

Article

Effective Hybrid Soft Computing Approach for Optimum Design of Shallow Foundations

Mohammad Khajehzadeh ¹, Suraparb Keawsawasvong ² and Moncef L. Nehdi ^{3,*}

¹ Department of Civil Engineering, Anar Branch, Islamic Azad University, Anar 7741943615, Iran; mohammad.khajehzadeh@anariau.ac.ir

² Department of Civil Engineering, Thammasat School of Engineering, Thammasat University, Pathumthani 12120, Thailand; ksurapar@engr.tu.ac.th

³ Department of Civil Engineering, McMaster University, Hamilton, ON L8S 4M6, Canada

* Correspondence: nehdim@mcmaster.ca; Tel.: +905-525-9140 (ext. 23824)

Abstract: In this study, an effective intelligent system based on artificial neural networks (ANNs) and a modified rat swarm optimizer (MRSO) was developed to predict the ultimate bearing capacity of shallow foundations and their optimum design using the predicted bearing capacity value. To provide the neural network with adequate training and testing data, an extensive literature review was used to compile a database comprising 97 datasets retrieved from load tests both on large-scale and smaller-scale sized footings. To refine the network architecture, several trial and error experiments were performed using various numbers of neurons in the hidden layer. Accordingly, the optimal architecture of the ANN was $5 \times 10 \times 1$. The performance and prediction capacity of the developed model were appraised using the root mean square error (RMSE) and correlation coefficient (R). According to the obtained results, the ANN model with a RMSE value equal to 0.0249 and R value equal to 0.9908 was a reliable, simple and valid computational model for estimating the load bearing capacity of footings. The developed ANN model was applied to a case study of spread footing optimization, and the results revealed that the proposed model is competent to provide better optimal solutions and to outperform traditional existing methods.

Keywords: neural network; rat swarm; spread footing; optimization; bearing capacity



Citation: Khajehzadeh, M.; Keawsawasvong, S.; Nehdi, M.L. Effective Hybrid Soft Computing Approach for Optimum Design of Shallow Foundations. *Sustainability* **2022**, *14*, 1847. <https://doi.org/10.3390/su14031847>

Academic Editor: Muntasir Billah

Received: 30 December 2021

Accepted: 31 January 2022

Published: 6 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A spread footing is a geotechnical structure that transfers loads to the soil immediately beneath it. It is one of the most significant and sensitive structural components, and thus it has received a lot of attention in recent studies. Structures' functionality can be jeopardized unless the effective loads are successfully transmitted to the earth by a well-designed foundation. As a result, the proper design of shallow footings is paramount to ensure the resilience of the structures they support. Spread footing structures are widely used and they typically involve large volumes of materials in their construction. Therefore, economical design of these structures is essential. However, in most geotechnical engineering optimization problems, the objective function is discontinuous and has a large number of design variables. These difficulties and complexity in geotechnical engineering challenges, such as shallow foundations, pile optimization, and slope stability and liquefaction, have prompted concerted research to develop new optimization techniques and approaches for the solution of these problems [1–3].

The ultimate bearing capacity and control of foundation settlement are two requirements that must be met by every foundation design. The bearing capacity is the resistance of the foundation when maximum pressure is applied by the foundation to the soil without triggering shear failure in the soil. By taking into account these criteria, several models and techniques have been developed in laboratory and in situ experiments to evaluate the

ultimate bearing capacity, such as the theories presented by Terzaghi, Meyerhof, Hansen, Vesic, and others [4].

For both square and rectangular footings, the ultimate bearing capacity is determined by the foundation size. As a result, in terms of behavior and stress distribution, laboratory-made miniature models of footings differ from real-world footings. Therefore, when using the results of extremely small-scale model footing tests instead of full-scale behaviors, caution should be taken. Testing the actual size footing is necessary to understand true soil-foundation behavior; nevertheless, this is a time-consuming, experimentally challenging and expensive procedure. As a result of the scale effect, a majority of researchers have only focused on small-scale foundations of various sizes in the laboratory to establish final bearing capacity [5]. In addition, researchers are attempting to evaluate reliable approaches for predicting ultimate bearing capacity based on load test data from real-size foundations as well as smaller-scale model footings. Because of the variability of soils and the limitations of laboratory and field testing, a better approach for estimating bearing capacity is necessary.

Artificial neural networks (ANNs), which simulate the structure and learning mechanism of biological neural networks, are one of the most common prediction methods. ANNs are a class of parallel processing structures that work together to solve problems using highly interconnected but simple computing units called neurons. This enables the evaluation of non-linear correlations between any of the soil and foundation characteristics, as well as provides faster and more accurate results than earlier techniques. ANNs have recently been used to solve a variety of geotechnical engineering applications such as bearing capacity estimation [6,7], rock burst hazard prediction in underground projects [8], slope stability evaluation [9–11], concrete compressive strength prediction [12], and estimation of rock modulus [13]. This suggests that ANNs can be utilized for forecasting as well as prediction of events by simulating exceedingly complex functions [14]. The training procedure is one of the most important aspects of neural networks. The goal of this approach is to find the best possible connection weights and biases to attain the minimal amount of the objective function, which can be specified as root mean squared error (RMSE) or sum of squared errors (SSE). Generally, training algorithms can be divided into two categories: classic deterministic and recent metaheuristic algorithms. Classic optimization algorithms based on mathematical concepts take a long time or may not obtain the optimum solution at all. To address the aforementioned drawback, in the previous couple of decades, several metaheuristic optimization algorithms have been developed and applied for training ANNs. Some of this research includes: application of genetic algorithms [15], particle swarm optimization [16] and imperialistic competitive algorithms [9]. Although metaheuristics methods can produce acceptable results, there is no algorithm that can outperform others in solving all optimization problems. As a result, research has been conducted in an effort to modify the original algorithms' performance and efficiency in some aspects and to apply them to a specific application. [17–22].

Rat swarm optimizer (RSO) is a relatively new metaheuristic optimization approach developed by Dhiman in 2020 [23]. Compared with other metaheuristics, RSO has a simple concept and structure and does not have complicated mathematical functions. The RSO algorithm mimics the following and attacking performances of rats in nature. Like the other population-based techniques, RSO, without any information about the solution, utilizes random initialization to generate the candidate solutions. Compared to the other metaheuristics, RSO possesses several advantages. It has a very simple structure, a fast convergence rate, and can be easily understood and utilized. However, like other metaheuristic algorithms, RSO commonly suffers from getting trapped in local minima when the objective function is complex and includes a rather large number of variables.

This paper presents an effective modified version of the RSO algorithm to overcome the mentioned weaknesses and implements a new algorithm for the optimum design of spread footings. In addition, an ANN model is created and trained using the proposed, modified RSO for predicting the ultimate bearing capacity of spread footings.

2. Foundation Optimization

Reinforced spread footings, as a key geotechnical construction, must securely and reliably support the superstructure, maintain stability against excessive settlement and failure of the soil's bearing capacity, and restrict concrete stresses. Aside from these design goals, spread footings must meet a number of requirements: it must have enough shear and moment capacities in both long and short dimensions, the foundation's bearing capacity cannot be exceeded, and the steel reinforcement design must comply with all design codes.

Mathematically, the general form of a constraint optimization problem can be expressed as follows:

$$\begin{aligned} & \text{minimize } f(X) \\ & \text{subject to} \\ & g_i(X) \leq 0, i = 1, 2, \dots, p, \\ & h_j(X) = 0, j = 1, 2, \dots, m, \\ & X^L \leq X \leq X^U \end{aligned} \quad (1)$$

where X is n dimensional vector of design variables, $f(X)$ is the objective function, $g(X)$ and $h(X)$ are inequality and equality constraints, respectively. Boundary constraints, X^L and X^U , are two n -dimensional vectors containing the design variables' lower and upper bounds, respectively.

In the problem of foundation optimization, it is required that the objective function, design constraint, and design variables be identified, as presented in the following sub-sections.

2.1. Objective Function

The total cost of spread footing construction is used as the objective function in this study, and it may be represented mathematically as follows:

$$f(X) = C_c V_c + C_e V_e + C_b V_b + C_f A_f + C_s W_s \quad (2)$$

In Equation (2), C_c , C_e , C_b , C_f and C_s are the unit cost of concrete, excavation, backfill, formwork, and reinforcement, respectively. Table 1 shows the unit expenses that are being considered [24].

Table 1. Unit cost of spread footing construction [24].

