Hindawi
Complexity
Volume 2021, Article ID 6639032, 19 pages
https://doi.org/10.1155/2021/6639032

Research Article

Analysis of Beam-Column Designs by Varying Axial Load with Internal Forces and Bending Rigidity Using a New Soft Computing Technique

Wen Huang,[1] **Tianhua Jiang**,[1] **Xiucheng Zhang**,[2,3] **Naveed Ahmad Khan**,[4] **and Muhammad Sulaiman**[4]

[1]*School of Urban Construction, Wuhan University of Science and Technology, Wuhan 430074, China*
[2]*Fujian University Engineering Research Center of Disaster Prevention and Reduction of Southeast Coastal Engineering Structure (JDGC03), Putian 351100, China*
[3]*Putian College Institute of Civil Engineering, Putian 351100, China*
[4]*Department of Mathematics, Abdul Wali Khan University Mardan, Mardan 23200, Khyber Pakhtunkhwa, Pakistan*

Correspondence should be addressed to Wen Huang; huangwen@wust.edu.cn

Received 28 December 2020; Revised 22 January 2021; Accepted 10 February 2021; Published 5 March 2021

Academic Editor: Harish Garg

Design problems in structural engineering are often modeled as differential equations. These problems are posed as initial or boundary value problems with several possible variations in structural designs. In this paper, we have derived a mathematical model that represents different structures of beam-columns by varying axial load with or without internal forces including bending rigidity. We have also developed a novel solver, the LeNN-NM algorithm, which consists of weighted Legendre polynomials, and a single path following optimizer, the Nelder–Mead (NM) algorithm. To evaluate the performance of our solver, we have considered three design problems representing beam-columns. The values of performance indicators, MAD, TIC, NSE, and ENSE, are calculated for a hundred simulations. The outcome of our statistical analysis points to the superiority of the LeNN-NM algorithm. Graphical illustrations are presented to further elaborate on our claims.

1. Introduction

Most of the problems arising in numerical science and engineering such as physics, biology, economics, mathematics, and astronomy are modeled as differential equations with initial or boundary values. The modeling enables us to analyze and understand the concept of a particular event in a precise way. Thus, finding an exact and analytical solution for such models gained immense importance in recent years. Generally, finding exact solutions to the physical problems modeled as differential equations is a difficult task and, in most cases, the analytic solution does not exist. Therefore, it becomes necessary to study numerical methods for finding solutions to the differential equations representing physical phenomenon.

Among various proposed methods for solving ODE's, to solve the two-point Boundary value problem (BVP), a finite difference method [1] was proposed. For solving fourth-order and sixth-order BVP's, the homotopy perturbation method [2, 3] is proposed. Multipoints' BVP's are solved by optimal homotopy ssymptotic proposed by Ali [4]. Also, for the solution of two-points and multipoints' BVP's of higher order, a domain decomposition method [5–7] was developed. Jacobi polynomials were used by Doha [8] to propose a spectral Galerkin algorithm for two-points' BVP's of third and fifth order. For the solution of even order DEs, Chebyshev polynomials are used by Doha [9] to develop spectral Galerkin algorithm. Shannon wavelet method [10], Haar wavelets method [11], Saadatmandi and Dehghan technique [12], a fifth-fourth continuous block implicit hybrid method [13], and spectral Bernstein residual method [14] were proposed for the solution of higher-order differential equations. To solve higher-order linear and nonlinear differential equations with two boundary values, variational methods [15–17] were introduced. In recent years, many techniques have been developed to solve nonlinear stiff models, i.e., blood flow of Ree Eyring fluid [18] and numerical study of the DNA dynamics arising in oscillator

chain of the Peyrard–Bishop model [19]. Besides the advantages that these methods give a good approximation to the solution, it is required to have an initial guess near a required solution and the approximate solution is differentiable and continuous in the domain, and hence, it seriously affects the stability of these numerical techniques.

Artificial neural networks are one of the intelligent techniques in solving different types of differential equations. The results obtained by the approximation through neural networks are close enough to the analytic or exact solution. In general, neural networks are widely used to solve fractional differential equations, integro differential equations (IDEs), partial differential equations (PDEs), and ordinary differential equations (ODEs). Feedforward artificial neural networks are used to solve two-point boundary value problems (BVP's) of fourth order. Higher-order ODE's with boundary values are solved by the radial basis activation function in neural networks as presented by Mai-Duy [20].

According to no free lunch theorem, researchers are compelled to hybridize the strengths of two or more techniques so that high-quality solutions are calculated with fewer efforts. In recent times, metaheuristic techniques have gained the attention of researchers in various fields of science and engineering. Impacts of different crossover operators are investigated for handling multiobjective problems [21]. A plant propagation algorithm (PPA) and its modified version were developed to solve design engineering problems [22]. In electrical engineering, several methodologies are used to solve complex optimization problems [23]. Optimal design and temperature distribution of heat fin is solved by using hybridization of artificial neural networks and metaheuristic algorithms [24, 25] and, furthermore, oscillatory behavior of heart beat [26]. Most of the real-application problems are highly nonlinear ODEs and provide less information about the continuity and differentiability of resulting solutions in solution space. The authors of this paper are addressing this issue and hence developed a novel solver that combines weighted Legendre polynomials and an efficient single path following optimizer the Nelder–Mead (NM) algorithm. The outcome of our experiments dictates that the LeNN-NM algorithm can handle ODEs and provide quality solutions.

Salient features of this research are summarized as follows:

(i) We have derived a mathematical model that represents different structures of beam-columns. By varying axial load with or without internal forces including bending rigidity, we have studied different cases of the problem.

(ii) We have also developed a novel solver, the LeNN-NM algorithm, which consists of weighted Legendre polynomials and a single path following optimizer the Nelder–Mead (NM) algorithm.

(iii) A graphical overview of the novel LeNN-NM algorithm is presented in Figure 1.

(iv) Three versions of the structural designs of beam-columns are analyzed with the help of LeNN-NM algorithm, see Problems 1–3.

(v) To examine the quality of solutions calculated by LeNN-NM algorithm, we have calculated values of performance indicators mean absolute deviation (MAD), Theil's inequality coefficient (TIC), Nash Sutcliffe efficiency (NSE), and error in Nash Sutcliffe efficiency (ENSE).

(vi) Series solutions are presented in equations (30), (31), and (32), which may be used by researchers for further investigation.

(vii) Our solutions are in strong agreement with reference solutions and absolute errors are less than the reference errors, see Tables 1–3.

(viii) Graphical illustrations for performance indicators are presented in Figure 2. Lower values of MAD, TIC, and ENSE show that our solutions are of good quality.

(ix) We have repeated our simulations for hundred times and overall performance is depicted in terms of fitness values, MAD, TIC, and ENSE, for 100 runs, see Figure 3.

(x) The success rate of the LeNN-NM algorithm is shown by plotting the frequency graphs of fitness values, MAD, TIC, and ENSE, with normal distribution fittings. All these analyses dictate that the LeNN-NM algorithm is stable, efficient, and compatible for solving design engineering problems.

2. Mathematical Model Representing the Design of Beam-Columns

Most of the problems related to the physical phenomenon are modeled as differential equations. Higher-order differential equations appear in mathematical models of the elastic stability theory. Hence, most of the physical problems are posed with nonlinear and complex ODEs. To understand and analyze the design of beam-column, a differential relation should be developed between the various cross-sections of the physical problem.

Cross-section analysis of a beam-column problem is shown in Figure 1. An element at a distance dx, as shown in the figure, is taken from the cross-section of beam-column and is subject to both spread load q perpendicular to the axis and the axial load P, see Figure 1. Internal forces may arise in the element, and mathematically, it can be written as

$$q = -\frac{\mathrm{d}V}{\mathrm{d}x}. \tag{1}$$

Equilibrium equation in y-direction can be written as an ordinary differential equation given in [29–31]:

$$-V + q\mathrm{d}x + (V + \mathrm{d}V) = 0. \tag{2}$$

The sum of forces acting on each surface of the cross-section element will be the same because of the equilibrium state. Mathematically, it can be written as

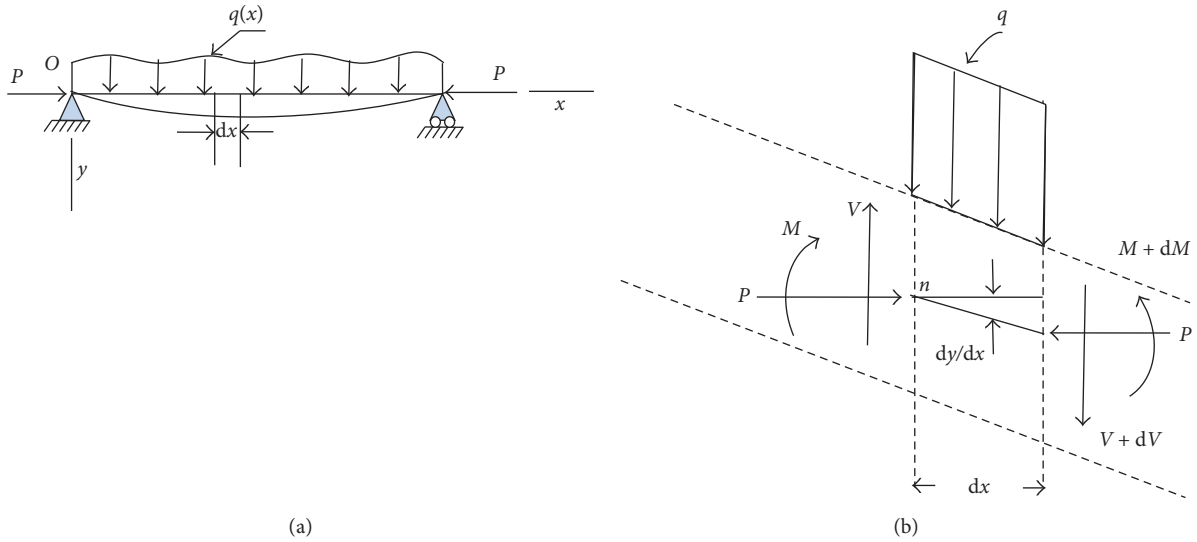


Figure 1: Cross-sectional presentation of the beam-column design.

