective function (a) CBs sorted in increasing order; (b) colliding object pairs (Kaveh [22])

$$v_i = 0, \quad i = 1,2,\ldots,\frac{n}{2}, \tag{16}$$

$$v_i = x_i - x_{i-\frac{n}{2}}, \quad i = \frac{n}{2}+1, \frac{n}{2}+2, \ldots, n, \tag{17}$$

where, v_i and x_i are the velocity and position vector of the ith CB in this group, respectively; $x_{i-\frac{n}{2}}$ is the ith CB pair position of xi in the previous group.

4. After collision, velocity of the bodies in each group is obtained using Eqs. (16) and (17). The velocity of each moving CB after the collision is calculated by Eq. (18).

$$v_i = \frac{\left(m_i - \varepsilon m_{i-\frac{n}{2}}\right) v_i}{m_i + m_{i-\frac{n}{2}}}, \quad i = \frac{n}{2}+1, \frac{n}{2}+2, \ldots, n. \tag{18}$$

The velocity of each stationary CB after the collision is also calculated by Eq. (19).

$$v_i = \frac{\left(m_{i+\frac{n}{2}} + \varepsilon m_{i-\frac{n}{2}}\right) v_{i+\frac{n}{2}}}{m_i + m_{i-\frac{n}{2}}}, \quad i = 1,2,\ldots,\frac{n}{2}. \tag{19}$$

Where v_i is the velocity of the ith moving CB before the collision, $v_{i+\frac{n}{2}}$ is the velocities of the ith moving CB pair before the collision, v_i' is the velocity of the ith moving CB after the collision. In addition m_i is mass of the ith CB, $x_{i-\frac{n}{2}}$ is mass of the ith CB pair and ε is the Coefficient of Restitution (COR), which is defined as the ratio of the separation velocity of two bodies after collision to the approach velocity of two bodies before collision. For most of the real objects, it is between 0 and 1. Therefore, to control exploration and exploitation rates, COR decreases linearly from unity to zero and is defined as Eq. (20).

$$\varepsilon = 1 - \frac{iter}{iter_{max}}. \tag{20}$$

5. The new position of each CB is obtained using the new velocity of the moving bodies after collision with constant bodies using the following equation:

$$x_i^{new} = x_{i-\frac{n}{2}} + \left(rand.v_i\right) \quad i = \frac{n}{2}+1, \frac{n}{2}+2, \ldots, n, \tag{21}$$

$$x_i^{new} = x_i + \left(rand.v_i\right) \quad i = 1,2,\ldots,\frac{n}{2}, \tag{22}$$

where x_i^{new} is the new position of the moving bodies and stationary bodies in Eqs. (21) and (22), respectively. In addition, rand is a random vector uniformly distributed in the range (1, –1).

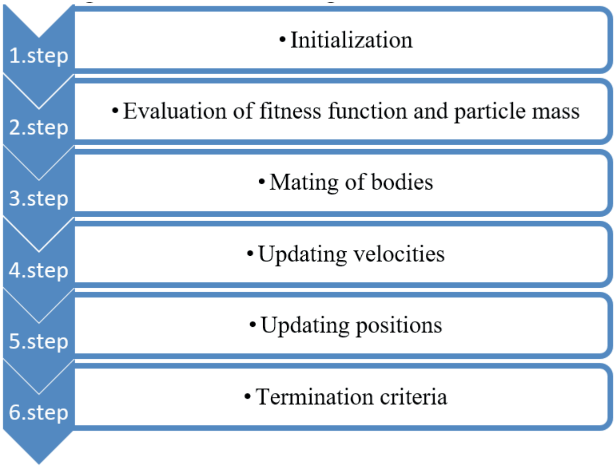


Fig. 4 Flowchart of the CBO algorithm

6. In this step, the optimization is repeated from step 2 until termination criteria is achieved such as reaching maximum iteration number. The general flowchart of this algorithm is shown in Fig. 4.

2.3 Enhanced Colliding Bodies Optimization

Enhanced Colliding Bodies Optimization (ECBO) is an improved version of the CBO algorithm, which improves the response of CBO. The main advantage of this method in comparison with CBO is the ability to utilize system memory better and thus increase its convergence speed. In addition, changes in some of the CBs parameters have increased the systems reliability and help ECBO to escape from the local optima. In short, ECBO increases the convergence speed by keeping some of the best CBs in memory and randomly changes some of the bodies features to prevent premature convergence [21]. In addition to the steps mentioned in the CBO section, this algorithm has two more steps as following:

1. First step involves storing some CB vectors that historically had best values and saves their related mass and objective function values. This step will reduce the computational time and increase the efficiency of the algorithm and is done after the second step of the CBO algorithm. Then, by creating Colliding Memory (CM), the best answers in population are stored and some of the worst answers are omitted. Finally, CBs are sorted according to their masses in a decreasing order.

2. This step is put after the fifth step of the CBO algorithm and in fact, this is the main difference between this algorithm and the original CBO. In this step, a parameter like *Pro* within (0, 1) is introduced and it is specified to determine whether a component of each CB must be changed or not. For each CB, *Pro* is compared with $rn_i (i = 1, 2, \ldots, n)$

which is a random number uniformly distributed within (0, 1). If $rn_i < Pro$, one dimension of the *i*th CB is selected randomly and its value is reproduced by Eq. (23).

$$x_{i,j} = x_{j,min} + random.\left(x_{j,max} - x_{j,min}\right), \tag{23}$$

where $x_{i,j}$ is the *j*th variable of the *i*th CB. $x_{j,min}$ and $x_{j,max}$ are the lower and upper limits of the *j*th variable respectively. In order to protect the structures of CBs, only one dimension is changed. This step helps all CBs navigate across the entire search space and help the algorithm escape from local optima.

2.4 Vibrating particles system algorithm

Vibrating Particles System (VPS) is based on the damped free vibration of Single Degree Of Freedom (SDOF) system. Fig. 5 shows a typical representation of a SDOF oscillator. The mass (m) is attached to the spring (stiffness k) and the damper (damping c). The system is forced by the random vibration function (F) in the y-direction only. In this algorithm, each candidate solution is considered as a particle that wants to approach its equilibrium position [22].The particles are initialized randomly in a n-dimensional search space and gradually approach to their equilibrium positions. The VPS steps are introduced as follows:

1. The parameters of the VPS are determined, and then the initial location of all particles in an n-dimensional search space is randomly selected.

2. The objective function value is calculated for each particle.

3. For each particle, three equilibrium positions are considered with different weights that the particle tends to approach.

a) Best position achieved in the entire population (*HB*).
b) A good particle (*GP*).
c) A bad particle (*BP*).

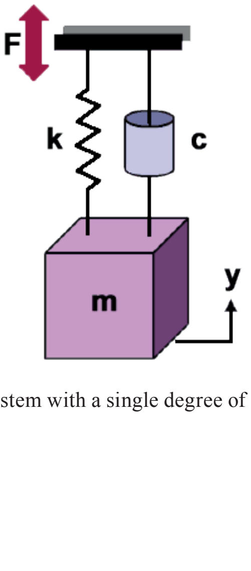


Fig. 5 A system with a single degree of freedom

In order to select the *GP* and *BP* for each candidate solution, the current population is sorted according to their objective function values in an increasing order, and then *GP* is chosen from the first half and *BP* is chosen from the second half randomly.