Item	Unit	Unit Cost (Euros)
Excavation	m ³	25.16
Formwork	m ²	51.97
Reinforcement	kg	2.16
Concrete	m ³	173.96
Compacted backfill	m ³	3.97

2.2. Design Variables

The design factors for the spread footing model are shown in Figure 1. There are two types of design variables: those that define geometrical parameters and those that describe reinforcing steel. The dimensions of the foundation are represented by four geometric design variables, as illustrated in Figure 1. X_1 is the foundation's length, X_2 is the foundation's width, X_3 is the foundation's thickness and X_4 is the depth of embedment. Moreover, the steel reinforcement has two design variables: X_5 is the longitudinal reinforcement and X_6 is the transverse reinforcement.

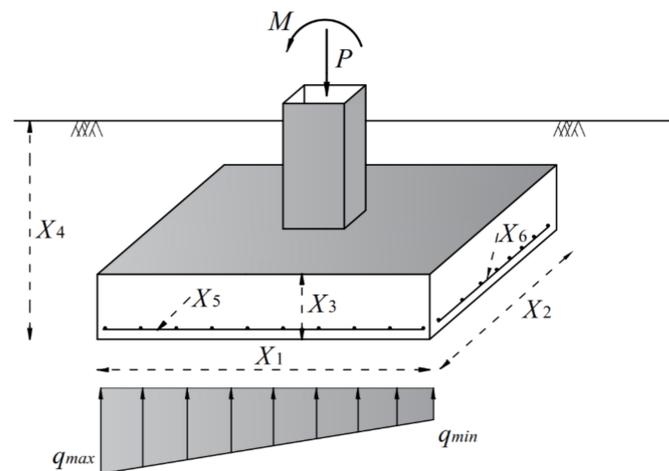


Figure 1. Design variables of the footing.

2.3. Design Constraints

The forces operating on the footing are depicted in Figure 1. M and P denote the axial load and moment imparted to the footing in this figure. The minimum and maximum bearing pressures on the foundation's base are q_{\min} and q_{\max} , respectively. The next subsections go over the design restrictions that must be taken into account when optimizing the spread footing.

Bearing capacity: The foundation's bearing capacity must be sufficient to withstand the forces acting along the base. The maximum stress should be less than the soil's bearing capacity to ensure a safe design:

$$q_{\max} \leq \frac{q_{ult}}{FS} \quad (3)$$

where q_{ult} denotes the foundation's ultimate bearing capacity and q_{\max} is the maximum contact pressure at the boundary between the foundation's bottom and the underlying soil. In this study, an ANN model is established to estimate the q_{ult} in Section 6.

The lowest and highest applied bearing pressures on the foundation's base are calculated as follows:

$$q_{\max} = \frac{P}{X_1 X_2} \left(1 \pm \frac{6e}{X_1} \right) \quad (4)$$

where e denotes the eccentricity, which is defined as the ratio of the overturning moments (M) to the total vertical forces (P).

Eccentricity: The following requirements must be met such that tensile forces at the bottom of the footing are avoided:

$$e \leq \frac{X_1}{6} \quad (5)$$

Settlement: According to the following inequalities, foundation settlement should be kept within a legal range:

$$\delta \leq \delta_{all} \quad (6)$$

where δ_{all} is the permitted settlement and δ is the foundation's immediate settlement. The settlement can be estimated as follows using the elastic solution proposed by Poulos and Davis [25]:

$$\delta = \frac{P(1 - \nu^2)}{\kappa_z E \sqrt{X_1 X_2}} \quad (7)$$

where κ_z is the shape factor, ν is the Poisson ratio and E is modulus of elasticity. In this research, the shape factor proposed by Wang and Kulhawy [24] is used as follows:

$$\kappa_z = -0.0017(X_2/X_1)^2 + 0.0597(X_2/X_1) + 0.9843 \quad (8)$$

One-way shear: The footing must be viewed as a wide beam for one-way shear. According to ACI [26], the shear strength of concrete measured along a vertical plane extending the whole width of the base and located at a distance equal to the effective depth of the footing (V_u) should be less than the nominal shear strength of concrete:

$$V_u \leq \frac{1}{6} \phi_V \sqrt{f'_c} b d \quad (9)$$

where ϕ_V is the shear strength reduction factor of 0.75 [26], f'_c is the concrete compression strength, and b is the section width.

Two-way shear: The tendency of the column to punch through the footing slab is called punching shear. According to Equation (10), the maximum shearing force in the upward direction (V_u) should be less than the nominal punching shear strength to avoid such a failure.

$$V_u \leq \min \left\{ \left(1 + \frac{2}{\beta_c}\right) / 6, \left(\frac{\alpha_s d}{b_0} + 2\right) / 12, \frac{1}{3} \right\} \phi_V \sqrt{f'_c} b_0 d \quad (10)$$

where b_0 is the crucial section's perimeter taken at $d/2$ from the column's face, d denotes the depth at which steel reinforcement is placed, β_c is the ratio of a column section's long side to its short side and α_s is equal to 40 for interior columns.

Bending moment: The nominal flexural strength of the reinforced concrete foundation section should be less than the moment capacity [26]:

$$M_u \leq \phi_M A_S f_y \left(d - \frac{a}{2}\right) \quad (11)$$

where M_u denotes the bending moment of the reaction stresses due to the applied load at the column's face, ϕ_M presents the flexure strength reduction factor equal to 0.9 [26], A_S denotes the area of steel reinforcement and f_y is the yield strength of steel.

Reinforcements limitation: In each direction of the footing, the amount of steel reinforcement must fulfill minimum and maximum reinforcement area limitations according to the following inequality [26]:

$$\rho_{\min} b d \leq A_S \leq \rho_{\max} b d \quad (12)$$

where A_S is the cross section of steel reinforcement, ρ_{\min} and ρ_{\max} are the minimum and maximum reinforcement ratios based on the following equations [26]:

$$\rho_{\min} = \max \left\{ \frac{1.4}{f_y}, 0.25 \frac{\sqrt{f'_c}}{f_y} \right\} \quad (13)$$

$$\rho_{\max} = 0.85 \beta_1 \frac{f'_c}{f_y} \left(\frac{600}{600 + f_y} \right) \quad (14)$$

Limitation of embedment's depth: The depth of embedment should be limited between 0.5 and 2. Therefore:

$$0.5 \leq X_4 \leq 2 \quad (15)$$

To address the above-mentioned limitations and transform a constrained optimization to an unconstrained one, a penalty function method is used in this paper, according to:

$$F(X) = f(X) + r \sum_{i=1}^p \max\{0, g_i(X)\}^l \quad (16)$$

where $F(X)$ is the penalized objective function, $f(X)$ is the problem's original objective function in Equation (2) and r is a penalty factor.

3. Modified Rat Swarm Optimizer

Rat Swarm Optimizer (RSO) is a novel metaheuristic algorithm inspired by the following and attacking behaviors of rats [23]. Rats are regional animals that live in a swarm of both males and females. In many circumstances, the rats' behavior is extremely aggressive, which may result in the death of several animals. In this approach, the following and aggressive actions of rats are mathematically modelled to perform optimization [23]. Similar to the other population-based optimization techniques, the rat swarm optimizer starts with a set of random solutions which represent the rat's position in the search space. This random population is estimated repeatedly by an objective function and improved based on following and aggressive behaviors of rats. In the original version of the RSO technique, the initial positions of eligible solutions (rats' positions) are determined randomly in the search space as follows:

$$x_i = x_{i \min} + rand \times (x_{i \max} - x_{i \min}), i = 1, 2, \dots, N \quad (17)$$

where $x_{i \min}$ and $x_{i \max}$ are the i th variable's lower and upper limits, respectively. Generally, rats follow the bait in a group through their social painful behavior. Mathematically, to describe this performance of rats, it is assumed that the greatest search agent has the knowledge of bait placement. Therefore, the other search agents can inform their locations with respect to the greatest search agent obtained until now. The following equation has been suggested to represent the attacking process of rats using bait and produce the rat's updated next position [23]:

$$\vec{P}_i(t+1) = \left| \vec{P}_r(t) - \vec{P} \right| \quad (18)$$

where, $\vec{P}_i(t+1)$ defines the updated positions of i th rats, $\vec{P}_r(t)$ is the best optimal solution founded so far and t denotes the iteration number. In the above equation, \vec{P} can be obtained using Equation (19).

$$\vec{P} = A \times \vec{P}_i(t) + C \times (\vec{P}_r(t) - \vec{P}_i(t)) \quad (19)$$

where, $\vec{P}_i(t)$ defines the positions of i th rats, and parameters A and C are calculated as follows:

$$A = R - t \times \left(\frac{R}{t_{max}} \right), t = 1, 2, 3, \dots, t_{max} \quad (20)$$

$$C = 2 \times rand \quad (21)$$

The parameter R is a random number between [1,5], C is a random number between [0,2] [23]. t is the current iteration of optimization process and t_{max} is the maximum number of iterations. Equation (18) updates the locations of search agents and saves the best solution. Even though the performance of RSO to obtain the global optima is better than other evolutionary algorithms like Moth-fame Optimization (MFO), Grey Wolf Optimizer (GWO), and Gravitational Search Algorithm (GSA) [23], the algorithm may face some difficulty in finding better results through exploring complex functions.