Table 1: Comparison of approximate solutions obtained by LeNN-NM algorithm with the exact solution and RK-5 method [27] along with absolute errors for Problem 1.

t	Exact solution	LeNN-NM ($n = 11$)	RK-5 [27]	Absolute errors (RK-5 [27])	Absolute errors (LeNN-NM)
0	0	0	0	0	7.38E−06
(π/20)	0.001954	0.001954	0.001838	1.16E−04	1.86E−06
(π/10)	0.003738	0.003738	0.003585	1.53E−04	5.34E−06
(3π/20)	0.005331	0.005331	0.005210	1.21E−04	5.12E−06
(4π/20)	0.006715	0.006715	0.006674	4.18E−05	2.82E−06
(5π/20)	0.007881	0.007881	0.007933	5.21E−05	2.50E−07
(6π/20)	0.008822	0.008822	0.008938	1.17E−04	2.98E−06
(7π/20)	0.009537	0.009537	0.009749	2.12E−04	4.41E−06
(8π/20)	0.010031	0.010031	0.010073	4.21E−05	3.73E−06
(9π/20)	0.010316	0.010316	0.010311	5.37E−06	2.68E−07
(π/2)	0.010407	0.010407	0.010406	4.59E−07	6.36E−06

Table 2: Comparison of approximate solutions obtained by LeNN-NM algorithm with the exact solution and spline method [28] along with absolute errors for Problem 2.

t	Exact solution	LeNN-NM ($n = 11$)	Absolute errors (spline method) [28]	Absolute errors (LeNN-NM)
0	0	0	0.00E + 00	1.62E−06
0.2	0.195424	0.195424	1.96E−05	9.08E−06
0.4	0.358038	0.358038	3.21E−05	1.32E−05
0.6	0.437309	0.437308	3.57E−05	3.32E−06
0.8	0.356087	0.356086	2.68E−05	2.69E−06
1	0	0	0.00E + 00	3.92E−06

Table 3: Result comparison of LeNN-NM algorithm with the exact solution and spline Method [28] along with absolute errors for Problem 3.

t	Exact solution	LeNN-NM ($n = 11$)	Absolute errors (spline method) [28]	Absolute errors (LeNN-NM)
0	1	1	0.00E + 00	0.00E + 00
0.2	0.755705	0.755705	2.68E−07	2.72E−07
0.4	0.54174	0.54174	1.45E−07	3.53E−07
0.6	0.349517	0.349517	2.90E−07	3.02E−07
0.8	0.17132	0.17132	1.94E−07	1.71E−07
1	0	−2.89E−06	2.14E−06	0.00E + 00

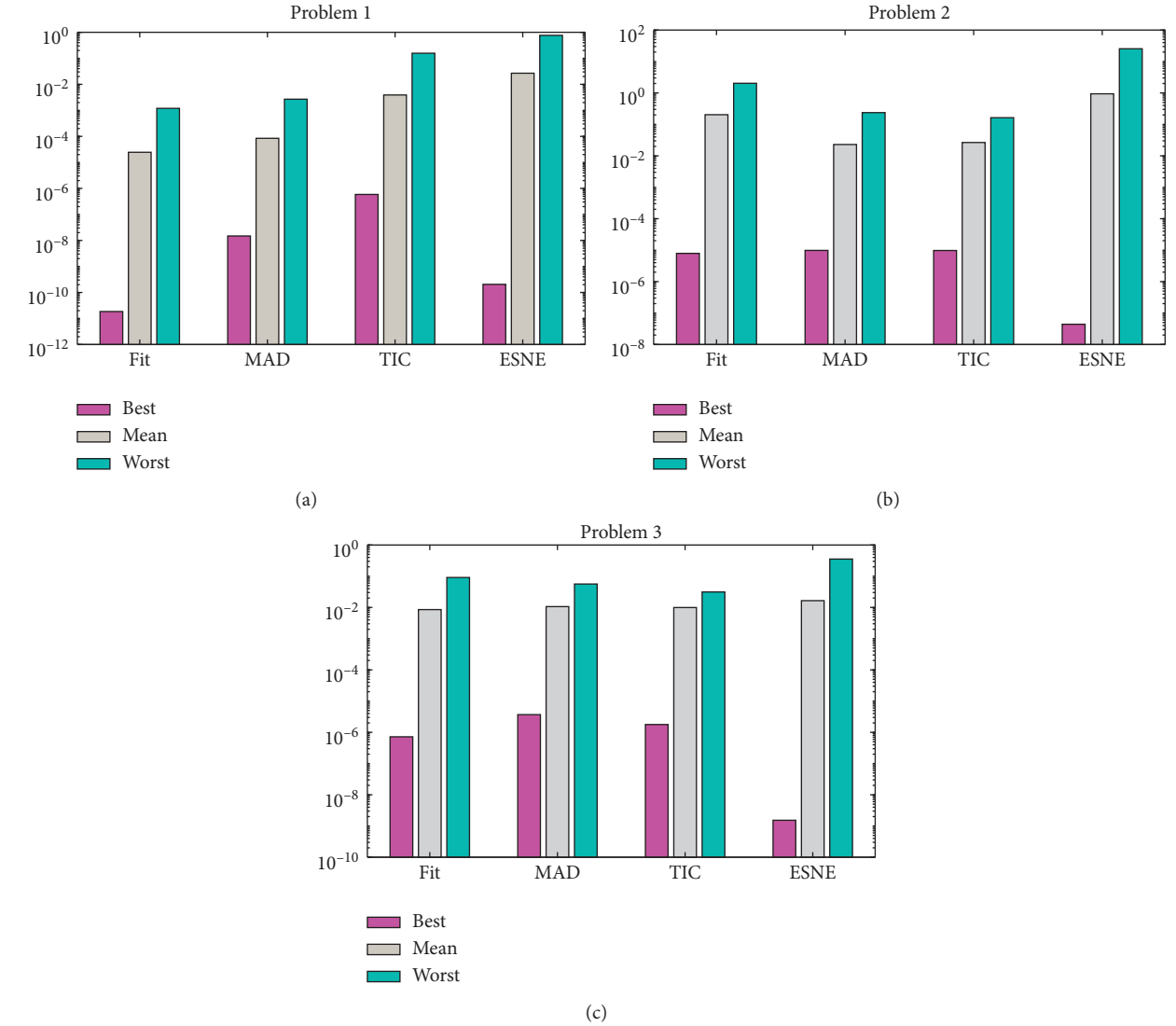


Figure 2: Performance measures for beam equations presented in Problems 1–3.

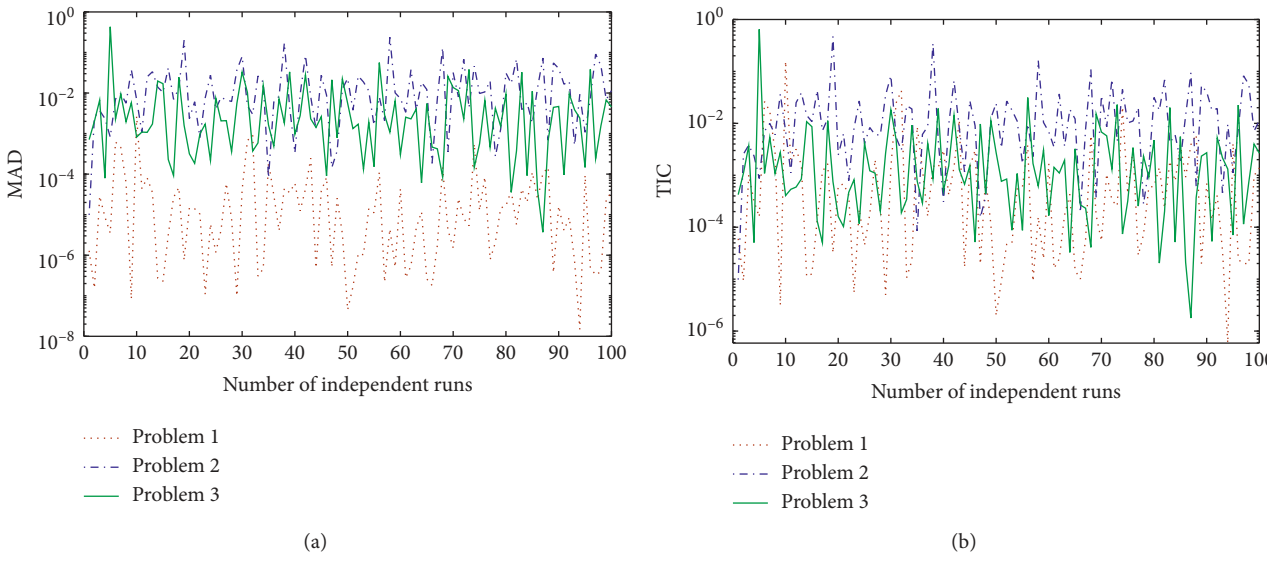
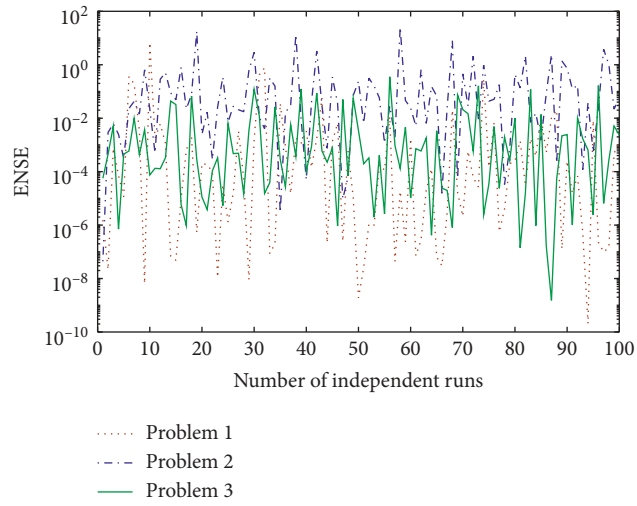


Figure 3: Continued.



(c)

Figure 3: Convergence of performance indices during 100 independent runs for beam equations presented in Problems 1–3.

$$M + q\mathrm{d}x\frac{\mathrm{d}x}{2} + (V + \mathrm{d}V)dx - (M + \mathrm{d}M) + P\frac{\mathrm{d}y}{\mathrm{d}x} = 0, \quad (3)$$

where M is the bending moment that tries to bend the cross-section element and V is the shear force acting on the surface of the element.

If we assume that rotations are small and if we neglect the terms of second order in $\mathrm{d}x$, then equation (3) becomes

$$V = \frac{\mathrm{d}M}{\mathrm{d}x} - P\frac{\mathrm{d}y}{\mathrm{d}x}. \quad (4)$$

Since the rotations are assumed to be small and $d^2/\mathrm{d}x^2 = -M/\mathrm{EI}$, then equation (4) can be written as

$$-V = \mathrm{EL}\frac{\mathrm{d}^3y}{\mathrm{d}x^3} + P\frac{\mathrm{d}y}{\mathrm{d}x}, \quad (5)$$

where bending rigidity is denoted by EL. Taking derivative on both sides of equation (5) with respect to x, then we get differential equation of fourth order for the elastic curve and is given as

$$\mathrm{EL}\frac{\mathrm{d}^4y}{\mathrm{d}x^4} + P\frac{\mathrm{d}y}{\mathrm{d}x} = q(x). \quad (6)$$

3. Approximate Solutions and Weighted Legendre Polynomials

Higher-order boundary value problems occur in several areas of applied mathematics, fluid mechanics, elasticity, and quantum mechanics, as well as other branches of science and engineering.