In order to model the damping level in the vibration in the optimization algorithm, a descending function is used (Eq. (24)).

$$D = \left(\frac{iteration}{iteration_{max}}\right)^{-\alpha}. \tag{24}$$

Where *iteration* is the current iteration number and $iteration_{max}$ is the total number of iterations in the optimization process and *α* is a constant value. Based on mentioned points, location of each particle is updated using Eq. (25–27).

$$x_j^i = w_1.\left[D.A.Rand1 + HB^j\right] + w_2.\left[D.A.Rand2 + GP^j\right] + w_3.\left[D.A.Rand3 + BP^j\right]. \tag{25}$$

$$A = \left[w_1.\left(HB^j - x_i^j\right)\right] + \left[w_2.\left(GP^j - x_i^j\right)\right] + \left[w_3.\left(BP^j - x_i^j\right)\right]. \tag{26}$$

$$w_1 + w_2 + w_3 = 1, \tag{27}$$

where x_i^j is the *j*th variable of particle *i*; w_1, w_2, and w_3 are three parameters to measure the relative importance of *HB*, *GP*, and *BP*, respectively; and *Rand*1, *Rand*2 and *Rand*3 are random numbers uniformly distributed in the range of (0, 1).

A parameter such as p is defined in the range of (0, 1) to determine whether BP affects the position of the particle in each position updating or not. So, for each particle, the value of *p* is compared with the value of the rand parameter and if $p < Rand$, then $w_3 = 0$ and $w_2 = 1 - w_1$. In this algorithm, three essential concepts consisting of self-adaptation, cooperation, and competition are considered. Self-adaptation is provided by moving particles toward *HB* and each particle can affect other particles positions, so cooperation between the particles is also supplied. Because of the *p* parameter, the influence of *GP* is more than *BP*, and therefore the competition is provided.

1. Particles are moving in the search space to find the best possible position, but sometimes may move out of the boundaries of the search space, so they must be controlled and regenerated by harmony search-based side constraint handling approach [30]. In this technique, there is a possibility like HMCR (harmony memory considering rate) that specifies violating component should be regenerated

considering the corresponding component with historically best position of a random particle, or it should be determined randomly in the search space. Moreover, if the component of a historically best position is selected, there is a possibility like PAR (pitch adjusting rate) that specifies whether this value should be changed with the neighboring value or not.

2. The second to fourth steps are repeated until reaching specified iteration number. Additional stopping criteria can also be considered according to the nature of the optimization problem. In this study, the optimization process is terminated after a fixed number of iterations. The general flowchart of this algorithm is shown in Fig. 6.

2.5 Genetic Algorithm

The Genetic Algorithm (GA) is a random-based classical evolutionary algorithm which is based on Darwin theory of evolution [31]. GA is a method for solving both constrained and unconstrained optimization problems based on a natural selection process that mimics biological evolution. This algorithm repeatedly modifies a population of individual solutions slowly. At each step, GA randomly selects individuals from the current population and uses them as parents to produce the children for the next generation. Over successive generations, the population evolves toward an optimal solution.

The genetic algorithm consists of five steps: Initialization, Fitness assignment, selection, crossover and mutation. This algorithm works on a population consisting of some solutions where the population size is the number of solutions. Each solution is called individual and each individual solution has a chromosome, which is represented as a set of parameters that included a set of genes. Similar to natural selection, the central concept of GA selection is fitness. The chromosomes that are more fit have a better chance for survival. Fitness is a function that measures the quality of the solution represented by the chromosome. In essence, each chromosome within the population represents the input parameters. General flow chart of the GA is illustrated in Fig. 7.

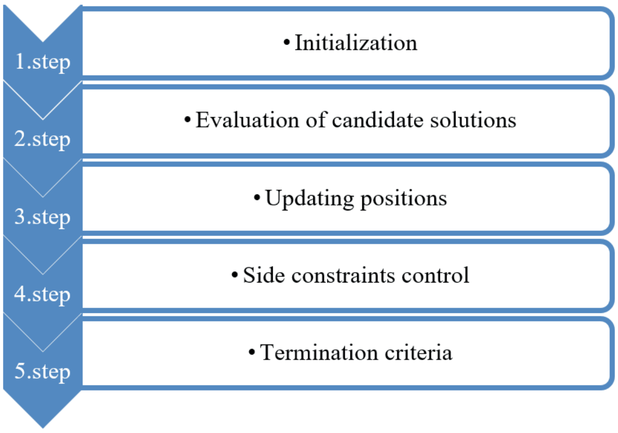


Fig. 6 Flowchart of VPS

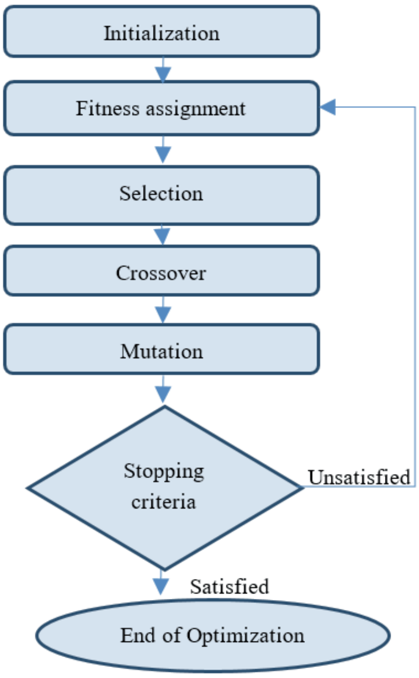


Fig. 7 Flowchart of GA

2.6 Particle swarm optimization

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Eberhart and Kennedy inspired by social behavior of bird flocking or fish schooling [32]. Similar to other metaheuristic methods.

PSO is initialized with a population of random designs, named particles that are updated in each generation to search the optimum. In this method, each particle represents a candidate solution and has a location and velocity vector that changes during optimization. In each generation, each particle updates its speed and position according to its and the total population best position. This process is repeated until the evolution of the entire society converges to the best position of the search space. The PSO algorithm uses simple moving rules for particle characteristics (Eq. (28–29)).

$$X_i^{k+1} = X_i^k + V_i^{k+1}, \tag{28}$$

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 \left(P_i^k - X_i^k\right) + c_2 r_2 \left(P_g^k - X_i^k\right). \tag{29}$$

The new particle position X_i^{k+1} is achieved by adding a new speed V_i^{k+1} to the current position X_i^k. V_i^k, P_i^k and P_g^k are previous velocity, the best position obtained by each particle and the best solution that any particle has found so far, respectively. ω is a fixed weight to control the effect of the previous velocity. r_1 and r_2 are two random numbers uniformly distributed in the range of (0, 1), and c_1 and c_2 are two learning factors which control the influence of the cognitive and social components. General flow chart of PSO algorithm is shown in Fig. 8.

2.7 Proposed models

In the adaptive neural-fuzzy inference system, least square error (LSE) and steep descend error (SDE) determines the primary and secondary parameters. One of the disadvantages of gradient-based methods is the possibility of being trapped in local optimal areas. For this reason, the use of meta-heuristic methods such as CBO, ECBO and VPS are used as an alternative and useful method for the referred approaches. In this study, initial parameters ω_i, σ_i, c_i in Eq. (5) which are related to the membership function are optimized using the mentioned three algorithms. Each of these parameters contains n particles, which are equal to the number of membership functions. The obtained parameters p_i, q_i, r_i in Eq. (8) can also be trained during the optimization algorithm. At last, to estimate the maximum scour depth near the bridge pier using CBO, ECBO and VPS algorithms, the weight parameter ω_i is determined.