To increase the performance and efficiency of RSO, this research presents a modified version of the algorithm using the idea of opposition-based learning (OBL). As mentioned

before, RSO, as a member of population-based optimization algorithms, starts with a set of initial solutions and tries to improve performance toward the best solution. In the absence of a priori knowledge about the solution, the random initialization method is used to generate candidate solutions (rats' initial positions) based on Equation (17). Obviously, the performance and convergence speed are directly related to the distance of the initial solutions from the best solution. In other words, the algorithm has a better performance if the randomly generated solutions have a lower value than the objective function. According to this idea and in order to improve the convergence speed and chance of finding the global optima of the standard RSO, this paper proposes a modified version of the algorithm called modified rat swarm optimization (MRSO). In the new MRSO, in the first iteration of the algorithm after generating the initial random solutions (i.e., rats' positions) using Equation (17), the opposite positions of each solution will be generated based on the concept of opposite number. To describe the new population initialization, it is required to define the concept of opposite number. Let's consider an N -dimensional vector X as follows:

$$X = (x_1, x_2, \dots, x_N) \quad (22)$$

where $x_i \in [x_{i \min}, x_{i \max}]$. Then, the opposite point of x_i , which denoted by \bar{x}_i , is defined by:

$$\bar{x}_i = (x_{i \max} + x_{i \min}) - x_i, \quad i = 1, 2, \dots, N \quad (23)$$

To apply the concept of the opposite number in the population initialization of the MRSO, consider x_i to be a randomly generated solution in N -dimensional problems space (i.e., candidate solution). For this random solution, its opposite will be generated using Equation (23) and denoted by \bar{x}_i . Then, both solutions (i.e., x_i and \bar{x}_i) will be evaluated by the objective function $f(\cdot)$. Therefore, if $f(\bar{x}_i)$ is better than $f(x_i)$ (i.e., $f(\bar{x}_i) < f(x_i)$), the agent x_i will be replaced by \bar{x}_i ; otherwise, we continue with x_i . Hence, in the first iteration, the initial solution and its opposite are evaluated simultaneously to continue with better (fitter) starting agents.

Despite the fact that MRSO is capable of expressing an efficient performance when compared to the traditional method, it can still get stuck in local optima and is not ideal for extremely difficult problems. In other words, during the search process, occasionally some agents fall into a local minimum and do not move for several iterations. To overcome these weaknesses and to increase the exploration and search capability, in the proposed MRSO at each iteration, the worst solution yielding the largest fitness value (in minimization problems) is replaced by a new solution according to the following equation:

$$x_{worst} = \begin{cases} rand_1 \times \vec{P}_r(t) & \text{if } rand_3 \leq 0.5 \\ (x_{i \max} + x_{i \min}) - x_i & \text{if } rand_3 > 0.5 \end{cases} \quad (24)$$

where, x_{worst} is the solution with the maximum value of the objective function, $rand_1$, $rand_2$ and $rand_3$ are random numbers between 0 and 1. The new approach exchanges the position vector of a least ranked rat with its opposite or based on the best solution found so far ($\vec{P}_r(t)$) in each generation. This process attempts to modify the result by preserving population diversity and exploring new locations across the problem search area.

In summary, the suggested MRSO algorithm's phases are implemented as follows: first, the initial random solutions and their opposites are generated, and then these solutions are evaluated according to the objective function to start the algorithm with fitter (better) solutions. Second, the population updating phase is conducted by updating the current solutions and then these solutions are evaluated again to replace the worst solution with a new one. Algorithm 1 shows the pseudo code for the proposed MRSO.

Algorithm 1 Modified Rat Swarm Optimization.

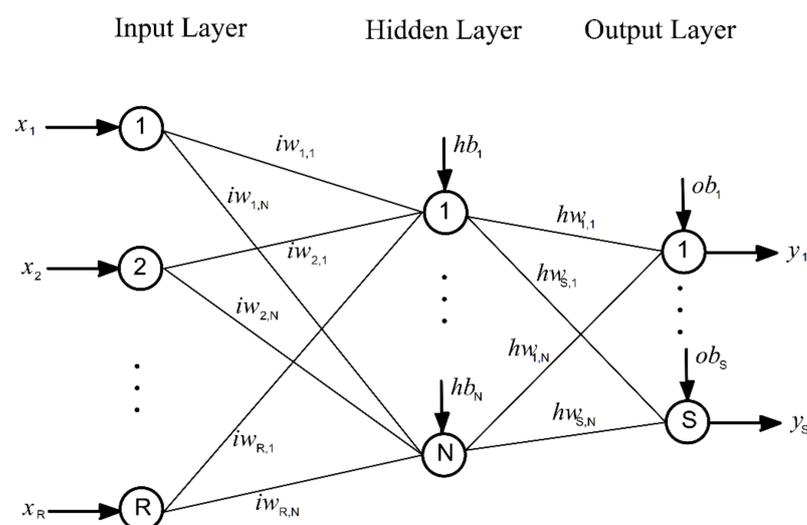
```

Define algorithm parameters:  $N, t_{max}$ 
For  $i=1$  to  $N$  //generate initial population
Initialize the rats' position,  $x_i$ , using Equation (17)
Evaluate opposite of rats' position,  $\bar{x}_i$ , based on Equation (23)
If  $f(\bar{x}_i) < f(x_i)$ 
Replace  $x_i$  with  $\bar{x}_i$ 
End if
End for
Initialize parameters  $A, C$ , and  $R$  //algorithm process
Calculate the fitness value of each search agent
 $\vec{P}_r \leftarrow$  best search agent
While  $t < t_{max}$  //rats' movement
For  $i=1$  to  $N$ 
Update parameters  $A$  and  $C$  by Equations (20) and (21)
Update the positions of search agents using Equation (18)
Calculate the fitness value of each search agent
If the search agent goes beyond the boundary limits adjust it
End for
Change the worst agent with a new one using Equation (24)
Update best agent  $\vec{P}_r$ 
 $t = t + 1$ 
End While

```

4. Artificial Neural Network

Artificial Neural Networks (ANNs) are parallel connectionist structures that are used to model the human brain's functional network of neurons. ANNs are made up of neurons, which are extremely complicated mathematical processing units [27]. Weights and biases connect neurons. The input layer, hidden layer, and output layer all make up an ANN network [27]. Figure 2 illustrates a typical ANN model with one hidden layer. The input layer neurons receive data and transfer the data to the hidden layer, which conducts computation and sends the results to the last layer. The output layer is made up of neurons that send the system's output to the user.

**Figure 2.** Typical ANN model.

The number of hidden layers used in an ANN is determined by the complexity of the problem. If the network structure is small, it cannot reach a good level of effectiveness. Moreover, if it is too large, it will lead to redundant internal connections, loss of general-

izability, and unnecessary complexity. According to the results of previous studies, one or two hidden layers are found to be sufficient and suitable for most situations and can solve any complex function [13,28]. The ANN models in this work are created using one hidden layer.

Furthermore, the most crucial task in the ANN architecture is determining neuron numbers in the hidden layer, which is dependent on the nature of the problem [13]. As suggested by Caudill [29], the number of neurons in the hidden layer required to map any function with R inputs has an upper limit of $2R+1$.

