Ann Math Artif Intell (2011) 63:185–207
DOI 10.1007/s10472-011-9272-8

Numerical treatment of nonlinear Emden–Fowler equation using stochastic technique

Junaid Ali Khan · Muhammad Asif Zahoor Raja · Ijaz Mansoor Qureshi

Published online: 2 December 2011

Abstract The article is based on the approximate solution of a well known Lane–Emden–Fowler (LEF) equation. A trial solution of the model is formulated as an artificial feed-forward neural network containing unknown weights which are optimized in an unsupervised way. The proposed scheme is tested successfully on various test cases of initial value problems of LEF equations. The reliability and effectiveness is validated through comprehensive statistical analysis.

Keywords Emden–Fowler equation · Artificial neural networks · Global search technique · Local search technique

Mathematics Subject Classifications (2010) 34A08 · 34A34 · 34A12 · 65L05 · 68T20 · 65C20

1 Introduction

Many problems in mathematical physics, chemical physics and astrophysics are modeled by second order non-linear ordinary differential equations (ODEs). A second order non-linear ODE of the type Emden–Fowler is used to model the phenomena of the theory of stellar structure, the thermal behavior of a spherical

J. A. Khan (✉) · M. A. Z. Raja
Department of Electronic Engineering, Facility of Engineering and Technology,
International Islamic University, Sector H-10, Islamabad, Pakistan
e-mail: junaid.phdee17@iiu.edu.pk

M. A. Z. Raja
e-mail: asif.phdee10@iiu.edu.pk

I. M. Qureshi
Department of Electrical Engineering, Air University,
Sector E-9, Islamabad, Pakistan
e-mail: imqureshi@mail.au.edu.pk

cloud of gas, isothermal gas sphere and theory of thermionic currents [1–3]. These models are difficult to solve analytically and some time it is impossible, the solution of Emden–Fowler is even more challenging due to the singularity behavior at the origin [4]. In this regard applications are made to the relevant numerical methods. The generic form of Emden–Fowler taken in this paper is as follow

$$\ddot{y} + \frac{r}{x}\dot{y} + a\, f(x)\, g(y) = k(x), \quad r \geq 0, \tag{1}$$

with the following initial conditions

$$y(0) = c \quad ; \quad \dot{y}(0) = b\,, \tag{2}$$

where c is a constant and, $f(x)$ and $g(y)$ are some given functions of x and y respectively. The case, when $f(x) = 1$ and $a = 1$, the expression (1) reduces to Lane–Emden equation, which, with specified $g(y)$, was used to model various phenomenas of engineering [2].

The vast applications and importance of Emden–Fowler equation attracts a number of researchers to find out its numerical solution. In this regard, Adomian decomposition methods were used for approximate analytic solution to Lane–Emden equations by Shawagfeh [5] and Wazwaz [6]. Recently Wazwaz [7] applied Adomian decomposition method (ADM) to solve time dependent Emden–Fowler type of equations. The effect of competing parameters on the existence and non existence of positive solutions of singular Emden–Fowler equation is presented by Guedda [8]. Oscillation criteria for a class of second order Emden–Fowler delay dynamic equations on time scale has been solved by mean of Riccati transformation technique [9]. Variation iteration method (VIM) is used for approximate analytic solutions for time dependent Emden–Fowler type equation by Batiha [10]. Homtopy perturbation method (HPM) is another powerful and convenient analytical technique given by He [11]. A special class of time dependent Emden–Fowler equation was solved analytically in [12, 13] using the ADM, HPM and Homtopy analysis method (HAM) [14], respectively. Recently, the HAM was successfully applied to Cauchy reaction diffusion problems by Bataineh et al. [15]; they also extended the applicability of HAM to the system of ODEs. After a comprehensive survey it has been investigated that a number of researchers analytically and numerically solved various forms of Emden–Fowler equation by various methods and in different applications [2, 3, 16]. A lot of work has been done to solve this notorious problem of highly stochastic nature but no one yet try to approximate the solution with the help of heuristic computation like evolutionary computing, swarm intelligence, genetic programming, artificial immune systems, simulating annealing etc. That is the motivation to cope with different heuristic techniques to solve linear and non linear forms of Emden–Fowler equation.

The mathematical models based on ANN methods have universal capability to solve a variety of problems associated with differential equations (DEs) [17–19]. An unsupervised feed forward ANN is used for finding the general solution of magnetohydrodynamic plasma equilibrium problem represented by ODEs [20]. Radial basis functions neural network were exploited to design a moving mass attitude control system [21] to control a vehicle with three axis stabilization in intra-atmospheric space. Recently some of the developed solutions of the differential equation incorporate the ANN with evolutionary computation techniques [22–24]. It is well known that the computational techniques like genetic algorithms avoid to get

stuck in local minima and it maintain the diversity in searching of candidate solution [25, 26]. Moreover, there is a need to see the applicability of the stochastic solvers on even complex form of non-linear systems having singularity at $x = 0$.

In this paper, an investigation and analysis is carried out for successful modeling of Emden–Fowler type of non-linear ODE using feed-forward ANN assisted with heuristic computation. According to the best of author literature survey, this paper represents the first application of heuristic computational technique for the numerical approximation of Emden Fowler equation. The modeling of the equation is performed by ANN networks by defining an unsupervised error. The unknown weights of these networks are highly stochastic by nature, therefore it has been tuned with genetic algorithm (GA) and simulating annealing (SA) hybridized with interior point algorithm (IPA) for efficient local search. We systematically discuss various models of $f(x)$ and $g(y)$ to verify the efficiency of stochastic numerical solvers. A number of Monte Carlo simulations are performed to determine the effectiveness of the given scheme. Moreover the reliability of the scheme is carried out with superior statistical analysis.

The remainder of this paper is organized as follows. The neural network mathematical modeling along with formulation of the fitness function is revealed in Section 2. The learning procedure for the adaptive parameters of the neural network is introduced in Section 3. A detailed computer simulations and discussion on the results for various numerical problems is presented in Section 4. In the last section we conclude our main findings along with some directions for future research.

2 Designed methodology

The designed methodology consist of two parts, in the first part artificial neural network mapping along with the feed-forward architecture has been revealed. In the second part learning methodology for both local and global optimization has been narrated along with the logical steps.

2.1 Artificial neural network modeling

In this sub-section, an approximate mathematical model has been developed using feed-forward ANN. As a feed-forward NN is a universal function approximator [27–29]. Any network suitably trained to approximate a mapping satisfying some ODE will have an output function that will also approximate the DE [30]. For this following continuous mapping is employed,

$$z(x) = \sum_{i=1}^{m} \alpha_i \varphi\left(w_i x + b_i\right) \tag{3}$$

$$\dot{z}(x) = \sum_{i=1}^{m} \alpha_i \dot{\varphi}\left(w_i x + b_i\right) \tag{4}$$

$$\ddot{z}(x) = \sum_{i=1}^{m} \alpha_i \ddot{\varphi}\left(w_i x + b_i\right) \tag{5}$$

For z, $\dot{z}$ and $\ddot{z}$ respectively, where ϕ being the activation function normally taken as log sigmoid for hidden layers and linear function for output layer.

$$\varphi(t) = \frac{1}{1 + e^{-t}} \tag{6}$$

$$\varphi(t) = t \tag{7}$$

where α_i, w_i, and β_i are real-valued bounded adaptive parameters and m is the number of neurons in the ANN architecture.

2.2 Fitness evaluation function

The unsupervised error function e is formulated by the linear combination of network (3) to (5) for any problem in the form given is (1) as

$$e = \frac{1}{1 + e_j} \quad j = 1, 2, \ldots\ldots \tag{8}$$

where j is the cycle index, and the function e_j is defined as:

$$e_j = e_1 + e_2|_j \tag{9}$$

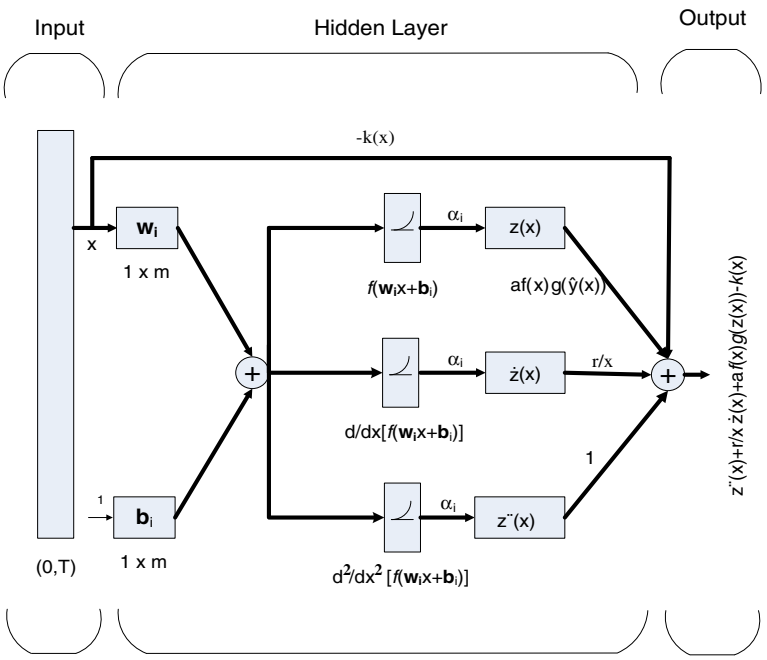

Fig. 1 Artificial neural network architecture of Emden–Fowler equation

where e_1 is error associated with the equation is given as the mean of sum of square error.

$$e_1 = \frac{1}{s}\sum_{i=1}^{s}\left[\ddot{z}\,(t_i) + \frac{2}{t_i}\dot{z}\,(t_i) + az\,(t_i)\right]^2 \tag{10}$$

where $s = mh$, m is the total number of steps and h defines the size of step, t is taken between $(0, T)$. Greater the value of s more will be the accuracy but at the cost of greater computational complexity of the algorithm. Setting the value of s is bit fiddly, because it is a parameter that decides a compromise between accuracy and computational cost.