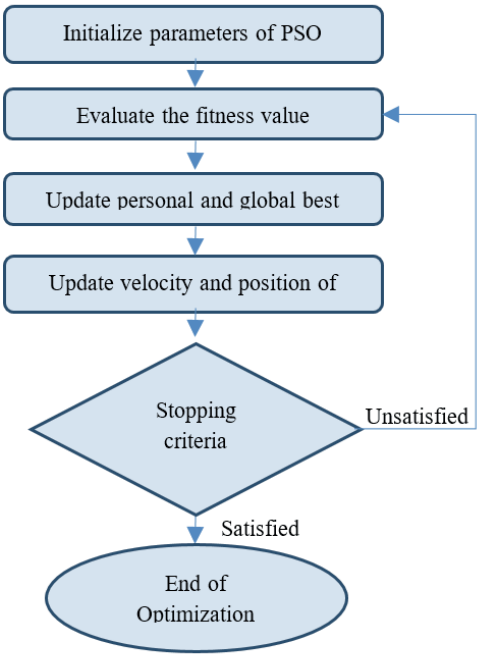

Initialize parameters of PSO
Evaluate the fitness value
Update personal and global best
Update velocity and position of
Stopping criteria
Unsatisfied
Satisfied
End of Optimization

Fig. 8 Flowchart of PSO

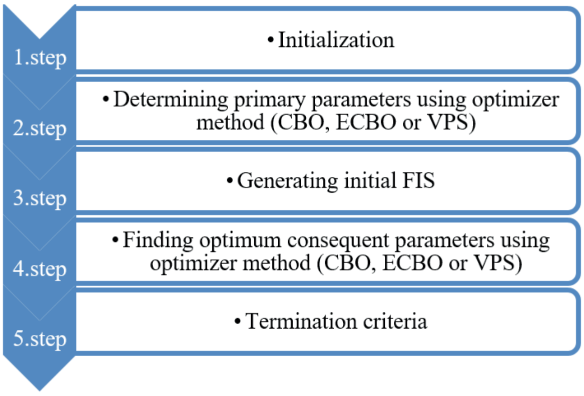

1.step
2.step
3.step
4.step
5.step
• Initialization
• Determining primary parameters using optimizer method (CBO, ECBO or VPS)
• Generating initial FIS
• Finding optimum consequent parameters using optimizer method (CBO, ECBO or VPS)
• Termination criteria

Fig. 9 Flowchart of the proposed models

To fulfill this aim, at first a random search space is generated for ANFIS parameters. Then, these parameters should be updated by using the utilized optimization algorithm. This process should be repeated until a stopping criteria is satisfied. Fig. 9 summarizes the proposed model using an optimization algorithm (CBO, ECBO or VPS). The aim is to minimize one of the goodness functions such as Root Mean Square Error (RMSE), Coefficient of Determination (R^2), Mean Absolute Error (MAE) or Mean Absolute Percentage Error (MAPE). These functions are described briefly in Section 3.1. Afterwards, the error value between ANFIS and the target value is determined that is used as an index to be optimized by the utilized optimization methods. This means that the searching process should be continued until a good error value is obtained.

The main limitation of using different algorithms is that special parameters (i.e., algorithm-specific parameters) are required for a good working of these algorithms. Proper tuning of these parameters is essential for the searching of the optimum solution by these algorithms. A change in the algorithm-specific parameters changes the effectiveness of the algorithm. Therefore in this study the ANFIS parameters were automatically found using the optimization algorithms. Therefore, control parameters of the final algorithm are just limited to the number of iteration and optimization algorithm parameters. This helps the user utilize ANFIS directly without the need for tuning some special parameters.

2.8 Utilized data

In the present study, due to the simplifications carried out in most laboratory works and complexities of the local scouring phenomenon, the FHWA dataset were used [19]. This dataset consisted of 439 measurement at 79 bridge in 17 states [5]. In this paper, 237 of 439 measurements were used, and the rest were omitted for the following reasons.

The data related to scouring in cohesive soils was removed because the time required to reach the maximum scour depth was so longer compared to granular soils [33]. The data related to bridges with large accumulation of debris from past floods was also removed due to the high scouring caused by them. In addition, the data related to the pier group due to differences in the nature of the pier layouts and all data with zero or indeterminate scour depths, as well as data with unknown bed material was removed. Similar to Pal et al. [34] and based on the above points, the present study consists of seven input parameters: pier shape (S_h), depth of flow (h), flow velocity (V), pier width (W), pier angle with flow (S_k), grain size of bed material (d_{50}) and gradation of bed material (g) that were used to determine the scour depth (h_s) at bridge pier. The specification of the used data in this study is shown in Table 1.

3 Numerical results

3.1 Evaluation functions

In this study, five evaluation functions are used including Root Mean Square Error (RMSE), Coefficient of Determination (R^2), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). These functions are described as follows:

RMSE is the difference between the predicted value of the model and the real value and it is good tool for evaluating prediction ability of the model. In this method because of the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors. Therefore, RMSE is more useful when large errors are especially undesirable (Eq. (30)).

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{\left(x_i - y_i\right)^2}{n}}. \tag{30}$$

Coefficient of determination represents the correlation of real data with predicted values and shows mathematically how much predicted data is close to the fitted regression line. R^2 values indicate the degree of adaptation between predicted data and actual values. The values close to one represent the high accuracy of the model and greater coverage of the dependent variable by the independent variables (Eq. (31)).

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum_{i=1}^{n}\left(x_i - y_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2}. \tag{31}$$

MAE measures the average magnitude of the errors without considering their direction and all the differences between predicted values and real ones are considered with the same weight (Eq. (32)).

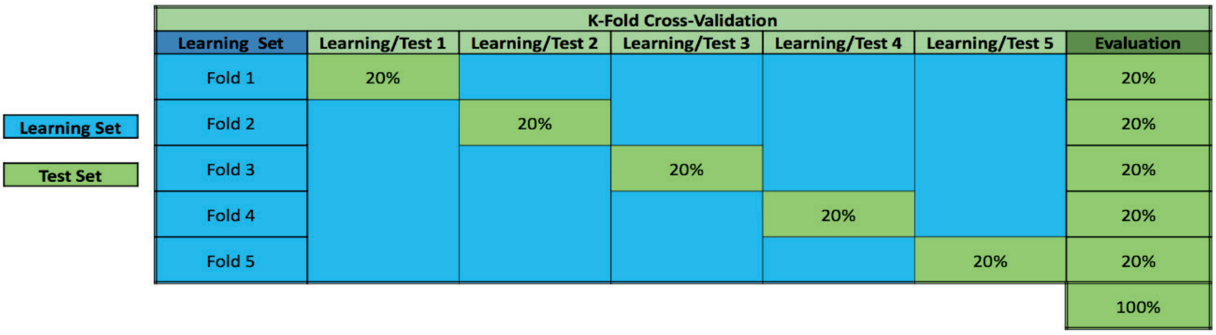
$$MAE = \frac{\sum_{i=1}^{n}\left(x_i - y_i\right)}{n}. \tag{32}$$