In the network, the information passes from the input layer to the hidden layer and then to the output layer. As shown in Figure 2, in the presence of a bias, each neuron evaluates the sum of the weighted inputs and sends this sum through an activation function to generate the output. To get the optimum performance in training and testing, different transfer functions such as log-sigmoid and tan-sigmoid are examined. This process can be expressed as follows:

$$ho_j = f_j\left(\sum_{i=1}^R iw_{j,i}x_i + hb_j\right) \quad (25)$$

where $iw_{j,i}$ is the weight connected between neurons $i = (1, 2, \dots, R)$ and $j = (1, 2, \dots, N)$, hb_j is a bias in hidden layer, R is the total number of neurons in the input layer, x_i is the corresponding input data and f is the transfer function.

In the output layer, the output of the neuron is obtained by the following equation:

$$y_k = f_k\left(\sum_{j=1}^N hw_{k,j}ho_j + ob_k\right) \quad (26)$$

where $hw_{k,j}$ is the weight connected between neurons $j = (1, 2, \dots, N)$ and $k = (1, 2, \dots, S)$, ob_k is a bias in output layer, N denotes the total number of neurons in the hidden layer, and S represents the total number of neurons in the output layer.

Following the formation of the ANN's structure, training with known input and output data sets is carried out to determine the network's appropriate weights and biases. The term "network training" refers to the process of determining the best values for the network's weights and biases. Various techniques are typically used to determine the appropriate weights and biases for the ANN.

The ANN training is an unconstrained optimization problem involving the minimization of the global error by adjusting the values of synaptic weights and biases. A learning algorithm iteratively updates the values of the network parameters for provided training data, which consists of input–output vectors, to approach the target. This update procedure is commonly performed by back-propagating the error signal, layer by layer, and adjusting the parameters with respect to the magnitude of error signal. Back-propagation is the most widely utilized of various developed learning algorithms, and it has been used to represent many phenomena in the field of geotechnical engineering with considerable success. It requires less memory than the other algorithms and typically reaches an acceptable error level rapidly, although it can take a long time to converge properly on an error minimum. The network is trained by adjusting the weights and biases based on the differences between the actual and desired output values.

The prediction performance of the overall ANN model can be assessed by the correlation coefficient, R and the root mean squared error (RMSE). The coefficient of correlation is a measure that is used to determine the relative correlation and goodness-of-fit between the predicted and observed data. The following guide is suggested for values of R between 0.0 and 1.0 [30]:

- $R \geq 0.8$ strong correlation exists between two sets of variables;
- $0.2 < R < 0.8$ correlation exists between the two sets of variables; and
- $R \leq 0.2$ weak correlation exists between the two sets of variables.

The RMSE is the most widely accepted measure of error and has the advantage that large errors receive much greater attention than small errors [29]. If R is 1.00 and RMSE is 0, the model's predictive capability is treated as excellent. Therefore, a well-trained ANN model should have a R value near to 1 and low RMSE values.

The RMSE criteria is derived based on the neural network's prediction of the training dataset, which can be obtained using the following equation:

$$RMSE = \sqrt{\frac{1}{M} \sum_{i=1}^M (y - \hat{y})^2} \quad (27)$$

where y and \hat{y} are the actual and the predicted values obtained by the model and M is total number of samples.

The ideal solution, which is considered as the weights and biases of the network associated with the minimized value of RMSE, is finally attained by continuous iterations. At the end of the training phase, the neurons' associated trained weights and biases are stored in the network's memory. The neural network is then tested using a different set of data in the next phase. In the testing phase, using the trained parameters, the network produces the target output values for the test data.

5. Performance Verification of MRSO

The effectiveness of the proposed method (MRSO) will be investigated in this section. On a set of benchmark test functions from the literature, the performance of MRSO is compared to that of the standard version of the algorithm (RSO) as well as some well-known metaheuristic algorithms. These are all minimization problems that can be used to evaluate new optimization algorithms' robustness and search efficiency. The mathematical formula and characteristics of these test functions are shown in Table 2.

Table 2. Benchmark functions' details.

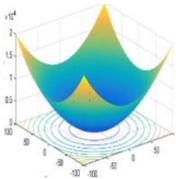
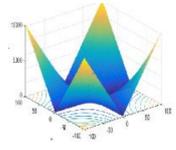
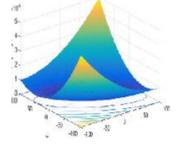
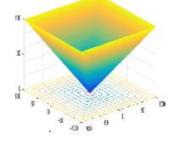
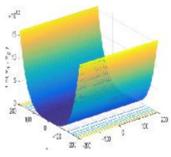
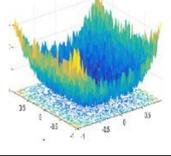
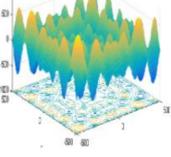
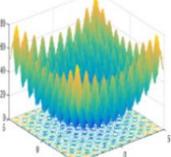
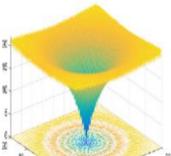
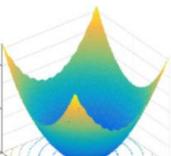
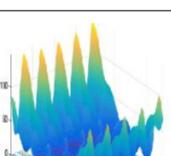
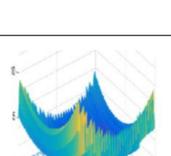
Function	Range	F_{min}	n (Dim)	3D View
$F_1(X) = \sum_{i=1}^n x_i^2$	$[-100, 100]^n$	0	30	
$F_2(X) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-10, 10]^n$	0	30	
$F_3(X) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	$[-100, 100]^n$	0	30	
$F_4(X) = \max_i \{ x_i , 1 \leq i \leq n\}$	$[-100, 100]^n$	0	30	

Table 2. Cont.

Function	Range	F_{min}	n (Dim)	3D View
$F_5(X) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]^n$	0	30	
$F_6(X) = \sum_{i=1}^n ix_i^4 + random[0,1)$	$[-1.28, 1.28]^n$	0	30	
$F_7(X) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	$[-500, 500]^n$	$428.9829 \times n$	30	
$F_8(X) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^n$	0	30	
$F_9(X) = -20 \exp \left(\frac{-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}}{-\exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20} + e \right)$	$[-32, 32]^n$	0	30	
$F_{10}(X) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^n$	0	30	
$F_{11}(X) = \frac{\pi}{n} \{ 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \} + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 4}{4} u(x_i, a, k, m)$ $= \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	$[-50, 50]^n$	0	30	
$F_{12}(X) = 0.1 \{ \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	$[-50, 50]^n$	0	30	

The performance and robustness of the proposed MRSO are compared with the original RSO and some well-established optimization techniques, including Particle Swarm Optimization (PSO) [31], Moth-flame Optimization (MFO) [32] and Multi-Verse Optimizer (MVO) [33]. According to the literature [23] and to have a fair comparison between the selected methods, for all algorithms, the size of solutions (N) and maximum iteration number (t_{max}) are considered equal to 50 and 1000, respectively.

Because metaheuristics approaches are stochastic, a single run's results may be incorrect. As a result, statistical analysis should be performed to provide a fair comparison and evaluate the effectiveness of the algorithms. To address this issue, 30 independent runs for the stated algorithms are performed, with the results presented in Table 3.

Table 3. Comparison of different methods in solving test functions.

