Neural Comput & Applic (2014) 25:1723–1739
DOI 10.1007/s00521-014-1664-3

ORIGINAL ARTICLE

Solution of the 2-dimensional Bratu problem using neural network, swarm intelligence and sequential quadratic programming

Muhammad Asif Zahoor Raja · Siraj-ul-Islam Ahmad · Raza Samar

Received: 6 March 2013 / Accepted: 5 July 2014 / Published online: 19 July 2014

Abstract In this paper, stochastic techniques have been developed to solve the 2-dimensional Bratu equations with the help of feed-forward artificial neural networks, optimized with particle swarm optimization (PSO) and sequential quadratic programming (SQP) algorithms. A hybrid of the above two algorithms, referred to as the PSO-SQP method is also studied. The original 2-dimensional equations are solved by first transforming them into equivalent one-dimensional boundary value problems (BVPs). These are then modeled using neural networks. The optimization problem for training the weights of the network has been addressed using particle swarm techniques for global search, integrated with an SQP method for rapid local convergence. The methodology is evaluated by applying on three different test cases of BVPs for the Bratu equations. Monte Carlo simulations and extensive analyses are carried out to validate the accuracy, convergence and effectiveness of the schemes. A comparative study of proposed results is made with available exact solution, as well as, reported numerical results.

Keywords 2-Dimensional Bratu equations · Neural networks · Particle swarm optimization · Nonlinear ODEs · Memetic computing · Sequential quadratic programming

M. A. Z. Raja (✉)
Department of Electrical Engineering, COMSATS Institute of Information Technology, Attock Campus, Attock, Pakistan
e-mail: Muhammad.asif@ciit-attock.edu.pk; rasifzahoor@yahoo.com; asif.phdee10@iiu.edu.pk

S. I. Ahmad
Pakistan Institute of Engineering and Applied Science, Nilore, Islamabad, Pakistan
e-mail: sirajisl@yahoo.co.uk

S. I. Ahmad
Department of Physics, COMSATS Institute of Information Technology, Islamabad, Pakistan

R. Samar
Mohammad Ali Jinnah University, Islamabad, Pakistan
e-mail: raza.samar@gmail.com; rsamar@jinnah.edu.pk

1 Introduction

Artificial intelligence techniques based on neural networks optimized with efficient global and local search methodologies have been extensively used to solve variety of the linear and nonlinear systems based on ordinary and partial differential equations [1–5]. For example, stochastic solvers based on neural networks optimized with evolutionary computing and swarm intelligence algorithms have been employed to solve the nonlinear oscillatory systems with both stiff and non-stiff scenarios [6, 7], nonlinear magnetohydrodynamics (MHD) problems [8], fluid dynamics problems based on nonlinear Jaffery–Hamel flow equations [9], the nonlinear Schrodinger equations [10], one-dimensional Bratu's problems arise in fuel ignition model [11, 12], Troesch's problem arising in the study of confinement of a plasma column by radiation pressure [13, 14] and nonlinear singular systems based on Lane Emden flower equations [15]. A good a survey article is referred for interested readers [16] that summarized the importance, history, recent development and future progress of research in this area. Applicability domain of these methods has been extended to solve effectively the differential equations of fractional order as well [17, 18]. For instance, the fractional Riccati differential equation and the Bagley–Torvik fractional system [19, 20] are other significant applications of such solvers. Recently, the evolutionary computing approach based on genetic algorithms (GAs) and interior-point method (IPM) is applied to 2-dimensional (2D) Bratu's problems by transforming into equivalent one-dimensional ordinary differential equation [21]. This study presents the extension of previous works by applying the hybrid or memetic computing technique based on artificial neural networks (ANN), particle swarm optimization (PSO) and sequential quadratic programming (SQP) algorithms to solve the transformed boundary value problem (BVP) of 2D Bratu's type equation.

The PSO algorithms since their introduction by Kennedy and Eberhart [22] have been used extensively in diverse fields as an alternate to genetic algorithms to get better optimal results. PSO techniques belong to a class of global optimization techniques which were developed on the basis of mathematical modeling of bird and fish behavior. Discrete and continuous versions of the algorithms have been developed along with many variants, and these are broadly used as expert systems for constrained and unconstrained optimization problems. A few examples in which PSO methods have performed exceedingly well include mobile communication, sensor networks, inventory control, multiprocessor scheduling, controls, stock market prediction [23–26]. These optimizers are good candidate to explore for finding the optimal design parameter of neural networks model to difficult nonlinear singular BVPs of differential equation.

In the present study, alternate stochastic numerical solution is developed for the transformed form of 2D Bratu's equation, whose generic n-dimensional form is given as:

$$\begin{cases} \Delta u + \mu e^{u} = 0, & \text{in } \Omega \\ u = 0, & \text{on } \Omega \end{cases} \tag{1}$$

Here Δ is the Laplacian operator, u represents the solution to the equation, and μ is a real number. Usually the domain Ω is taken to be the unit interval [0, 1] in $\Re$, or the unit square [0, 1]·[0, 1] in $\Re^2$, or the unit cube [0, 1]·[0, 1]·[0, 1] in $\Re^3$. In case $\Omega = B_1$ is the unit ball in $\Re^n$, $n > 1$, the n-dimensional Bratu equation (1) is transformed into an equivalent one-dimensional nonlinear BVP with singularity at origin as follows [21, 27–29]:

$$\begin{cases} u''(r) + \dfrac{n-1}{r}u'(r) + \mu e^{u(r)} = 0, & 0 \le r \le 1 \\ u(1) = u'(0) = 0, \end{cases} \tag{2}$$

Detailed derivation of this transformation is given in [29]. 2D Bratu's problem are derived from (2) by taking $n = 2$ as:

$$\begin{cases} u''(r) + \dfrac{1}{r}u'(r) + \mu e^{u(r)} = 0, & 0 \le r \le 1 \\ u(1) = u'(0) = 0. \end{cases} \tag{3}$$

Bratu problem (1–3) arises in the simplification of solid fuel ignition models in the field of thermal combustion theory. These problems have a long history, few famous generalization are the "Liouville–Gelfand" or "Liouville–Gelgand–Bratu" problem in recognition of the great French mathematicians Liouville and Gelfand [30–32]. The problem comes up extensively in various physical problems in applied science and engineering, such as chemical reactor theory, nanotechnology, radiative heat transfer, and the Chandrasekhar model for the expansion of the universe [31–35]. Recent articles in which solutions of Bratu-type equations have been given include the non-polynomial spline method of Jalilian [36], the one-point pseudospectral collocation method [37] of Boyd and the Lie-group shooting method (LGSM) by Abbasbandy et al. [38.]

In this study, the strength of ANNs has been exploited once again to develop an approximate mathematical model for the Bratu-type BVP. The accuracy of the model is subject to the tuning of the adjustable parameters, i.e., the weights of the ANNs. A swarm intelligence technique based on the PSO algorithm, the SQP method and a hybrid scheme PSO-SQP are used for the training of the weights of the ANNs. The viability and reliability of this method has been validated by a large number of independent runs of the algorithms along with their detailed statistical analysis. Comparison of the results is made with the reported exact and numerical solution.

2 Neural network modeling

In this section, a detailed description of modeling of BVPs of Bratu-type equations using ANNs is presented. We also formulate an unsupervised fitness function for the network.

2.1 Mathematical model

An approximate mathematical model for the Bratu equations is developed with the help of feed-forward ANNs by exploiting its strength of universal function approximation capability. It is well known that the solution $u(r)$ to the equation and its kth derivative $u^{(k)}$ can be modeled by the following continuous neural network mapping [1, 5, 21]:

$$\begin{cases} \hat{u}(r) = \sum_{i=1}^{m} \alpha_i f(w_i r + \beta_i), \\ u^{(k)}(r) = \sum_{i=1}^{m} \alpha_i f^{(k)}(w_i r + \beta_i), \end{cases} \tag{4}$$

Here α_i, w_i and β_i are the adaptive (real-valued) network parameters, and m is the number of neurons in the network. f is the activation function normally taken as log-sigmoid function for hidden layers and linear function is used for the output layers:

$$\begin{cases} u(t) = \dfrac{1}{1+e^{-t}}, & \text{for the hidden layers} \\ u(t) = t, & \text{for the output layer} \end{cases} \tag{5}$$

Using the log-sigmoid function as given above in Eq. (4), the updated form of the solution $u(r)$ to Eq. (3), and its first (u') and second (u'') derivatives, respectively, are written as:

$$\begin{cases} \hat{u}(r) = \sum_{i=1}^{m} \alpha_i \left(\dfrac{1}{1+e^{-(w_i r+\beta_i)}} \right) \\ \hat{u}'(r) = \sum_{i=1}^{m} \alpha_i w_i \left(\dfrac{e^{-(w_i r+\beta_i)}}{\left(1+e^{-(w_i r+\beta_i)}\right)^2} \right) \\ \hat{u}''(r) = \sum_{i=1}^{m} \alpha_i w_i^2 \left(\dfrac{2e^{-2(w_i r+\beta_i)}}{\left(1+e^{-(w_i r+\beta_i)}\right)^2} - \dfrac{e^{-(w_i r+\beta_i)}}{\left(1+e^{-(w_i r+\beta_i)}\right)^2} \right) \end{cases} \tag{6}$$

The arbitrary combination of the networks given in above set of equations is used to develop the approximate model of the second-order differential equations, in-particularly, Eq. (3).

2.2 Fitness function

The fitness function for the Bratu equations in terms of the unsupervised error function ε is defined as the sum of the mean squared errors:

$$\varepsilon = \varepsilon_1 + \varepsilon_2, \tag{7}$$

and the error ε_1 associated with the differential equations is formulated as:

$$\begin{cases} \varepsilon_1 = \dfrac{1}{K+1} \sum_{k=0}^{K} \left(\hat{u}''_k + \dfrac{1}{r} \hat{u}'_k + \mu e^{\hat{u}_k} \right)^2, & r \in (0,\ 1) \\ k = 1/h, \quad \hat{u}_k = \hat{u}(r_k), \quad r_k = kh \end{cases} \tag{8}$$

Here the interval $r \in (0,\ 1)$ is divided into K steps $r \in (r_0 = 0,\ r_1,\ r_2,\ \ldots,\ r_k = 1)$ with step size h, and $\hat{u}(r)$, $\hat{u}'$ and $\hat{u}''$ are the outputs of the neural networks as given by the set of Eq. (6).

Similarly, the error ε_2 due to the boundary conditions can be written as:

$$\varepsilon_2 = \frac{1}{2}\left((\hat{u}_K)^2 + (\hat{u}'_0)^2 \right) \tag{9}$$

It can be seen that for weights α_i, w_i and β_i for which the functions ε_1 and ε_2 approach zero, the value of the unsupervised error ε also approaches 0. The ANN thus approximates the solution $u(r)$ of Eq. (3) with $\hat{u}(r)$ as given by (6). The architecture of the network is presented graphically in Fig. 1 and will be referred to as a differential equation artificial neural network (DE-ANN).