MAPE is a tool for evaluating prediction accuracy of forecasting models and expresses accuracy of model as a percentage. This method is scale sensitive and when working with low-volume data, answers are unreliable. MAPE functions best when there are no extremes to the data (including zeros). It should be considered with zeros or near-zeros, MAPE can give a distorted picture of error. The error on a near-zero item can be infinitely high, causing a distortion to the overall error rate when it is averaged in (Eq. (33)).

$$MAPE = \frac{100}{n} \times \sum_{i=1}^{n} \frac{\left(x_i - y_i\right)}{x}. \tag{33}$$

3.2 k-fold cross validation

To evaluate the models efficiency, 80 percent of existing data was used for training and 20 percent was used for test of the model. As a result, from 237 available data, 190 were used for model training and 47 were used for testing of the models. In order to investigate the model's dependence on selected data, k-fold cross validation was employed for two times to limit problems like overfitting



K-Fold Cross-Validation

Learning Set	Learning/Test 1	Learning/Test 2	Learning/Test 3	Learning/Test 4	Learning/Test 5	Evaluation
Fold 1	20%					20%
Fold 2		20%				20%
Fold 3			20%			20%
Fold 4				20%		20%
Fold 5					20%	20%
						100%

Fig. 10 5-fold cross validation

and give an insight on how the model will generalize to an independent dataset as illustrated in Fig. 10. By using k-fold cross validation method and RMSE, R2, MAE and MAPE evaluation functions, efficiency and applicability of the proposed models were examined and the results were presented in Tables 2–4.

3.3 Results of the new models

Tables 2–4 shows that the A-VPS model has better results than the other models. The A-ECBO model takes the second place while the A-CBO model is the worst one. Comparison of the obtained results shows that the A-VPS model, in contrast to the other two models, obtained identical results for

Table 1 Input data

	Notation	Parameters	Minimum	Maximum	Average	Standard deviation
Pier width (m)	W	x_1	0.3	5.5	1.555	1.179
Flow velocity (m/s)	V	x_2	0	5.4	1.529	0.819
Depth of flow (m)	h	x_3	0.2	22.5	4.313	3.899
Pier shape	S_h	x_4	0.7	1.3	0.980	0.204
Pier angle with flow (Deg)	S_k	x_5	0	85	9.274	18.360
Grain size of bed material	d_{50}	x_6	0.12	95	17.968	25.132
Gradation of bed material	g	x_7	1.2	21.8	3.728	3.139
Scour depth (m)	h_s	y	0.1	7.7	1.124	1.326

Table 2 Results of the A-CBO model for 10 independent runs

		1	2	3	4	5	6	7	8	9	10	Average	Standard deviation
Train	R^2	0.846	0.836	0.836	0.874	0.820	0.822	0.843	0.863	0.779	0.881	0.840	0.028
	MAPE	56.5	39.0	17.0	42.7	67.9	68.3	64.5	61.0	68.1	44.2	52.9	16.022
	RMSE	0.493	0.559	0.741	0.493	0.564	0.605	0.536	0.502	0.596	0.418	0.551	0.083
	MAE	0.320	0.295	0.350	0.249	0.372	0.421	0.352	0.337	0.422	0.291	0.341	0.053
Test	R^2	0.666	0.645	0.645	0.665	0.830	0.689	0.766	0.708	0.781	0.753	0.715	0.061
	MAPE	56.7	69.6	69.6	54.7	49.9	73.9	82.7	55.8	65.1	50.1	62.8	10.469
	RMSE	0.892	0.705	0.705	0.687	0.552	0.607	0.820	0.702	0.684	0.788	0.714	0.094
	MAE	0.489	0.466	0.466	0.422	0.379	0.450	0.589	0.456	0.487	0.503	0.471	0.052

Table 3 Results of the A-ECBO model for 10 independent runs

		1	2	3	4	5	6	7	8	9	10	Average	Standard deviation
Train	R^2	0.883	0.862	0.873	0.853	0.863	0.883	0.834	0.842	0.870	0.765	0.853	0.033
	MAPE	49.6	54.6	24.2	59.8	61.0	49.6	59.0	66.1	41.6	61.4	52.7	11.735
	RMSE	0.458	0.519	0.395	0.461	0.502	0.458	0.526	0.490	0.458	0.663	0.493	0.067
	MAE	0.299	0.365	0.303	0.329	0.337	0.299	0.345	0.357	0.254	0.390	0.328	0.038
Test	R^2	0.796	0.760	0.895	0.821	0.708	0.758	0.759	0.804	0.839	0.746	0.789	0.051
	MAPE	47.7	76.5	61.9	55.6	55.8	47.7	53.8	65.2	66.1	65.2	59.6	8.603
	RMSE	0.484	0.585	0.612	0.660	0.702	0.637	0.786	0.676	0.580	0.671	0.639	0.077
	MAE	0.331	0.427	0.353	0.411	0.456	0.381	0.462	0.511	0.389	0.442	0.416	0.052

Table 4 Results of the A-VPS model for 10 independent runs

		1	2	3	4	5	6	7	8	9	10	Average	Standard deviation
Train	R^2	0.866	0.836	0.882	0.859	0.857	0.888	0.796	0.897	0.871	0.890	0.864	0.029
	MAPE	47.6	48.2	24.3	16.9	53.5	50.1	59.2	51.8	53.0	46.2	45.1	12.837
	RMSE	0.491	0.531	0.427	0.483	0.515	0.454	0.599	0.402	0.492	0.438	0.483	0.054
	MAE	0.278	0.318	0.281	0.331	0.319	0.299	0.397	0.284	0.327	0.270	0.310	0.036
Test	R^2	0.917	0.893	0.899	0.801	0.885	0.816	0.812	0.929	0.788	0.872	0.861	0.049
	MAPE	45.6	63.0	48.9	59.8	52.3	58.2	66.9	54.1	58.8	48.4	55.6	6.522
	RMSE	0.474	0.491	0.405	0.518	0.425	0.545	0.621	0.395	0.559	0.509	0.494	0.068
	MAE	0.342	0.353	0.286	0.371	0.308	0.363	0.456	0.312	0.384	0.319	0.349	0.046

teaching and testing data sets, which indicates the models power in predicting the maximum scour depth and its stability and reliability. The values obtained by the A-VPS model for the coefficient of determination and its standard deviation are 0.861 and 0.049, respectively which indicate that predicted values are very close to the actual ones. Figs. 11–13 shows a comparison of the actual and predicted values obtained by the A-CBO, A-ECBO and A-VPS models for the best testing fold. As shown in Fig. 14, although there is no significant difference between the models in the training data set but in the testing data set, proposed models obtain good results. Fig. 15 shows how the number of iterations affects the models accuracy. As shown in the Fig. 15 A-VPS has the highest convergence rate and A-ECBO has the second highest convergence rate. Convergence history of the methods shows that almost after the 550th iteration, changes are very small in all models, but for accurate investigating of the model performance iteration continued until 1000. The optimal training times for the A-CBO, A-ECBO and A-VPS models were 92, 61, and 27 s, respectively.

A statistical analysis of the obtained results by the A-ECBO model in comparison to the A-CBO model shows that the performance of the A-ECBO model is improved. For example, when comparing the values of MAPE, R^2 and RMSE obtained by these two models, MAPE decreased by 5 %, coefficient of determination (R^2) increased by 10 % and RMSE decreased by 10.5 % for the ECBO method. In addition, comparing the obtained values by the A-ECBO model for training and testing data sets shows that it is more consistent than the A-CBO model.