Function	Statistics	MRSO	RSO	PSO	MFO	MVO
F_1	Mean	0.000	6.09×10^{-32}	4.98×10^{-9}	3.15×10^{-4}	2.81×10^{-1}
	Std.	0.000	5.67×10^{-35}	1.40×10^{-8}	5.99×10^{-4}	1.11×10^{-1}
F_2	Mean	0.000	0.000	7.29×10^{-4}	$3.71 \times 10^{+1}$	3.96×10^{-1}
	Std.	0.000	0.000	1.84×10^{-3}	$2.16 \times 10^{+1}$	1.41×10^{-1}
F_3	Mean	0.000	1.10×10^{-18}	1.40×10	$4.42 \times 10^{+3}$	4.31×10
	Std.	0.000	4.47×10^{-19}	7.13	$3.71 \times 10^{+3}$	8.97×10
F_4	Mean	0.000	4.67×10^{-7}	6.00×10^{-1}	6.70×10	8.80×10^{-1}
	Std.	0.000	1.96×10^{-8}	1.72×10^{-1}	1.06×10	2.50×10^{-1}
F_5	Mean	4.71×10^{-3}	6.13	4.93×10	$3.50 \times 10^{+3}$	$1.18 \times 10^{+2}$
	Std.	0.000	7.97×10^{-1}	3.89×10	$3.98 \times 10^{+3}$	$1.43 \times 10^{+2}$
F_6	Mean	6.32×10^{-7}	9.49×10^{-6}	6.92×10^{-2}	3.22×10^{-1}	2.02×10^{-2}
	Std.	4.75×10^{-7}	1.83×10^{-5}	2.87×10^{-2}	2.93×10^{-1}	7.43×10^{-3}
F_7	Mean	$-1.25 \times 10^{+4}$	$-8.57 \times 10^{+3}$	$-6.01 \times 10^{+3}$	$-8.04 \times 10^{+3}$	$-6.92 \times 10^{+3}$
	Std.	2.60	$4.23 \times 10^{+2}$	$1.30 \times 10^{+3}$	$8.80 \times 10^{+2}$	$9.19 \times 10^{+2}$
F_8	Mean	0.000	$1.57 \times 10^{+2}$	$4.72 \times 10^{+1}$	$1.63 \times 10^{+2}$	$1.01 \times 10^{+2}$
	Std.	0.000	7.39×10	1.03×10	3.74×10	1.89×10
F_9	Mean	8.88×10^{-16}	7.40×10^{-17}	3.86×10^{-2}	1.60×10	1.15×10
	Std.	0.000	6.42	2.11×10^{-1}	6.18×10	7.87×10^{-1}
F_{10}	Mean	0.000	0.000	5.50×10^{-3}	5.03×10^{-2}	5.74×10^{-1}
	Std.	0.000	0.000	7.39×10^{-3}	1.74×10^{-1}	1.12×10^{-1}
F_{11}	Mean	2.90×10^{-3}	5.52×10^{-1}	1.05×10^{-2}	1.26×10	1.27×10
	Std.	4.00×10^{-3}	8.40	2.06×10^{-2}	1.83×10	1.02×10
F_{12}	Mean	2.15×10^{-2}	6.05×10^{-2}	4.03×10^{-1}	7.24×10^{-1}	6.60×10^{-1}
	Std.	3.72×10^{-2}	7.43×10^{-1}	5.39×10^{-1}	1.48×10	4.33×10^{-2}

Table 3 shows that, when compared to conventional RSO and alternative optimization methods for all functions, MRSO can deliver better solutions in terms of the mean value of the objective functions. The results also reveal that the MRSO algorithm's standard deviations are substantially smaller than those of the other techniques, indicating that the algorithm is stable. Based on the findings, it can be inferred that MRSO outperforms the standard algorithm as well as alternative optimization methods.

6. ANN for Prediction of Factor of Safety

In this section, an ANN model is developed and tested to forecast the shallow foundations' ultimate bearing capacity. The determination of characteristics that affect bearing capacity is one of the more significant processes in model creation for bearing capacity estimation. Regardless of whether the values of bearing capacity produced using different traditional approaches vary significantly, the essential form of the equation is almost the same. The general form of the bearing capacity formula for foundations rested on cohesionless soil is presented in Equation (28).

$$q_{ult} = \gamma DN_q S_q d_q + 0.5 \gamma BN_\gamma S_\gamma d_\gamma \quad (28)$$

where γ is the unit weight of the soil, B denotes the width of the footing and D is the depth of soil above the foundation base. N_q and N_γ are bearing capacity factors that are dependent on the internal friction angle of the base soil, d_q and d_γ are depth factors, and s_q and s_γ are shape factors [4].

Several researchers have introduced different equations for the evaluation of these factors. The equations for bearing capacity factors, shape factors and depth factors based on Meyerhof's theory are presented in the following equations:

$$N_q = e^{\pi \tan \phi} \tan^2 \left(45 + \frac{\phi}{2} \right) \quad (29)$$

$$N_\gamma = (N_q - 1) \tan(1.4\phi) \quad (30)$$

$$s_q = s_\gamma = 1 + 0.1 \tan^2 \left(45 + \frac{\phi}{2} \right) \frac{B}{L} \quad (31)$$

$$d_q = d_\gamma = 1 + 0.1 \tan^2 \left(45 + \frac{\phi}{2} \right) \frac{D}{B} \quad (32)$$

As shown in Equations (28)–(32), the bearing capacity of a foundation is governed by a range of physical characteristics of the foundation and the soil in which it is embedded. Among the variables related to the foundation geometry, the following are the primary elements influencing the bearing capacity: footing's width (B), footing's length (L), footing's shape, and depth of embedment (D). In addition, the angle of shearing resistance of the soil (ϕ) and the unit weights (γ) are the most important parameters with regard to the soil that affect the bearing capacity. Based on the above, the five input parameters for the ANN model creation in this study include: footing's width (B), depth of embedment (D), geometry of footing (L/B), soil's unit weight (γ), and friction angle (ϕ). The single output variable of the ANN is the footing's ultimate bearing capacity (q_{ult}). The proposed ANN model to predict the q_{ult} is shown in Figure 3.

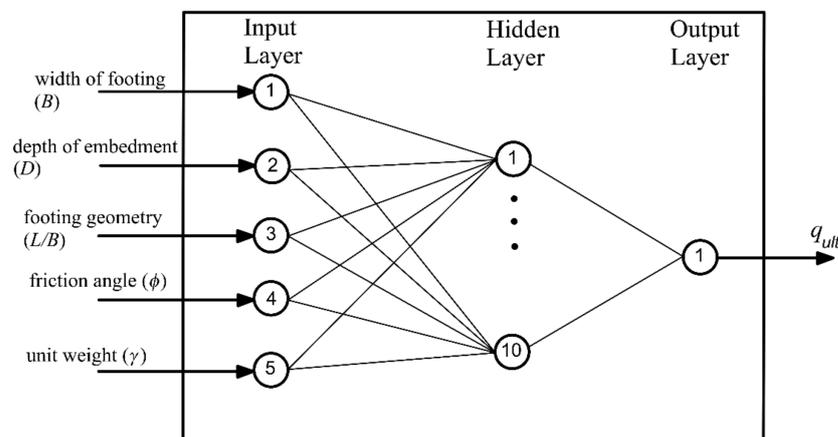


Figure 3. Neural model for q_{ult} prediction.

The data for calibrating and validating the neural network model was gathered from prior experimental research in the literature, which included load test data on real-sized foundations, as well as the footing and soil information. There are 97 data sets in all, with 47 dealing with load tests on large-scale footings and 50 with lower scale model footings.

The data used are presented in Appendix A. These data were implemented for network training and validating.

Neural network training can be made more efficient, if an adequate normalization is performed for the network input and output variables previous to the training process. Normalizing the data generally eliminates variations in input and output parameters, speeds up learning, leads to faster convergence, and significantly reduces calculation

time [34]. In the current study, before training, the following normalization expression is used to scale the input and output data to lie between -1 and 1 .

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (33)$$

where X represents the measured value, X_{norm} is the normalized value of the measured parameter, X_{min} and X_{max} are the minimum and maximum values for the measured parameters. In the training process of the network, the available dataset is split into two sections: training and testing. The training data is used to adjust the network parameters and the testing data is used to verify the performance of the model. Usually, 80% of the data are suggested for model training and 20% for testing [35]. Accordingly, we randomly divided the dataset into training and testing sub-classes that have the respective amounts of 0.8 (i.e., 78 data sets) and 0.2 (i.e., 19 data sets).

An experimental investigation was used to identify the optimal number of neurons in the model's hidden layer. In this experiment, the performance of the model (i.e., RMSE and R) was evaluated with a different number of neurons by considering tan-sigmoid as the transfer function. The network's parameters were evaluated using both MRSO and RSO algorithms to minimize the error between the real and predicted values of q_{ult} . To compare the performance of these algorithms for training the network, 30 independent runs were conducted using both algorithms and the best values of the Root Mean Square Error (RMSE) and the correlation coefficient (R) for all training samples are collected and presented in Table 4. From Table 4, it is shown that the optimum number of the neurons in the hidden layer is equal to 10. Moreover, the results indicate that the best values of RMSE and R obtained by MRSO are lower than those evaluated by RSO and the new algorithm apparently performs better than RSO in the model training.

Table 4. Effects of different neurons numbers in predicting q_{ult} .

