

Research Article

A New Stochastic Technique for Painlevé Equation-I Using Neural Network Optimized with Swarm Intelligence

Muhammad Asif Zahoor Raja,^{1,2,3} Junaid Ali Khan,^{1,2,3}
Siraj-ul-Islam Ahmad,^{2,4} and Ijaz Mansoor Qureshi^{1,2}

¹Department of Electronic Engineering, International Islamic University, Islamabad, Pakistan

²Center for Computational Intelligence, P.O. Box 2300, Islamabad, Pakistan

³Department of Electrical Engineering, COMSATS Institute of Information Technology, Attock, Pakistan

⁴Pakistan Institute of Engineering and Applied Science, Nilore, Pakistan

Correspondence should be addressed to Muhammad Asif Zahoor Raja, asif.phdee10@iiu.edu.pk

Received 6 March 2012; Revised 9 April 2012; Accepted 13 April 2012

Academic Editor: Christian W. Dawson

Copyright © 2012 Muhammad Asif Zahoor Raja et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A methodology for solution of Painlevé equation-I is presented using computational intelligence technique based on neural networks and particle swarm optimization hybridized with active set algorithm. The mathematical model of the equation is developed with the help of linear combination of feed-forward artificial neural networks that define the unsupervised error of the model. This error is minimized subject to the availability of appropriate weights of the networks. The learning of the weights is carried out using particle swarm optimization algorithm used as a tool for viable global search method, hybridized with active set algorithm for rapid local convergence. The accuracy, convergence rate, and computational complexity of the scheme are analyzed based on large number of independents runs and their comprehensive statistical analysis. The comparative studies of the results obtained are made with MATHEMATICA solutions, as well as, with variational iteration method and homotopy perturbation method.

1. Introduction

The history of Painlevé equations is more than one century old. These are six second-order nonlinear irreducible equations that define new transcendental functions known as Painlevé Transcendents. These functions describe different physical processes and have been extensively used in both pure [1] and applied mathematics [2], along with theoretical physics [3]. For instance, Painlevé Transcendents are used in solutions to Korteweg-de Vries (KdV), cylindrical KdV and Bussinesq equations [4, 5], bifurcations in nonlinear non-integral models [6], matrix models of quantum gravity with continuous limits [7, 8], among others. The history, importance, and applications of Painlevé transcendents can be seen elsewhere [9, 10]. In recent publications, Painlevé equation-I (PE-I) is used in Tronquée [11] and hyperasymptotic [12] solutions, as well as in modeling the viscous shocks in Hele-shaw flow and Stokes phenomena [13]. Beside this, many

researchers have employed PE-I in diverse fields of applied science and engineering [14–22].

In this article, our investigation is confined to find out solution of initial value problem (IVP) of nonlinear second-order PE-I, written in the following form:

$$\begin{aligned}y''(t) &= 6y(t)^2 + t, \quad t \in (0, 1), \\ y(0) &= 0, \quad y'(0) = 1.\end{aligned}\tag{1}$$

In the present study, the strength of feed forward artificial neural networks (ANNs) is exploited for approximate mathematical model of PE-I. The real strength of such model to solve the differential equations can be achieved by using modern stochastic solvers for optimization of weights based on particle swarm optimization (PSO) technique hybrid with local search methods. For example, the solution of linear and nonlinear differential equations and their systems is reported

extensively by many researchers [23–26]. Recently, solution of well-known fractional-order systems in engineering based on Bagley-Torvik and Riccati equations are other illustrative examples of such approaches [27, 28].

In this paper, ANNs supported with PSO and active set algorithm (ASA) are used to find the solution of IVP of PE-I. Monte-Carlo simulations of proposed scheme are performed and analyzed statistically to verify and validate its effectiveness. Comparison with standard MATHEMATICA, as well as homotopy perturbation method (HPM) [29, 30] and variational iteration method (VIM) [29, 30] results is also carried out for the given scheme.

The organization of the paper is as follows: the objective function creation with neural network mathematical modeling and its training methodology are introduced in Section 2. A brief introduction to particle swarm optimization algorithm is revealed in Section 3. A detailed application of the designed method along with some discussion on the results is presented in Section 4. Comparative analysis of the results is presented in Section 5. Last section concludes the findings along with some directions for future research.