Consider a general form of fourth-order ODE as

$$\begin{aligned} y'''' &= f(t, y, y', y'', y'''), \quad a \le t \le b, \\ y(a) &= \alpha_1, \\ y'(b) &= \alpha_2, \\ y''(c) &= \alpha_3, \\ y'''(d) &= \alpha_4, \end{aligned} \quad (7)$$

where α_1, α_2,α_3, and α_4 are constants, t is the independent variable, y is the dependent variable, and a and b are bounds on the independent variable.

Prior to the discussion of the approximate solution in equation (1) for the differential equation, the weighted Legendre polynomials are discussed first. $Ln(t)$ denotes the Legendre polynomials. Here, n denotes order of Legendre polynomials. These polynomials are orthogonal and thus form a set of orthogonal polynomials on the basis of $[-1, 1]$. In Table 4, the first ten Legendre polynomials are given.

The recursive formula given below generates higher-order Legendre polynomials:

$$L_{n+1}(t) = \frac{1}{n+1}\left[(2n+1)tL_n(t) - nL_{n-1}(t)\right]. \quad (8)$$

Approximate series solution for equation (7) is considered as

$$y_{\mathrm{appox}}(t) = \sum_{n=0}^{N} \zeta_n L_n(\psi_n t + \theta_n), \quad (9)$$

where ζ_n, ψ_n, and θ_n are unknown parameters. Since, nth order continuous derivatives of equation (9) exist, so first, second, third, and fourth derivative of equation (9) is given by

Table 4: First ten Legendre polynomials.

n	$L_n(t)$
0	1
1	t
2	$(1/2)(3t^2-1)$
3	$(1/2)(5t^3-3t)$
4	$(1/8)(35t^4-30t^2+3)$
5	$(1/8)(63t^5-70t^3+15t)$
6	$(1/16)(231t^6-315t^4+105t^2-5)$
7	$(1/16)(429t^7-693t^5+315t^3-35t)$
8	$(1/128)(6435t^8-12012t^6+6930t^4-1260t^2+35)$
9	$(1/128)(12155t^9-25740t^7+18018t^5-4620t^3+315t)$
10	$(1/256)(46189t^{10}-109395t^8+90090t^6-30030t^4+3465t^2-63)$

$$y'_{\text{appox}}(t) = \sum_{n=1}^{N} \zeta_n L'_n(\psi_n t + \theta_n), \tag{10}$$

$$y''_{\text{appox}}(t) = \sum_{n=4}^{N} \omega_n L''_n(\psi_n t + \theta_n), \tag{11}$$

$$y'''_{\text{appox}}(t) = \sum_{n=4}^{N} \zeta_n L'''_n(\psi_n t + \theta_n), \tag{12}$$

$$y''''_{\text{appox}}(t) = \sum_{n=4}^{N} \zeta_n L'''_n(\psi_n t + \theta_n), \tag{13}$$

where ζ_n, ψ_n, and θ_n are real-valued unknown parameters. Plugging equations (10)–(13) in equation (7),

$$\begin{aligned} y''''_{\text{approx}} &= f\left(t, y_{\text{approx}}, y'_{\text{approx}}, y''_{\text{approx}}, y'''_{\text{approx}}\right), \\ y_{\text{approx}}(a_1) &= b_1, \\ y'_{\text{approx}}(a_2) &= b_2, \\ y_{\text{approx}}(a_3) &= b_3, \\ y'_{\text{approx}}(a_4) &= b_4, \end{aligned} \tag{14}$$

where $a \le t \le b$. Equation (7) is now converted into an equivalent algebraic system of equations. Equation (14) can be solved for unknown parameters ζ_n, ψ_n, and θ_n using an efficient solver such as the LeNN-NM algorithm. All the symbols and abbreviations used in the paper are defined in Table 5.

4. Nelder–Mead Algorithm

Nelder–Mead (NM) algorithm is a single-path following optimizer and is used in this paper to optimize the fitness function developed for each problem. To reduce a function, NM algorithm sets up the simplex by using $n+1$ points. Such points are defined as the vector directions on an n-dimensional search space. In recent times, NM algorithm is used to find numerical solution of the dynamical model of Li-ion batteries for electric vehicle [32], nonlinear Muskingum models [33], economic load dispatch problem with valve point loading effect [34], optimization of TIG welding parameters [35], optimization of noisy CNLS problems [36], and parameter identification of chaotic systems [37]. Implementation of NM algorithm is based on four basic operators [38]. Below, we present details of these operators:

4.1. Reflection. Reflection point is determined by

$$X_r = \overline{X} - \alpha\left(X_{n+1-\overline{X}}\right), \tag{15}$$

where $\overline{X}$ is called centroid and is defined as

$$\overline{X} = \sum_{i=1}^{n} \frac{X_i}{n}. \tag{16}$$

Reflection coefficient is denoted by α. "X_r" is accepted, and iterations are terminated if $f(X_1) \le f(X_r) < f(X_n)$.

4.2. Expansion. To calculate expansion point, equation (17) is used:

$$X_e = \overline{X} - \beta(\overline{X} - X_r). \tag{17}$$

Equation (17) will be evaluated if value of function at X_r is less then X_1. Expansion coefficient is denoted by β. If $f(X_e) \le f(X_r)$, then we accept "X_e," and the process of iteration is terminated. X_r will be accepted if the above condition is not satisfied.

4.3. Contraction. If $f(X_r) \ge f(X_n)$, then the process of contraction occurs.

(1) Outside contraction is performed by equation (18) if $f(X_r) < f(X_{n+1})$:

$$X_c = \overline{X} + \gamma(X_{n+1} - \overline{X}), \tag{18}$$

where "γ" is coefficient of contraction. We accept "X_c" if $f(X_c) \le f(X_r)$. Otherwise, NM algorithm will move to the shrinking step.

(2) Inside contraction is performed $X_c = \overline{X} - \gamma(\overline{X} - X_{n+1})$ if $f(X_r) \ge f(X_{n+1})$. Otherwise, NM algorithm will move to the shrink step.

4.4. Shrink. The process of shrinking is modeled by

Table 5: Notations and abbreviations used in this paper.

Abbreviation	Description	Abbreviation	Description
LeNN	Legendre neural networks	α	Reflection coefficient
NM	Nelder–Mead algorithm	β	Expansion coefficient
MAD	Mean absolute deviation	γ	Contraction coefficient
TIC	Theil's inequality coefficient	δ	Shrink coefficient.
NSE	Nash Sutcliffe efficiency	M	Bending moment
ENSE	Error in Nash Sutcliffe efficiency	V	Shear force
P	Axial load	EL	Bending rigidity
q	Load perpendicular to axis	L_n	Legendre polynomials
ζ_n, ψ_n, and θ_n	Neurons in LeNN	X_r	Reflection point
X_e	Expansion point	$\overline{X}$	Centroid

$$V_i = X_1 - \delta(X_1 - X_i), \quad i = 2, \ldots, n+1, \tag{19}$$

where "δ" denotes shrink coefficient. The subsequent shrinkage of the simplex is expressed as $V = \{X_1, V_1, V_2, \ldots, V_{n+1}\}$ for the succeeding iterations.

5. Our Proposed LeNN-NM Algorithm

Steps for the proposed technique are summarized as follows:

Step 1: initialize a random population of weights and define a fitness function. Approximate solution is constructed by using activation function given by equation (9).

Step 2: solution along with higher derivatives are approximated at arbitrary generated search points.

Step 3: put approximate solution along with higher derivatives in a given differential equation.

Step 4: an equivalent set of system of algebraic equations corresponding to given differential equation will be generated by Step 3.

Step 5: optimize the system unknown parameters ζ_n, ψ_n, and θ_n using LeNN.

Step 6: Nelder–Mead algorithm will start the process of optimizing the system by taking ζ_n, ψ_n, and θ_n as its initial guess.

Step 7: Nelder–Mead algorithm evaluates the fitness function, and the results will be displayed when the stopping criteria are achieved.

Step 8: best values for the weights ζ_n, ψ_n, and θ_n obtained will be plugged in approximate solution equation (9).

Step 9: it will be the solution.

Working strategy of proposed algorithm is also shown in Figure 4. Parameters' setting for LeNN and NM algorithm is given in Table 6.

6. Performance Indicators

To study the performance of LeNN-NM technique for solving Beam equations, the performance indications such as mean absolute deviation (MAD), Theil's inequality coefficient (TIC), and error in Nash Sutcliffe efficiency (ENSE) are implemented. The formulation of these performance indices are given by [38]

$$\begin{aligned}
\mathrm{MAD} &= \frac{1}{n}\sum_{m=1}^{n}\left|y(t) - y_{\mathrm{approx}}(t)\right|,\\
\mathrm{TIC} &= \frac{\sqrt{(1/n)\sum_{n=1}^{n}\left(y(t) - y_{\mathrm{approx}}(t)\right)^2}}{\left(\sqrt{(1/n)\sum_{n=1}^{n}\left(y(t)\right)^2} + \sqrt{(1/n)\sum_{n=1}^{n}\left(y_{\mathrm{approx}}(t)\right)^2}\right)},\\
\mathrm{NSE} &= \left\{1 - \frac{\sum_{n=1}^{n}\left(\left(y(t) - y_{\mathrm{approx}}(t)\right)^2\right.}{\sum_{n=1}^{n}\left(\left(y(t) - \overline{y}(t)^2\right)\right.},\right.
\end{aligned} \tag{20}$$

where

$$\begin{aligned}
\overline{y}(t) &= \frac{1}{n}\sum_{m=1}^{n} y(t),\\
\mathrm{ENSE} &= 1 - \mathrm{NSE},
\end{aligned} \tag{21}$$

where n represents the number the mesh points.