Neurons Number	Algorithm	RMSE	R
1	MRSO	0.152	0.8043
	RSO	0.348	0.7539
2	MRSO	0.135	0.8275
	RSO	0.248	0.7902
3	MRSO	0.106	0.8910
	RSO	0.181	0.8386
4	MRSO	0.087	0.9102
	RSO	0.142	0.8779
5	MRSO	0.069	0.9444
	RSO	0.105	0.8991
6	MRSO	0.056	0.9505
	RSO	0.095	0.9272
7	MRSO	0.044	0.9727
	RSO	0.078	0.9403
8	MRSO	0.034	0.9828
	RSO	0.060	0.9764
9	MRSO	0.0291	0.9888
	RSO	0.0414	0.9795
10	MRSO	0.0289	0.9908
	RSO	0.0342	0.9875
11	MRSO	0.0314	0.9817
	RSO	0.0408	0.9786

Following determination of the model's structure, the parameters of the ANN model (i.e., the connecting weights and biases) need to be determined. In this study, the proposed MRSO as an effective optimization algorithm was implemented to determine the weights and biases of the network associated with the minimum value of RMSE. Thereafter, using the trained parameters, the network produced the target output values (i.e., q_{ult}).

Figures 4 and 5 display the actual values of q_{ult} together with their predicted values obtained by MRSO and RSO for training and testing datasets, respectively. According to these figures, the MRSO model can predict better results for both training and testing datasets and it can be introduced as a new hybrid model in this field.

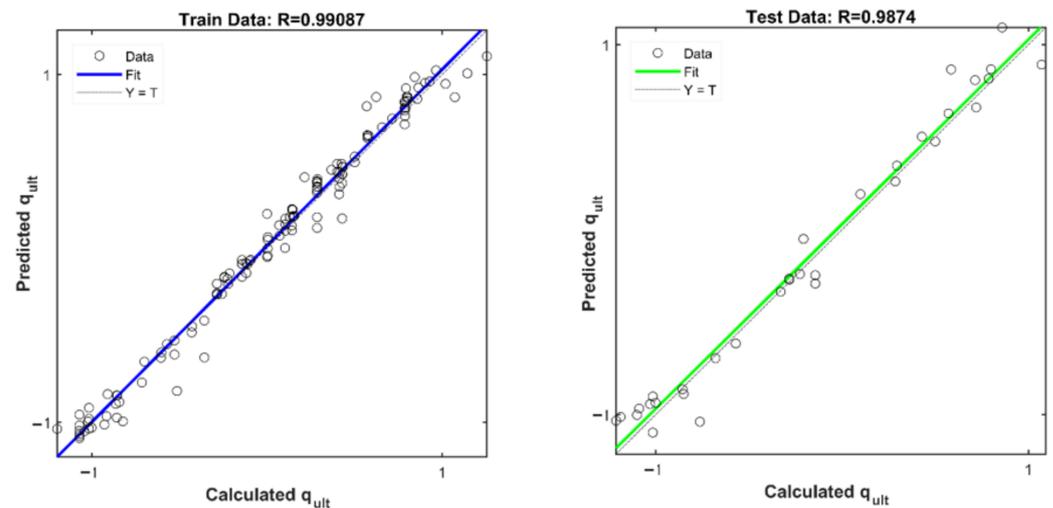


Figure 4. Relationship between real and predicted q_{ult} from the ANN model trained by MRSO.

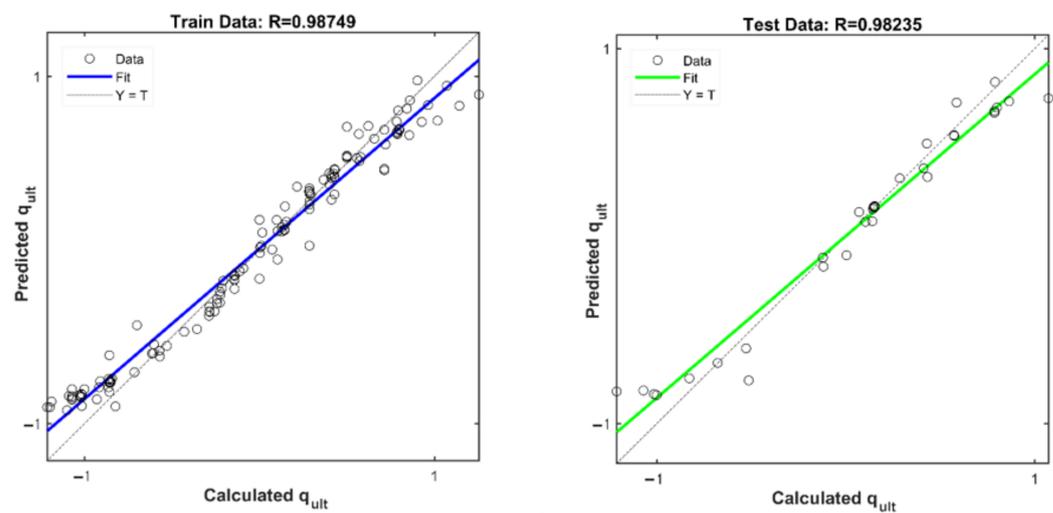


Figure 5. Relationship between real and predicted q_{ult} from the ANN model trained by RSO.

In addition, to obtain a consistent data division of the developed model, several combinations of the training and testing sets are considered. This is to ensure the reliability and repeatability of the trained network when it comes to forecasting the testing dataset.

In this experiment, the data presented in Appendix A is partitioned into five groups. Each group is randomly split into two groups: the first group is used for training the neural network model, and the other (20% of the data set) for testing. Then, the mean value of the RMSEs obtained on five different testing subsets is estimated, which is equal to 0.0298 in this experiment. The results of this experiment could verify the reliability of the proposed model.

7. Model Application

In this section, the optimum design of an interior spread footing in dry sand is conducted using the proposed MRSO. This problem was solved previously by Khajezadeh et al. [36] using the modified particle swarm optimization (MPSO) technique. Khajezadeh et al. [36] applied Meyerhof's method for evaluation of ultimate bearing capacity. However, in the current study, the ultimate bearing capacity is predicted using the developed ANN model based on in situ test results. Other input parameters for the case study are given in Table 5.

Table 5. Input parameters for the case study.

Parameter	Unit	Value for Example
Effective friction angle of base soil	degree	30
Unit weight of base soil	kN/m ³	18
Young's modulus	MPa	35
Poisson's ratio	–	0.3
Vertical load	kN	3480
Moment	kN m	840
Concrete cover	cm	7.0
Yield strength of reinforcing steel	MPa	400
Compressive strength of concrete	MPa	30
Factor of safety for bearing capacity	–	3.0
Allowable settlement of footing	m	0.04

The problem is solved by the presented procedure. In order to verify the efficiency of the proposed method, the analysis results are compared with the standard RSO as well as modified particle swarm optimization (MPSO). The best results of the analyses for the minimum cost are presented in Table 6.

Table 6. Optimization result for spread footing optimization.

Design Variable	Unit	Optimum Values MRSO (Current Study)	Optimum Values RSO (Current Study)	Optimum Values MPSO [36]
X_1	m	5.30	5.75	5.75
X_2	m	1.90	1.82	1.70
X_3	m	0.503	0.505	0.67
X_4	m	1.90	1.82	1.70
X_5	cm ²	135	149.7	160
X_6	cm ²	25	23.1	23
Objective function	Euros	2756	2845	2926

The findings presented in Table 6 show that the optimum design evaluated by the proposed methodology in the current study using the ANN model and MRSO algorithm is lower than those evaluated by standard RSO and MPSO techniques. According to the results, the best price obtained by MRSO is 2756 USD, almost 3.2% lower than the best price calculated by RSO, which means the new method could provide a better solution. In addition, the difference between the best prices evaluated by the MRSO and MPSO is almost 6.2%, which indicates that the predicted ultimate bearing capacity using the proposed ANN model can provide acceptable results compared with the traditional methods for bearing capacity estimation.