Similarly e_2 is linked with initial conditions are written as,

$$e_2 = \frac{1}{N}\sum_{k=0}^{N-1}(z(0) - c_k)^2 + \frac{1}{N}\sum_{k=0}^{N-1}(\dot{z}(0) - b_k)^2 \tag{11}$$

where N is the number of initial conditions. It is quite evident that subject to the availability of unknown weights for which the function e_j approaches zero then the value of unsupervised error $e$approaches to 1, hence, $z(t)$ approaches the solution $y(t)$. The linear combinations of networks from (3) to (5) can approximately model the differential equation given in (1). It is named as differential equation neural network (DE-NN). The DE-NN architecture for Emden–Fowler is given in the Fig. 1.

3 Learning procedure

In this section, our intent is to provide the necessary details about learning procedure for unknown weight of the neural network architecture. The learning methodologies are based on algorithms like simulating annealing and genetic algorithm hybridized with interior point algorithm.

3.1 Genetic algorithm

The genetic algorithm (GA) is a heuristic search that mimics the process of natural evolution [31]. This heuristic is routinely used to generate useful solutions to optimization and search problems which are computationally complex to solve [32]. GAs belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques, such as inheritance, mutation, selection, and crossover [33]. Commonly, the algorithm terminates, when, either a maximum number of generations has been produced, or a satisfactory fitness level has been reached. The major advantage of GA is that, it is robust, simple, efficient and does not trapped in poor region of search space like classical numerical methods [34]. The generic flow diagram of the evolutionary algorithm used for optimization is provided in Fig. 2.

3.2 Simulating annealing

The method was independently described by Scott Kirkpatrick, C. Daniel Gelatt and Mario P. Vecchi in 1983 [35], and by Vlado Černý in 1985 [36]. SA is a generic

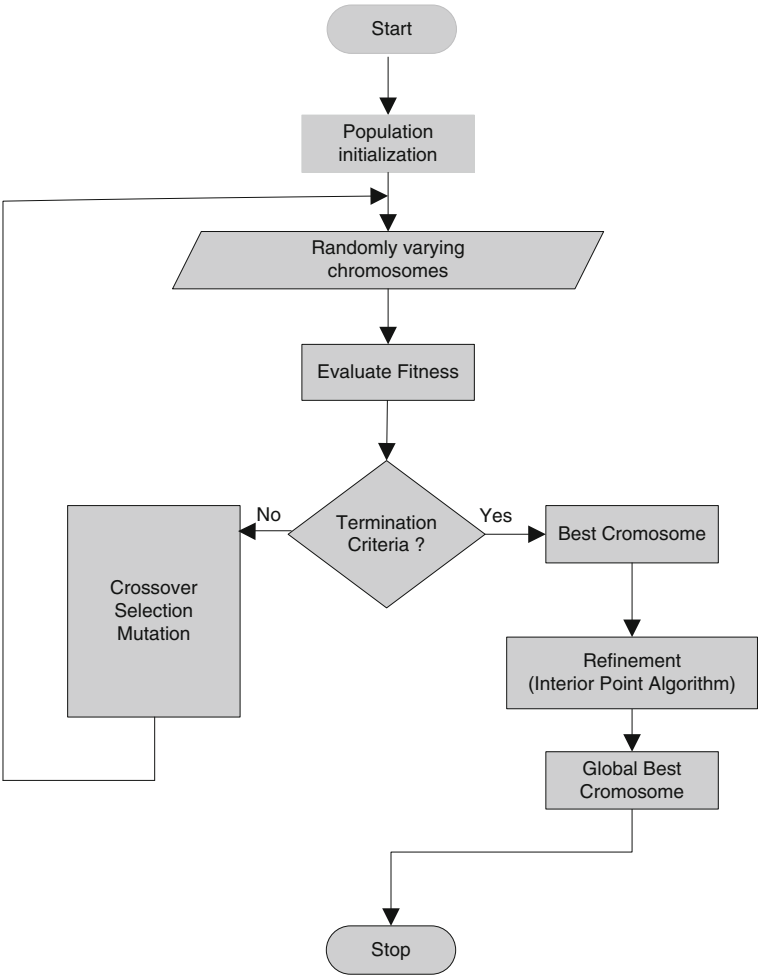

Fig. 2 Generic flow diagram of genetic algorithm

probabilistic metaheuristic for the global optimization problem of applied mathematics, namely locating a good approximation to the global optimum of a given function in a large search space. The name and inspiration of SA come from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The heat causes the atoms to become unstuck from their initial positions (a local minimum of the internal energy) [37]. The converging capabilities of SA are remarkable in pattern analysis and intelligent system designs [38]. In the simulated annealing (SA) method, each point s of the search space is analogous to a state of some physical system, and the function $E(s)$ to be minimized is analogous to the internal energy of the system in that state. The goal is to bring the system, from an arbitrary *initial state*, to a state with the minimum possible energy. At each step, the SA heuristic considers some neighboring state s' of the current state s, and probabilistically decides between moving the system

to state s' or staying in state s. These probabilities ultimately lead the system to move to states of lower energy. Typically this step is repeated until the system reaches a state that is good enough for the application, or until a given computation budget has been exhausted [39].

3.3 Interior point algorithm

Interior point methods (IPA) also referred to as barrier methods are a certain class of algorithms to solve linear and nonlinear convex optimization problems. These algorithms have been inspired by Karmarkar's algorithm, developed by Narendra Karmarkar in 1984 for linear programming [40]. The basic elements of the method consist of a self-concordant barrier function used to encode the convex set [41]. Contrary to the simplex method, it reaches an optimal solution by traversing the interior of the feasible region [42].

The algorithm runs iteratively for the optimization of adaptive parameters, the structure of the given algorithm is described briefly in following steps:

Step (1) Initialization population
An initial population is generated in a bounded range with the help of a random number generator in the following way

$$X_l = [x_{l1}, x_{l2}, \ldots, x_{\ln}]; \quad l = 1 \; to \; m$$

where

$$x_{lk} = (U - L) * r_{lk} + L; \quad k = 1 \; to \; n$$

where x_{ik} is the kth element of a chromosome X_l of length n, and U and L are upper and lower bounds for elements of population. r_{lk} is a random number between 0 and 1.

Step (2) Generation of subpopulations
Divide the population in q subpopulations, each with $\frac{m}{q}$ chromosomes.

Step (3) Ranking
The individuals in each subpopulation are ranked using user defined objective function that calculates the fitness of all individuals. The individuals are arranged such that the member X_1 is considered to be dominating on the member X_2 if X_1 is no worse than X_2 in all objectives and X_1 is strictly better than X_2 in at least one objective.

Step (4) (Selection)
Half chromosomes from each subpopulation are selected as parents by taking $\frac{m}{2q} - 2$ top ranked individuals and 2 from the worst ranked individuals.

Step (5) (Crossover)
Crossover of two parents X_p and X_q generates two new off-springs X_a and X_b with the following elements

$$x_{ak} = \begin{cases} x_{qk} & k > i \\ x_{pk} & k \leq i \end{cases}$$

$$x_{bk} = \begin{cases} x_{qk} & k \leq i \\ x_{pk} & k > i \end{cases}$$

where i is a random integer in the range 1 to $n-1$ and x_{lk} is the kth element of chromosome X_l. In order to search the entire space the crossover is also held for some of the pairs of elite and busted individuals.

Step (6) (Mutation)
Few new mutated chromosomes are included by small random constant value addition in some elements of chromosomes selected at random to increase the diversity in the search space.

Step (7) Formulate new subpopulation by combining initial subpopulation and generated population in steps (4)–(6).

Step (8) (Termination Criteria)
For all $\frac{m}{q}$ sub-population repeat steps (3)–(6) until the termination criteria meets.

Step (9) (Local Search)
The chromosome attain from the above step are given as the start point to interior point algorithm for further refinement.