2. Mathematical Modeling

The Feed-forward ANNs is well known to be used as a universal function approximator for any continuous function along with its derivatives on a compact set. In the modeling of differential equation through ANN methodology, following continuous mapping is used for the solution $y(t)$, first-order derivative $y'(t)$ and second-order derivative $y''(t)$, respectively, given as [31–34]

$$\begin{aligned}\hat{y}(t) &= \sum_{i=1}^m \alpha_i f(w_i t + \beta_i), \\ \hat{y}'(t) &= \sum_{i=1}^m \alpha_i f'(w_i t + \beta_i), \\ \hat{y}''(t) &= \sum_{i=1}^m \alpha_i f''(w_i t + \beta_i),\end{aligned}\quad (2)$$

where α_i , w_i , and β_i are real-valued bounded adaptive parameters, m is the number of neurons, and f being the activation function taken as log sigmoid for hidden layers:

$$f(x) = \frac{1}{1 + e^{-x}}. \quad (3)$$

The linear combination of the networks represented by (2) can arbitrarily model the nonlinear PE-I. It is called differential equation neural network (DE-NN) of the equation and its architecture is shown in Figure 1.

The objective or fitness function e for solving PE-I has been formulated by defining an unsupervised error, given as the sum of mean square errors:

$$e = e_1 + e_2|_j, \quad j = 1, 2, 3, \dots, M, \quad (4)$$

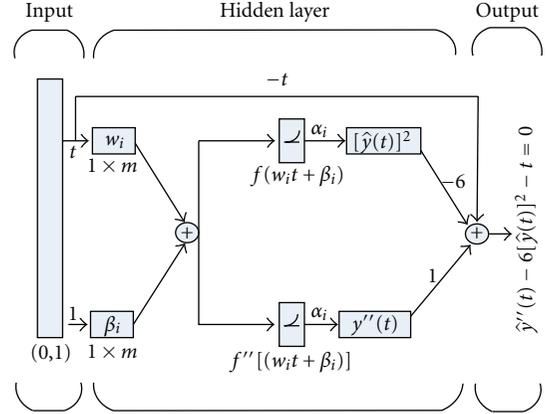


FIGURE 1: DE-NN architecture for Painlevé equation I.

where j is the iteration number, M represents the total number of iterations, and the error term e_1 is related with the differential equation and is written as

$$e_1 = \frac{1}{N+1} \sum_{m=0}^N (\hat{y}_m'' - 6\hat{y}_m^2 - t_m)^2, \quad t \in (0, 1), \quad (5)$$

$$N = \frac{1}{h}, \quad \hat{y}_m = \hat{y}(t_m), \quad t_m = mh.$$

Here, the interval $t \in (0, 1)$ is divided into N number of steps $t \in (t_0 = 0, t_1, t_2, \dots, t_N = 1)$ with step size h and $\hat{y}(t)$ and $\hat{y}'(t)$ are DE-NN given in set of (2).

Similarly, the error term e_2 is due to initial conditions and it is given as

$$e_2 = \frac{1}{2} (\hat{y}_0^2 + (\hat{y}'_0 - 1)^2). \quad (6)$$

It is quite clear that, with the availability of weights α_i , w_i , and β_i in DE-NN model for which the values of functions e_1 and e_2 approach zero, the value of unsupervised error e also approaches zero. Therefore, the solution $y(t)$ of the PE-I is approximated by $\hat{y}(t)$ as given in (2).

3. Learning Methodology

The learning procedures are introduced here for finding the unknown weights of DE-NN networks representing PE-I using PSO and ASA algorithms.

The swarm intelligence techniques often referred to as PSO algorithm was first introduced by Kennedy and Eberhart [35]. It is a kind of global optimization technique which was based on the model of social behavior of bird flocking and fish schooling. Its discrete and continuous versions have been developed and used in different optimization problems of applied science and engineering. Few examples include the application to mobile communications, sensor networks, inventory control, multiprocessor scheduling, controls, stock market prediction, and steganographic methods [36–39] and so forth.

In PSO algorithm, each single candidate solution to an optimization problem is taken as a particle in the search