8. Summary and Conclusions

In this research, an effective optimization method based on the rat swarm optimizer, namely modified rat swarm optimization (MRSO), was developed and applied for neural network training as well as optimal design of spread foundations. In the proposed modified RSO, both the initial random solutions and their opposites were evaluated in the first iteration of the algorithm, and if the opposite solution's fitness was lower than the random

one, the opposite solution would be selected. As a result, the algorithm begins with better solutions instead of random ones. Furthermore, the new algorithm replaces the worst solution with a better one at each iteration to improve the algorithm's exploration capabilities as well as its performance and convergence rate. In addition, efforts were made to develop an ANN model that could be used to estimate the ultimate bearing capacity of shallow foundations on granular soil. To prepare a suitable database for the development of the ANN model, data was retrieved from 97 load tests on footings (with sizes that match to actual footings and smaller model footings) from the literature. After preparing the training database, the proposed MRSO algorithm was implemented for training the network model. Based on the obtained results of this study, the following conclusions can be drawn:

- The performance comparison of the proposed MRSO algorithm on a set of benchmark functions reveals that the MRSO outperforms the standard RSO and other algorithms.
- The most optimal network for q_{ult} estimation is a three-layer neural network with 10 neurons in the hidden layer.
- The developed ANN model can be applied for ultimate bearing capacity estimation with RMSE equal to 0.0249 and a correlation coefficient equal to 0.9908.
- The new MRSO algorithm was successfully applied to a case study of spread footing optimization from the literature.
- According to the numerical experiment, the MRSO algorithm outperforms the other methods and may provide a cheaper design for spread foundations.

Author Contributions: M.K., methodology, software, investigation, data curation. S.K., writing—original draft preparation, resources, investigation. M.L.N., supervision, project administration, methodology, validation, funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest: This research work abides by the highest standards of ethics, professionalism and collegiality. All authors have no explicit or implicit conflict of interest of any kind related to this manuscript.

Appendix A

Table A1. Database for Bearing Capacity Prediction.

Sources	B (m)	D (m)	L/B	γ (kN/m ³)	ϕ (deg)	q_{ult} (kPa)
Muhs et al. [37]	0.6	0.3	2	9.85	34.9	270
	0.6	0	2	10.2	37.7	200
	0.6	0.3	2	10.2	37.7	570
	0.6	0	2	10.85	44.8	860
	0.6	0.3	2	10.85	44.8	1760
Weiß [38]	0.5	0	1	10.2	37.7	154
	0.5	0	1	10.2	37.7	165
	0.5	0	2	10.2	37.7	203
	0.5	0	2	10.2	37.7	195
	0.5	0	3	10.2	37.7	214
	0.52	0	3.85	10.2	37.7	186
	0.5	0.3	1	10.2	37.7	681
	0.5	0.3	2	10.2	37.7	542
	0.5	0.3	2	10.2	37.7	530
	0.5	0.3	3	10.2	37.7	402
	0.52	0.3	3.85	10.2	37.7	413

Table A1. Cont.

Sources	B (m)	D (m)	L/B	γ (kN/m ³)	ϕ (deg)	qult (kPa)
Muhs and Weiß [39]	0.5	0	1	11.7	37	111
	0.5	0	1	11.7	37	132
	0.5	0	2	11.7	37	143
	0.5	0.013	1	11.7	37	137
	0.5	0.029	4	11.7	37	109
	0.5	0.127	4	11.7	37	187
	0.5	0.3	1	11.7	37	406
	0.5	0.3	1	11.7	37	446
	0.5	0.3	4	11.7	37	322
	0.5	0.5	2	11.7	37	565
	0.5	0.5	4	11.7	37	425
	0.5	0	1	12.41	44	782
	0.5	0	4	12.41	44	797
	0.5	0.3	1	12.41	44	1940
	0.5	0.3	1	12.41	44	2266
	0.5	0.5	2	12.41	44	2847
	0.5	0.5	4	12.41	44	2033
	0.5	0.49	4	12.27	42	1492
	0.5	0	1	11.77	37	123
	0.5	0	2	11.77	37	134
0.5	0.3	1	11.77	37	370	
0.5	0.5	2	11.77	37	464	
0.5	0	4	12	40	461	
0.5	0.5	4	12	40	1140	
Muhs and Weiß [40]	1	0.2	3	11.97	39	710
	1	0	3	11.93	40	630
Briaud and Gibben[41]	0.991	0.711	1	15.8	32	1773.7
	3.004	0.762	1	15.8	32	1019.4
	2.489	0.762	1	15.8	32	1158
	1.492	0.762	1	15.8	32	1540
	3.016	0.889	1	15.8	32	1161.2
Gandhi [42]	0.0585	0.029	5.95	15.7	34	58.5
	0.0585	0.058	5.95	15.7	34	70.91
	0.0585	0.029	5.95	16.1	37	82.5
	0.0585	0.058	5.95	16.1	37	98.93
	0.0585	0.029	5.95	16.5	39.5	121.5
	0.0585	0.058	5.95	16.5	39.5	142.9
	0.0585	0.029	5.95	16.8	41.5	157.5
	0.0585	0.058	5.95	16.8	41.5	184.9
	0.0585	0.029	5.95	17.1	42.5	180.5
	0.0585	0.058	5.95	17.1	42.5	211
	0.094	0.047	6	15.7	34	74.7
	0.094	0.094	6	15.7	34	91.5
	0.094	0.047	6	16.1	37	104.8
	0.094	0.094	6	16.1	37	127.5
	0.094	0.047	6	16.5	39.5	155.8
	0.094	0.094	6	16.5	39.5	185.6
	0.094	0.047	6	16.8	41.5	206.8
	0.094	0.094	6	16.8	41.5	244.6
	0.094	0.047	6	17.1	42.5	235.6
	0.094	0.094	6	17.1	42.5	279.6
0.152	0.075	5.95	15.7	34	98.2	
0.152	0.15	5.95	15.7	34	122.3	
0.152	0.075	5.95	16.1	37	143.3	
0.152	0.15	5.95	16.1	37	176.4	
0.152	0.075	5.95	16.5	39.5	211.2	
0.152	0.15	5.95	16.5	39.5	254.5	

Table A1. Cont.

Sources	B (m)	D (m)	L/B	γ (kN/m ³)	ϕ (deg)	qult (kPa)
	0.152	0.075	5.95	16.8	41.5	285.3
	0.152	0.15	5.95	16.8	41.5	342.5
	0.152	0.075	5.95	17.1	42.5	335.3
	0.152	0.15	5.95	17.1	42.5	400.6
	0.094	0.047	1	15.7	34	67.7
	0.094	0.094	1	15.7	34	90.5
	0.094	0.047	1	16.1	37	98.8
	0.094	0.094	1	16.1	37	131.5
	0.094	0.047	1	16.5	39.5	147.8
	0.094	0.094	1	16.5	39.5	191.6
	0.094	0.047	1	16.8	41.5	196.8
	0.094	0.094	1	16.8	41.5	253.6
	0.094	0.047	1	17.1	42.5	228.8
	0.094	0.094	1	17.1	42.5	295.6
	0.152	0.075	1	15.7	34	91.2
	0.152	0.15	1	15.7	34	124.4
	0.152	0.075	1	16.1	37	135.2
	0.152	0.15	1	16.1	37	182.4
	0.152	0.075	1	16.5	39.5	201.2
	0.152	0.15	1	16.5	39.5	264.5
	0.152	0.075	1	16.8	41.5	276.3
	0.152	0.15	1	16.8	41.5	361.5
	0.152	0.075	1	17.1	42.5	325.3
	0.152	0.15	1	17.1	42.5	423.6