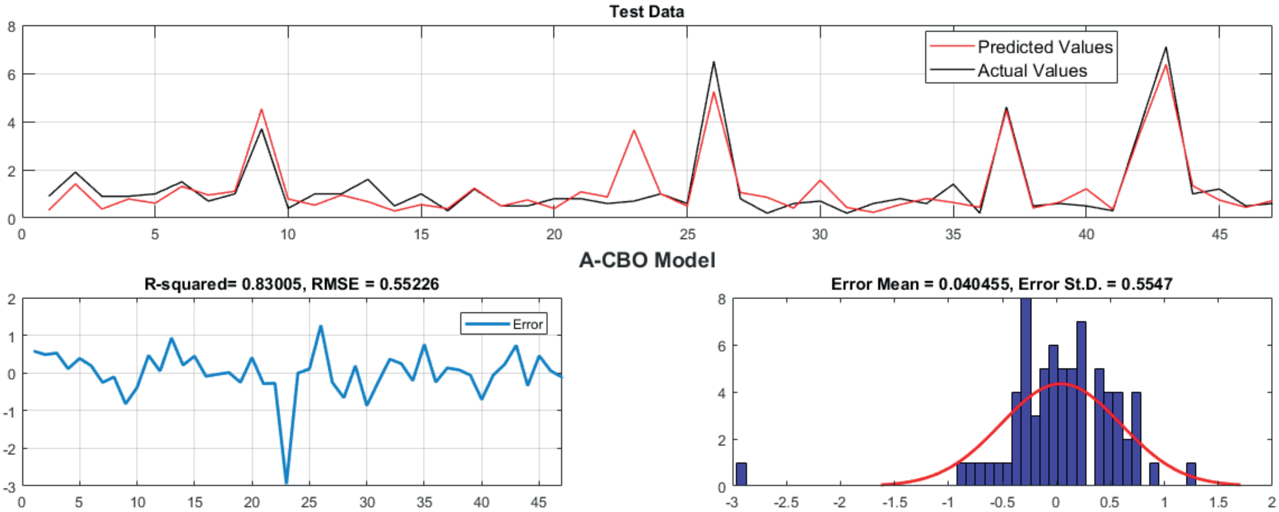


Fig. 11 Actual and predicted values obtained by A-CBO model for the best testing fold

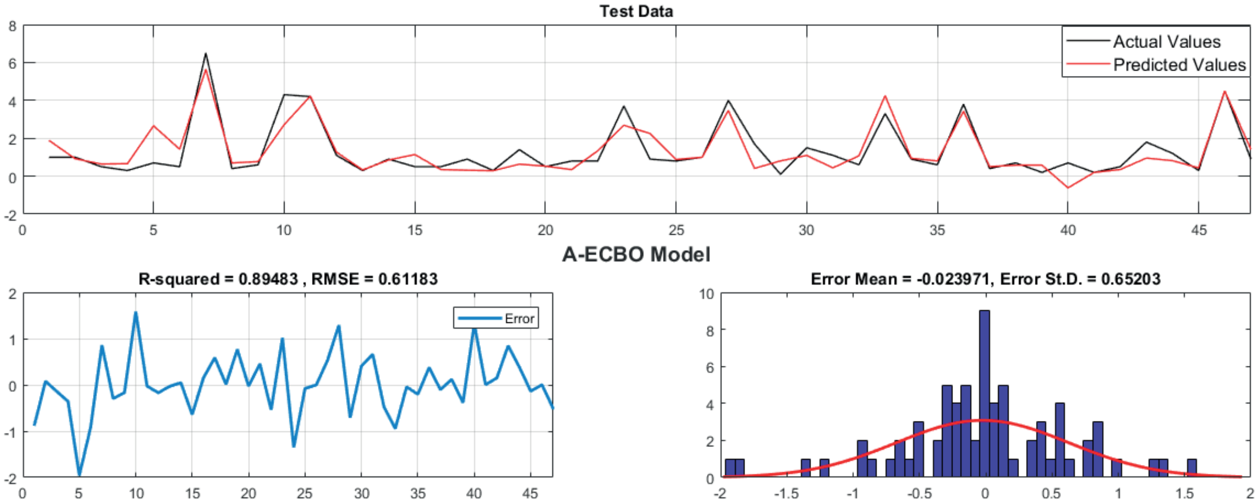


Fig. 12 Actual and predicted values obtained by A-ECBO model for the best testing fold

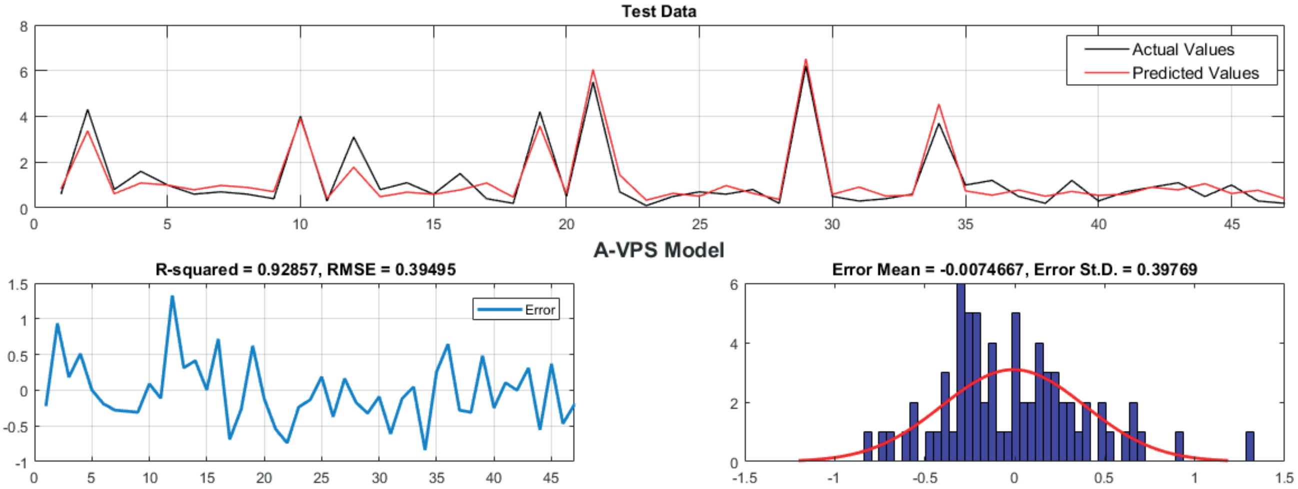


Fig. 13 Actual and predicted values obtained by A-VPS model for the best testing fold