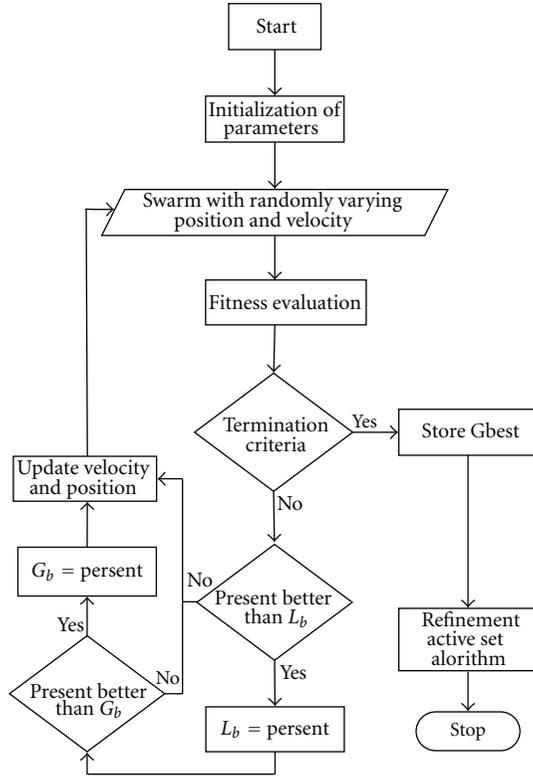


FIGURE 2: Flowchart of particle swarm optimization aided with actives set algorithm.

space. The exploration of a problem space in PSO algorithm is made with a population of particles called a swarm. All particles in the swarm have own fitness values associated with problem specific objective function. The PSO algorithm is initialized with a swarm of particles randomly and is used to search for optimal solution iteratively. The broader spread of initial swarm results in optimal performance of the algorithm. The position and the velocity of each particle are updated according to its known previous local best position \mathbf{P}_{lb}^{n-1} and the global best position of all particles \mathbf{P}_{gb}^{n-1} in the swarm so far. The updating formula for each particle velocity and position in continuous standard PSO is written as

$$\begin{aligned} \mathbf{v}_i^n &= \omega \mathbf{v}_i^{n-1} + c_1 \mathbf{r}_1 (\mathbf{P}_{lb}^{n-1} - \mathbf{x}_i^{n-1}) + c_2 \mathbf{r}_2 (\mathbf{P}_{gb}^{n-1} - \mathbf{x}_i^{n-1}) \\ \mathbf{x}_i^n &= \mathbf{x}_i^{n-1} + \mathbf{v}_i^{n-1}, \end{aligned} \quad (7)$$

where \mathbf{x}_i is vector to represent i th particle of the swarm, $i = 1, 2, \dots, p$, p is an integer giving the total number of particles in a swarm. The \mathbf{v}_i is the velocity vector associated with i th particle, c_1 and c_2 are the local and global social acceleration constants, respectively, ω is the inertia weight linearly decreasing over the course of search between 0 and 1, and \mathbf{r}_1 and \mathbf{r}_2 are random vectors with elements distributed between 0 and 1.

The elements of velocity are bounded as $v_i \in [-v_{\max}, v_{\max}]$, where v_{\max} is maximum velocity. If the velocity goes beyond its maximum value, it will be set to v_{\max} . This parameter has its own importance and controls the

convergence rate of the algorithm. The execution of the algorithm is terminated on the criterion to maximum flights/cycles is completed or a specific value of the fitness is achieved. The generic flowchart showing the process of proposed algorithm is presented in Figure 2.

The algorithm is given in the following steps.

Step 1. Initialization. The initial swarm of particle is generated randomly using bounded real values. Each particle represents the candidate solution with as many members as number of unknown parameters in DE-NN networks. The scattering in values of initial swarm is maintained for better search space for the algorithm. Initialize the values of parameters as given in Table 1 before execution of algorithm.

Step 2. Fitness Evaluation. Calculate fitness for each particle by using the fitness function given in (4), (5), and (6).

Step 3. Termination Criteria. Terminate the algorithm if any of following criteria satisfies the following:

- (i) predefined value of fitness achieved that is, $e \leq 10^{-10}$,
- (ii) number of maximum flights/cycles is executed.

If yes, then go to Step 6.

Step 4. Ranking. Rank each particle on the basis of minimum of the fitness function values.

Step 5. Renewal. Update the velocity and position are updated using the equations given in (7). Repeat the algorithm from Step 2 to Step 5 until total number of flights is reached.

Step 6. Refinement. MATLAB built-in formulation is used for running active set algorithm (ASA) for fine tuning of the results. The global best particle of PSO is used as a start point of the algorithm and other parameter settings for ASA is also provided in Table 1.

Step 7. Statistical Analysis. Store the value of global best particle along with its fitness value and time of executions for algorithm. Repeat Step 1 to 6 for sufficient large number of runs for reliable statistical analysis.