4 Simulation and results

In this section, we shell consider three test problems based on generalized Emden–Fowler equation. In order to prove the applicability and effectiveness of the proposed scheme a non-linear homogenous Emden Fowler equation has been considered as a Problem 1. The comparison is made with exact, Homtopy analysis method (HAM) as well as by stochastic solvers. Moreover the statistical analysis is also carried out for solvers to see the reliability of the algorithm. In Problem 2, another non-linear homogenous Emden–Fowler equation is solved by the proposed method as well as Homtopy Perturbation method (HPM) and exact solution. In the last example the results are compared with the proposed methodology, various solvers and with exact method. The error differences of the SA and GA are shown in the form of the graphs to have a close look on the strengths of the stochastic solvers.

Case 1 $f(x) = x^m,\ g(y) = e^y$

Problem 1 Let us consider a non-linear, homogeneous Emden–Fowler type equation [6, 7, 11]

$$\ddot{y} + \frac{5}{x}\dot{y} + 8a\left(e^y + 2e^{y/2}\right) = 0, \tag{12}$$

with initial and boundary conditions,

$$y(0) = 0, \quad \dot{y}(0) = 0,$$

The exact solution for the this equation is given as

$$y(t) = -2\ln\left(1 + ax^2\right) \tag{13}$$

The following results of Homtopy analysis method (HAM) upto third term given in [11] is used to take the analytic result of the expression (12)

$$u_1 = 2ahx^2$$
$$u_2 = 2ahx^2 + 2ah^2x^2 + a^2h^2x^4$$
$$u_2 = 2ahx^2 + 4ah^2x^2 + 2ah^3x^2 + 2a^2h^2x^4 + 2a^2h^3x^4 + \frac{2}{3}a^3h^3x^6$$

By considering the value of $h = -1$ and $a = 1$, the above relation is converted into the following limiting solution of third term

$$u(x) = -2ax^2 + a^2x^4 - \frac{2}{3}a^3x^6 \tag{14}$$

The neural network modeling of the equation in (12) is done by DE-NN networks that consist of ten neurons (i.e., $N = 10$), it means 30 adaptive parameters are optimized during the training. The real numbered values of the weights are restricted in the interval $[-15, 15]$.The training is performed for inputs between 0 to 1 with a problem specific fitness function. The fitness evaluation function formulated for (12) is given as

$$e_j = \frac{1}{11}\sum_{i=1}^{11}\left[\ddot{z}\,(t_i) + \frac{5}{t_i}\dot{z}\,(t_i) + 8\left(e^{z(t_i)} + 2e^{z(t_i)/2}\right)\right]^2 + \frac{1}{2}\left[\{z(0)\}^2 + \{\dot{z}(0)\}^2\right]\Bigg|_j, \quad j = 1, 2, 3\ldots \tag{15}$$

where, j is the number of generations and $z(t)$, $\dot{z}(t)$ and $\ddot{z}(t)$ are the networks provided in (3) to (5) respectively. In order to compute the minimum value of the fitness function, an initial population with N number of chromosome is generated. The scheme runs iteratively for specified number of generations or fitness function $e_j \leq 10^{-9}$, which ever comes earlier. The parameter setting for the global optimization by SA and GA is listed in Table 1. The best chromosome achieved by global search is passed as a start point to IPA, which is an efficient and reliable local search technique. The parameter setting for IPA is also given in Table 1. One can find the solution for the above problem for any input time t between 0 and 1 by the best refined weights achieved after the hybridization. For the approximate numerical solution of (12) the proposed technique has been applied by considering the value of $r = 5$, with some value of $a = 1, 2$ and 3 respectively. The comparisons of the results for SA and GA are made with exact solution and limiting solution of third term approximation for HAM method.

The results along with error difference of SA, GA and HAM methods with exact solution are summarized in Table 3. This is obvious from the given table that the accuracy of third term HAM method decreases as the time increases from 0 to 1 while the results obtained by SA and GA are consistent and are in a good agreement with the exact solution. By taking the limiting solution on large number of terms, we can get results at a good accuracy but on the cost of high computations. However, equivalent precision in the results are found with reduced computational cost, when

Table 1 Parameters setting for SA, IPA, and GA

GA		IPA		SA	
Parameters	Setting	Parameters	Setting	Parameters	Setting
Population size	240	Subproblem algorithm	Idl factorization	Temperature update function	Exponential temperature update
Chromosome size	30	Chromosome size	30	Chromosome size	30
No. of runs	2000	Hessian	BFGS	Mesh size	(1, 10)
Selection	Stochastic uniform	Minimum perturbation	1e-8	Annealing function	Fast annealing
Reproduction	Elite count of 2 crossover fraction 0.7	Maximum perturbation	0.1	Re-annealing interval	200
Mutation	Adaptive feasible	Maximum evaluation function	3000	Maximum evaluation function	90000
Hybridization	IPA	Maximum iteration	1000	Hybridization	IPA

compared with evolutionary computational intelligence based on neural networks and Genetic Algorithm hybrid with Interior point Algorithm. Another advantage of the given scheme is that it involves mathematically less expensive operations like crossover, selection and mutation.

One of the best set chromosome of DE-NN trained heuristically by SA and GA are provided in Table 2. The weights in Table 2 is used to optimize the Problem 1 and the results are listed in Table 3 for $a = 1$.

In order to have a reliability in the proposed method a comprehensive simulation is performed by considering $a = 2$ and $a = 3$ respectively on the same inputs from 0 to 1. The real valued chromosomes for adaptive parameters α_i, w_i and β_i of $a = 1, 2, 3$ are represented in Fig. 3 in the form of multiple bars.

The results achieved by the chromosomes represented in Fig. 3 are thoroughly given in Fig. 4. Figure 4a provides the result for $t \in [0,1]$ with a step size of 0.1

Table 2 DE-NN weights trained by different solvers for Problem 1

I	wi		α_i		β_i	
	SA	GA	SA	GA	SA	GA
1	1.4194776	1.3016022	−1.5326960	−1.1591736	−1.2697695	−1.2774433
2	−0.3377788	−0.6914482	0.0867564	0.758588	0.1556725	−1.5518364
3	1.3075257	1.3758173	0.5201725	−2.5515528	0.5014446	−2.2615302
4	1.1603475	−3.8639086	−1.4711290	−1.1223040	−1.7152615	−2.2908400
5	−1.2964929	2.5369194	−0.6763090	−0.8775756	0.2254261	−0.6663938
6	3.0114602	−1.3861552	2.7874442	0.92781952	1.2373492	1.9764443
7	−2.5958861	−0.5048545	1.0323294	−0.2828672	1.0723367	1.0073925
8	1.1789274	−2.9993198	−3.1465080	−1.9515727	−1.3092351	−0.7214404
9	0.2404541	−0.1051222	−1.5441880	−0.1131359	0.2116409	−0.0237624
10	0.0920477	−2.1426376	−1.0836810	1.09664274	1.2447624	1.2280225

Table 3 Comparison of results for the solution of problem in Case 1

t	y(t)	$\hat{y}(t)$			$\lvert y(t) - \hat{y}(t) \rvert$		
		SA	GA	HAM	SA	GA	HAM
0.0	0.0000000	−1.55E-06	−2.24E-07	0.0000000	1.55E-06	2.24E-07	0.000000
0.1	−0.0199007	−0.0198783	−0.0198923	−0.0199007	2.24E-05	8.34E-06	4.96E-09
0.2	−0.0784414	−0.0784194	−0.0784337	−0.0784427	2.20E-05	7.71E-06	1.24E-06
0.3	−0.1723554	−0.1723453	−0.1723516	−0.1723860	1.01E-05	3.77E-06	3.06E-05
0.4	−0.2968400	−0.2968288	−0.2968353	−0.2971307	1.13E-05	4.73E-06	0.000291
0.5	−0.4462871	−0.4462706	−0.4462805	−0.4479167	1.65E-05	6.55E-06	0.001630
0.6	−0.6149694	−0.6149552	−0.6149643	−0.6215040	1.42E-05	5.07E-06	0.006535
0.7	−0.7975522	−0.7975441	−0.7975497	−0.8183327	8.18E-06	2.50E-06	0.020780
0.8	−0.9893925	−0.9893854	−0.9893899	−1.0451627	7.07E-06	2.59E-06	0.055770
0.9	−1.1866537	−1.1866447	−1.1866498	−1.3181940	9.02E-06	3.91E-06	0.131540
1.0	−1.3862944	−1.3862883	−1.3862922	−1.6666667	6.09E-06	2.18E-06	0.280372

for different values of the variable coefficient by GA hybrid with IPA. The above chromosomes are also used to find the behavior of their corresponding derivative of the solution as shown in the Fig. 4d. It can be seen from Fig. 4a and d that the results obtained by GA hybrid with IPA are overlapped to the exact solution as well as at its derivative. As there is no noticeable difference between the exact and proposed technique so the results of the same chromosomes are also achieved for $t \in [0,4]$. In Fig. 4b and e, it is evident that the error is starting to accumulate for input greater than 2 but still under a acceptable level of accuracy. Figure 4c and f presents the results of the results for the input time $t \in [0,1]$ with the same chromosomes as achieved in Fig. 3. This is worth mentioning that for Fig. 4c and f some of the error difference is there, but still the trend is the same. It has been seen in the solution of non-linear systems that the results start to diverge for the large input times but in the presented method, this is not the case. This is the point of attention that the results of the derivatives of the problem with the given method also validate the capability of stochastic solvers.