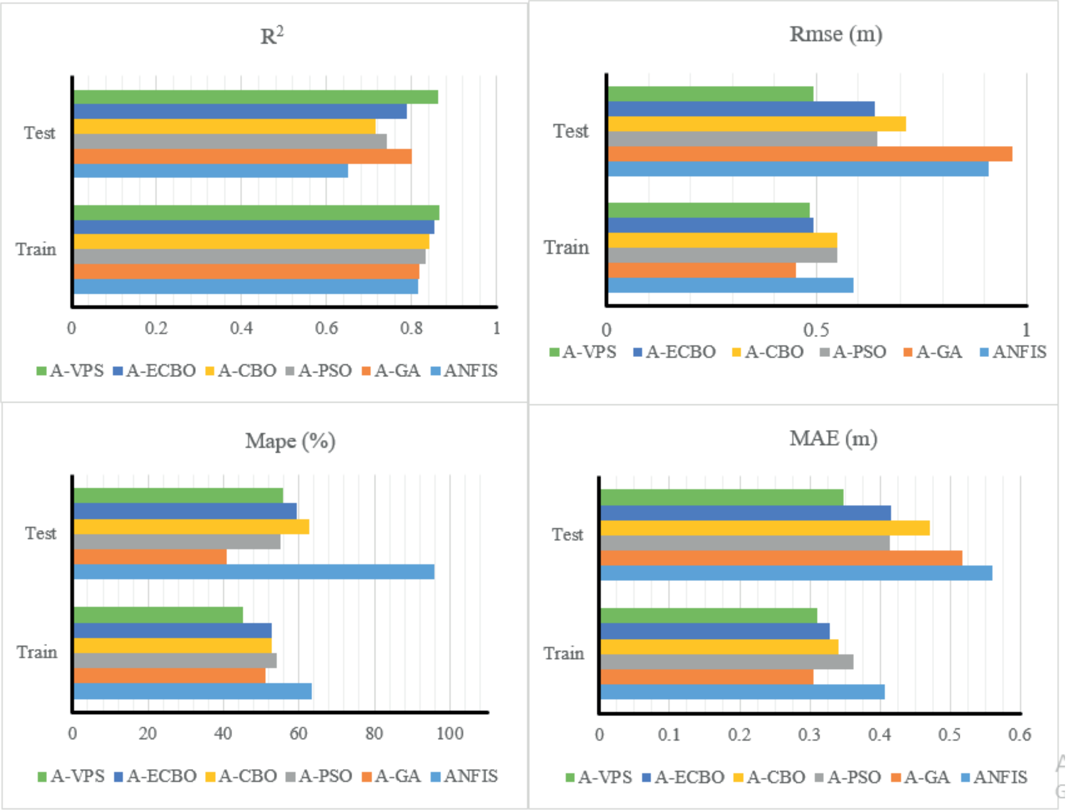


Fig. 14 Average evaluation function values

3.4 Comparison of results with the standard and modified ANFIS

To evaluate the performance of the proposed models and the effect of used optimization methods, the results of the proposed models are compared with results of the standard and modified versions of ANFIS using GA and PSO (Table 5). Since modeling process is similar for modified models, the dependence of the results on the type of data selection or creating the model is eliminated and models can be compared in the same conditions. Values of RMSE, MAE, MAPE and R^2 for all three models for the training and validation, were used to rank models according to accuracy (5–1, from best to worst score for each statistic), and the ranks were summed to obtain an overall accuracy score for each model. In general, the best values of RMSE, MAE, MAPE and R^2 were found for the A-VPS model. Comparing the results shows that the combination of all three optimization methods with ANFIS increases its

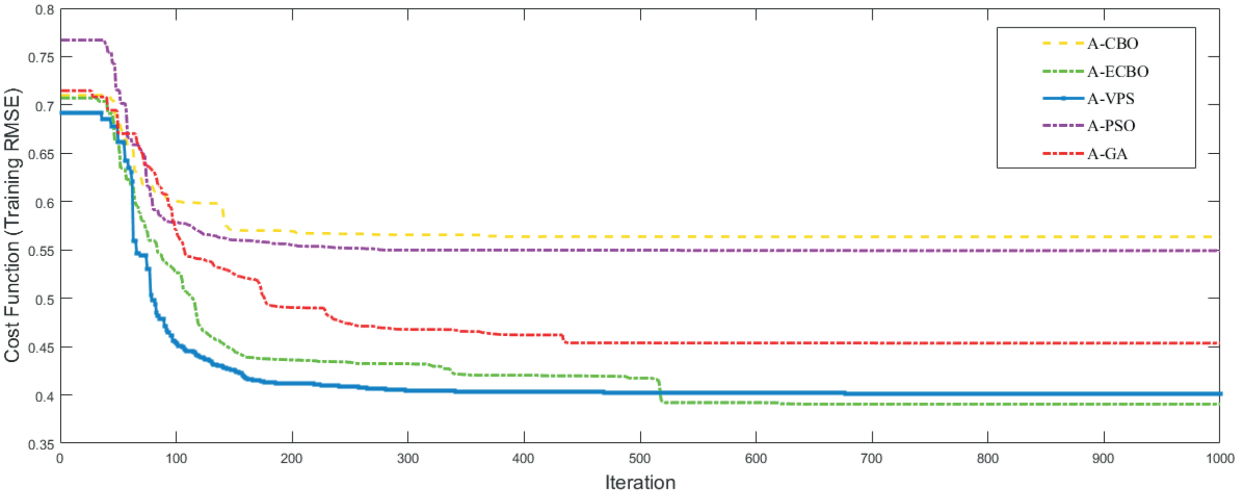


Fig. 15 Convergence history for the best testing fold

efficiency. Obtained results for MAPE and RMSE show that using VPS method reduced MAPE and RMSE about 42 % and 46 % respectively compared to standard ANFIS. The standard ANFIS shows good performance in the training step; however, it is unable to work well in the field of validation with new data. Results of modified versions of ANFIS show that A-VPS and A-CBO have the best and worst results, respectively. Even though GA and PSO improve ANFIS performance, ECBO and VPS obtain better results when considering determination coefficient and RMSE as benchmarks. For instance, VPS obtains 49 % better result than GA and 23 % than PSO considering RMSE. However, A-VPS model yields similar results in both data sets and this is due to the high power of the model in predicting the maximum scour depth. Therefore, the new methods are effective, since the model errors in both training and testing data sets are decreased and the reliability of the model is increased. Moreover, the average optimized results and standard deviation on averages results obtained by A-VPS are competitive with the other optimization methods. The convergence speed comparisons also reveal the fast converging feature of the A-VPS model.

3.5 Comparison of results with empirical relationships

One of the common methods for estimating the scour depth near bridge pier is the use of empirical relationships. A lot of research has been done by researchers in this field that has led to a variety of empirical relationships [11, 35–41]. The results of scour depth estimation with the empirical relationships are compared with the results of the proposed models (Table 6). Analysis of the results shows that the empirical relationships indicate a poor performance in scour depth estimation. Empirical equations in terms of coefficient of determination and RMSE do not reach good values. The negative values obtained for coefficient of determination indicate the weakness of Laursen and Toch, Mississippi and HEC-18 approaches because the negative values show that the model does not follow the trend of the data. Comparison between the results shows that the Froehlich has a better performance than other ones, however it has a higher RMSE (lower coefficient of determination as well) than the proposed models. The A-VPS model compared to the Froehlich relationship has 58 % higher R^2 and 45 % lower RMSE value.