3 Learning procedures

In this section, we give a brief introduction to the PSO and SQP algorithms, which are used to train the weights of the DE-ANN. The procedure for execution of the algorithms along with their parameter settings is described stepwise.

Swarm intelligence techniques often referred to as particle swarm optimization (PSO) algorithms were originated from the work of Kennedy and Eberhart [22]. In the standard working of PSO method, each candidate solution to an optimization problem is taken as a particle in the search space. The exploration exploitation of the entire search space is made with a population of particles referred to as a "swarm." All particles of the swarm have their own values of fitness according to the problem specific objectives. The particles of the swarm are initialized randomly; the position and velocity of every particle are updated in each flight, according to the previous best local $\mathbf{p}_{\mathrm{LB}}$ $(t-1)$ and global $\mathbf{p}_{\mathrm{GB}}$ $(t-1)$ positions of that particle. The generic updating formulae for particle velocity and position in the standard continuous PSO are written as:

$$\begin{cases} \mathbf{v}(t) = \omega \mathbf{v}(t-1) + c_1 \mathbf{rd}_1 (\mathbf{p}_{\mathrm{LB}}(t-1) - \mathbf{x}(t-1)) \\ \qquad + c_2 \mathbf{rd}_2 (\mathbf{p}_{\mathrm{GB}}(t-1) - \mathbf{x}(t-1)), \\ \mathbf{x}(t) = \mathbf{x}(t-1) + \mathbf{v}(t), \end{cases} \tag{10}$$

where the vector $\mathbf{x}$ represents a particle of the swarm $\mathbf{X}$, consist of sufficient number of particles, $\mathbf{v}$ is the velocity vector for the particle, ω is the inertia weight (linearly decreasing over the course of the search between 0 and 1), c_1 and c_2 are the local and global social acceleration constants, respectively, and $\mathbf{rd_1}$ and $\mathbf{rd}_2$ are random vectors with values between 0 and 1. The elements of velocity are bounded as $v_i \in [-v_{\max},\ v_{\max}]$, where $v_{\max}$ is the maximum velocity. If the velocity exceeds the limits, it will be reset to the respective lower or upper bound. The global learning capabilities of the PSO is exploited for finding the optimal results along with the combination with local search techniques based of efficient sequential quadratic programming (SQP) technique for further fine-tuning of results.

The SQP methods belong to a class of nonlinear programming techniques suited well for constrained optimization problems. Their importance and significance is well established on the basis of efficiency, accuracy and percentage of successful solutions to a large number of test problems. A broad introduction and review of mathematical programming techniques for large-scale numerical optimization is given by Nocedal and Wright [39]. A good source of reference material on SQL algorithms can be found in [40, 41]. The SQP methods have been used by

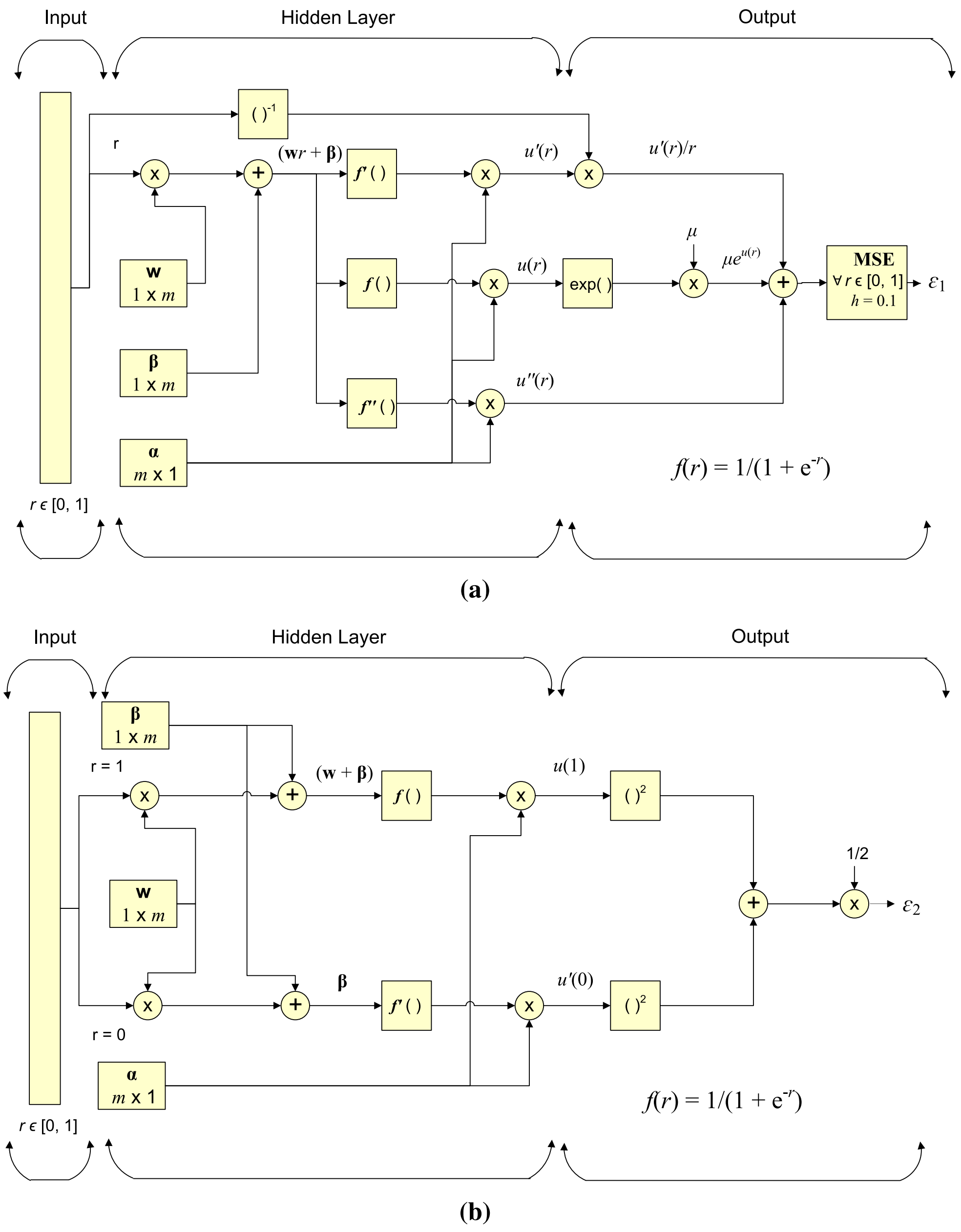

Fig. 1 Neural networks architecture for nonlinear singular system given in (3). **a** For error function ε_1 associated with differential equation, **b** for error function ε_1 associated with differential equation

many researchers in diverse areas of applied science and engineering. A few recently published articles can be seen in [42, 43].

Keeping in view the strengths of the PSO and SQP algorithms, our intention is to use these schemes along with their hybrid approach referred to as PSO-SQP, for training the weights of the DE-ANN. The generic flow diagram of the process of the hybrid PSO-SQP technique is shown graphically in Fig. 2, while the necessary detailed procedural steps are given as follows:

Step 1 *Initialization*: Initialize the swarm consists of particles generated randomly using bounded real values. Each particle has as many elements as the number of unknown adaptive parameters of the DE-ANN. The parameters are initialized as per settings given in Table 1, and then the execution of the algorithm is started.

Step 2 *Fitness evaluation*: Calculate fitness values for each particle using the fitness function given in (7) and subsequently the Eqs. (8) and (9).

Step 3 *Termination criteria:* Terminate the algorithm if any of following criterion is satisfied:

- Fitness value less than a pre-specified tolerance is achieved, e.g., $\varepsilon \leq 10^{-16}$.
- Maximum number of cycles is executed.
  If any of the above criteria is met, then go to step 5.

Step 4 *Renewal*: Update the velocity and position of the particles using the Eq. (10), and go to step 2 for the next execution cycle, i.e., flight.

Step 5 *Refinement*: MATLAB built-in function (*FMINCON* with algorithm SQP) is used for running the SQP method for fine-tuning of the results. Following procedural is adapted for SQP algorithm:

(a) *Initialization*: The best fitted particle from the PSO technique is taken as the start point or initial weight vector for the SQP algorithm: bounds and other declarations for program and tool initialization are given in Table 1.

(b) *Evaluation of fitness*: Calculate the value of fitness function, ε, as given in (7) and subsequently the Eqs. (8) and (9).

(c) *Termination criteria:* Terminate the iterative process of updating the weights of ANNs, if any of the following criteria fulfilled

- Predefined total number of iterations completed.
- Any value for function tolerance (TolFun), maximum function evaluation (MaxFunEvals), X-tolerance (TolX), or constraints tolerance (TolCon) is achieved as defined in Table 2.
  If termination criterion fulfilled, then go to step 6.

(d) *Updating of weights:* Step increment in SQP procedure is made and go to step 5(c).

Step 6 Storage: Store the values of the best global individual along with its fitness and time taken for this runs of the algorithm.

Step 7 Statistical analysis: Repeat steps 1–6 for a sufficiently large number of independent runs of the algorithm in order to have reliable results for subsequent statistical analysis.

4 Simulation and results

In this section, we present the results of simulations of the designed methodology applied to the nonlinear Bratu problem with singularity at origin for the three variants corresponding to different values of μ. The Bratu equation (3) has either zero, one or two solutions depending on whether $\mu > \mu_c$, $\mu = \mu_c$, or $\mu < \mu_c$, respectively, where μ_c represents the critical value of the coefficient μ ($\mu_c = 2$ [27–29]). Results for the three cases $\mu = 0.5$, 1 and 2 are given here and compared with the exact solution. A statistical analysis is also provided in each case.