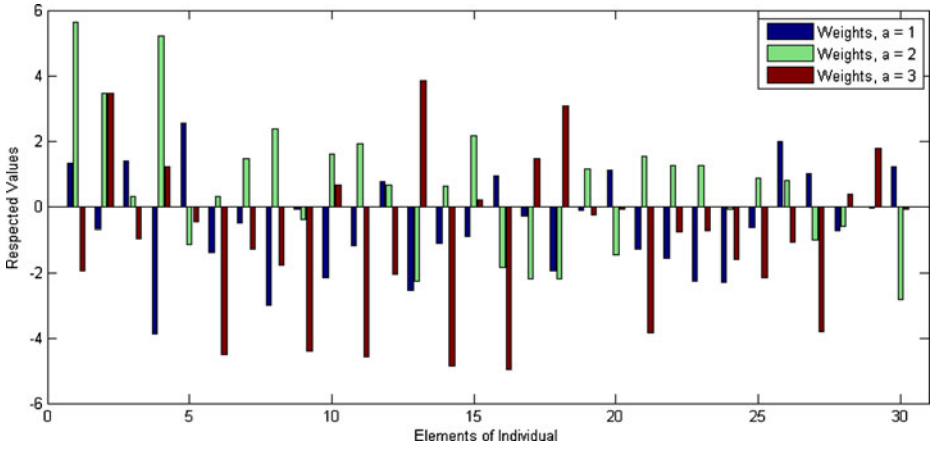

Fig. 3 Real valued chromosomes of adaptive weights for Problem 1 at $a = 1$, $a = 2$ an $a = 3$ respectively for interval between 0 to 1

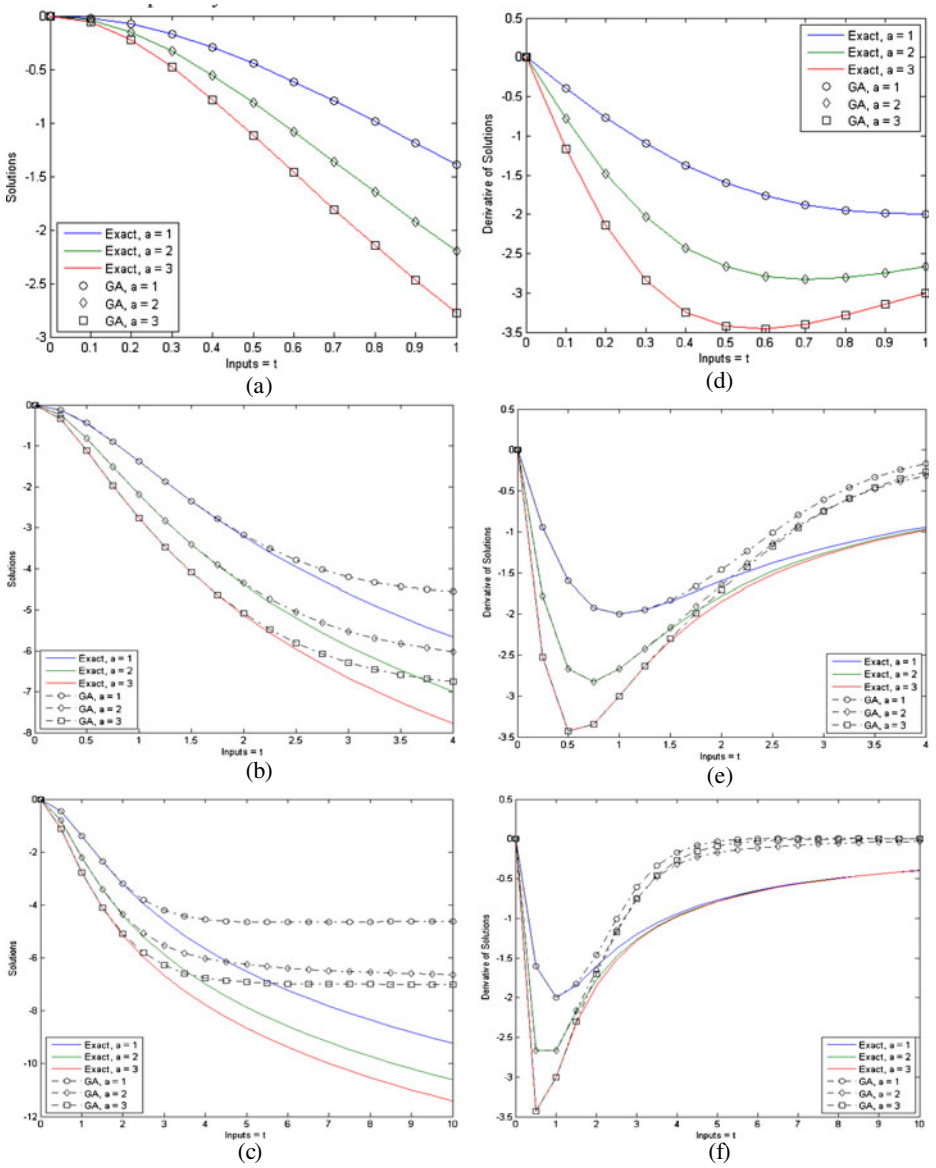

Fig. 4 The results of the Problem 1 and their derivatives for $a = 1,2,3$ with GA and exact solution

The results drawn in Fig. 4 are at the log scale as the error difference between exact and approximated methods is from 10^{-5} to 10^{-6}.

It can be concluded form Fig. 4 that the results are convergent for smaller training inputs and it is still consistent for the larger inputs. However the accuracy for a longer time intervals can be calculated by training for larger intervals. We have optimized the problem for larger interval from 0 to 4. Now this time the results remain convergent even for this large interval but at a cost of large computations.

The chromosomes obtained by training from 0 to 4 are also used to get results from 0 to 10 for both of the solution and its derivative. Figure 5a and b show the experimental results of the Problem 1 for training from 0 to 4, while Fig. 5c and d represents the results of its corresponding derivatives. It has been observed that the error started to grow after $t = 7$, however this error can be reduced by training for even larger intervals. It can be concluded that there is always a trade off between the computational cost and level of accuracy.

The chromosome attained in the training of Problem 1 from 0 to 4 are provided in the Fig. 6. These weights are in a bounded range, by using these tuned adaptive variable one can get the reported result. It is remarkable to mention that only 30 units of neurons are needed to solve such a complex non-linear problem, which is a small number as compared to the number of basic function required in the conventional calculations iteratively. To get a reliable statistical analysis for the efficiency of the given method a large number of simulations are performed. In this regard 125 independent runs are carried out for finding the weights of DE-NN optimized with SA and GA algorithms.

The matlab 7.7 version optimization tool box is used for SA, GA and IPA with setting given in the Table 1. The chromosomes obtained by SA and GA are given as a start point to IPA. All 125 independent run results are ranked in descending

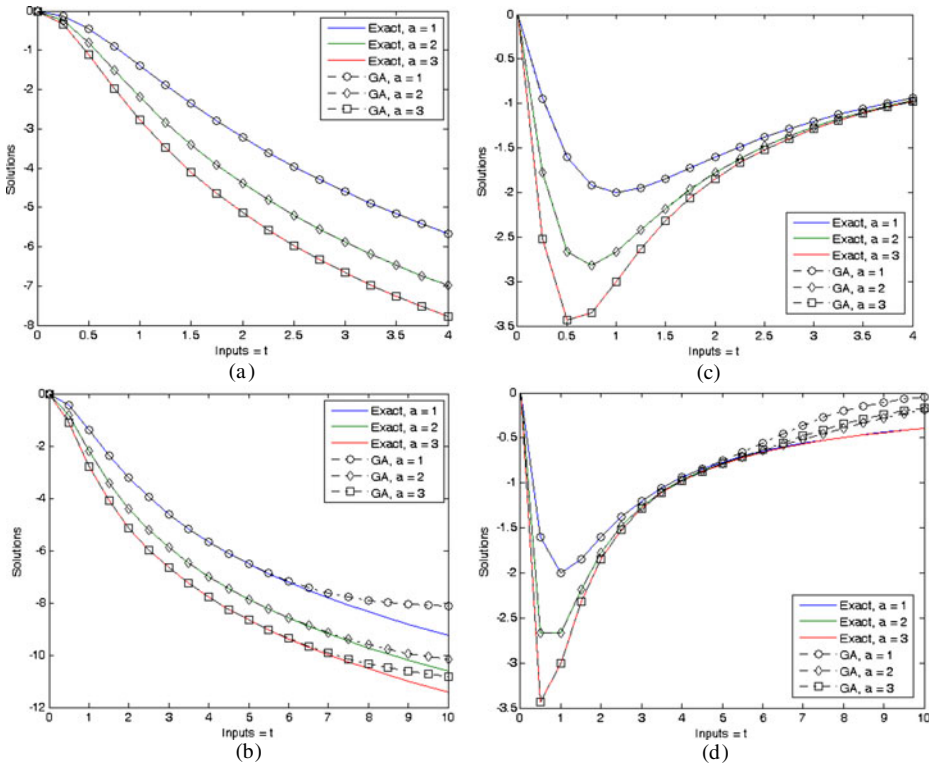

Fig. 5 The results of the problem and their derivatives for larger time spans with GA and exact solution