7. Numerical Applications

Problem 1. Consider the homogenous beam equation of the fourth order with bending rigidity [27], EL = 1, $P = -1$, and $q(t) = 0$:

$$\frac{d^4y}{dt^4} + \frac{d^2y}{dt^2} = 0, \quad 0 \le t \le \frac{\pi}{2}, \tag{22}$$

with boundary conditions

$$\begin{aligned}
y(0) &= 0,\\
y'(0) &= \frac{-1.1}{72 - 50\pi},\\
y''(0) &= \frac{1}{144 - 100\pi},\\
y'''(0) &= \frac{1.2}{144 - 100\pi}.
\end{aligned} \tag{23}$$

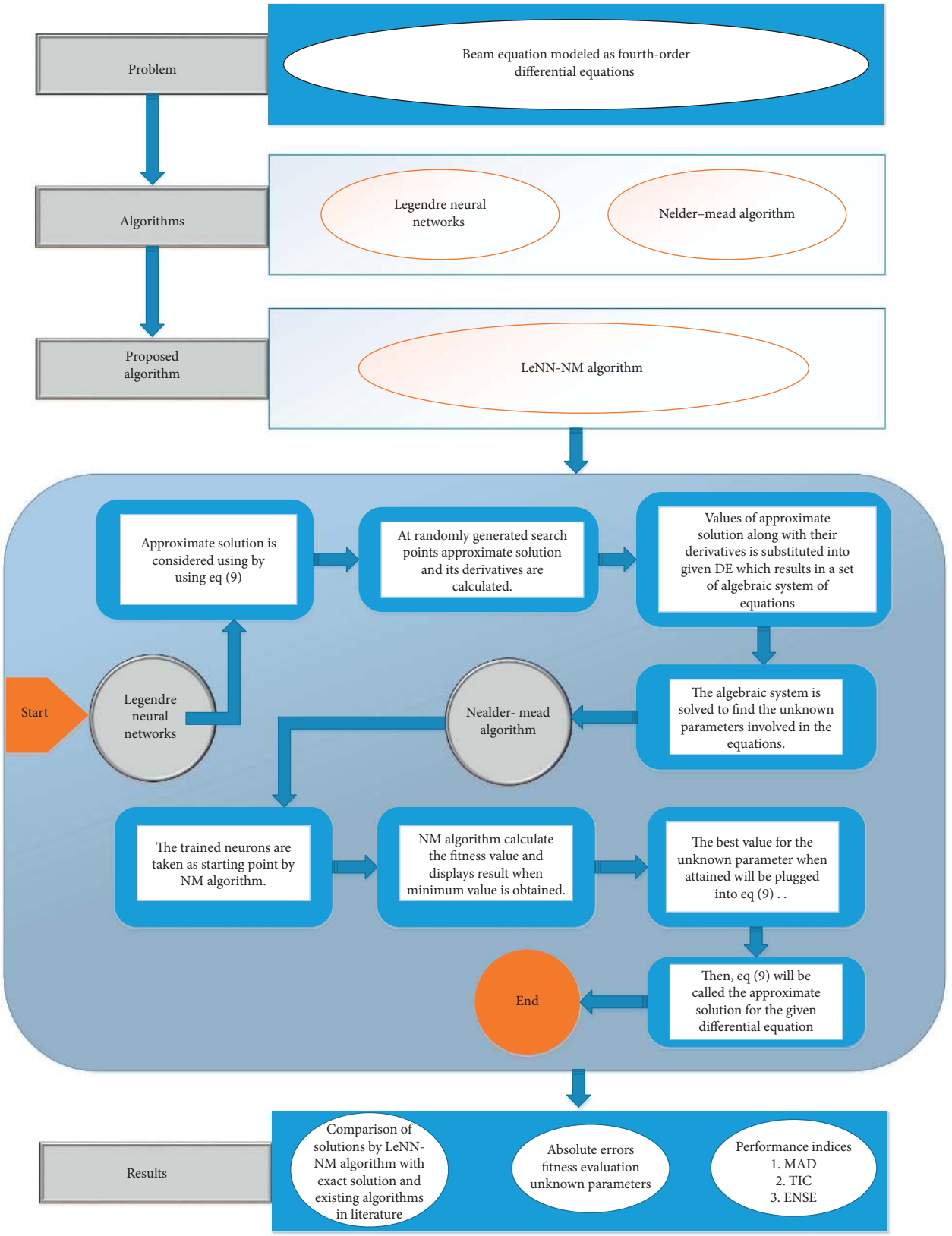


Figure 4: Flowchart of LeNN-NM algorithm for solving beam-column models.

Table 6: Setting of parameters for LeNN and NM algorithm.

Algorithm	Parameters	Settings	Parameters	Settings
LeNN	Limits	[−1, 1]	Max. iterations	6,000
	Candidate selection	Uniform	Search agents	40
Nelder–Mead algorithm	Initial weights	Global best from LeNN	Max evaluation	100,000
	X-tolerance TolX	1.0E−100	Max. iterations	2,000
	Scaling	Objective and constraints	'TolFun'	1.00E−100

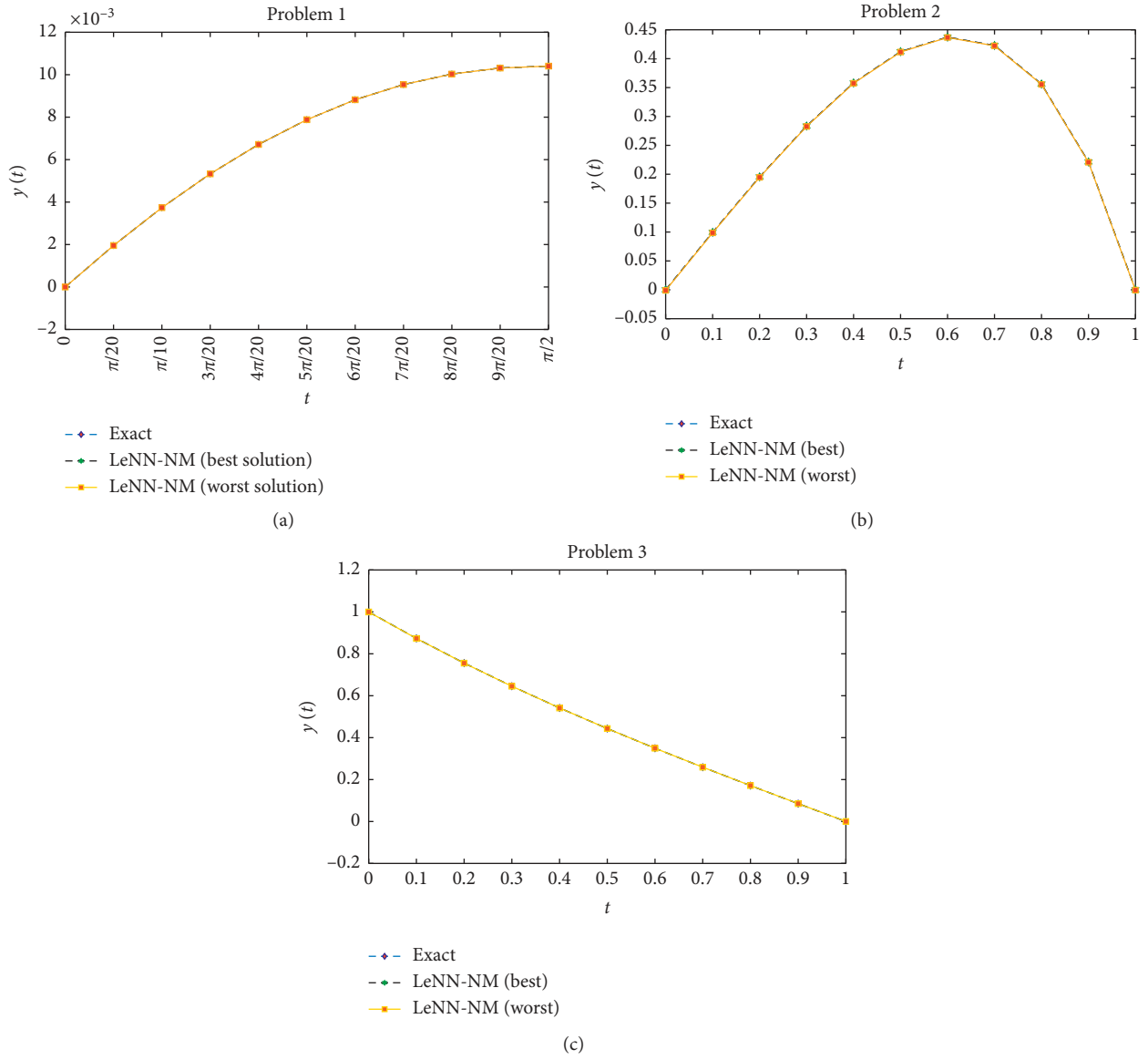


Figure 5: Best and worst approximate solution obtained by LeNN-NM algorithm along with exact solution for Problem 1, 2 and 3.

The exact solution for the physical problem modeled as equation (22) is given by

$$y(t) = \frac{1 - t - \cos(t) - 1.2\ \sin(t)}{144 - 100\pi}. \tag{24}$$

The result obtain by LeNN-NM algorithm along with absolute errors is given in Table 1 and graphically illustrated through Figures 5(a) and 6(a), respectively. The unknown parameters achieved by the proposed method are revealed in Table 7. The result of MAD, TIC, and ESNE for equation (22) is shown in Figures 2 and 3. Normal provability plots for fitness evaluation, MAD, TIC, and ENSE, are shown in Figures 7–10, respectively. Hence, the graphs and tables show the dominance of LeNN-NM algorithm in solving higher-order ordinary differential

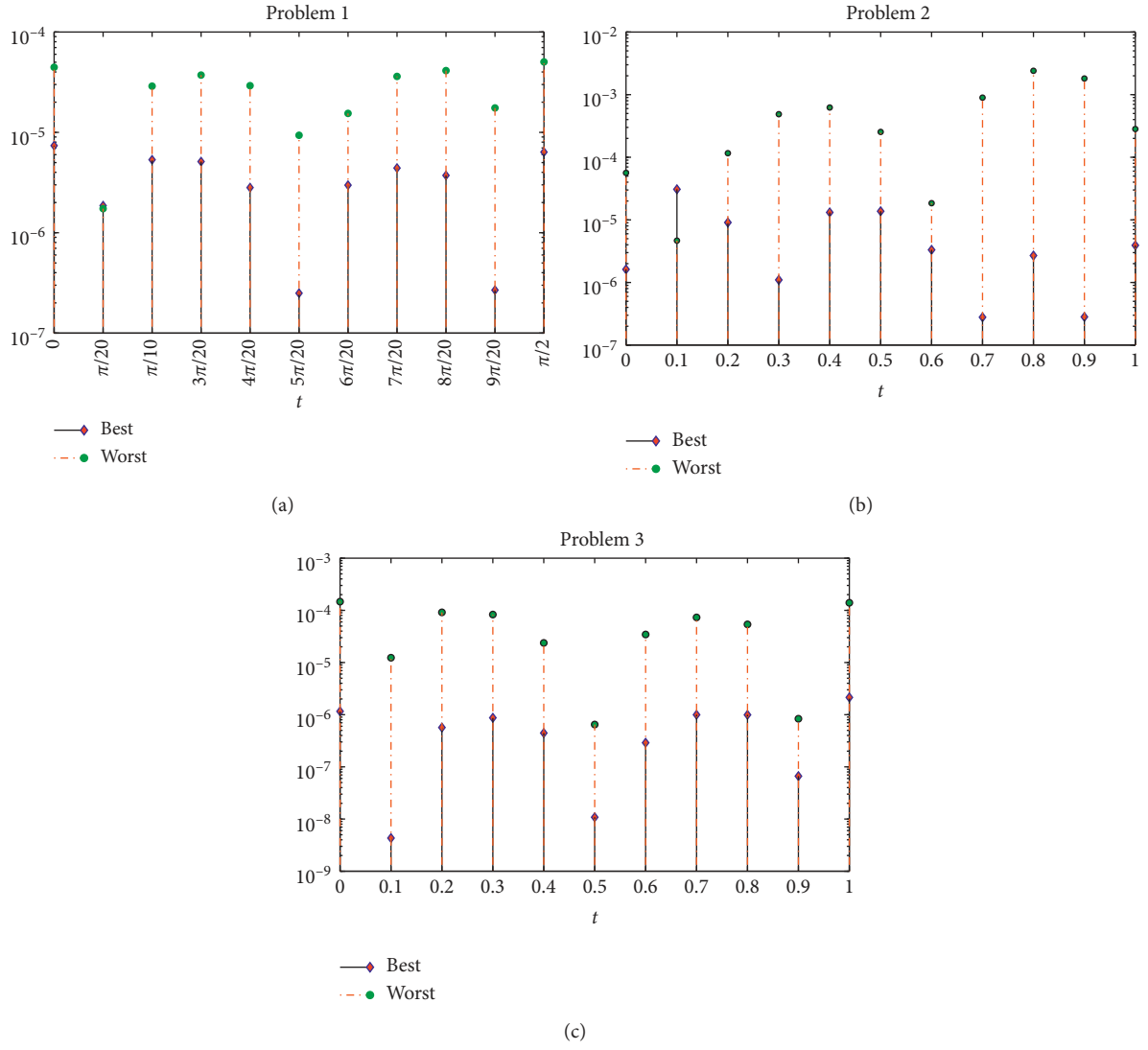


Figure 6: Comparison between best and worst absolute errors for Problems 1–3.

equations subjecting to multiple initial or boundary conditions.