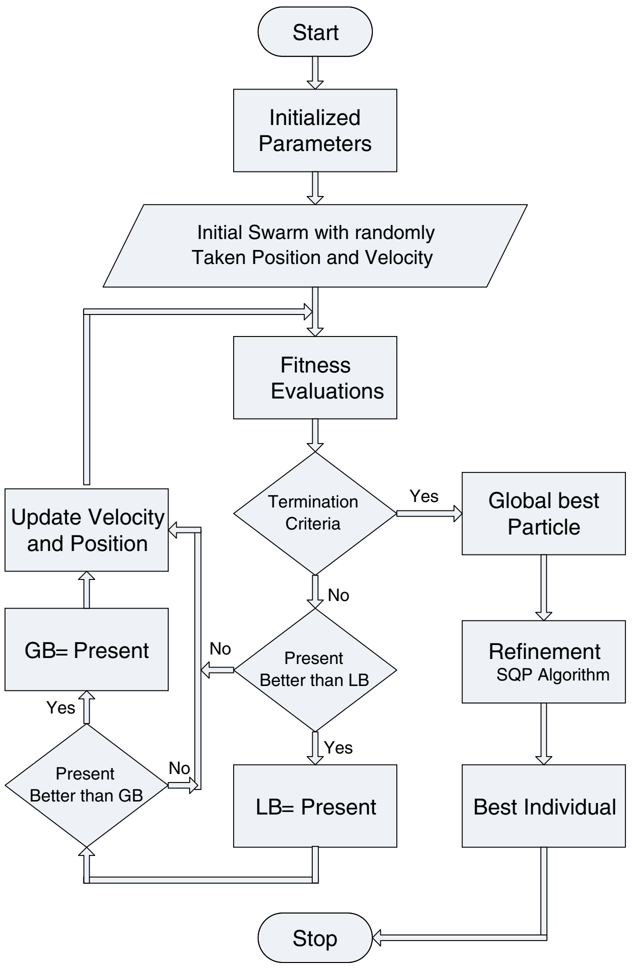

Fig. 2 Generic flow diagram of the hybrid PSO-SQP algorithm

4.1 Bratu equation for $\mu = 0.5$

Consider the BVP of the form (3), given in this case as: [21]

$$\begin{cases} u''(r) + \dfrac{1}{r}u'(r) + 0.5e^{u(r)} = 0, \quad 0 \leq r \leq 1 \\ u(1) = u'(0) = 0 \end{cases} \tag{11}$$

The exact solution of the above problem is given as: [21]

Table 1 Parameter settings for PSO and SQP techniques

PSO		SQP	
Parameters	Setting	Parameters	Setting
Swarm size	160	Start point	Best particle of PSO
Particle size	30	No. of variable	30
Flights	2,000	Iteration	1,000
c_1	Linear decreasing (2.5–0.5)	Maximum function evaluations (MaxFunEvals)	50,000
c_2	Linear increasing (0.5–2. 5)	Function tolerance (TolFun)	10^{-18}
ω	Linearly decreasing (0.9–0.4)	Nonlinear constraints tolerance(TolCon)	10^{-18}
$v_{\max}$	02	Derivative approximate	Finite forward difference
Population span	(−50, 50)	X-Tolerance (TolX)	10^{-12}
Velocity span	(−2, 2)	Bounds	(−50, 50)

$$u(r) = \log\left(\frac{16(7+4\sqrt{3})}{\left(7+4\sqrt{3}+r^2\right)^2}\right) \tag{12}$$

To solve this case of the Bratu equation using the proposed DE-ANN algorithm, we take 10 neurons in each hidden layer of the ANN, which results in 30 weights (α_i, w_i and β_i). The training set is taken from inputs $r \in (0, 1)$ with a step of $h = 0.1$, i.e., a total of 11 grid points in the entire domain. The fitness function ε for this problem is formulated using Eqs. (7), (8) and (9) as:

$$\varepsilon = \frac{1}{11}\sum_{k=0}^{10}\left(\hat{u}''_k + \frac{1}{r}\hat{u}'_k + 0.5e^{\hat{u}_k}\right)^2 + \frac{1}{2}\left((\hat{u}_{10})^2 + (\hat{u}'_0)^2\right) \tag{13}$$

The optimization of weights of the DE-ANN is carried out using the PSO, SQP and PSO-SQP algorithms for one hundred independent runs using the parameter settings as given in Table 1. One set of weights learned by the PSO, the SQP and the hybrid PSO-SQP methods with fitness values of 1.2417×10^{-08}, 1.3753×10^{-13}, and 2.6895×10^{-12}, respectively, are tabulated in Appendix Table 10. The solution obtained using first equation of set (6) and weights given in Table 10 for the 11 input points in $r \in (0, 1)$ with a step size of 0.1 is given in Table 2. The exact solution and reported results [21] of GA, IPA and GA-IPA to the problem are also tabulated in Table 2 for the same inputs. The maximum value of the absolute error (AE) is defined as absolute deviation from the exact solution and its values for PSO, SQP and PSO-SQP techniques are 4.37×10^{-06}, 2.53×10^{-07} and 2.31×10^{-07}, respectively, while for reported GA, IPA and GA-IPA, these are 6.15×10^{-05}, 7.56×10^{-07} and 5.55×10^{-07}, respectively. It is seen that the generally the accuracy of the present methods is superior to the reported results.

The accuracy and convergence of the proposed algorithms is examined by calculating the values of mean absolute error (MAE) and fitness achieved (FA) based on one hundred independent runs. Results are shown graphically in Fig. 3 on a semi-log scale in order to elaborate small differences in values. In Fig. 3a, c, the values of the fitness function ε and MAE are plotted, respectively, for each algorithm against the number of independent runs of the algorithms. The runs are renumbered so that FA and MAE are plotted in ascending order in Fig. 3b, d. The values of fitness function ε as given in (13) for the PSO algorithm lie in the range 10^{-05}–10^{-08}, while for both SQP and PSO-SQP it lies in the range 10^{-08}–10^{-13}. Similarly, the values of MAE are of the order of 10^{-03}–10^{-06} for the PSO technique, while for both SQP and PSO-SQP schemes it lies in the range 10^{-05}–10^{-08}. It is seen that there is no noticeable difference between SQP and PSO-SQP for this variant of BVP (3) and relatively better than that of PSO algorithm.

The statistical analysis of the results for each solver is now performed using the mean and standard deviation (STD) along with the minimum (MIN) and maximum (MAX) values of the absolute error. The results are tabulated in Table 3 along with reported results of GA-IPA for inputs between 0 and 1 with a step size of 0.2. It is seen that the mean value lies in the range 10^{-04}–10^{-05} for the PSO technique, while for both SQP and PSO-SQP algorithms it lies between 10^{-05} and 10^{-08}. It is also observed that there is no noticeable difference in the values of the statistical parameters for the SQP and PSO-SQP present and reported GA-IPA methods; however, the results of the hybrid PSO-SQP scheme are generally more accurate than those of the SQP technique. Also these are much better than the solutions given by the PSO algorithm. Moreover, the reported results of GA given in [21] are generally inferior to the result proposed PSO algorithm.

Table 2 Comparison of the results for $\mu = 0.5$

r	Reference	Present			Reported					
	$u(r)$	$\hat{u}(r)$			$\lvert u(r) - \hat{u}(r)\rvert$			$\lvert u(r) - \hat{u}(r)\rvert$		
	Exact	PSO	SQP	PSO-SQP	PSO	SQP	PSO-SQP	GA	IPA	GA-IPA
0.0	0.13867293	0.1386686	0.1386727	0.13867312	4.37×10^{-06}	2.53×10^{-07}	2.31×10^{-07}	6.15×10^{-05}	7.56×10^{-07}	5.55×10^{-07}
0.1	0.13723751	0.1372366	0.1372373	0.1372377	9.08×10^{-07}	1.69×10^{-07}	1.92×10^{-07}	4.67×10^{-05}	5.44×10^{-07}	3.64×10^{-07}
0.2	0.13293742	0.1329392	0.1329373	0.1329375	1.78×10^{-06}	9.86×10^{-08}	9.90×10^{-08}	2.89×10^{-05}	3.19×10^{-07}	2.52×10^{-07}
0.3	0.12579109	0.1257917	0.1257910	0.1257911	6.58×10^{-07}	7.86×10^{-08}	5.80×10^{-08}	1.98×10^{-05}	2.37×10^{-07}	1.94×10^{-07}
0.4	0.11582892	0.1158268	0.1158289	0.1158290	2.14×10^{-06}	6.73×10^{-08}	7.15×10^{-08}	1.64×10^{-05}	2.14×10^{-07}	1.56×10^{-07}
0.5	0.10309291	0.1030891	0.1030929	0.1030930	3.78×10^{-06}	4.90×10^{-08}	7.75×10^{-08}	1.41×10^{-05}	1.75×10^{-07}	1.14×10^{-07}
0.6	0.08763602	0.0876330	0.0876360	0.0876361	3.03×10^{-06}	3.19×10^{-08}	4.80×10^{-08}	1.06×10^{-05}	1.15×10^{-07}	7.59×10^{-08}
0.7	0.06952147	0.0695208	0.0695215	0.0695215	6.72×10^{-07}	2.31×10^{-08}	1.24×10^{-08}	6.16×10^{-06}	6.68×10^{-08}	5.75×10^{-08}
0.8	0.04882193	0.0488233	0.0488219	0.0488219	1.32×10^{-06}	1.87×10^{-08}	6.42×10^{-09}	1.81×10^{-06}	4.51×10^{-08}	4.24×10^{-08}
0.9	0.02561855	0.0256199	0.0256185	0.0256186	1.31×10^{-06}	1.24×10^{-08}	1.70×10^{-08}	1.53×10^{-06}	3.12×10^{-08}	1.30×10^{-08}
1.0	0.00000000	−2.798E−07	−5.770E−09	7.153E−10	2.80×10^{-07}	5.77×10^{-09}	7.15×10^{-10}	3.87×10^{-06}	5.39×10^{-11}	5.24×10^{-12}

4.2 Bratu equation for $\mu = 1.0$

The nonlinear BVP of the form (3) for $\mu = 1.0$ is given as: [21]

$$\begin{cases} u''(r) + \frac{1}{r}u'(r) + e^{u(r)} = 0, & 0 \le r \le 1 \\ u(1) = u'(0) = 0 \end{cases} \tag{14}$$

The exact solution of the above equation is given as: [21]

$$u(r) = \log\left(\frac{8\left(3+2\sqrt{2}\right)}{\left(3+2\sqrt{2}+r^2\right)^2}\right) \tag{15}$$

The given problem (14) is solved using the same methodology as applied for the case $\mu = 0.5$; however, the fitness function ε formulated in this case is:

$$\varepsilon = \frac{1}{11}\sum_{k=0}^{10}\left(\hat{u}''_k + \frac{1}{r}\hat{u}'_k + e^{\hat{u}_k}\right)^2 + \frac{1}{2}\left((\hat{u}_{10})^2 + (\hat{u}'_0)^2\right) \tag{16}$$

Again optimization of the weights of the DE-ANN is carried out with PSO, SQP, and PSO-SQP algorithms for a hundred independent runs. A set of trained weights is given in Appendix Table 11 for the three techniques with fitness values of 2.8207×10^{-09}, 1.2113×10^{-11} and 3.1141×10^{-11}, respectively. The solution to problem (14) is now determined using these weights for inputs $r \in (0, 1)$ with a step size of 0.1, and results are presented in Table 4 along with reported and exact solution. The maximum values of the AE for proposed PSO, SQP and PSO-SQP methods are 1.34×10^{-05}, 4.91×10^{-08} and 2.43×10^{-07}, respectively, while for reported GA, IPA and GA-IPA [21], these values are 1.94×10^{-04}, 6.38×10^{-07} and 2.03×10^{-07}, respectively. Generally the proposed results are relatively better than that of the reported results.

The values of the MAE and fitness function ε as given in Eq. (16) are computed for the 100 independent runs for the three techniques, and results are plotted in Fig. 4. The value of ε lies in the range 10^{-05}–10^{-08}, 10^{-02}–10^{-13} and 10^{-09}–10^{-13} for the PSO, SQP and PSO-SQP methods, respectively. Similarly, the value of MAE for the three respective algorithms lies in the range 10^{-03}–10^{-05}, 10^{00}–10^{-08} and 10^{-04}–10^{-08}. It is observed that for some independent runs the results of the SQP algorithm are not convergent, while the PSO and PSO-SQP methods remained convergent for all runs. As far as accuracy is concerned, the values of MAE for the convergent cases of the SQP and PSO-SQP algorithms are better than those of the PSO algorithm. Generally, the hybrid approach PSO-SQP provides the most accurate and convergent results for this case of the Bratu equation.