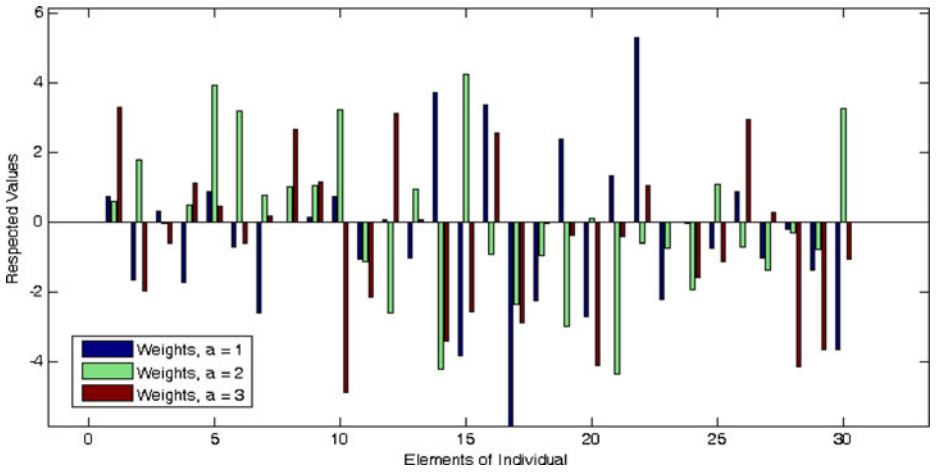

Fig. 6 Real valued chromosomes of adaptive weights for Problem 1 at $a = 1$,$a = 2$ an $a = 3$ respectively for interval between 0 to 4

order; the results of 25 runs are discarded for a better refinement. The discarded independent runs are ill with respect to the desired level of accuracy. From the remaining 100 runs the analysis is made on the criteria of best, worst, mean and standard deviation (STD). The best is defined as the minimum value of the function e_j while the worst is the maximum value of the problem specific fitness function.

The statistical results of the solution and its derivatives are given in the Table 4 for SA-IPA while Table 5 presents the statistical analysis for GA-IPA. The absolute of the error difference is taken with exact and approximated solutions from stochastic solvers.

The best values for the solution and its derivative lie in the range of 10^{-6} to 10^{-7} for SA and 10^{-6} to 10^{-8}.

The statistical analysis is made for 125 independent runs, best 100 value of the function e_j for GA, SA-IPA and GA-IPA are drawn in the decreasing order in Fig. 7. The results are drawn on log-scale as the stochastic solvers provide the results close to

Table 4 Statistical analysis of SA-IPA for the problem in Case 1

t	SA-IPA $\lvert y(t) - \hat{y}(t)\rvert$				SA-IPA $\lvert y'(t) - \hat{y}'(t)\rvert$			
	Best	Worst	Mean	STD	Best	Worst	Mean	STD
0	1.59E-06	0.3981116	0.0077061	0.0475609	4.93E-06	0.1375817	0.003319	0.018618
0.1	2.42E-06	0.4078750	0.0081924	0.0497579	2.48E-05	0.2161927	0.007198	0.028326
0.2	2.94E-05	0.4041971	0.0084656	0.0502946	4.10E-07	0.1519571	0.004509	0.019977
0.3	7.98E-06	0.3838130	0.0081235	0.0479361	5.62E-07	0.2809099	0.007577	0.038486
0.4	5.16E-06	0.3496018	0.0075024	0.0437811	5.26E-06	0.3936050	0.007966	0.046685
0.5	2.28E-07	0.3071916	0.0067488	0.0390909	4.11E-06	0.4438464	0.008318	0.048712
0.6	1.23E-06	0.2628983	0.0059721	0.0344089	1.95E-07	0.4325150	0.008369	0.047540
0.7	7.40E-06	0.2223111	0.0052307	0.0299886	3.79E-06	0.3728862	0.007862	0.043151
0.8	1.15E-06	0.1891149	0.0045437	0.0260048	4.18E-06	0.2898748	0.006915	0.037425
0.9	3.70E-06	0.1638794	0.0039304	0.0225083	9.74E-06	0.2224257	0.006256	0.032872
1.0	8.45E-07	0.1429506	0.0033790	0.0193419	4.53E-05	0.2277216	0.006040	0.031511

Table 5 Statistical analysis of GA-IPA for the problem in Case 1

t	GA-IPA $\lvert y(t) - \hat{y}(t)\rvert$				GA-IPA $\lvert y'(t) - \hat{y}'(t)\rvert$			
	Best	Worst	Mean	STD	Best	Worst	Mean	STD
0	8.32E-08	0.0001571	1.54E-05	2.75E-05	1.72E-07	0.0005165	4.27E-05	6.42E-05
0.1	6.76E-08	0.0001249	3.03E-05	2.55E-05	3.08E-06	0.0014745	0.000227	0.000216
0.2	1.27E-06	0.0001865	3.24E-05	3.12E-05	3.43E-06	0.0002540	0.000112	5.93E-05
0.3	2.70E-07	0.0001212	2.25E-05	2.50E-05	1.90E-07	0.0010292	0.000104	0.000149
0.4	1.36E-06	0.0001181	1.91E-05	2.10E-05	1.06E-07	0.0006177	5.97E-05	8.13E-05
0.5	3.76E-08	0.0001014	1.99E-05	1.84E-05	2.21E-07	0.0002282	5.47E-05	5.27E-05
0.6	1.00E-06	8.490E-05	1.95E-05	1.60E-05	1.78E-06	0.0006360	6.66E-05	8.41E-05
0.7	8.76E-07	0.0001115	1.69E-05	1.85E-05	1.44E-07	0.0004913	5.24E-05	6.24E-05
0.8	5.16E-08	0.0001376	1.52E-05	1.97E-05	3.24E-09	0.0001630	3.62E-05	3.62E-05
0.9	1.04E-07	0.0001167	1.42E-05	1.64E-05	7.56E-08	0.0004424	2.86E-05	6.04E-05
1.0	1.02E-08	8.100E-05	1.11E-05	1.24E-05	2.25E-06	0.0002694	5.17E-05	4.08E-05

each other. It has been noticed that GA-IPA characteristics curve is always superior to SA-IPA and GA in term of minimum of e_j.

Case 2 $f(x) = x^m,\ g(y) = y^n \ln(y)$

Problem 2 Another type of the nonlinear, homogeneous Emden–Fowler equation is taken to investigate the strengths of the proposed stochastic algorithm. Consider the equation of Emden–Fowler [6, 7]

$$\ddot{y} + \frac{8}{x}\dot{y} + 18ay + 4ay \ln y = 0 \tag{16}$$

subject to the following conditions

$$y(0) = 1,\ \dot{y}(0) = 0, \tag{17}$$

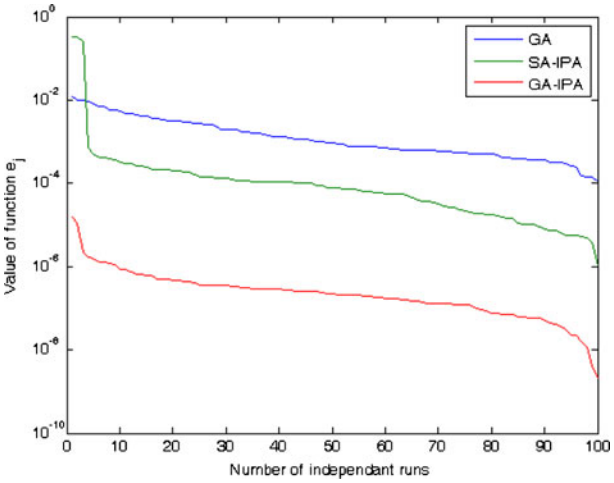

Fig. 7 Comparison of DE-NN networks optimized with stochastic solvers for 100 independent runs

The exact solution obtained Wazwaz [7] by Adomian decomposition method is as follow,

$$y(x) = e^{-ax^2} \tag{18}$$

The Problem 2 is also solved numerical by using Homtopy Perturbation method (HPM) method upto a limiting solution of fourth order and following relationships are obtained from [43] to take the analytic result of (16).

$$u_0 = 1$$
$$u_1 = -ax^2$$
$$u_2 = \frac{1}{2}a^2x^4$$
$$u_3 = -\frac{1}{3}a^3x^6$$

By considering the value of $a = 1$ the above relation is converted into the following relation

$$u(x) = 1 - x^2 + \frac{1}{2}x^4 - \frac{1}{3}x^6 \tag{19}$$

This problem is solved on the same manner as the previous one by considering the value of $r = 8$ and constant coefficient $a = 1$. The problem specific fitness function for (16) is given below as

$$e_j = \frac{1}{11}\sum_{i=1}^{11}\left[\ddot{z}\,(t_i) + \frac{8}{t_i}\dot{z}\,(t_i) + 18z\,(t_i) + 4\,z\,(t_i)\ln\left(z\,(t_i)\right)\right]^2 + \frac{1}{2}\left[\{z(0) - 1\}^2 + \{\dot{z}(0)\}^2\right]\Big|_j \tag{20}$$

where, $j = 1, 2, 3....$ is the number of generation used for the given algorithm. This time the numerical method used for the limiting solution is Homtopy perturbation method (HPM) of order four. The HAM method, exact solution and the proposed solution is compared to see the working of the proposed method. The results are summarized in Table 6 along with error difference of SA, GA and HPM methods with exact solution.