3.6 Sensitivity analysis

In order to determine the relative influence of the input parameters on the pier scour depth, 10 tests were performed for each parameter and the average value was calculated. The A-VPS model, known as the best among proposed models, was selected to be used in this section. To evaluate relative significance of the input parameters, they were eliminated in turn, and the effect of removing each one on the models was determined by using the evaluation functions. Table 7 summarizes the results of all tests performed with different inputs of the model. Comparison of the results shows that the width of the pier had the major influence on predicting the scour depth and gradation of bed materials had the least effect. Other effective parameters on scour depth ranking from higher to lower values include the depth of the flow, flow velocity, pier shape, pier angle with flow, grain size of bed material (d_{50}) and gradation of bed material. The comparison of the results shows that width of the pier has a significant effect on the prediction of scour depth because its removal has increased RMSE by 159 % and reduces determination coefficient by 33 %.

Table 5 Results of the modified and standard ANFIS-based methods

Model	Train				Test			
	R^2	MAPE	RMSE	MAE	R^2	MAPE	RMSE	MAE
ANFIS	0.814	63.3	0.588	0.407	0.651	95.8	0.910	0.560
A-GA	0.818	51.2	0.451	0.305	0.799	40.9	0.966	0.517
A-PSO	0.833	54.2	0.549	0.362	0.742	55.1	0.645	0.415
A-CBO	0.840	52.9	0.551	0.341	0.715	62.8	0.714	0.471
A-ECBO	0.853	52.7	0.493	0.328	0.789	59.6	0.639	0.416
A-VPS	0.864	45.1	0.483	0.310	0.861	55.6	0.494	0.349

Table 6 Comparison of the modified models with the empirical equations

	A-CBO	A-ECBO	A-VPS	Mississippi	HEC-18	Froehlich	Laursen and Toch
R^2	0.715	0.789	0.864	–2.24	–3.620	0.547	–1.52
RMSE	0.714	0.639	0.493	2.383	2.840	0.890	2.101

Table 7 Results of the sensitivity analysis

		Pier Shape	Pier Width	Skew	Velocity	Depth	D50	gradation
Train	R^2	0.863	0.899	0.880	0.851	0.932	0.882	0.856
	MAPE	47.0	54.2	47.3	43.4	42.2	42.0	59.4
	RMSE	0.455	0.402	0.469	0.515	0.323	0.471	0.501
	MAE	0.286	0.272	0.300	0.322	0.246	0.289	0.338
Test	R^2	0.696	0.577	0.766	0.687	0.659	0.795	0.815
	MAPE	67.7	73.5	79.8	77.0	57.7	59.0	53.2
	RMSE	0.898	1.277	0.582	0.751	0.967	0.546	0.576
	MAE	0.514	0.687	0.414	0.461	0.558	0.343	0.383

4 Conclusions

In this study, three hybrid models were developed to predict bridge scours. The proposed models operate automatically without human intervention, and enhances the ANFIS accuracy in predicting the scour depth under various natural conditions. The proposed models were based on ANFIS and CBO, ECBO or VPS algorithms. To develop the model, the United State Federal Highway Administration scour data set were used to train and validate the model. In order to investigate the models dependence on selected data, 5-fold cross validation was employed and finally the results of the proposed models were compared with standard and modified ANFIS as well as empirical equations.

Comparing the results showed that applied optimizations on the standard ANFIS were very effective and increase the model capability and reliability. The A-VPS model had better results than the other two proposed models and the A-ECBO model showed better performance than A-CBO model. For instance, using VPS method reduced MAPE and RMSE about 42 % and 46 % respectively and increased the determination coefficient by 33 % compared to standard ANFIS. Comparison of the performance of used optimization methods showed that although GA and PSO improved ANFIS, VPS and ECBO algorithm can obtain better results. A-VPS model obtained identical results for teaching and testing data sets, which indicates the models power in predicting the maximum scour depth and its reliability. In addition, comparison of the results of the proposed models with the results of empirical relations showed that all of the three proposed models could find better results. Results of empirical equations showed that some of the formulas are not suitable for estimating local scour due to the high value of errors. While proposed models showed good performance and they can estimate bridge pier scour with good accuracy.

References

[1] Chiew, Y.-M. "Scour Protection at Bridge Piers", Journal of Hydraulic Engineering, 118(9), pp. 1260–1269, 1992. https://doi.org/10.1061/(ASCE)0733-9429(1992)118:9(1260)

[2] Clopper, P. E., Lagasse, P. F., Zevenbergen, L. W. "Bridge Pier Scour Countermeasures", presented at World Environmental and Water Resources Congress 2007, Tampa, FL, USA, May 15–19, 2007. https://doi.org/10.1061/40927(243)380

[3] Lu, J.-Y., Hong, J.-H., Su, C.-C., Wang, C.-Y., Lai, J.-S. "Field Measurements and Simulation of Bridge Scour Depth Variations during Floods", Journal of Hydraulic Engineering, 134(6), pp. 810–821, 2008. https://doi.org/10.1061/(ASCE)0733-9429(2008)134:6(810)

[4] Sheppard, D. M., Melville, B., Demir, H. "Evaluation of Existing Equations for Local Scour at Bridge Piers", Journal of Hydraulic Engineering, 140(1), pp. 14–23, 2013. https://doi.org/10.1061/(ASCE)HY.1943-7900.0000800

[5] Mueller, D. S., Wagner, C. R. "Field Observations and Evaluations of Streambed Scour at Bridges", Office of Engineering Research and Development Federal Highway Administration, McLean, VA, USA, Rep. FHWA–RD–03–052, 2005. [online] Available at: https://www.fhwa.dot.gov/publications/research/infrastructure/hydraulics/03052/03052.pdf [Accessed: 10 June 2019]

[6] Bateni, S. M., Borghei, S., Jeng, D.-S. "Neural network and neuro-fuzzy assessments for scour depth around bridge piers", Engineering Applications of Artificial Intelligence, 20(3), pp. 401–414, 2007. https://doi.org/10.1016/j.engappai.2006.06.012

[7] Zounemat-Kermani, M., Beheshti, A.-A., Ataie-Ashtiani, B., Sabbagh-Yazdi, S.-R. "Estimation of current-induced scour depth around pile groups using neural network and adaptive neuro-fuzzy inference system", Applied Soft Computing, 9(2), pp. 746–755, 2009. https://doi.org/10.1016/j.asoc.2008.09.006

[8] Akib, S., Mohammadhassani, M., Jahangirzadeh, A. "Application of ANFIS and LR in prediction of scour depth in bridges", Computers & Fluids, 91, pp. 77–86, 2014. https://doi.org/10.1016/j.compfluid.2013.12.004

[9] Ebtehaj, I., Bonakdari, H., Zaji, A. H., Sharafi, H. "Sensitivity analysis of parameters affecting scour depth around bridge piers based on the non-tuned, rapid extreme learning machine method", Neural Computing and Applications, pp. 1–12, 2018. https://doi.org/10.1007/s00521-018-3696-6

[10] Balouchi, B., Nikoo, M. R., Adamowski, J. "Development of expert systems for the prediction of scour depth under live-bed conditions at river confluences: Application of different types of ANNs and the M5P model tree", Applied Soft Computing, 34, pp. 51–59, 2015. https://doi.org/10.1016/j.asoc.2015.04.040

[11] Cheng, M.-Y., Cao, M.-T., Wu, Y.-W. "Predicting Equilibrium Scour Depth at Bridge Piers Using Evolutionary Radial Basis Function Neural Network", Journal of Computing in Civil Engineering, 29(5), 2015. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000380

[12] Choi, S.-U., Cheong, S. "Prediction of local scour around bridge piers using artificial neural networks", JAWRA Journal of the American Water Resources Association, 42(2), pp. 487–494, 2006. https://doi.org/10.1111/j.1752-1688.2006.tb03852.x