The accuracy of the proposed algorithms is analyzed further on the basis of the statistical parameters calculated

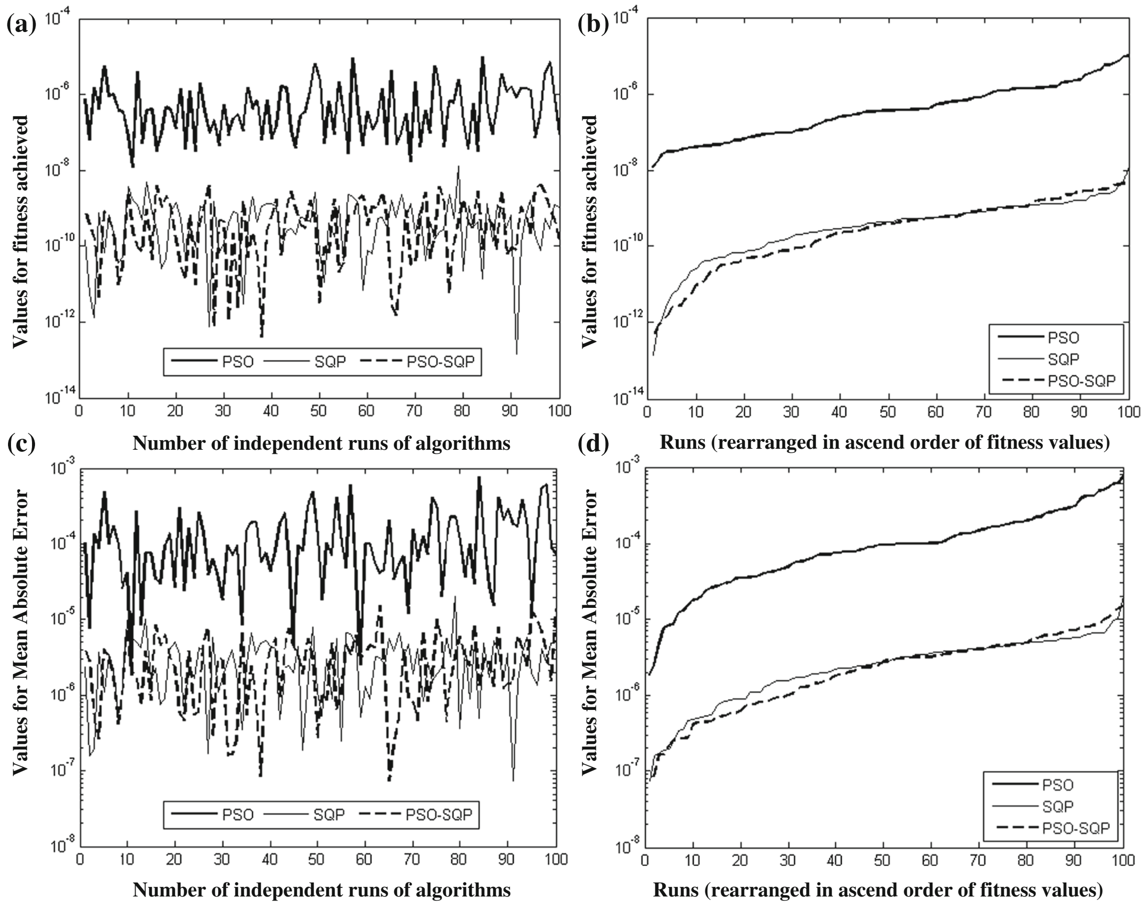

Fig. 3 The results for 100 independent runs of the algorithms for the Bratu equation for $\mu = 0.5$, **a**, **c** unarranged values of FA and MAE, respectively, **b**, **d** rearranged in ascending order of FA and MA values, respectively

for 100 independent runs of each solver. The results are presented in Table 5 for inputs 0–1 with a step size of 0.2. The reported results [21] of statistical analysis for hybrid approach GA-IPA are also tabulated in Table 5 for the same results. It can be seen that the values for MIN, MAX, mean and STD of the AE are lowest for the hybrid approach PSO-SQP. The mean values of the AE calculated for PSO, SQP and PSO-SQP algorithms lie in the range 10^{-04}–10^{-05}, 10^{-02}–10^{-03} and 10^{-05}–10^{-08}, respectively. The results of the SQP method get deteriorated due to convergence problems observed in a few runs, this is because the SQP technique is a local search method and therefore more probable to getting stuck in some local minimum. The PSO-SQP technique stands out as the most accurate and convergent of the three proposed schemes, and generally its results are in good agreement with reported GA-IPA technique.

4.3 Bratu equation for $\mu = 2.0$

The singular BVP of nonlinear Bratu equation (3) is given for this case as: [21]

$$\begin{cases} u''(r) + \frac{1}{r}u'(r) + 2e^{u(r)} = 0, & 0 \le r \le 1 \\ u(1) = u'(0) = 0 \end{cases} \tag{17}$$

The exact solution to the above equation is given as: [21]

$$u(r) = \ln\left(\frac{4}{(1+r^2)^2}\right) \tag{18}$$

Equation (17) is solved following the procedure outlined above for the two cases; however, the fitness function ε in this case is formulated as:

$$\varepsilon = \frac{1}{11}\sum_{k=0}^{10}\left(\hat{u}''_k + \frac{1}{r}\hat{u}'_k + 2e^{\hat{u}_k}\right)^2 + \frac{1}{2}\left((\hat{u}_{10})^2 + (\hat{u}'_0)^2\right) \tag{19}$$

Optimal weights for the DE-ANN are learned using the PSO, SQP and PSO-SQP algorithms for one hundred independent runs. One set of weights is tabulated in Appendix Table 12 for the three algorithms. The values of the unsupervised error (19) are 1.8234×10^{-06},

Table 3 Results of statistical analysis for Bratu equation for the case $\mu = 0.5$

Values	Results	Method	The values of $\lvert u(r) - \hat{u}(r)\rvert$ for inputs "r"					
			0.0	0.2	0.4	0.6	0.8	1.0
MIN	Present	PSO	2.3931×10^{-06}	1.7761×10^{-06}	7.3548×10^{-07}	5.1248×10^{-07}	9.6512×10^{-08}	1.4502×10^{-07}
		SQP	1.3164×10^{-07}	9.8641×10^{-08}	6.7338×10^{-08}	3.8394×10^{-09}	1.1050×10^{-08}	1.9818×10^{-11}
		PSO-SQP	2.3133×10^{-07}	9.9003×10^{-08}	7.1534×10^{-08}	5.4016×10^{-09}	4.9389×10^{-09}	6.7869×10^{-12}
	Reported	GA-IPA	5.9702×10^{-08}	8.0747×10^{-09}	8.9684×10^{-10}	1.7443×10^{-08}	1.7376×10^{-09}	5.2409×10^{-12}
MAX	Present	PSO	1.7317×10^{-03}	1.5237×10^{-03}	9.1294×10^{-04}	3.7419×10^{-04}	2.2017×10^{-04}	2.8661×10^{-04}
		SQP	5.6375×10^{-05}	3.9082×10^{-05}	1.5958×10^{-05}	7.5385×10^{-06}	5.6737×10^{-06}	1.6539×10^{-06}
		PSO-SQP	4.9699×10^{-05}	2.1432×10^{-05}	1.3545×10^{-05}	7.2996×10^{-06}	3.4741×10^{-06}	2.0528×10^{-06}
	Reported	GA-IPA	3.6165×10^{-05}	1.8703×10^{-05}	8.5487×10^{-06}	7.0653×10^{-06}	2.6151×10^{-06}	2.9205×10^{-07}
Mean	Present	PSO	3.5258×10^{-04}	2.5685×10^{-04}	1.3278×10^{-04}	6.4071×10^{-05}	3.3837×10^{-05}	2.0749×10^{-05}
		SQP	1.0033×10^{-05}	5.1390×10^{-06}	2.4053×10^{-06}	1.9168×10^{-06}	7.4070×10^{-07}	6.8769×10^{-08}
		PSO-SQP	1.0513×10^{-05}	5.0289×10^{-06}	2.6649×10^{-06}	1.9060×10^{-06}	7.4570×10^{-07}	5.5929×10^{-08}
	Reported	GA-IPA	1.1355×10^{-05}	5.7481×10^{-06}	2.6971×10^{-06}	2.1849×10^{-06}	8.0696×10^{-07}	9.8603×10^{-09}
STD	Present	PSO	3.4726×10^{-04}	2.9689×10^{-04}	1.6934×10^{-04}	7.1321×10^{-05}	4.0146×10^{-05}	4.1652×10^{-05}
		SQP	8.1517×10^{-06}	4.8419×10^{-06}	2.1052×10^{-06}	1.4176×10^{-06}	7.0135×10^{-07}	2.2911×10^{-07}
		PSO-SQP	9.8650×10^{-06}	4.6520×10^{-06}	2.6019×10^{-06}	1.7165×10^{-06}	7.2149×10^{-07}	2.3535×10^{-07}
	Reported	GA-IPA	8.7696×10^{-06}	4.6012×10^{-06}	2.0817×10^{-06}	1.7112×10^{-06}	6.5277×10^{-07}	3.5349×10^{-08}

1.8791×10^{-10} and 1.5703×10^{-10}, respectively, for the three proposed methods. The solution to equation (17) is approximated using weights in Table 12 for inputs $r \in (0, 1)$ with a step size of 0.1, and results are presented in Table 6 along with the reported results. The maximum values of the AE for the proposed PSO, SQP and PSO-SQP algorithms are 3.16×10^{-02}, 7.90×10^{-05} and 3.18×10^{-04}, respectively, while the reported GA, IPA and GA-IPA methods, these results are 1.19×10^{-01}, 5.89×10^{-04} and 2.82×10^{-04}, respectively.

The values of MAE and FA as given in expression (19) are also computed for 100 independent runs for the three proposed algorithms, and results shown graphically in Fig. 5. It can be seen that the value of ε achieved lies in the range 10^{-04}–10^{-06}, 10^{-00}–10^{-12} and 10^{-08}–10^{-12} for the PSO, SQP and PSO-SQP methods, respectively, while the values of MAE lies in the range 10^{-04}–10^{-06}, 10^{-00}–10^{-12} and 10^{-08}–10^{-12}, for the three respective algorithms. A few runs of the SQP method diverge in this case also, while both the PSO and PSO-SQP algorithms remain consistently convergent, the hybrid scheme being the most accurate.

The accuracy of the PSO, SQP and PSO-SQP solvers is further investigated based on statistical analysis for the 100 independent runs. The results of mean, STD, MIN and MAX values of the AE are given in Table 7 for inputs 0–1 with a step size of 0.2. The reported result for hybrid approach GA-IPA is also given in the table for the same inputs. It is seen that the values of MIN, MAX, mean and STD of the AE are generally lowest for the PSO-SQP hybrid technique. Mean values of the AE for PSO, SQP and PSO-SQP algorithms lie in the range 10^{-01}–10^{-03}, 10^{-01}–10^{-02} and 10^{-03}–10^{-06}, respectively. The MIN values of the AE for SQP and PSO-SQP methods are better than that of the PSO algorithm. Moreover, the values of statistical parameters of the proposed scheme PSO-SQP match closed with the reported results of GA-IPA.