The one of the best chromosomes tuned by SA and GA hybridized by IPA for constructed DE-NN architecture are narrated in Table 7. The real values of the adaptive parameters are between 10 to −10.

The error of the results achieved by SA and GA in comparison to exact solution is given on the training set t between 0 and 1 and results are shown in the Table 6.

The training of the weights of Problem 2 is also performed iteratively for 125 independent runs for a reliable statistical analysis. The results for best 100 run are given for GA, SA-IPA and GA-IPA respectively in Fig. 8. This is quite evident from the Fig. 8 that the results achieved by GA-IPA are not only consistent but also the value of fitness function e_j is minimum in other words the fitness has been maximized.

The statistical analysis for problem in Case 2 is also made on the same pattern as in Problem 1. The results of SA-IPA is given in Table 8 which clearly mention that best of the fitness value is in an acceptable range of 10^{-6} to 10^{-8} for SA-IPA

Table 6 Comparison of results for the solution of Problem 2

t	SA			GA		HPM	
	y(t)	$\hat{y}(t)$	$\lvert y(t) - \hat{y}(t)\rvert$	$\hat{y}(t)$	$\lvert y(t) - \hat{y}(t)\rvert$	$\hat{y}(t)$	$\lvert y(t) - \hat{y}(t)\rvert$
0.0	1.0000000	1.0000301	3.01E-05	0.9999945	5.46E-06	1.000000	0.000000000
0.1	0.9900498	0.9900765	2.66E-05	0.9900465	3.30E-06	0.990050	4.15835E-10
0.2	0.9607894	0.9608197	3.02E-05	0.9607871	2.29E-06	0.960789	1.05819E-07
0.3	0.9139312	0.9139616	3.04E-05	0.9139282	2.94E-06	0.913929	2.68527E-06
0.4	0.8521438	0.8521687	2.49E-05	0.8521405	3.26E-06	0.852117	2.64556E-05
0.5	0.7788008	0.7788180	1.72E-05	0.7787980	2.75E-06	0.778646	0.000154950
0.6	0.6976763	0.6976891	1.28E-05	0.6976743	2.03E-06	0.697024	0.000652326
0.7	0.6126264	0.6126415	1.51E-05	0.6126246	1.76E-06	0.610442	0.002184561
0.8	0.5272924	0.5273142	2.18E-05	0.5272907	1.76E-06	0.521109	0.006183091
0.9	0.4448581	0.4448837	2.56E-05	0.4448567	1.36E-06	0.429477	0.015381566
1.0	0.3678794	0.3679015	2.20E-05	0.3678787	7.83E-07	0.333333	0.034546108

while for GA-IPA the 10^{-7} to 10^{-8}. This is the point to be notice that the best value of the derivative of Problem 2 is also in the range of 10^{-7} to 10^{-8} for GA-IPA (Table 9).

Case 3 $f(x) = x^m,\ g(y) = y^n$

Problem 3 Consider the following non homogeneous Emden–Fowler equation [17, 22]

$$\ddot{y} + \frac{8}{x}\dot{y} + xy = x^5 - x^4 + 44x^2 - 30x \tag{21}$$

where the initial conditions are given as

$$y(0) = 0,\ \dot{y}(0) = 0 \tag{22}$$

The exact solution of the problem is [10, 36]

$$y(x) = x^4 - x^3 \tag{23}$$

Table 7 DE-NN weights trained by different solvers for Problem 2

I	SA			GA		
	w_i	α_i	b_i	w_i	α_i	b_i
1	1.5201467	8.8901147	0.9027935	1.4716276	−0.4755607	1.6002860
2	−1.5685892	−7.3210845	7.9901269	2.4783652	0.4507208	1.3647668
3	−2.5359836	0.8868132	4.6943234	0.9541985	−0.3475089	1.3352805
4	−1.7266383	6.4666929	0.7572675	2.5578620	1.3607064	1.3871906
5	−5.0384262	−2.4886952	−7.4792495	−2.3116898	1.6004335	1.5874304
6	1.3903107	−7.0050581	2.0164435	0.3587036	−0.4600746	1.1781354
7	4.3444801	−5.7561590	7.8112571	2.4280146	−0.1688584	−0.6624523
8	0.7428610	8.4312150	5.5882520	−0.6946742	−0.8847606	1.7750085
9	−0.6469537	−7.7219384	1.6934410	2.8451283	−0.3666042	−0.9588452
10	3.9783800	6.7728618	9.1006489	1.9122634	0.4222301	−0.4777037

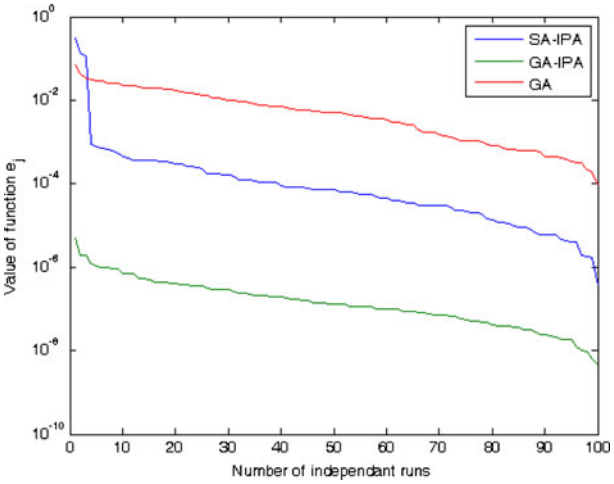

Fig. 8 Comparison of GA, Sa-IPA and GA-IPA for 100 independent runs for Problem 2

The (21) is the special form of expression (1) by taking $r = 8$ and $a = 1$. The problem is also solved on the same pattern as the previous problems, with a number of variations, the number of neurons in each hidden layer are taken as $m = 8$ which results in 24 unknown adaptive weights (α_i, w_i and b_i). These parameters are restricted to real numbers between $(-10, 10)$. A randomly generated initial population consists of a set of 100 chromosomes. Each chromosome has a length of 24 parameters. The input of the training set is taken from time $t \in (0, 1)$ with a step size of 0.1. It means that $s = 11$ in the expression (10) is to be used, so the fitness evaluation function subject to the Problem 3 in this case can be given as:

$$e_j = \frac{1}{11}\sum_{i=1}^{11}\left[\ddot{z}\,(t_i) + \frac{8}{t_i}\dot{z}\,(t_i) + t_i z\,(t_i) - (t_i)^5 + (t_i)^4 - 44\,(t_i)^2 + 30t_i\right]^2 + \frac{1}{2}\left[\{z(0)\}^2 + \{\dot{z}(0)\}^2\right]\Big|_j, \quad j = 1, 2, 3\ldots \tag{24}$$

Table 8 Statistical analysis of SA-IPA for the problem in Case 2

t	SA-IPA $\lvert y(t) - \hat{y}(t)\rvert$				SA-IPA $\lvert y'(t) - \hat{y}'(t)\rvert$			
	Best	Worst	Mean	STD	Best	Worst	Mean	STD
0	5.44E-06	0.4784636	0.0105718	0.0659547	2.14E-05	0.1322593	0.0027999	0.0153775
0.1	2.50E-07	0.4829271	0.0105837	0.0657393	1.86E-05	0.1258985	0.0037624	0.0133579
0.2	4.62E-06	0.4707706	0.0103640	0.0639148	8.60E-06	0.1982206	0.0046631	0.0256108
0.3	5.26E-06	0.4441879	0.0098226	0.0606821	9.22E-06	0.3281620	0.0072667	0.0399769
0.4	1.61E-06	0.4063106	0.0090563	0.0559941	2.34E-06	0.4230523	0.0092947	0.0538737
0.5	1.40E-05	0.3608972	0.0080880	0.0500828	1.34E-06	0.4784944	0.0108722	0.0636673
0.6	8.72E-06	0.3119360	0.0070140	0.0434912	1.27E-05	0.4943658	0.0114913	0.0672539
0.7	6.55E-06	0.2631978	0.0059292	0.0368166	8.44E-06	0.4751860	0.0110547	0.0655619
0.8	4.42E-07	0.2177561	0.0049074	0.0304788	9.15E-07	0.4304859	0.0101328	0.0607597
0.9	9.06E-08	0.1774355	0.0039639	0.0247030	1.64E-06	0.4002562	0.0092000	0.0546916
1.0	2.17E-06	0.1420642	0.0031492	0.0195324	5.92E-06	0.3587941	0.0081473	0.0490579