Problem 2. Consider the nonhomogenous linear beam equation of the fourth order with bending rigidity [28], EL = 1, $P = -2$, and $q(t) = -8e^t$, for

$$\frac{d^4y}{dt^4} - 2\frac{d^2y}{dt^2} + y = -8e^t, \quad t \in [0,1], \tag{25}$$

with boundary conditions

$$\begin{aligned} y(0) &= y(1) = 0, \\ y''(0) &= 0, \\ y''(1) &= -4e. \end{aligned} \tag{26}$$

The exact solution for equation (25) is $y(t) = t(1-t)e^t$. The result obtained by LeNN-NM algorithm is compared with the spline method [28] and given in Table 2. Figures 5(b) and 6(b) show that the solution obtained by proposed algorithm overlaps the exact solution with minimum absolute errors. Unknown neurons in LeNN structure for optimization of Problem 2 are shown in Figure 11. The result of MAD, TIC, and ESNE for equation (24) is shown in Figures 2 and 3. Normal provability plots for fitness evaluation, MAD, TIC, and ENSE, are shown in Figures 7–10, respectively. The statistical analysis shows the dominance of the proposed method.

Problem 3. Consider the homogenous linear beam equation of the fourth order with bending rigidity [28], EL = 1, $P = 0$, and $q(t) = 0$:

$$\frac{d^4y}{dt^4} - y = 0, \quad t \in [0,1], \tag{27}$$

with boundary conditions

$$y(0) = y''(0) = 1,$$
$$y(1) = y''(1) = 0. \tag{28}$$

The exact solution for equation (27) is given by

$$y(t) = \frac{1}{2\sinh(1)}\left(e^{1-t} - e^{t-1}\right). \tag{29}$$

The result obtained by LeNN-NM algorithm along with absolute errors are given in Table 3 and graphically presented through Figures 5(c) and 6(c), respectively. The unknown parameters achieved by the proposed method are revealed in Table 7. Convergence of the fitness value during 100 independent runs of the proposed algorithm is shown in Figure 12. The result of MAD, TIC, and ESNE for equation (27) is shown in Figures 2 and 3. Normal provability plots for fitness evaluation and performance measures are shown through Figures 7–10, respectively.

Approximate series solution for Problem 1 is given as

$$\begin{aligned}
\widehat{y}_{P1} &= 0.360045 + (-0.040634t - 0.039017)(0.093106) \\
&+ \left(\frac{3(0.094271t - 0.041539)^2 - 1}{2}\right)(0.552727) \\
&+ \left(\frac{5(-0.04037t - 0.01940)^3 - 3(-0.04037t - 0.01940)}{2}\right)(0.160012) \\
&+ \left(\frac{35(-0.1294t - 0.0835)^4 - 30(-0.1294t - 0.0835)^2 + 3}{8}\right)(-0.2591) \\
&+ \left(\frac{63(0.022929t - 0.112845)^5 - 70(0.022929t - 0.112845)^3}{8}\right. \\
&\left. + \frac{15(0.022929t - 0.112845)}{8}\right)(0.087927) \\
&+ \left(\frac{231(-0.0002t - 0.0212)^6 - 315(-0.0002t - 0.0212)^4}{16}\right. \\
&\left. + \frac{105(-0.0002t - 0.0212)^2 - 5}{16}\right)(0.3193) \\
&+ \left(\frac{429(0.2662t - 0.0298)^7 - 693(0.2662t - 0.0298)^5}{16}\right. \\
&\left. + \frac{315(0.2662t - 0.0298)^2 - 35(0.2662t - 0.0298)}{16}\right)(0.5709) \\
&+ \left(\frac{6435(0.2662t - 0.0298)^8 - 12012(0.2662t - 0.0298)^6}{128}\right. \\
&\left. + \frac{6930(0.2662t - 0.0298)^4 - 1260(0.2662t - 0.0298)^2 + 35}{128}\right)(0.571) \\
&+ \left(\frac{12155(-0.01207t + 0.1319)^9 - 25740(-0.01207t + 0.1319)^7}{128} + + \frac{18018(-0.01207t + 0.1319)^5 - 4620(-0.01207t + 0.1319)^3}{128}\right. \\
&\left. + \frac{315(-0.01207t + 0.1319)}{128}\right)(-0.11928) \\
&+ \left(\frac{46189(-0.00597t + 0.00983)^{10} - 109395(-0.00597t + 0.00983)^8}{256} + \frac{90090(-0.00597t + 0.00983)^6 - 30030(-0.00597t + 0.00983)^4}{256}\right. \\
&\left. + \frac{3465(-0.00597t + 0.00983)^2 - 63}{256}\right)(-0.13750).
\end{aligned} \tag{30}$$

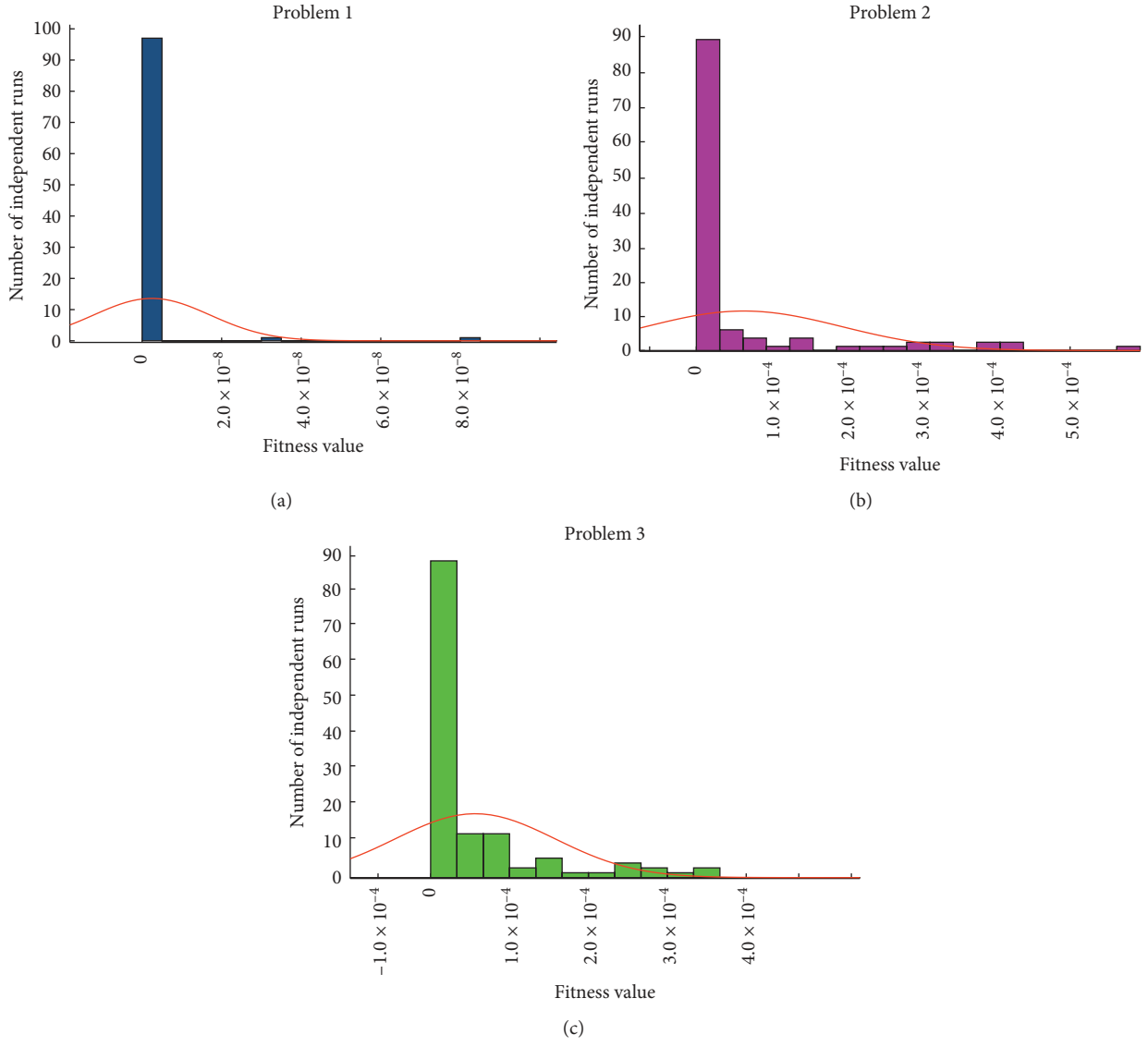


Figure 7: Normal probability plots of fitness values obtained during 100 experiments for Problems 1–3.