5 Comparative analyses of the solvers

In this section, a comparative analysis is presented for the numerical computations carried out in the last section based on values of the MAE, global mean absolute error (GMAE), mean fitness achieved (MFA), convergence rate and computational time taken for each algorithm.

The reliability and effectiveness of the proposed solvers have been analyzed by computing the values of MAE and FA for 100 independent runs of each algorithm. The accuracy and convergence of the results based on values of MAE $\leq 10^{-03}$, 10^{-04}, 10^{-05} and 10^{-06} are provided in Table 8. Percentage acceptability of the solution $\hat{u}(r)$ for the three cases: $\mu = 0.5$, 1.0 and 2.0 are also given in the table. It is seen that acceptability of the results based on MAE $\leq 10^{-04}$ for the case of $\mu = 1.0$ is >90 % for all the proposed schemes. It is also observed that with increase in the value of the coefficient μ the problem becomes stiffer and the accuracy decreases. Reliable solutions are still provided by SQP and PSO-SQP methods even for the case $\mu = 2.0$, i.e., when μ is close to its critical value; the accuracy of the solution, however, degrades close to the critical value.

Table 4 Comparison of the results for $\mu = 1.0$

r	Reference	Present						Reported		
	$u(r)$	$\hat{u}(r)$			$\lvert u(r) - \hat{u}(r)\rvert$			$\lvert u(r) - \hat{u}(r)\rvert$		
	Exact	PSO	SQP	PSO-SQP	PSO	SQP	PSO-SQP	GA	IPA	GA-IPA
0.0	0.31669437	0.3166810	0.3166943	0.3166945	1.34×10^{-05}	4.34×10^{-08}	1.63×10^{-07}	1.94×10^{-04}	6.38×10^{-07}	2.03×10^{-07}
0.1	0.31326585	0.3132576	0.3132659	0.3132661	8.21×10^{-06}	4.21×10^{-08}	2.43×10^{-07}	1.71×10^{-04}	4.47×10^{-07}	1.56×10^{-07}
0.2	0.30301542	0.3030126	0.3030154	0.3030155	2.83×10^{-06}	9.34×10^{-09}	8.74×10^{-08}	1.24×10^{-04}	2.89×10^{-07}	9.46×10^{-08}
0.3	0.28604727	0.2860454	0.2860472	0.2860472	1.90×10^{-06}	4.91×10^{-08}	2.65×10^{-08}	8.65×10^{-05}	2.37×10^{-07}	6.68×10^{-08}
0.4	0.26253113	0.2625281	0.2625311	0.2625312	3.07×10^{-06}	2.18×10^{-08}	3.51×10^{-08}	6.59×10^{-05}	1.96×10^{-07}	6.08×10^{-08}
0.5	0.23269678	0.2326933	0.2326968	0.2326969	3.45×10^{-06}	3.17×10^{-08}	1.22×10^{-07}	5.19×10^{-05}	1.43×10^{-07}	5.36×10^{-08}
0.6	0.19682681	0.1968245	0.1968268	0.1968269	2.27×10^{-06}	2.46×10^{-08}	9.67×10^{-08}	3.63×10^{-05}	1.01×10^{-07}	3.76×10^{-08}
0.7	0.15524811	0.1552475	0.1552481	0.1552481	5.87×10^{-07}	2.43×10^{-08}	2.73×10^{-10}	1.94×10^{-05}	7.72×10^{-08}	2.14×10^{-08}
0.8	0.10832276	0.1083231	0.1083227	0.1083227	3.74×10^{-07}	2.75×10^{-08}	2.42×10^{-08}	5.42×10^{-06}	5.25×10^{-08}	1.35×10^{-08}
0.9	0.05643860	0.0564392	0.0564386	0.0564386	5.48×10^{-07}	1.74×10^{-08}	3.48×10^{-08}	3.92×10^{-06}	2.03×10^{-08}	9.60×10^{-09}
1.0	0.00000000	7.996E−07	−2.859E−09	2.610E−10	8.00×10^{-07}	2.86×10^{-09}	2.61×10^{-10}	1.15×10^{-05}	7.65×10^{-11}	3.01×10^{-11}

The values of the fitness function ε as given in Eqs. (13), (16) and (19) are also calculated for 100 independent runs using weights optimized by the PSO, SQP and PSO-SQP algorithms. Results for the solvers are given in Table 8 based on values of FA or the unsupervised error $\varepsilon \leq 10^{-03}$, 10^{-05}, 10^{-07} and 10^{-09}. It is seen that the PSO-SQP algorithm gives consistently better and convergent results as compared to the other techniques. The rate of convergence decreases with increase in the value of the coefficient μ, especially when it is near its critical value.

Two additional performance factors GMAE and MFA are introduced here in order to better compare the results of the proposed solvers:

$$\begin{cases} \text{GMAE} = \dfrac{1}{R}\left(\sum_{j=1}^{R}\left(\dfrac{1}{p}\left(\sum_{i=1}^{P}\lvert u_i - \hat{u}_i^j\right)\right)\right) \\ \text{MFA} = \dfrac{1}{R}\left(\sum_{j=1}^{R}\varepsilon^j\right) \end{cases} \quad (20)$$

Here P and R are integers, representing the total number of grid points, and the total number of independent runs of the solver, respectively. u_i denotes the exact solution of the Bratu equation for the ith input, $\hat{u}_i^j$ is the approximate solution for the ith input and the jth independent run, and ε^j represents the fitness achieved for the jth run of the algorithm. In-depth evaluation of the performance of the algorithms is analyzed by these operators because smaller the values of GMAE and MF obtained by the algorithms mean that consistently accurate and convergent results are determined by the algorithm.

In our experimentation, we have taken 11 inputs or grid point, i.e., $P = 11$ and 100 independent runs, i.e., $R = 100$. The values of the GMAE and MFA are computed for the Bratu Problems for $\mu = 0.5$, 1.0 and 2.0 for the three proposed methods. The proposed results are provided in Table 9 along with the values of the respective standard deviations (STD). The reported results of GA, IPA and GA-IPA are also embedded in the same table for both global performance operators. It is evident that results of the PSO-SQP hybrid approach are the best when compared to the PSO and SQP methods. Divergence is observed for a few runs of the SQP technique, and this has a significant impact on values of GMAE and MFA. The solution $\hat{u}(r)$ by the PSO-SQP hybrid approach matches with the exact solution $u(r)$ with an accuracy of up to 9, 8, and 5 places of decimal for the $\mu = 0.5$, 1.0 and 2.0 cases, respectively. Whereas comparing with the reported results, it is observed that proposed PSO results are consistently better than that of GAs for all three variants of Bratu problem, while the results for both hybrid approaches

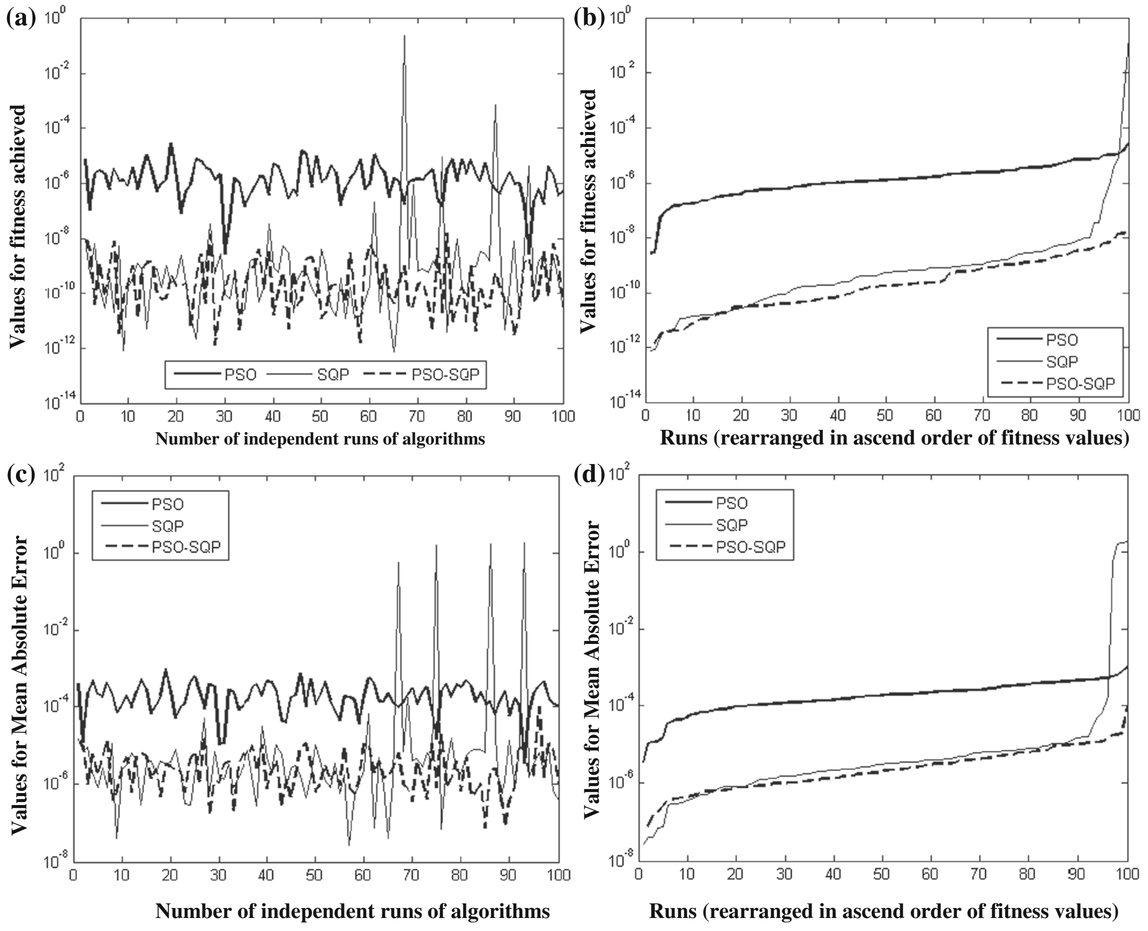

Fig. 4 The results for 100 independent runs of the algorithms for the Bratu equation for the case $\mu = 1.0$, **a**, **c** unarranged values of FA and MAE, respectively, **b**, **d** rearranged in ascending order of FA and MAE values, respectively

Table 5 Results of statistical analysis for Bratu equation for the case $\mu = 1.0$