Table 9 Statistical analysis of GA-IPA for the problem in Case 2

t	GA-IPA $\lvert y(t) = \hat{y}(t)\rvert$				GA-IPA $\lvert y'(t) = \hat{y}'(t)\rvert$			
	Best	Worst	Mean	STD	Best	Worst	Mean	STD
0	3.62E-08	0.0003170	1.75E-05	3.81E-05	1.51E-08	8.36E-05	2.26E-05	1.74E-05
0.1	1.07E-07	0.0003079	1.74E-05	3.81E-05	9.39E-08	0.0002902	8.89E-05	6.27E-05
0.2	3.23E-07	0.0002956	1.81E-05	3.72E-05	2.52E-07	0.0002698	5.11E-05	4.02E-05
0.3	2.89E-08	0.0002844	1.49E-05	3.55E-05	1.48E-07	0.0002478	5.79E-05	4.79E-05
0.4	4.29E-08	0.0002660	1.36E-05	3.20E-05	4.32E-07	0.0002475	4.13E-05	4.42E-05
0.5	4.40E-08	0.0002344	1.17E-05	2.83E-05	3.39E-07	0.0003781	5.17E-05	5.27E-05
0.6	2.23E-07	0.0001933	1.20E-05	2.43E-05	4.80E-07	0.0004222	2.64E-05	4.61E-05
0.7	1.10E-07	0.0001541	1.04E-05	2.09E-05	2.18E-07	0.0003448	4.36E-05	5.26E-05
0.8	1.66E-08	0.0001263	8.49E-06	1.81E-05	4.47E-07	0.0002431	3.66E-05	4.05E-05
0.9	5.51E-08	0.0001091	6.87E-06	1.47E-05	1.07E-07	0.0003073	3.35E-05	4.81E-05
1.0	4.67E-08	8.990E-05	4.92E-06	1.15E-05	3.15E-07	0.0002595	3.39E-05	4.36E-05

where j be the number of generations, $\ddot{z}$, $\dot{z}$ and z are the networks given in (5), (4) and (3) respectively. Our scheme runs iteratively in order to find the minimum of fitness function e_j, with stoppage criteria as 2000 number of runs or fitness value $e_j \leq 10^{-10}$ whichever comes earlier. One of the best unknown weights trained stochastically by DE-NN algorithms for the step size of 0.1 for SA and GA are provided in Fig. 9a and b respectively in the bounded range. These weights can be used to obtain the solution of the equation for any real input time t between 0 and 1. The trained weights by SA and GA hybridized by IPA provide the results so close to each other that it was merely difficult to differentiate between the accuracy level of both of the stochastic solvers, so the error of each method with the exact solution has been computed at each time step. The absolute error of the exact solution with the results achieved by the training of SA-IPA and GA-IPA are given in the Fig. 8. The Fig. 10 is drawn on the log scale to have a clear view on the difference of both the errors. This obvious from the Fig. 9 that the SA-IPA provides the results upto an error of 10^{-4} and GA-IPA in the range of 10^{-6} to 10^{-8}, which clarify the strengths of the genetic algorithm in optimization of complex problems.

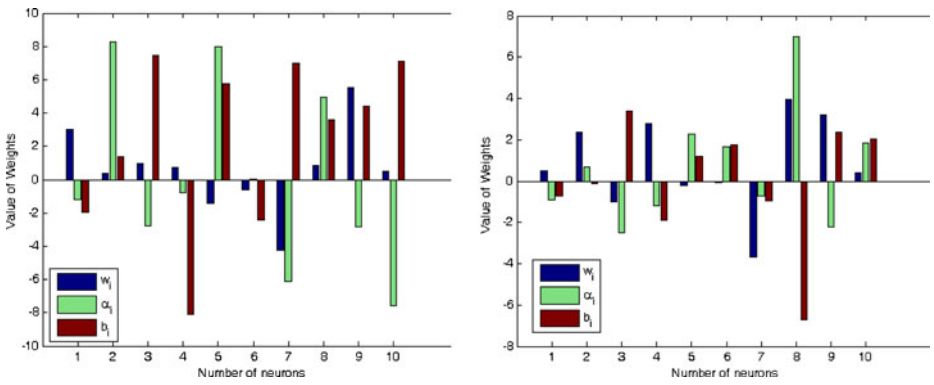

Fig. 9 The unknown weights trained by DE-NN algorithm for Problem 3, **a** for SA and **b** for GA

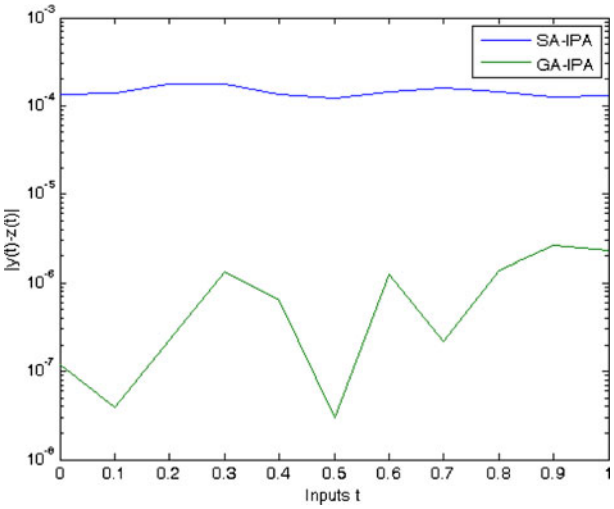

Fig. 10 The difference of the absolute error with the exact for SA-IPA and GA-IPA $|y(t) - z(t)|$

The training of the weights of Problem 3 is highly stochastic in nature so the statistical analysis is made to see the consistency and accuracy for the stochastic optimizer. 125 independent runs are for made for both SA and GA in collaboration of IPA. The results for best 100 run are given in Fig. 11. The statistical analysis, based on 100 independent runs of our scheme, is provided in the Table 10. It is clear from the Table that the values of the best and worst have much difference, while the mean and standard deviation results are not at wide spread. Moreover, the average accuracy is obtained in the range of 10^{-5} to 10^{-6}. On the basis of these result it can be stated that the proposed method is applicable to solve such problems of Emden–Fowler at a good level of accuracy.

The best and mean value of the GA-IPA is much better than that of SA-IPA as evident from the Table 10.

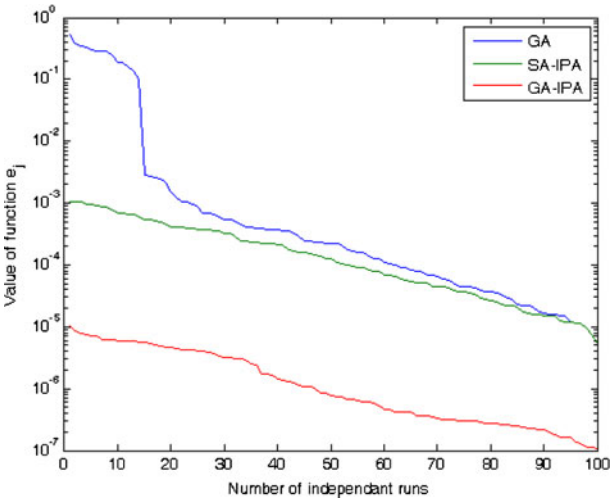

Fig. 11 Comparison of GA, Sa-IPA and GA-IPA for 100 independent runs

Table 10 Statistical analysis for $t \in (0,1)$ in Problem 3

T	Best		Worst		Mean		Standard deviation	
	SA-IPA	GA-IPA	SA-IPA	GA-IPA	SA-IPA	GA-IPA	SA-IPA	GA-IPA
0.0	5.29E-07	4.58E-09	0.512802	0.000239	0.034635	6.36E-06	0.102069	2.85E-05
0.1	1.36E-06	3.95E-08	0.51255	0.000254	0.034916	2.88E-05	0.102204	3.69E-05
0.2	1.51E-05	2.21E-07	0.519216	0.000255	0.035242	2.91E-05	0.103080	3.84E-05
0.3	2.10E-06	3.55E-07	0.526737	0.000243	0.035403	1.51E-05	0.104314	3.04E-05
0.4	9.50E-06	5.31E-07	0.531842	0.000242	0.035555	1.64E-05	0.105443	2.97E-05
0.5	4.66E-07	3.01E-08	0.533322	0.000254	0.035687	2.71E-05	0.106162	3.52E-05
0.6	4.96E-07	1.33E-07	0.531300	0.000257	0.035763	2.89E-05	0.106351	3.69E-05
0.7	2.81E-06	5.87E-08	0.526544	0.000244	0.035764	1.94E-05	0.106177	3.20E-05
0.8	5.92E-06	1.36E-08	0.523687	0.000238	0.035703	1.73E-05	0.105848	2.97E-05
0.9	2.64E-06	2.84E-07	0.518163	0.000248	0.035564	2.52E-05	0.105248	3.33E-05
1.0	3.32E-06	1.81E-07	0.509409	0.000238	0.035291	1.91E-05	0.104137	3.02E-05

5 Conclusion

A novel method appropriate for the numerical treatment of Emden–Fowler equation that relies upon the function approximation capabilities of neural network assisted with computational intelligence has been presented. The method has been applied successfully on various cases of Emden–Fowler that has the vast applicability. The accuracy of the proposed method is checked by comparing the results with exact and standard state of art deterministic numerical solver and is found to be in good agreement. A large number of Monti Carlo simulations are performed to validate the reliability and effectiveness of the stochastic solvers. It must also be expressed that the strength of designed scheme over such solver is that it can provide the results on continuous time finite domain of inputs instead of predefine discrete grid of points.