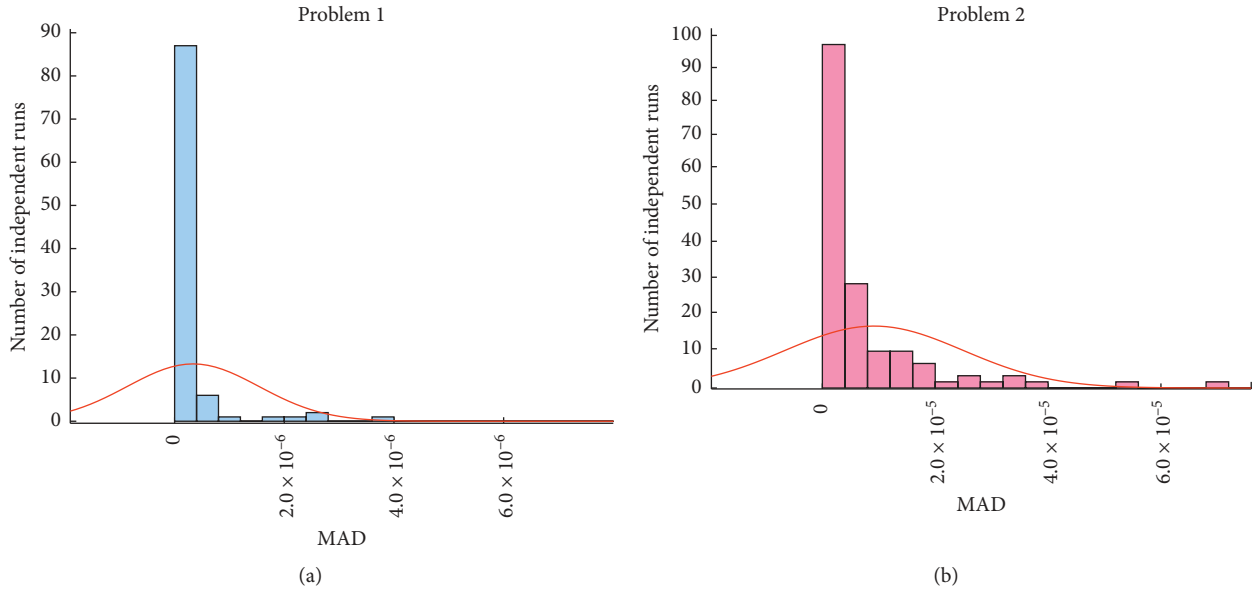
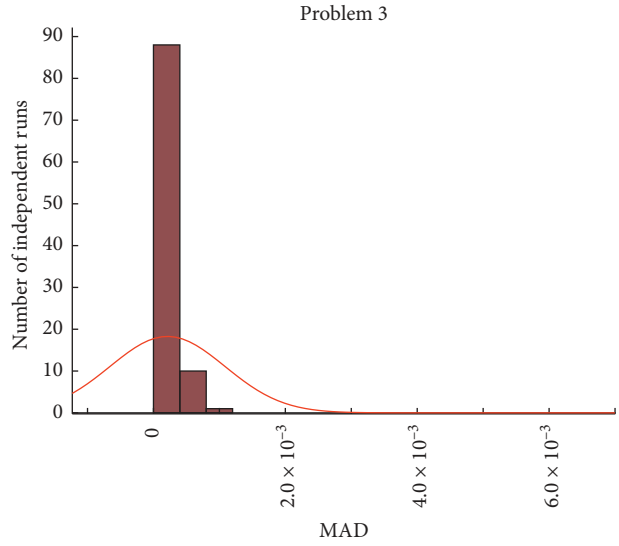
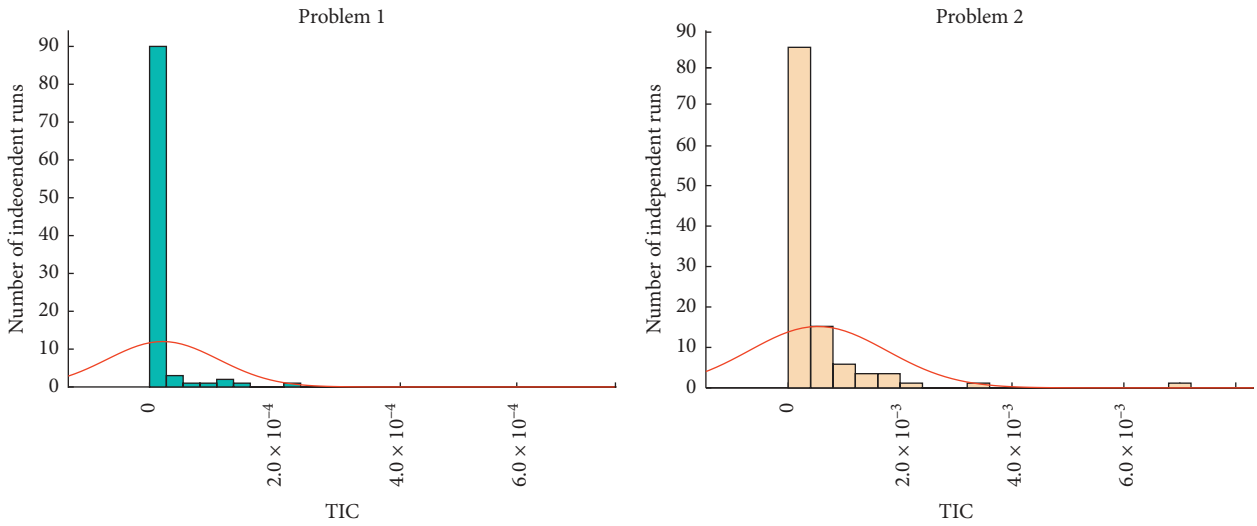


Figure 8: Continued.



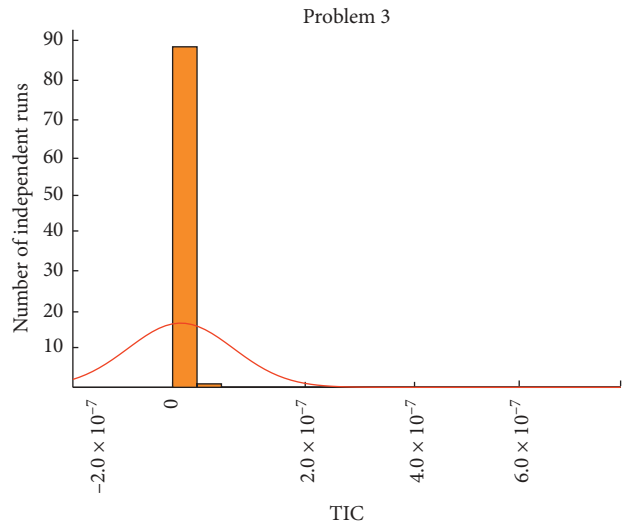
(c)

Figure 8: Normal probability plots of fitness values obtained during 100 experiments for Problem 1–3.



(a)

(b)



(c)

Figure 9: Normal probability plots of TIC obtained during 100 experiments for Problem 1–3.

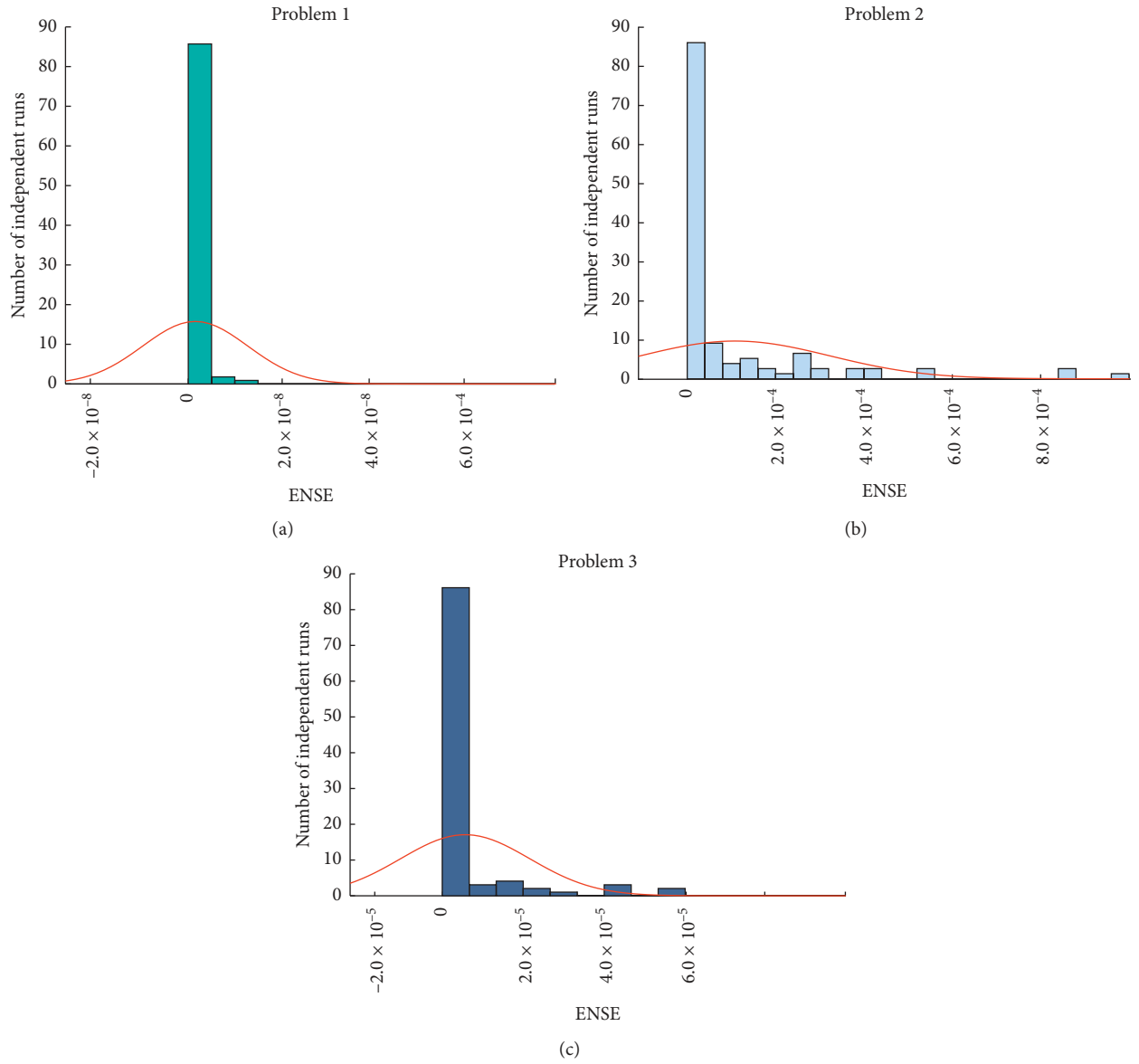


Figure 10: Normal probability plots of ENSE obtained during 100 experiments for Problem 1–3.

Table 7: Trained neurons obtained by LeNN-NM algorithm for Problems 1–3.