Values	Results	Method	The values of $\lvert u(r) - \hat{u}(r)\rvert$ at different inputs					
			$r = 0.0$	$r = 0.2$	$r = 0.4$	$r = 0.6$	$r = 0.8$	$r = 1.0$
MIN	Present	PSO	3.2667×10^{-06}	2.8270×10^{-06}	5.0908×10^{-07}	1.8653×10^{-06}	3.7419×10^{-07}	7.8398×10^{-08}
		SQP	4.3381×10^{-08}	9.3382×10^{-09}	2.1763×10^{-08}	5.3433×10^{-09}	7.0641×10^{-09}	3.6494×10^{-11}
		PSO-SQP	1.6282×10^{-07}	8.7389×10^{-08}	9.4888×10^{-09}	1.0336×10^{-09}	1.8894×10^{-08}	7.6198×10^{-11}
	Reported	GA-IPA	5.2717×10^{-08}	1.9111×10^{-08}	2.1108×10^{-09}	1.4094×10^{-08}	1.3454×10^{-08}	7.6877×10^{-12}
MAX	Present	PSO	2.6142×10^{-03}	1.8181×10^{-03}	7.9888×10^{-04}	4.4919×10^{-04}	3.0688×10^{-04}	1.5075×10^{-04}
		SQP	$5.5784 \times 10^{+00}$	$2.7598 \times 10^{+00}$	$1.6288 \times 10^{+00}$	9.3354×10^{-01}	4.5691×10^{-01}	3.4859×10^{-01}
		PSO-SQP	3.2576×10^{-04}	1.5646×10^{-04}	9.5847×10^{-05}	5.5280×10^{-05}	2.5164×10^{-05}	3.0950×10^{-06}
	Reported	GA-IPA	4.5748×10^{-05}	2.4065×10^{-05}	1.2106×10^{-05}	1.0012×10^{-05}	3.6815×10^{-06}	7.9060×10^{-08}
MEAN	Present	PSO	6.0529×10^{-04}	3.6810×10^{-04}	1.9008×10^{-04}	1.2911×10^{-04}	6.8861×10^{-05}	2.6410×10^{-05}
		SQP	1.6563×10^{-01}	8.4952×10^{-02}	5.2456×10^{-02}	3.2043×10^{-02}	1.6403×10^{-02}	3.5052×10^{-03}
		PSO-SQP	1.4748×10^{-05}	7.0362×10^{-06}	4.3207×10^{-06}	2.6839×10^{-06}	1.1176×10^{-06}	8.7145×10^{-08}
	Reported	GA-IPA	8.7919×10^{-06}	4.1689×10^{-06}	2.5828×10^{-06}	1.6417×10^{-06}	6.5676×10^{-07}	5.7928×10^{-09}
STD	Present	PSO	4.5368×10^{-04}	2.9543×10^{-04}	1.4751×10^{-04}	1.0359×10^{-04}	5.9743×10^{-05}	2.9600×10^{-05}
		SQP	9.1050×10^{-01}	4.5214×10^{-01}	2.7094×10^{-01}	1.6063×10^{-01}	8.0985×10^{-02}	3.4858×10^{-02}
		PSO-SQP	3.3870×10^{-05}	1.6399×10^{-05}	9.8797×10^{-06}	5.9101×10^{-06}	2.6100×10^{-06}	3.6048×10^{-07}
	Reported	GA-IPA	1.0230×10^{-05}	5.1223×10^{-06}	2.8765×10^{-06}	2.0861×10^{-06}	8.1067×10^{-07}	1.3193×10^{-08}

Table 6 Comparison of the results for $\mu = 2.0$

r	Reference	Present						Reported		
	$u(r)$	$\hat{u}(r)$			$\lvert u(r) - \hat{u}(r)\rvert$			$\lvert u(r) - \hat{u}(r)\rvert$		
	Exact	PSO	SQP	PSO-SQP	PSO	SQP	PSO-SQP	GA	IPA	GA-IPA
0.0	1.38629436	1.3546503	1.3863734	1.3859765	3.16×10^{-02}	7.90×10^{-05}	3.18×10^{-04}	1.19×10^{-01}	5.89×10^{-04}	2.82×10^{-04}
0.1	1.36639370	1.3355383	1.3664646	1.3660793	3.09×10^{-02}	7.09×10^{-05}	3.14×10^{-04}	1.17×10^{-01}	5.81×10^{-04}	2.78×10^{-04}
0.2	1.30785294	1.2789372	1.3079157	1.3075552	2.89×10^{-02}	6.28×10^{-05}	2.98×10^{-04}	1.10×10^{-01}	5.49×10^{-04}	2.62×10^{-04}
0.3	1.21393897	1.1878260	1.2139943	1.2136692	2.61×10^{-02}	5.53×10^{-05}	2.70×10^{-04}	9.98×10^{-02}	4.97×10^{-04}	2.37×10^{-04}
0.4	1.08945435	1.0668074	1.0895009	1.0892195	2.26×10^{-02}	4.65×10^{-05}	2.35×10^{-04}	8.71×10^{-02}	4.32×10^{-04}	2.06×10^{-04}
0.5	0.94000726	0.9212286	0.9400445	0.9398120	1.88×10^{-02}	3.72×10^{-05}	1.95×10^{-04}	7.28×10^{-02}	3.59×10^{-04}	1.71×10^{-04}
0.6	0.77132496	0.7565681	0.7713531	0.7711714	1.48×10^{-02}	2.81×10^{-05}	1.54×10^{-04}	5.78×10^{-02}	2.82×10^{-04}	1.34×10^{-04}
0.7	0.58874212	0.5779822	0.5887614	0.5886300	1.08×10^{-02}	1.93×10^{-05}	1.12×10^{-04}	4.30×10^{-02}	2.06×10^{-04}	9.80×10^{-05}
0.8	0.39690188	0.3899877	0.3969127	0.3968292	6.91×10^{-03}	1.08×10^{-05}	7.27×10^{-05}	2.86×10^{-02}	1.33×10^{-04}	6.32×10^{-05}
0.9	0.19964067	0.1963143	0.1996438	0.1996049	3.33×10^{-03}	3.16×10^{-06}	3.57×10^{-05}	1.52×10^{-02}	6.48×10^{-05}	3.06×10^{-05}
1.0	0.00000000	−6.387E−05	−3.686E−06	−1.777E−06	6.39×10^{-05}	3.69×10^{-06}	1.78×10^{-06}	2.74×10^{-03}	2.29×10^{-06}	7.74×10^{-07}

matched closed and better than other single algorithms used for optimization.

The computational complexity of the proposed algorithms is examined on the basis of average time taken for their execution, along with the values of STD. We have made the computational time analysis based on 100 independent runs of each solver, and results are given in Table 9 for the mean execution time (MET) and its STD. It is observed that the computational time taken by the PSO-SQP method is a bit longer than that taken by the PSO and SQP techniques, but on the other hand it has an invariable performance advantage in terms of accuracy and reliability. The computational time taken by proposed PSO, SQP and PSO-SQP is relatively less than that of reported results [21] for GA, IPA and GA-IPA, while obtaining the results with almost same level of accuracy and convergence. This analysis is carried out on a Dell Precision 390 Workstation, with Intel(R) Core(TM) 2 CPU 6000@2.40 GHz, 2.00 GB RAM, and running MATLAB version 2011a.

6 Conclusions

The nonlinear 2-dimensional Bratu equation has been solved alternately and effectively using DE-ANNs optimized with PSO, SQP and PSO-SQP algorithms, after the original problem is transformed into an equivalent one-dimensional nonlinear BVP with singularity at origin.

Comparison of the results with exact solutions shows that the PSO-SQP method gives an absolute error in the range 10^{-07}–10^{-10}, 10^{-07}–10^{-10} and 10^{-04}–10^{-06} for the cases $\mu = 0.5$, 1.0 and 2.0, respectively. In general the PSO-SQP hybrid scheme gave not only the most accurate from PSO and SQP but also relatively better than that of reported numerical results of GA, IPA and GA-IPAs.

The reliability and effectiveness of the proposed artificial intelligence techniques were validated by a large number of independent runs of the algorithms and their statistical analysis. The PSO-SQP method provided convergence in all independent runs for all three cases.

Comparative analysis of the three proposed algorithms showed that the PSO-SQP technique invariably provided the best mean fitness values and also the minimum value of the global mean absolute error for all problem cases, but it had a slightly longer computational time. The SQP technique provides results with comparative accuracy, but convergence is a problem. The results of the PSO algorithm converge for all problem cases, and the accuracy attained is lower than of the other two approaches.

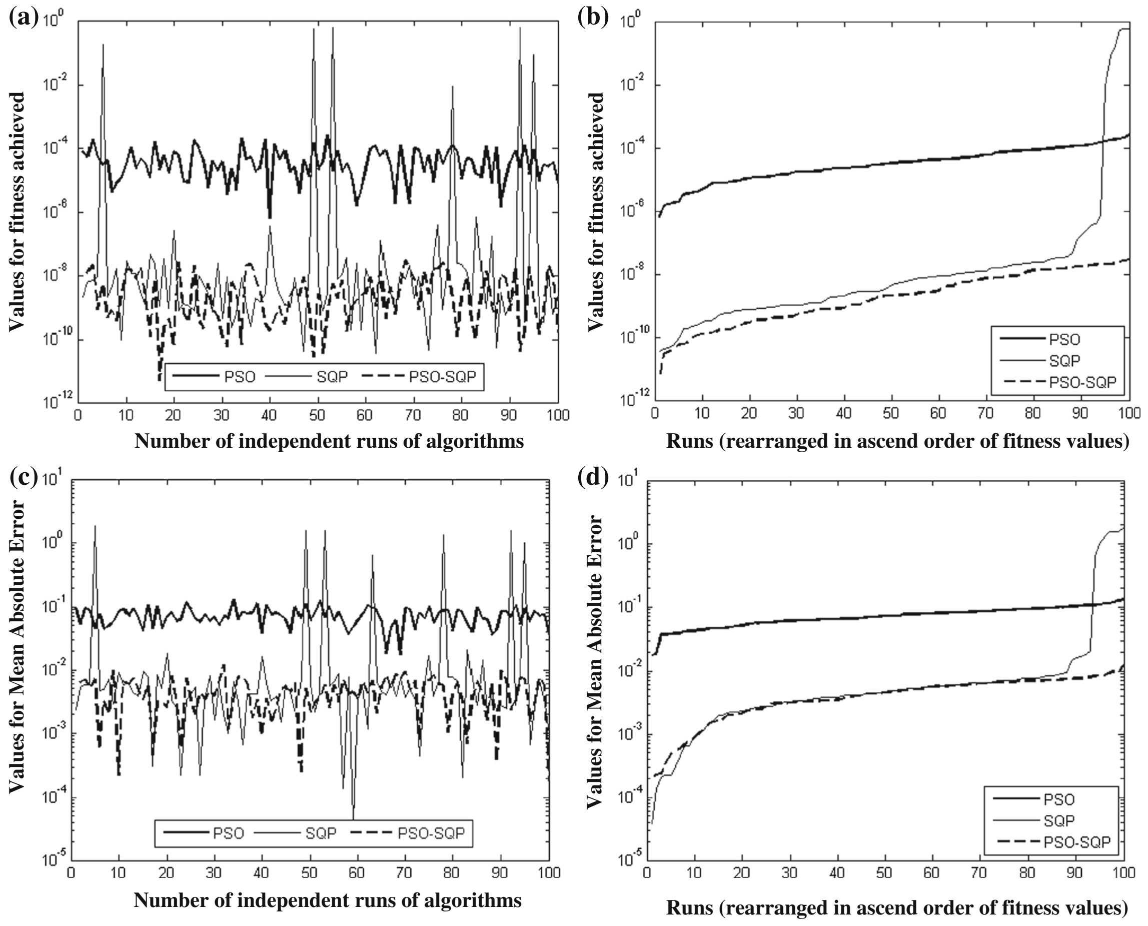

Fig. 5 The results for 100 independent runs of the algorithms for the Bratu equation for the case $\mu = 2.0$, **a**, **c** unarranged values of FA and MAE, respectively, **b**, **d** rearranged in ascending order of FA and MAE values, respectively