5.1 Future work

In future, we are supposed to improve the mathematical modeling by employing activation functions other than log-sigmoid or applying other neural network models. Another, possible improvement is to develop other population based optimization algorithms.

References

1. Chandrasekhar, S.: Introduction to the Study of Stellar Structure. Dover, New York (1967)
2. Davis, H.T.: Introduction to Nonlinear Differential and Integral Equations. Dover, New York (1962)
3. Richardson, O.W.: The Emission of Electricity from Hot Bodies. London (1921)
4. Wazwaz, A.M.: A new algorithm for solving differential equations of Lane–Emden type. Appl. Math. Comput. **118**, 287–310 (2001)
5. Shawagfeh, N.T.: Nonperturbative approximate solution for Lane–Emden equation. J. Math. Phys. **34**, 4364–4369 (1993)
6. Wazwaz, A.M.: The modified decomposition method for analytic treatment of differential equations. Appl. Math. Comput. **173**:165–176 (2006)
7. Wazwaz, A.M.: Adomian decomposition method for a reliable treatment of the Emden–Fowler equation. Appl. Math. Comput. **161**, 543–560 (2005)

8. Guedda, M.: Multiple positive solutions of a singular boundary value problem for a superliner Emden–Fowler equation. J. Math. Anal. Appl. **352**, 259–270 (2009)
9. Han, Z., Sun, S., Shi, B.: Oscillation criteria for a class of second order Emden–Fowler delay dynamic equations on time scales. J. Math. Anal. Appl. **334**, 847–858 (2007)
10. Batiha, B., Noorani, M.S.M., Hashim, I.: Numerical solutions of the nonlinear integro-differential equations. Int. J. Open Problems Compt. Math. **1**(1) (2008)
11. Dehgham, M., Shakeri, F.: Solution of an integro-differential equation arising in oscillating magnetic field using He's Homtopy perturbation method. Prog. Electromagn. Res. **78**, 361–376 (2008)
12. Wazwaz, A.M.: Analytical solution for the time-dependent Emden–Fowler type of equations by Adomian decomposition method. Appl. Math. Comput. **166**, 38–651 (2005)
13. Chowdhury, M.S.H., Hashim, I.: Solutions of time-dependent Emden Fowler type equations by homotopy-perturbation method. Phys. Lett. A **368**, 305–313 (2007)
14. Liao, S.J.: The proposed homotopy analysis techniques for the solution of nonlinear problems. Ph.D. dissertation, Shanghai Jiao Tong University, Shanghai (1992). In English
15. Bataineh, A.S., Noorani, M.S.M., Hashim, I.: Solutions of time-dependent Emden Fowler type equations by homotopy analysis method. Phys. Lett. A **371**, 72–82 (2007)
16. Sun, Y.G., Wong, J.S.W.: Oscillation criteria for second order forced ordinary differential equations with mixed nonlinearities. J. Math. Anal. Appl. **334**, 549–560 (2007)
17. Meada, A.J., Fernandez, A.A.: The numerical solution of linear ordinary differential equation by feedforward neural network. Math. Comput. Model. **19**, 1–25 (1994)
18. Raja, M.A.Z., Khan, J.A., Qureshi, I.M.: A new stochastic approach for solution of Riccati differential equation of fractional order. Ann. Math. Artif. Intell. **60**(3–4), 229–250 (2010). doi:10.1007/s10472-010-9222-x
19. Sivanandam, S.N., Visalakshi, P.: Multiprocessor scheduling using hybrid particle swarm optimization with dynamically varying inertia. Int. J. Comput. Sci. Appl. **4**(3), 95–106 (2007)
20. Monterola, C., Saloma, C.: Characterizing the dynamics of constrained physical systems with unsupervised neural network. Phys. Rev. E **57**, 1247R–1250R (1998)
21. Junaid, A.K., Raja, M.A.Z., Qureshi, I.M.: Swarm intelligence for the solution of problems in differential equations. In: Second International Conference on Environmental and Computer Science, pp. 141–147. ICECS (2009)
22. Rarisi, D.R. et al.: Solving differential equations with unsupervised neural networks. J. Chem. Eng. Process. **42**, 715–721 (2003)
23. Lewis, A., Mostaghim, S., Randall, M. (eds.): Biologically-Inspired Optimization Methods, Parallel Algorithms, System and Application, Studies in Computational Intelligence, vol. 210 (2009)
24. Junaid, A., Raja, M.A.Z., Qureshi, I.M.: Evolutionary computing approach for the solution of initial value problems in ordinary differential equations. WASET **55**, 578–581 (2009)
25. Raja, M.A.Z., Khan, J.A., Qureshi, I.M.: Swarm in telligent optimized neural networks for solving fractional differential equations. Int. J. Innov. Comput. I **7**(11), 6301–6318 (2011)
26. Raja, M.A.Z., Junaid, A.K., Qureshi, I.M.: Evolutionary computation technique for solving Riccati differential equation of arbitrary order. WASET **58**, 303–309 (2009)
27. Hetch-Nielsen, R.: Kolmogorov's mapping neural network existence theorem. In: 1st IEEE Inter. Conf. on Neural Networks, vol. 3, p. 11. San Diego, CA (1987)
28. Funahashi, K.I.: On the approximate realization of continuous mappings by neural networks. Neural Netw. **2**, 183 (1989)
29. Hornik, K., Stichcombr, M., White, H.: Universal approximation of an unknown mapping and its derivatives using multilayer feed forward networks. Neural Netw. **3**, 551–560 (1990)
30. Cybenko, G.: Approximation by superposition of a sigmoidal function. Math. Control. Signal. Syst. **2**, 303 (1989)
31. Goldberg, D.E.: Genetic Algorithms in Search Optimization and Machine Learning, p. 41. Addison Wesley. ISBN 0201157675 (1989)
32. Srinivas, M., Patnaik, L.: Adaptive probabilities of crossover and mutation in genetic algorithms. IEEE Trans. Syst. Man Cybern. **24**(4), 656–667 (1994)
33. Zhang, J., Chung, H., Lo, W.L.: Clustering-based adaptive crossover and mutation probabilities for genetic algorithms. IEEE Trans. Evol. Comput. **11**(3), 326–335 (2007)
34. Raja, M.A.Z., Khan, J.A., Qureshi, I.M.: Solution of fractional order system of Bagley–Torvik equation using evolutionary computational intelligence. Math. Probl. Eng. Article ID. 765075, 1–18 (2011)

35. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science. New Series **220**(4598), 671–680 (1983)
36. Cerny, V.: A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm. J. Optim. Theor. Appl. **45**, 41–51 (1985)
37. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equations of state calculations by fast computing machines. J. Chem. Phys. **21**(6), 1087–1092 (1953)
38. Granville, V., Krivanek, M., Rasson, J.-P.: Simulated annealing: a proof of convergence. IEEE Trans. Pattern Anal. Mach. Intell. **16**(6), 652–656 (1994). doi:10.1109/34.295910
39. De Vicente, J., Lanchares, J., Hermida, R.: Placement by thermodynamic simulated annealing. Phys. Lett. A **317**(5–6), 415–423 (2003)
40. Byrd, R.H., Gilbert, J.C., Nocedal, J.: A trust region method based on interior point techniques for nonlinear programming. Math. Program. **89**(1), 149–185 (2000)
41. Byrd, R.H., Hribar, M.E., Nocedal, J.: An interior point algorithm for large-scale nonlinear programming. SIAM J. Optim. **9**(4), 877–900 (1999)
42. Waltz, R.A., Morales, J.L., Nocedal, J., Orban, D.: An interior algorithm for nonlinear optimization that combines line search and trust region steps. Math. Program. **107**(3), 391–408 (2006)
43. Chowdhury, M.S.H., Hashim, I.: Solutions of Emden–Fowler equations by homtopy perturbation method. Nonlin. Anal.: Real World Appl. **10**, 104–115 (2009)