	Problem 1			Problem 2			Problem 3		
Index	ζ_n	ψ_n	θ_n	ζ_n	ψ_n	θ_n	ζ_n	ψ_n	θ_n
1	0.095393	−0.00751	0.360046	−0.08189	−1.87335	−0.25355	0.301072	0.050994	0.081955
2	−0.04063	−0.03902	0.093106	−0.59112	0.017422	−1.02846	0.11172	0.147113	−0.32471
3	0.094272	−0.04154	0.552728	0.024504	−0.42908	−0.05271	0.070749	−0.85036	1.372923
4	−0.04037	−0.0194	0.160012	−0.52688	−0.72334	0.014351	0.121341	0.712113	0.608768
5	−0.12942	−0.08346	−0.25919	0.109931	−0.78767	−0.5388	−0.01146	−0.09546	0.091558
6	0.022929	−0.11285	0.087927	0.042851	−0.66665	−0.29808	0.463903	0.224941	−0.46237
7	−0.05836	−0.09943	−0.02973	−0.01336	0.132638	−0.02625	0.729139	−0.04045	0.176563
8	−0.00019	−0.02121	0.319345	−0.02041	−0.04962	−0.80236	0.4657	0.061354	0.227969
9	0.266188	−0.0298	0.570869	0.270267	−0.27792	−0.35499	0.717546	0.043049	−0.3886
10	−0.01207	0.1319	−0.11928	0.125709	0.131585	−0.75108	0.012412	0.138911	0.165863
11	−0.00597	0.009826	−0.1375	−0.36437	−0.18466	−0.93043	0.511784	0.103503	−0.37962

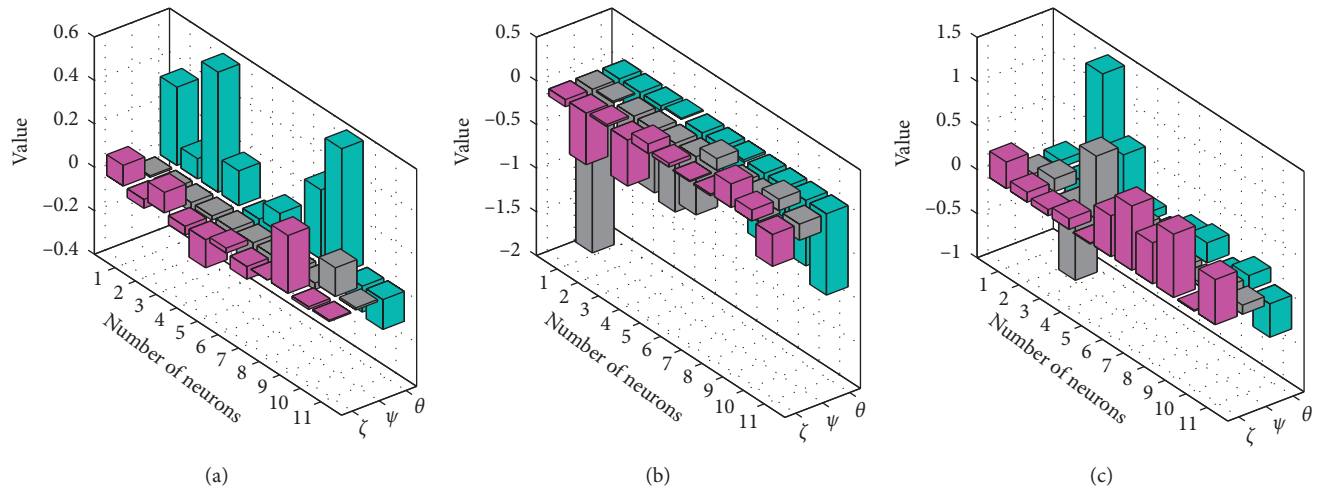


Figure 11: Trained neurons obtained by LeNN-NM algorithm for Problems 1–3. (a) Weights for Problem 1. (b) Weights for Problem 2. (c) Weights for Problem 3.

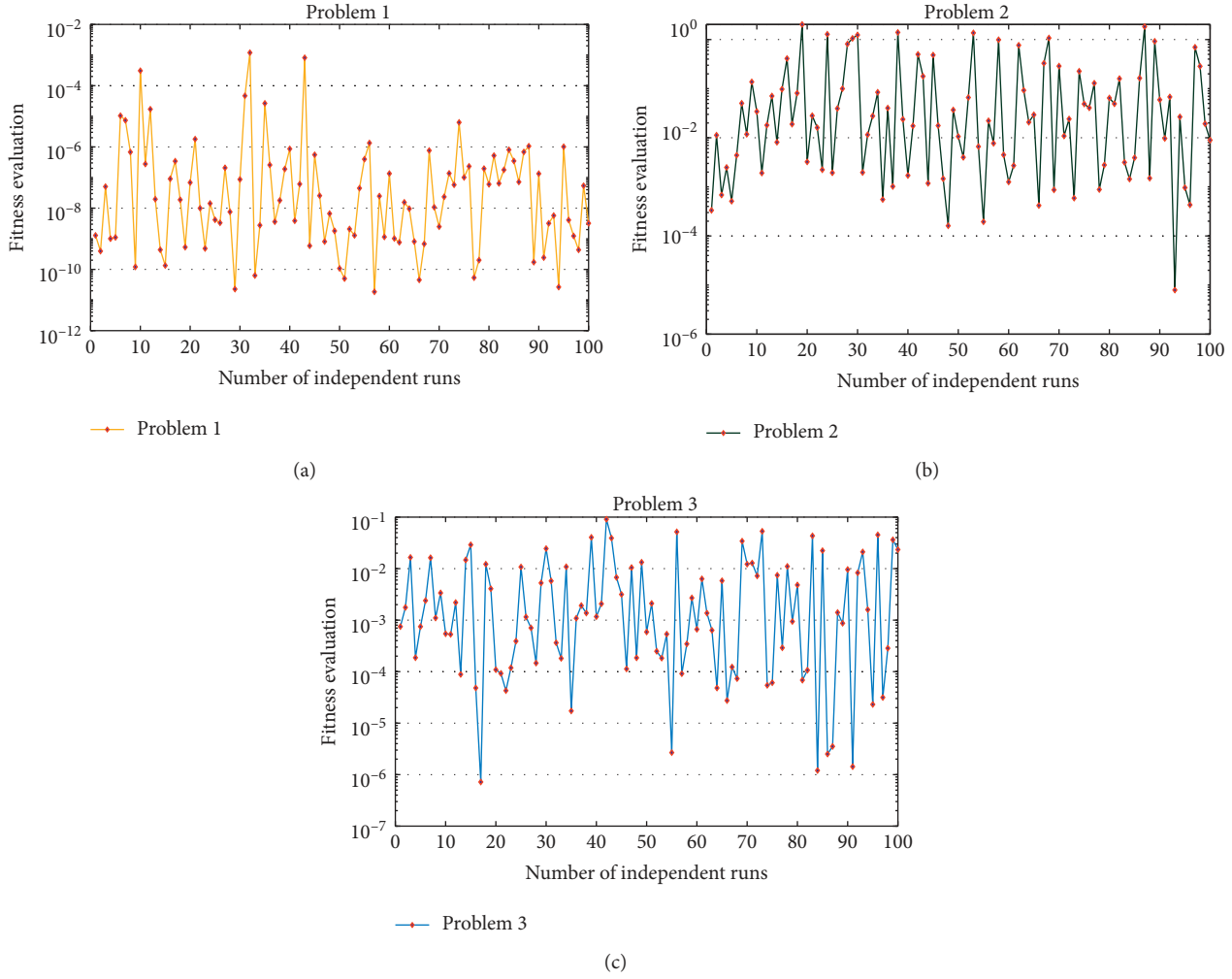


Figure 12: Convergence analysis of Problems 1–3 by LeNN-NM algorithm.

Approximate series solution for Problem 2 is given as

$$\begin{aligned}
\widehat{y}_{P2} &= -0.253550 + (-0.591120t + 0.017422)(-1.028463) \\
&+\left(\frac{3(0.024503t - 0.429078)^2 - 1}{2}\right)(-0.052705) \\
&+\left(\frac{5(-0.526876t - 0.723342)^3 - 3(-0.526876t - 0.723342)}{2}\right)(0.014351) \\
&+\left(\frac{35(0.1099t - 0.7878)^4 - 30(0.1099t - 0.7878)^2 + 3}{8}\right)(-0.5388) \\
&+\left(\frac{63(0.042850t - 0.666649)^5 - 70(0.042850t - 0.666649)^3}{8} + \frac{15(0.042850t - 0.666649)}{8}\right)(-0.298076) \\
&+\left(\frac{231(-0.0134t + 0.1326)^6 - 315(-0.0134t + 0.1326)^4}{16} + \frac{105(-0.0134t + 0.1326)^2 - 5}{16}\right)(-0.0262) \\
&+\left(\frac{429(-0.0204t - 0.0496)^7 - 693(-0.0204t - 0.0496)^5}{16} + \frac{315(-0.0204t - 0.0496)^2 - 35(-0.0204t - 0.0496)}{16}\right)(-0.8023) \\
&+\left(\begin{array}{c} \dfrac{6435(0.2703t - 0.2779)^8 - 12012(0.2703t - 0.2779)^6}{128} \\ \\ +\dfrac{6930(0.2703t - 0.2779)^4 - 1260(0.2703t - 0.2779)^2 + 35}{128} \end{array}\right)(-0.3549) \\
&+\left(\begin{array}{c} \dfrac{12155(0.1257t + 0.1316)^9 - 25740(0.1257t + 0.1316)^7}{128} + \dfrac{18018(0.1257t + 0.1316)^5 - 4620(0.1257t + 0.1316)^3}{128} \\ \\ +\dfrac{315(0.1257t + 0.1316)}{128} \end{array}\right)(-0.75106) \\
&+\left(\begin{array}{c} \dfrac{46189(-0.3644t - 0.1846)^{10} - 109395(-0.3644t - 0.1846)^8}{256} + \dfrac{90090(-0.3644t - 0.1846)^6 - 30030(-0.3644t - 0.1846)^4}{256} \\ \\ +\dfrac{3465(-0.3644t - 0.1846)^2 - 63}{256} \end{array}\right)(-0.930432).
\end{aligned} \tag{31}$$

Approximate series solution for Problem 3 is given as

$$
\begin{aligned}
\widehat{y}_{P3} &= 0.081955 + (0.111719t + 0.147113)(-0.324711) \\
&+ \left(\frac{3(0.070749t - 0.850358)^2 - 1}{2}\right)(1.372923) \\
&+ \left(\frac{5(0.121341t + 0.712113)^3 - 3(0.121341t + 0.712113)}{2}\right)(0.608767) \\
&+ \left(\frac{35(-0.0115t - 0.0955)^4 - 30(-0.0115t - 0.0955)^2 + 3}{8}\right)(0.0916) \\
&+ \left(\frac{63(0.463903t + 0.224941)^5 - 70(0.463903t + 0.224941)^3}{8} + \frac{15(0.463903t + 0.224941)}{8}\right)(-0.462369) \\
&+ \left(\frac{231(0.7291t - 0.04045)^6 - 315(0.7291t - 0.04045)^4}{16} + \frac{105(0.7291t - 0.04045)^2 - 5}{16}\right)(0.1766) \\
&+ \left(\begin{array}{c} \dfrac{429(0.4657t + 0.06135)^7 - 693(0.4657t + 0.06135)^5}{16} \\ + \dfrac{315(0.4657t + 0.06135)^2 - 35(0.4657t + 0.06135)}{16} \end{array}\right)(0.2279) \\
&+ \left(\begin{array}{c} \dfrac{6435(0.7175t + 0.04305)^8 - 12012(0.7175t + 0.04305)^6}{128} \\ + \dfrac{6930(0.7175t + 0.04305)^4 - 1260(0.7175t + 0.04305)^2 + 35}{128} \end{array}\right)(-0.3886) \\
&+ \left(\begin{array}{c} \dfrac{12155(0.0124t + 0.1389)^9 - 25740(0.0124t + 0.1389)^7}{128} + \dfrac{18018(0.0124t + 0.1389)^5 - 4620(0.0124t + 0.1389)^3}{128} \\ + \dfrac{315(0.0124t + 0.1389)}{128} \end{array}\right)(0.16586) \\
&+ \left(\begin{array}{c} \dfrac{46189(0.5118t + 0.1035)^{10} - 109395(0.5118t + 0.1035)^8}{256} + \dfrac{90090(0.5118t + 0.1035)^6 - 30030(0.5118t + 0.1035)^4}{256} \\ + \dfrac{3465(0.5118t + 0.1035)^2 - 63}{256} \end{array}\right)(-0.379615).
\end{aligned}
\tag{32}
$$