[13] Choi, S.-U., Choi, B., Lee, S. "Prediction of local scour around bridge piers using the ANFIS method", Neural Computing and Applications, 28(2), pp. 335–344, 2017. https://doi.org/10.1007/s00521-015-2062-1

[14] Etemad-Shahidi, A., Yasa, R., Kazeminezhad, M. H. "Prediction of wave-induced scour depth under submarine pipelines using machine learning approach", Applied Ocean Research, 33(1), pp. 54–59, 2011. https://doi.org/10.1016/j.apor.2010.11.002

[15] Firat, M., Gungor, M. "Generalized Regression Neural Networks and Feed Forward Neural Networks for prediction of scour depth around bridge piers", Advances in Engineering Software, 40(8), pp. 731–737, 2009. https://doi.org/10.1016/j.advengsoft.2008.12.001

[16] Azamathulla, H. M., Ab Ghani, A., Zakaria, N. A., Guven, A. "Genetic Programming to Predict Bridge Pier Scour", Journal of Hydraulic Engineering, 136(3), pp. 165–169, 2009. https://doi.org/10.1061/(ASCE)HY.1943-7900.0000133

[17] Najafzadeh, M., Saberi-Movahed, F., Sarkamaryan, S. "NF-GMDH-Based self-organized systems to predict bridge pier scour depth under debris flow effects", Marine Georesources & Geotechnology, 36(5), pp. 586–602, 2018. https://doi.org/10.1080/1064119X.2017.1355944

[18] Chou, J.-S., Pham, A.-D. "Hybrid computational model for predicting bridge scour depth near piers and abutments", Automation in Construction, 48, pp. 88–96, 2014. https://doi.org/10.1016/j.autcon.2014.08.006

[19] Chou, J.-S., Pham, A.-D. "Nature-inspired metaheuristic optimization in least squares support vector regression for obtaining bridge scour information", Information Sciences, 399, pp. 64–80, 2017. https://doi.org/10.1016/j.ins.2017.02.051

[20] Kaveh, A., Mahdavi, V. R. "Colliding bodies optimization: A novel meta-heuristic method", Computers & Structures, 139, pp. 18–27, 2014. https://doi.org/10.1016/j.compstruc.2014.04.005

[21] Kaveh, A., Ilchi Ghazaan, M. "Enhanced colliding bodies optimization for design problems with continuous and discrete variables", Advances in Engineering Software, 77, pp. 66–75, 2014. https://doi.org/10.1016/j.advengsoft.2014.08.003

[22] Kaveh, A., Ilchi Ghazaan, M. "A new meta-heuristic algorithm: vibrating particles system", Scientia Iranica, 24(2), pp. 551–566, 2017. https://doi.org/10.24200/sci.2017.2417

[23] Kaveh, A., Ilchi Ghazaan, M. "A comparative study of CBO and ECBO for optimal design of skeletal structures", Computers & Structures, 153, pp. 137–147, 2015. https://doi.org/10.1016/j.compstruc.2015.02.028

[24] Kaveh, A. "Advances in Metaheuristic Algorithms for Optimal Design of Structures", 1st ed., Springer, Cham, Switzerland, 2017. https://doi.org/10.1007/978-3-319-46173-1

[25] Kaveh, A., Rezaei, M. "Topology and geometry optimization of single-layer domes utilizing CBO and ECBO", Scientia Iranica, 23(2), pp. 535–547, 2016. https://doi.org/10.24200/sci.2016.2137

[26] Kaveh, A., Ilchi Ghazaan, M. "Vibrating particles system algorithm for truss optimization with multiple natural frequency constraints", Acta Mechanica, 228(1), pp. 307–322, 2017. https://doi.org/10.1007/s00707-016-1725-z

[27] Rabiei, M. H., Aalami, M. T., Talatahari, S. "Reservoir operation optimization using cbo ECBO and VPS algorithms", International Journal of Optimization in Civil Engineering, 8(3), pp. 489–509, 2018. [onine] Available at: http://ijoce.iust.ac.ir/browse.php?a_id=358&sid=1&slc_lang=en [Accessed: 10 June 2019]

[28] Jang, J.-S. R. "Fuzzy Modeling Using Generalized Neural Networks Kalman Filter Algorithm", In: The 9th AAAI Conference on Artificial Intelligence, Anaheim, CA, USA, 1991, pp. 762–767. [onine] Available at: https://www.aaai.org/Papers/AAAI/1991/AAAI91-119.pdf [Accessed: 10 June 2019]

[29] Haznedar, B., Kalinli, A. "Training ANFIS structure using simulated annealing algorithm for dynamic systems identification", Neurocomputing, 302, pp. 66–74, 2018. https://doi.org/10.1016/j.neucom.2018.04.006

[30] Kaveh, A., Talatahari, S. "Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures", Computers & Structures, 87(5–6), pp. 267–283, 2009. https://doi.org/10.1016/j.compstruc.2009.01.003

[31] Holland, J. H. "Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence", MIT Press, Cambridge, MA, USA, 1992. https://doi.org/10.7551/mitpress/1090.001.0001

[32] Eberhart, R., Shi, Y., Eberhart, R. "Swarm Intelligence", Morgan Kaufmann, San Francisco, CA, USA, 2001.

[33] Najafzadeh, M., Barani, G.-A., Azamathulla, H. M. "GMDH to predict scour depth around a pier in cohesive soils", Applied Ocean Research, 40, pp. 35–41, 2013. https://doi.org/10.1016/j.apor.2012.12.004

[34] Pal, M., Singh, N. K., Tiwari, N. K. "Support vector regression based modeling of pier scour using field data", Engineering Applications of Artificial Intelligence, 24(5), pp. 911–916, 2011. https://doi.org/10.1016/j.engappai.2010.11.002

[35] Richardson, E. V., Davis, S. R. "Evaluating Scour at Bridges", Federal Highway Administration Hydraulic Engineering Circular, 3rd ed., FHWA, Washington, DC, USA, Rep. FHWA-IP-90-017, 1995.

[36] Van Wilson Jr, K. "Scour at Selected Bridge Sites in Mississippi", US Geological Survey, Earth Science Information Center, Jackson, MS, USA, Rep. 94-4241, 1995. https://doi.org/10.3133/wri944241

[37] Laursen, E. M., Toch, A. "Scour Around Bridge Piers and Abutments", Iowa Institute of Hydraulic Research , Iowa City, IA, USA, Rep. PB–C-8314, 1956.

[38] Froehlich, D. C. "Analysis of Onsite Measurements of Scour at Piers", In: Hydraulic Engineering: Proceedings of the 1988 National Conference, Colorado Springs, CO, USA, 1988, pp. 534–539.

[39] Shen, H.W., Schneider, V. R., Karaki, S. "Local Scour Around Bridge Piers", Journal of the Hydraulics Division, 95(6), pp. 1919–1940, 1969.

[40] Breusers, H. N. C., Nicollet, G., Shen, H. W. "Local Scour Around Cylindrical Piers", Journal of Hydraulic Research, 15(3), pp. 211–252, 1977. https://doi.org/10.1080/00221687709499645

[41] Melville, B.W. "Local Scour at Bridge Sites", PhD Thesis, University of Auckland, 1975.