Table 7 Results of statistical analysis for the Bratu equation for the case $\mu = 2$

Mode	Result	Method	The values of $\lvert u(r) - \hat{u}(r)\rvert$ at inputs					
			$r = 0.0$	$r = 0.2$	$r = 0.4$	$r = 0.6$	$r = 0.8$	$r = 1.0$
MIN	Present	PSO	3.1644×10^{-02}	2.8916×10^{-02}	2.2647×10^{-02}	1.4757×10^{-02}	6.9142×10^{-03}	2.6287×10^{-05}
		SQP	7.9030×10^{-05}	6.2781×10^{-05}	4.6502×10^{-05}	2.8102×10^{-05}	1.0837×10^{-05}	1.4000×10^{-07}
		PSO-SQP	2.7613×10^{-04}	2.9778×10^{-04}	2.3482×10^{-04}	1.5356×10^{-04}	7.2689×10^{-05}	5.3990×10^{-09}
	Reported	GA-IPA	2.8192×10^{-04}	2.6208×10^{-04}	2.0618×10^{-04}	1.3436×10^{-04}	6.3182×10^{-05}	1.0834×10^{-08}
MAX	Present	PSO	2.2435×10^{-01}	2.0611×10^{-01}	1.6448×10^{-01}	1.1115×10^{-01}	5.6770×10^{-02}	1.0430×10^{-02}
		SQP	$3.9156 \times 10^{+00}$	$2.5305 \times 10^{+00}$	$1.9007 \times 10^{+00}$	$1.4673 \times 10^{+00}$	$1.0929 \times 10^{+00}$	6.9601×10^{-01}
		PSO-SQP	2.1909×10^{-02}	2.0013×10^{-02}	1.5659×10^{-02}	1.0168×10^{-02}	4.7427×10^{-03}	3.8055×10^{-05}
	Reported	GA-IPA	2.3849×10^{-02}	2.1773×10^{-02}	1.7032×10^{-02}	1.1058×10^{-02}	5.1574×10^{-03}	3.0502×10^{-05}
MEAN	Present	PSO	1.2493×10^{-01}	1.1485×10^{-01}	9.0999×10^{-02}	6.0563×10^{-02}	2.9944×10^{-02}	3.0177×10^{-03}
		SQP	1.7400×10^{-01}	1.3766×10^{-01}	1.1165×10^{-01}	8.4003×10^{-02}	5.5999×10^{-02}	2.8862×10^{-02}
		PSO-SQP	8.1968×10^{-03}	7.5476×10^{-03}	5.9205×10^{-03}	3.8491×10^{-03}	1.7994×10^{-03}	7.3180×10^{-06}
	Reported	GA-IPA	5.5033×10^{-03}	5.0718×10^{-03}	3.9787×10^{-03}	2.5866×10^{-03}	1.2086×10^{-03}	3.5608×10^{-06}
STD	Present	PSO	3.9262×10^{-02}	3.6164×10^{-02}	2.9000×10^{-02}	1.9728×10^{-02}	1.0243×10^{-02}	1.9726×10^{-03}
		SQP	6.4638×10^{-01}	4.9099×10^{-01}	4.0057×10^{-01}	3.0835×10^{-01}	2.1561×10^{-01}	1.2679×10^{-01}
		PSO-SQP	4.5553×10^{-03}	4.1766×10^{-03}	3.2727×10^{-03}	2.1263×10^{-03}	9.9327×10^{-04}	7.7895×10^{-06}
	Reported	GA-IPA	3.7403×10^{-03}	3.4287×10^{-03}	2.6862×10^{-03}	1.7451×10^{-03}	8.1490×10^{-04}	4.8212×10^{-06}

Table 8 The accuracy and convergence of the results

μ	DENN with	% Convergence rate on MAE				% Convergence rate on FA			
		$\leq 10^{-03}$	$\leq 10^{-04}$	$\leq 10^{-05}$	$\leq 10^{-06}$	$\leq 10^{-03}$	$\leq 10^{-05}$	$\leq 10^{-07}$	$\leq 10^{-09}$
0.5	PSO	100	098	061	008	100	095	011	000
	SQP	096	096	096	095	096	096	096	085
	PSO-SQP	100	100	100	100	100	100	100	094
1.0	PSO	100	073	009	000	100	053	000	000
	SQP	093	093	092	088	093	093	090	046
	PSO-SQP	100	100	100	100	100	100	100	085
2.0	PSO	000	000	000	000	083	000	000	000
	SQP	076	053	015	001	094	088	058	005
	PSO-SQP	098	087	051	016	100	100	091	021

Table 9 The comparative analysis of the results

μ	Present							Reported		
	Method	MET (s)		GMAE		MFA		Method	GMAE	MFA
		Values	STD	Values	STD	Values	STD		Values	Values
0.5	PSO	189.62	01.41	1.3977E−04	1.4969E−04	1.0823E−06	1.8768E−06	GA	5.2776E−04	8.1764E−06
	SQP	061.64	20.23	3.2246E−06	2.7311E−06	8.1281E−10	1.4769E−09	IPA	3.5447E−06	9.9468E−10
	PSO-SQP	253.66	22.82	3.3020E−06	3.0501E−06	8.0752E−10	1.0707E−09	GA-IPA	3.6193E−06	1.0033E−09
1.0	PSO	192.22	02.28	2.2337E−04	1.6819E−04	2.5448E−06	3.6554E−06	GA	4.9151E−04	6.0596E−06
	SQP	067.88	22.37	5.5788E−02	2.9179E−01	2.3573E−03	2.3505E−02	IPA	1.3545E−02	2.3134E−03
	PSO-SQP	258.05	21.42	4.7218E−06	1.0663E−05	1.1925E−09	2.6939E−09	GA-IPA	2.8288E−06	7.6463E−10
2.0	PSO	200.54	17.96	7.1605E−02	2.2981E−02	5.0675E−05	5.1661E−05	GA	1.1213E−01	1.3666E−04
	SQP	070.89	24.93	9.8289E−02	3.5599E−01	1.9818E−02	9.9705E−02	IPA	2.0813E−02	2.5183E−05
	PSO-SQP	276.60	28.80	4.6127E−03	2.5528E−03	5.7273E−09	7.2995E−09	GA-IPA	3.0988E−03	1.9371E−09

Comparison of the given solution based on global performance operators shows the results due to PSO is relatively better than GA, while the hybrid approach PSO-SQP achieve same level of accuracy and convergence as GA-IPA but with relatively lower in computational complexity.

Other optimization techniques, such as ant/bee colony optimization, genetic programming, differential evolution can also be applied in order to solve the BVPs for Bratu-type equations; this can be a topic of further research on the subject.

Appendix

One set of design parameter of DE-ANN optimized with PSO, SQP and PSO-SQP are tabulated in Tables 10, 11 and 12 in case of Bratu equation given in problems 1, 2 and 3, respectively. The weights given in the appendix are provided in term of real number up to 14 decimal points in order to exactly reproduced the results given in the main body of manuscript and avoid unnecessary rounding of error problems.

Table 10 Parameters obtained for the DE-ANN for the case $\mu = 0.5$

	PSO	SQP	PSO-SQP		PSO	SQP	PSO−SQP
w_1	−0.02170652016785	−0.50123241282682	0.11414941604848	w_6	0.69008979899790	0.54936023320203	−2.67256047331421
w_2	−0.02129823570861	1.50920949013498	0.00945782645750	w_7	0.11808544438502	−0.60900787220513	−0.54887902658642
w_3	0.70821739651183	−1.02997451859911	0.09518123469863	w_8	−0.51104368989452	−0.69293453209569	−0.74297160119701
w_4	2.06298224551873	−0.41498298051969	1.21741634145650	w_9	1.06686016038758	−0.59004577705466	0.29704706318016
w_5	0.37643468612459	0.08971142377240	1.16185776623735	w_{10}	−1.37557156260837	1.42384472965381	0.66548770986634
α_1	−0.48219066534649	−1.46316696881502	−1.25351875380670	α_6	−1.36419624282671	−1.65729608631845	0.01128767507190
α_2	−0.10070099367921	0.08747049503638	−1.30598110093932	α_7	−0.57530121402771	0.59775318590470	0.17732260392336
α_3	−1.59709239624532	−1.02007616426084	0.39277398417113	α_8	7.03890791931011	1.97263291647418	0.32683841845551
α_4	−0.08682078422861	0.95843325472851	0.75381521371952	α_9	1.32949911602930	0.61629613398454	1.20583543727520
α_5	0.65408174223207	−2.40582287111887	0.74609267484446	α_{10}	−0.31972513327534	1.27941091261145	−2.93566266525124
β_1	3.88555127173816	−1.31340981046395	0.80582680506676	β_6	−1.44187942108545	1.20238084384915	−0.87654238027157
β_2	4.13661295446621	0.36856379539027	0.67493201167701	β_7	−1.56340066097894	−0.63011634344328	0.80757553031600
β_3	−1.72490172237778	−0.29025893373694	−0.94376590832748	β_8	−4.05788594315673	2.17432863347987	1.37628081167054
β_4	0.12990565439962	0.82761083271472	1.13258099092714	β_9	1.25878372892867	0.03416864115513	−0.30429013649610
β_5	−0.08354741090233	1.60564015331439	2.45664390993803	β_{10}	−0.64733432609433	2.20368058804433	−1.85941313508288