References

- Kaveh, A.; Seddighian, M.R. Optimization of Slope Critical Surfaces Considering Seepage and Seismic Effects Using Finite Element Method and Five Meta-Heuristic Algorithms. *Period. Polytech. Civ. Eng.* **2021**, *65*, 425–436. [\[CrossRef\]](#)
- Chan, C.M.; Zhang, L.; Ng, J.T. Optimization of pile groups using hybrid genetic algorithms. *J. Geotech. Geoenviron. Eng.* **2009**, *135*, 497–505. [\[CrossRef\]](#)
- Gandomi, A.H.; Kashani, A.R. Construction cost minimization of shallow foundation using recent swarm intelligence techniques. *IEEE Trans. Ind. Inform.* **2017**, *14*, 1099–1106. [\[CrossRef\]](#)
- Das, B.M. *Principles of Foundation Engineering*; Cengage Learning: Boston, MA, USA, 2015.
- Taylor, R.E. *Geotechnical Centrifuge Technology*; CRC Press: London, UK, 2018.
- Ahmad, M.; Ahmad, F.; Wróblewski, P.; Al-Mansob, R.A.; Olczak, P.; Kamiński, P.; Safdar, M.; Rai, P. Prediction of Ultimate Bearing Capacity of Shallow Foundations on Cohesionless Soils: A Gaussian Process Regression Approach. *Appl. Sci.* **2021**, *11*, 10317. [\[CrossRef\]](#)
- Moayedi, H.; Moatamediyan, A.; Nguyen, H.; Bui, X.-N.; Bui, D.T.; Rashid, A.S.A. Prediction of ultimate bearing capacity through various novel evolutionary and neural network models. *Eng. Comput.* **2020**, *36*, 671–687. [\[CrossRef\]](#)
- Ahmad, M.; Hu, J.-L.; Hadzima-Nyarko, M.; Ahmad, F.; Tang, X.-W.; Rahman, Z.U.; Nawaz, A.; Abrar, M. Rockburst Hazard Prediction in Underground Projects Using Two Intelligent Classification Techniques: A Comparative Study. *Symmetry* **2021**, *13*, 632. [\[CrossRef\]](#)
- Gao, W.; Raftari, M.; Rashid, A.S.A.; Mu'azu, M.A.; Jusoh, W.A.W. A predictive model based on an optimized ANN combined with ICA for predicting the stability of slopes. *Eng. Comput.* **2020**, *36*, 325–344. [\[CrossRef\]](#)
- Khajehzadeh, M.; Taha, M.R.; Keawsawasvong, S.; Mirzaei, H.; Jebeli, M. An Effective Artificial Intelligence Approach for Slope Stability Evaluation. *IEEE Access* **2022**, *10*, 5660–5671. [\[CrossRef\]](#)
- Abdalla, J.A.; Attom, M.F.; Hawileh, R. Prediction of minimum factor of safety against slope failure in clayey soils using artificial neural network. *Environ. Earth Sci.* **2015**, *73*, 5463–5477. [\[CrossRef\]](#)
- Ahmad, M.; Hu, J.-L.; Ahmad, F.; Tang, X.-W.; Amjad, M.; Iqbal, M.J.; Asim, M.; Farooq, A. Supervised Learning Methods for Modeling Concrete Compressive Strength Prediction at High Temperature. *Materials* **2021**, *14*, 1983. [\[CrossRef\]](#)
- Sonmez, H.; Gokceoglu, C.; Nefeslioglu, H.; Kayabasi, A. Estimation of rock modulus: For intact rocks with an artificial neural network and for rock masses with a new empirical equation. *Int. J. Rock Mech. Min. Sci.* **2006**, *43*, 224–235. [\[CrossRef\]](#)
- Choobasti, A.; Farrokhzad, F.; Barari, A. Prediction of slope stability using artificial neural network (case study: Noabad, Mazandaran, Iran). *Arab. J. Geosci.* **2009**, *2*, 311–319. [\[CrossRef\]](#)
- Wang, H.; Moayedi, H.; Kok Foong, L. Genetic algorithm hybridized with multilayer perceptron to have an economical slope stability design. *Eng. Comput.* **2021**, *37*, 3067–3078. [\[CrossRef\]](#)

16. Rukhaiyar, S.; Alam, M.; Samadhiya, N. A PSO-ANN hybrid model for predicting factor of safety of slope. *Int. J. Geotech. Eng.* **2018**, *12*, 556–566. [[CrossRef](#)]
17. Koessler, E.; Almomani, A. Hybrid particle swarm optimization and pattern search algorithm. *Optim. Eng.* **2021**, *22*, 1539–1555. [[CrossRef](#)]
18. Gao, W. Modified ant colony optimization with improved tour construction and pheromone updating strategies for traveling salesman problem. *Soft Comput.* **2021**, *25*, 3263–3289. [[CrossRef](#)]
19. Delice, Y.; Aydoğan, E.K.; Özcan, U.; İlkay, M.S. A modified particle swarm optimization algorithm to mixed-model two-sided assembly line balancing. *J. Intell. Manuf.* **2017**, *28*, 23–36. [[CrossRef](#)]
20. Kiran, S.H.; Dash, S.S.; Subramani, C. Performance of two modified optimization techniques for power system voltage stability problems. *Alex. Eng. J.* **2016**, *55*, 2525–2530. [[CrossRef](#)]
21. Khajehzadeh, M.; Taha, M.R.; El-Shafie, A.; Eslami, M. Search for critical failure surface in slope stability analysis by gravitational search algorithm. *Int. J. Phys. Sci.* **2011**, *6*, 5012–5021.
22. Khajehzadeh, M.; Taha, M.R.; Eslami, M. Multi-objective optimization of foundation using global-local gravitational search algorithm. *Struct. Eng. Mech. Int. J.* **2014**, *50*, 257–273. [[CrossRef](#)]
23. Dhiman, G.; Garg, M.; Nagar, A.; Kumar, V.; Dehghani, M. A novel algorithm for global optimization: Rat swarm optimizer. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 8457–8482. [[CrossRef](#)]
24. Wang, Y.; Kulhawy, F.H. Economic design optimization of foundations. *J. Geotech. Geoenviron. Eng.* **2008**, *134*, 1097–1105. [[CrossRef](#)]
25. Poulos, H.G.; Davis, E.H. *Elastic Solutions for Soil and Rock Mechanics*; Wiley: New York, NY, USA, 1974.
26. ACI 318-05 *Building Code Requirements for Structural Concrete and Commentary*; American Concrete Institute International: Farmington Hills, MI, USA, 2005.
27. Anderson, J.A. *An Introduction to Neural Networks*; MIT Press: Cambridge, MA, USA, 1995.
28. Baheer, I. Selection of methodology for modeling hysteresis behavior of soils using neural networks. *Comput.-Aided Civ. Infrastruct. Eng.* **2000**, *5*, 445–463. [[CrossRef](#)]
29. Caudill, M. Neural networks primer, Part III. *AI Expert* **1988**, *3*, 53–59.
30. Smith, G.N. *Probability and Statistics in Civil Engineering*; Collins Professional and Technical Books: London, UK, 1986.
31. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; pp. 1942–1948.
32. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl.-Based Syst.* **2015**, *89*, 228–249. [[CrossRef](#)]
33. Mirjalili, S.; Mirjalili, S.M.; Hatamlou, A. Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **2016**, *27*, 495–513. [[CrossRef](#)]
34. Sola, J.; Sevilla, J. Importance of input data normalization for the application of neural networks to complex industrial problems. *IEEE Trans. Nucl. Sci.* **1997**, *44*, 1464–1468. [[CrossRef](#)]
35. Moayedi, H.; Abdullahi, M.a.M.; Nguyen, H.; Rashid, A.S.A. Comparison of dragonfly algorithm and Harris hawks optimization evolutionary data mining techniques for the assessment of bearing capacity of footings over two-layer foundation soils. *Eng. Comput.* **2021**, *37*, 437–447. [[CrossRef](#)]
36. Khajehzadeh, M.; Taha, M.R.; El-Shafie, A.; Eslami, M. Modified particle swarm optimization for optimum design of spread footing and retaining wall. *J. Zhejiang Univ. Sci. A* **2011**, *12*, 415–427. [[CrossRef](#)]
37. Muhs, H.; Elmiger, R.; Weiß, K. *Sohlreibung und Grenztragfähigkeit Unter Lotrecht und Schräg Belasteten Einzelfundamenten*; Deutsche Forschungsgesellschaft für Bodenmechanik (DEGEBO): Berlin, Germany, 1969.
38. Weiß, K. *Der Einfluß der Fundamentform auf die Grenztragfähigkeit Flachgegründeter Fundamente, Untersuchungen Ausgef von Klaus Weiß: Mit 14 Zahlentaf*; Deutsche Forschungsgesellschaft für Bodenmechanik (DEGEBO): Berlin, Germany, 1970.
39. Muhs, H.; Weiss, K. *Untersuchung von Grenztragfähigkeit und Setzungsverhalten Flachgegründeter Einzelfundamente im Ungleichförmig-nichtbindigen Boden*; Deutsche Forschungsgesellschaft für Bodenmechanik (DEGEBO): Berlin, Germany, 1971.
40. Muhs, H. Inclined load tests on shallow strip footings. In Proceedings of the 8th International Conference on Soil Mechanism and Foundation Engineering, Moscow, Russia, 6–11 August 1973; Volume II, pp. 173–179.
41. Briaud, J.-L.; Gibbens, R. Behavior of five large spread footings in sand. *J. Geotech. Geoenviron. Eng.* **1999**, *125*, 787–796. [[CrossRef](#)]
42. Gandhi, G. Study of Bearing Capacity Factors Developed from Lab. Experiments on Shallow Footings on Cohesionless Soils. Ph.D. Thesis, Shri GS Institute of Tech and Science, Indore, India, 2003.