8. Conclusion

In this paper, we have modeled physical problems arising in the beam-column theory as the fourth ordinary differential equation. Moreover, the paper presents a technique which is named as LeNN-NM algorithm for solving an ordinary differential equation of higher order subjecting to single or multi-initial boundary conditions. The different cases of beam equations are studied and have been solved by the proposed technique. The results given in Tables 1–3 show that LeNN-NM algorithm converges rapidly and dominates the existing algorithm in literature for finding solutions to higher-order differential equations such as beam equations. The analysis reveals that the proposed algorithm looks like a promising methodology to be exploited as an alternate, accurate, reliable, and robust computing framework for solving a variety of the problems arising in astrophysics, atomic physics, plasma physics, nonlinear optic, electric machines, nanotechnology, fuel ignition model, fluid dynamics, bioinformatics, and financial mathematics. The proposed method gives the direction of using the proposed algorithm for solving fractional differential equations.

Data Availability

The data used to support the findings of the study are available within the article.

Conflicts of Interest

The authors have no conflicts of interest to declare.

Authors' Contributions

All authors have seen and agreed with the contents of the manuscript.

Acknowledgments

This work was sponsored in part by the Open Project Program of Fujian University Engineering Research Center of Disaster Prevention and Reduction of Southeast Coastal Engineering Structure (no. 2019004).

References

[1] M. M. Chawla and C. P. Katti, "Finite difference methods for two-point boundary value problems involving high order differential equations," *Bit*, vol. 19, no. 1, pp. 27–33, 1979.

[2] S. T. Mohyud-Din and M. A. Noor, "Homotopy perturbation method for solving fourth-order boundary value problems," *Mathematical Problems in Engineering*, vol. 2007, Article ID 098602, 15 pages, 2007.

[3] M. A. Noor and S. T. Mohyud-Din, "Homotopy perturbation method for solving sixth-order boundary value problems," *Computers & Mathematics with Applications*, vol. 55, no. 12, pp. 2953–2972, 2008.

[4] J. Ali, S. Islam, S. Islam, and G. Zaman, "The solution of multipoint boundary value problems by the optimal homotopy asymptotic method," *Computers & Mathematics with Applications*, vol. 59, no. 6, pp. 2000–2006, 2010.

[5] M. Tatari and M. Dehghan, "The use of the Adomian decomposition method for solving multipoint boundary value problems," *Physica Scripta*, vol. 73, no. 6, pp. 672–676, 2006.

[6] A.-M. Wazwaz, "A new algorithm for calculating Adomian polynomials for nonlinear operators," *Applied Mathematics and Computation*, vol. 111, no. 1, pp. 33–51, 2000.

[7] A.-M. Wazwaz, "The numerical solution of sixth-order boundary value problems by the modified decomposition method," *Applied Mathematics and Computation*, vol. 118, no. 2-3, pp. 311–325, 2001.

[8] E. H. Doha, A. H. Bhrawy, and R. M. Hafez, "A Jacobi-Jacobi dual-Petrov-Galerkin method for third- and fifth-order differential equations," *Mathematical and Computer Modelling*, vol. 53, no. 9-10, pp. 1820–1832, 2011.

[9] E. H. Doha, W. M. Abd-Elhameed, and M. A. Bassuony, "New algorithms for solving high even-order differential equations using third and fourth chebyshev-galerkin methods," *Journal of Computational Physics*, vol. 236, pp. 563–579, 2013.

[10] Z. Shi and F. Li, "Numerical solution of high-order differential equations by using periodized Shannon wavelets," *Applied Mathematical Modelling*, vol. 38, no. 7-8, pp. 2235–2248, 2014.

[11] I. Aziz, M. Siraj-Ul-Islam, and M. Nisar, "An efficient numerical algorithm based on Haar wavelet for solving a class of linear and nonlinear nonlocal boundary-value problems," *Calcolo*, vol. 53, no. 4, pp. 621–633, 2016.

[12] A. Saadatmandi and M. Dehghan, "The use of Sinc-collocation method for solving multi-point boundary value problems," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 2, pp. 593–601, 2012.

[13] A. L. Osa and O. E. Olaoluwa, "A fifth-fourth continuous block implicit hybrid method for the solution of third order initial value problems in ordinary differential equations," *Applied and Computational Mathematics*, vol. 8, pp. 50–57, 2019.

[14] M. Nwonye, "Spectral-Bernstein residual method for the solution of boundary value problem governing deflection of a beam via Matlab," *Journal of the Nigerian Mathematical Society*, vol. 38, pp. 211–222, 2019.

[15] M. A. Noor and S. T. Mohyud-Din, "Variational iteration technique for solving higher order boundary value problems," *Applied Mathematics and Computation*, vol. 189, pp. 1929–1942, 2007.

[16] L. Xu, "The variational iteration method for fourth order boundary value problems," *Chaos, Solitons & Fractals*, vol. 39, no. 3, pp. 1386–1394, 2009.

[17] M. A. Noor and S. T. Mohyud-Din, "Modified variational iteration method for solving fourth-order boundary value problems," *Journal of Applied Mathematics and Computing*, vol. 29, no. 1-2, pp. 81–94, 2009.

[18] N. A. Shah and I. Khan, "Heat transfer analysis in a second grade fluid over and oscillating vertical plate using fractional Caputo–Fabrizio derivatives," *The European Physical Journal C*, vol. 76, no. 7, p. 362, 2016.

[19] K. K. Ali, C. Cattani Gómez-Aguilar, J. F. Gómez-Aguilar, D. Baleanu, and M. S. Osman, "Analytical and numerical study of the DNA dynamics arising in oscillator-chain of Peyrard-Bishop model," *Chaos, Solitons & Fractals*, vol. 139, Article ID 110089, 2020.

[20] N. Mai-Duy, "Solving high order ordinary differential equations with radial basis function networks," *International Journal for Numerical Methods in Engineering*, vol. 62, no. 6, pp. 824–852, 2005.

[21] W. K. Mashwani, A. Salhi, M. A. Jan, R. A. Khanum, and M. Sulaiman, "Impact analysis of crossovers in a multi-objective evolutionary algorithm," *Science International*, vol. 27, no. 6, pp. 4943–4956, 2015.

[22] M. Sulaiman and A. Salhi, "A seed-based plant propagation algorithm: the feeding station model," *The Scientific World Journal*, vol. 2015, Article ID 904364, 16 pages, 2015.

[23] M. Sulaiman, S. Ahmad, J. Iqbal, A. Khan, and R. Khan, "Optimal operation of the hybrid electricity generation system using multiverse optimization algorithm," *Computational Intelligence and Neuroscience*, vol. 2019, Article ID 6192980, 12 pages, 2019.

[24] A. Ahmad, M. Sulaiman, A. Alhindi, and A. J. Aljohani, "Analysis of temperature profiles in longitudinal fin designs by a novel neuroevolutionary approach," *IEEE Access*, vol. 8, pp. 113285–113308, 2020.

[25] W. Waseem, M. Sulaiman, S. Islam et al., "A study of changes in temperature profile of porous fin model using cuckoo search algorithm," *Alexandria Engineering Journal*, vol. 59, no. 1, pp. 11–24, 2020.

[26] A. Khan, M. Sulaiman, H. Alhakami, and A. Alhindi, "Analysis of oscillatory behavior of heart by using a novel neuroevolutionary approach," *IEEE Access*, vol. 8, pp. 86674–86695, 2020.

[27] N. Ahamad and S. Sharan, "Study of numerical solution of fourth order ordinary differential equations by fifth order runge-kutta method," 2019.

[28] O. A. Taiwo and O. M. Ogunlaran, "A non-polynomial spline method for solving linear fourth-order boundary-value problems," *International Journal of Physical Sciences*, vol. 6, pp. 3246–3254, 2011.

[29] S. Timoshenko and J. M. Gere, *Theory of Elastic Stability*, Dover Publications, Mineola,NY, USA, 1961.
[30] S. P. Timoshenko and J. M. Gere, *Theory of Elastic Stability*, Courier Corporation, Chelmsford, MA, USA, 2009.
[31] W. Chen and T. Atsuta, *In-plane behavior and design, Theory of Beam-Columns*, Vol. 1, J. Ross Publishing, Fort Lauderdale, FL, USA, 1976.
[32] T. Mesbahi, F. Khenfri, N. Rizoug, K. Chaaban, P. Bartholomeüs, and P. Le Moigne, "Dynamical modeling of Li-ion batteries for electric vehicle applications based on hybrid particle swarm-nelder-mead (PSO-NM) optimization algorithm," *Electric Power Systems Research*, vol. 131, pp. 195–204, 2016.
[33] R. Barati, "Parameter estimation of nonlinear Muskingum models using Nelder-Mead simplex algorithm," *Journal of Hydrologic Engineering*, vol. 16, no. 11, pp. 946–954, 2011.
[34] K. M. Hasan and M. A. Zahoor Raja, "Design of reduced search space strategy based on integration of Nelder–Mead method and pattern search algorithm with application to economic load dispatch problem," *Neural Computing and Applications*, vol. 30, no. 12, pp. 3693–3705, 2018.
[35] R. Kshirsagar, S. Jones, J. Lawrence, and J. Tabor, "Optimization of TIG welding parameters using a hybrid Nelder Mead-Evolutionary algorithms method," *Journal of Manufacturing and Materials Processing*, vol. 4, no. 1, p. 10, 2020.
[36] M. Žic and S. Pereverzyev, "Optimizing noisy CNLS problems by using Nelder-Mead algorithm: a new method to compute simplex step efficiency," *Journal of Electroanalytical Chemistry*, vol. 851, Article ID 113439, 2019.
[37] L. Wang, Y. Xu, and L. Li, "Parameter identification of chaotic systems by hybrid Nelder-Mead simplex search and differential evolution algorithm," *Expert Systems with Applications*, vol. 38, no. 4, pp. 3238–3245, 2011.
[38] N. A. Khan, M. Sulaiman, A. J. Aljohani, P. Kumam, and H. Alrabaiah, "Analysis of multi-phase flow through porous media for imbibition phenomena by using the LeNN-WOA-NM algorithm," *IEEE Access*, vol. 8, pp. 196425–196458, 2020.