Table 11 Parameters obtained for the DE-ANN for the case $\mu = 1.0$

	PSO	SQP	PSO-SQP		PSO	SQP	PSO-SQP
w_1	−0.30258379389371	−0.51531910221876	0.47379135995192	w_6	−1.30886830651457	1.30627192688922	−0.27291319324932
w_2	−0.92585025648513	−1.34589687212741	1.61977647469550	w_7	−1.43056002384781	0.08324889323814	−3.46409158178194
w_3	−1.58478247624979	1.37939260796679	1.47027048702151	w_8	1.52735752455202	−0.08106780714547	−0.56019138921933
w_4	1.66865132974912	0.08105226460098	−1.86676608911151	w_9	0.65891813410832	0.02274892625359	−0.72249866567810
w_5	1.21307513720293	−1.04251835073344	−1.31525335720724	w_{10}	−0.26756038105210	−1.20801626119937	0.58256801273946
α_1	−0.15690019784790	−0.02809019741611	0.73024157776081	α_6	−3.57819162023613	−1.36800469008067	0.10471837346499
α_2	0.92992042734608	−0.01132618657092	−1.31298351875736	α_7	0.71285908845065	−1.53695244162223	−0.24929899929521
α_3	1.20364398805907	1.72724335299198	−1.01566238168357	α_8	−0.84208541185277	0.38240361594108	0.23144727579148
α_4	−0.79618304784575	−1.74118173744816	−0.74756201949077	α_9	1.41863384614765	0.75559677122857	0.77802892796974
α_5	−2.01275333083932	−0.96244098135316	−0.41561512104809	α_{10}	−0.92411326793568	2.07916346723579	−0.14308384949312
β_1	2.06527662718265	−2.57015678163694	7.87124498840759	β_6	−0.43513300234208	−1.01727808525711	−3.10736075548446
β_2	0.04681213713145	1.47503766058894	−3.54810695706341	β_7	2.79156081473534	1.41768227482959	−2.95027965686779
β_3	−0.17784383334495	1.82872330197934	−1.47988537283966	β_8	−0.06373544900010	−0.45029451084048	−3.53528264256870
β_4	−1.64250340026114	1.76575715098621	−1.01454988489563	β_9	1.52522957730885	−0.87343834192110	−0.90292984320700
β_5	−3.42883636347546	−0.34309286776144	−1.00052849040577	β_{10}	−0.27304416474780	2.75729796961535	0.86208016966853

Table 12 Parameters obtained for the DE-ANN for the case $\mu = 2.0$

	PSO	SQP	PSO-SQP		PSO	SQP	PSO–SQP
w_1	1.76610776295627	0.05006607988409	−0.86680068209824	w_6	−0.79903153441604	1.73545629658980	−3.12238430118853
w_2	0.80197972145023	−0.12044429590395	1.13899635649731	w_7	0.18517493334125	−0.74310653125208	1.05631186520537
w_3	−3.92297936122884	−0.11505482879237	3.28465452645337	w_8	−1.35851931980316	−2.81201276728751	−0.66909748253739
w_4	−0.80991201805645	−3.63954867460042	3.78698028284898	w_9	2.91614180899783	−0.79460831166918	−1.43124159313922
w_5	0.58550142696244	−0.13901559550640	1.19129981306938	w_{10}	1.24844954843601	−0.43922690555465	−1.71608966905517
α_1	−3.61373046727112	1.09338235143650	4.02579926674003	α_6	0.06053125368859	−7.45728842218260	−3.40022819997215
α_2	−1.37791460894206	1.99035986602046	−1.89741796274795	α_7	0.49908073702095	0.35678527822752	−2.10823761937265
α_3	−1.50031649926726	1.40198750032686	1.11814680441676	α_8	0.83629354399484	−5.48673682782354	0.12500772158309
α_4	0.63892288444442	−4.15139032582301	0.20053843154669	α_9	−0.94767511665988	6.75477206156006	0.22472967172414
α_5	2.24578315915327	1.37974248016551	−1.55153465460332	α_{10}	0.32605937269222	4.21532275831197	2.30253137703912
β_1	−2.22325201348367	−1.75315794234628	0.78550588964204	β_6	2.45663648823465	1.06879875519506	−2.01850870251858
β_2	−1.99513577112041	−0.97180487975142	6.28824188735270	β_7	0.17836148384121	−3.28955504049115	−1.99393947534105
β_3	−1.26300887535434	−1.63507234369991	0.32978754221151	β_8	−0.86765496596572	−0.74565365238905	0.86751116394617
β_4	1.41390536055180	−2.90128842250249	0.79903555662378	β_9	−1.46561390543726	0.44569269404670	2.90581319416184
β_5	−0.17281413832295	−1.03200552208397	1.93670823987675	β_{10}	1.49411023459954	1.57007995888730	0.49820395092181

References

1. Parisi DR, Mariani MC, Laborde MA (2003) Solving differential equations with unsupervised neural networks. Chem Eng Process 42(8–9):715–721
2. Khan JA, Raja MAZ, Qureshi IM (2011) Stochastic computational approach for complex non-linear ordinary differential equations. Chin Phys Lett 28(2):020206–020209
3. Yazdi HS, Pourreza R (2010) Unsupervised adaptive neural-fuzzy inference system for solving differential equations. Appl Soft Comput 10(1):267–275
4. Shirvany Y, Hayati M, Moradian R (2009) Multilayer perceptron neural networks with novel unsupervised training method for numerical solution of the partial differential equations. Appl Soft Comput 9(1):20–29
5. Beidokhti RS, Malek A (2009) Solving initial-boundary value problems for systems of partial differential equations using neural networks and optimization techniques. J Franklin Inst 346(9):898–913
6. Khan JA, Raja MAZ, Qureshi IM (2011) Hybrid evolutionary computational approach: application to van der Pol oscillator. Int J Phys Sci 6(31):7247–7261. doi:10.5897/IJPS11.922
7. Khan JA, Raja MAZ, Qureshi IM Novel (2011) Approach for van der Pol oscillator on the continuous time domain. Chin Phys Lett 28:110205. doi:10.1088/0256-307X/28/11/110205
8. Raja MAZ, Samar R (2014) Numerical treatment for nonlinear MHD Jeffery–Hamel problem using neural networks optimized with interior point algorithm. Neurocomputing 124:178–193. doi:10.1016/j.neucom.2013.07.013
9. Raja MAZ, Samar R (2014) Numerical treatment of nonlinear MHD Jeffery–Hamel problems using stochastic algorithms. Comput Fluids 91:28–46
10. Monterola Christopher, Saloma Caesar (2001) Solving the nonlinear Schrodinger equation with an unsupervised neural network. Opt Express 9(2):16
11. Raja MAZ, Ahmad SI (2014) Numerical treatment for solving one-dimensional Bratu problem using neural networks. Neural Comput Appl 24(3–4):549–561. doi:10.1007/s00521-012-1261-2
12. Raja MAZ (2014) Solution of one-dimension Bratu equation arising in fuel ignition model using ANN optimized with PSO and SQP. Connect Sci. doi:10.1080/09540091.2014.907555
13. Raja MAZ (2014) Stochastic numerical techniques for solving Troesch's problem. Inf Sci. doi:10.1016/j.ins.2014.04.036
14. Raja MAZ (2014) Unsupervised neural networks for solving Troesch's problem. Chin Phys B 23(1):018903
15. Khan JA, Raja MAZ, Qureshi IM (2011) Numerical treatment of nonlinear Emden–Fowler equation using stochastic technique. Ann Math Artif Intell 63(2):185–207
16. Kumar Manoj, Yadav Neha (2011) Multilayer perceptrons and radial basis function neural network methods for the solution of differential equations: a survey. Comput Math Appl 62(10):3796–3811
17. Raja MAZ, Khan JA, Qureshi IM (2011) Swarm intelligent optimized neural networks for solving fractional differential equations. Int J Innov Comput Inf Control 7(11):6301–6318
18. Raja MAZ, Khan JA, Qureshi IM (2010) Evolutionary computational intelligence in solving the fractional differential equations. Lecture notes in computer science, vol 5990, part 1. Springer, ACIIDS Hue City, Vietnam, pp 231–240
19. Raja MAZ, Khan JA, Qureshi IM (2011) Solution of fractional order system of Bagley–Torvik equation using evolutionary computational intelligence. Math Probl Eng 2011:01–18, Article ID. 765075
20. Raja MAZ, Khan JA, Qureshi IM (2010) A new stochastic approach for solution of Riccati differential equation of fractional order. Ann Math Artif Intell 60(3–4):229–250

21. Raja MAZ, Ahmad SI, Samar R (2013) Neural network optimized with evolutionary computing technique for solving the 2-dimensional Bratu problem. Neural Comput Appl 23(7–8):2199–2210. doi:10.1007/s00521-012-1170-4
22. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks, Perth, Australia, IEEE Service Center, vol 4. Piscataway, NJ, pp 1942–1948
23. Sivanandam SN, Visalakshi P (2007) Multiprocessor scheduling using hybrid particle swarm optimization with dynamically varying inertia. Int J Comput Sci Appl 4(3):95–106
24. Li G-D, Masuda S, Yamaguchi D, Nagai M (2009) The optimal GNN-PID control system using particle swarm optimization algorithm. Int J Innov Comput Inf Control 5(10):3457–3470
25. de A Araujo R (2010) Swarm-based translation-invariant morphological prediction method for financial time series forecasting. Inf Sci 180(24):4784–4805
26. Li X, Wang J (2007) A steganographic method based upon JPEG and particle swarm optimization algorithm. Inf Sci 177(15): 3099–3109
27. Syam MI, Hamdan A (2006) An efficient method for solving Bratu equations. Appl Math Comput 176:704–713
28. Syam MI (2007) The modified Broyden-variational method for solving nonlinear elliptic differential equations. Chaos Solitons Fractals 32:392–404
29. Gidas B, Ni W, Nirenberg L (1979) Symmetry and related properties via the maximum principle. Commun Math Phys 68:209–243
30. Bratu G (1914) Sur les équations intégrales non linéaires. Bull Soc Math France 43:113–142
31. Gelfand IM (1963) Some problems in the theory of quasi-linear equations. Trans Am Math Soc Ser 2:295–381
32. Jacobsen J, Schmitt K (2002) The Liouville–Bratu–Gelfand problem for radial operators. J Differ Equ 184:283–298
33. Buckmire R (2003) On exact and numerical solutions of the one-dimensional planar Bratu problem. http://faculty.oxy.edu/ron/research/bratu/bratu.pdf
34. Frank-Kamenetski DA (1955) Diffusion and heat exchange in chemical kinetics. Princeton University Press, Princeton, NJ
35. Wan YQ, Guo Q, Pan N (2004) Thermo-electro-hydrodynamic model for electrospinning process. Int J Nonlinear Sci Numer Simul 5(1):5–8
36. Jalilian R (2010) Non-polynomial spline method for solving Bratu's problem. Comput Phys Commun 181:1868–1872
37. Boyd JP (2011) One-point pseudospectral collocation for the one-dimensional Bratu equation. Appl Math Comput 217:5553–5565
38. Abbasbandy S, Hashemi MS, Liu C-S (2011) The Lie-group shooting method for solving the Bratu equation. Commun Nonlinear Sci Numer Simul 16:4238–4249
39. Nocedal J, Wright SJ (1999) Numerical optimization. Springer Series in Operations Research, Springer, Berlin
40. Fletcher R (1987) Practical methods of optimization. Wiley, New York
41. Schittkowski K (1985) NLQPL: a FORTRAN-subroutine solving constrained nonlinear programming problems. Ann Oper Res 5:485–500
42. Sivasubramani S, Swarup KS (2011) Sequential quadratic programming based differential evolution algorithm for optimal power flow problem. IET Gener Transm Distrib 5(11):1149–1154
43. Aleem SHEA, Zobaa AF, Abdel Aziz MM (2012) Optimal C-type passive filter based on minimization of the voltage harmonic distortion for nonlinear loads. IEEE Trans Ind Electron 59(1):281–289