4. Simulation and Results

The well-known analytic solvers like adomian decomposition method (ADM), variational iteration method (VIM), homotopy perturbation method (HPM), homotopy analysis method (HAM), and their modified versions are used extensively to provide the solutions of strong nonlinear Painlevé equations [29, 30, 40–42]. The generic solution is provided in all these mentioned methods in the form of infinite series, but the results are determined using finite number of terms, whose accuracy can be enhanced with the increase in number of terms.

Before describing the designed approach, a brief description of VIM and HPM has been provided for PE-I. The approximate solution of $y(t)$ for PE-I (1) in case of both

TABLE 1: Parameter settings for ASA and PSO algorithms.

PSO		ASA	
Parameters	Setting	Parameters	Setting
Swarm Size	160	Start Point	Best Particle of PSO
Particle size	30	No. of variable	30
Flights	2000	Iteration	1000
c_1	Linear decreasing (2.5 to 0.5)	Maximum function Evaluations (MaxFunEvals)	50000
c_2	Linear increasing (0.5 to 2.5)	Function tolerance (TolFun)	10^{-18}
ω	Linearly decreasing (0.9 to 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)

VIM and HPM is represented in the form of a function $u(t)$. To describe the VIM, consider the following nonlinear differential equation:

$$L(u) + N(u) = g(t), \quad (8)$$

where L and N are linear and nonlinear operators, respectively, and $g(t)$ is an inhomogeneous term. In standard VIM, the correct functional can be constructed as

$$u_{n+1}(t) = u_n(t) + \int_0^t \lambda(\tau) (Lu_n(\tau) + N\tilde{u}_n(\tau) - g(\tau)) d\tau, \quad (9)$$

here, $\lambda(\tau)$ represents a general Lagrangian multiplier, n and $n+1$ are the present and next approximation, \tilde{u} represents restricted variation, that is, $\delta\tilde{u} = 0$. With suitable choice of u_0 , the fixed point of the correction functional (9) can be considered as an approximate solution of the equation and successive approximation u_{n+1} of the solution u will be readily obtained upon by determining the Lagrange multiplier. Consequently, the solution is given by $u = \lim_{n \rightarrow \infty} u_n$. For interesting readers, the detail description of VIM along with its applicability to various kinds of differential equations can be seen in [43–46].

The governing iterative formulation for solving the PE-I using VIM with the optimal value for $\lambda(\tau) = \tau - t$ can be written as

$$u_{n+1}(t) = u_n(t) + \int_0^t (\tau - t) (u_n''(\tau) - 6\tilde{u}_n^2(\tau) - \tau) d\tau, \quad (10)$$

The approximate solution $u(t)$ by VIM for first three iterations using $u_0(t) = t + (1/6)t^3$ can be written as

$$\begin{aligned} u_1(t) &= t + \frac{1}{6}t^3 + \frac{1}{2}t^4 + \frac{1}{15}t^6 + \frac{1}{336}t^8, \\ u_2(t) &= t + \frac{1}{6}t^3 + \frac{1}{2}t^4 + \frac{1}{15}t^6 + \frac{1}{7}t^7 + \dots + \frac{187}{764400}t^{14} \\ &\quad + \frac{1}{100800}t^{16} + \frac{1}{5757696}t^{18}, \end{aligned}$$

$$\begin{aligned} u_3(t) &= t + \frac{1}{6}t^3 + \frac{1}{2}t^4 + \frac{1}{15}t^6 + \frac{1}{7}t^7 + \dots \\ &\quad + \frac{1187}{1210325276160000}t^{14} + \frac{1}{60939454466400}t^{36} \\ &\quad + \frac{1}{7768399149858816}t^{38}. \end{aligned} \quad (11)$$

On the other hand to explain the HPM, let us consider the following general equation of type:

$$A(u) - g(t) = 0, \quad t \in \Omega, \quad (12)$$

where A represents the differential operator, Ω is the domain and $g(t)$ is a known analytical function. The operator A can consist of linear part L and a nonlinear part N . In standard HPM, the Homotopy $v(r, p) : \Omega \times [0, 1] \rightarrow R$ can be constructed which satisfies

$$H(v, p) = (1 - p)(L(v) - L(u_0)) + p(A(v) - g(t)) = 0, \quad (13)$$

here, $p \in [0, 1]$ is an embedding parameter and u_0 is an initial approximation of (12). By using small embedding parameter p , the v th solution of (12) can be given as power series in p , that is,

$$v = v_0 + pv_1 + p^2v_2 + \dots, \quad (14)$$

for $p = 1$, the approximate solution for (12) is given as

$$u = \lim_{p \rightarrow 1} v = v_0 + v_1 + v_2 + \dots \quad (15)$$

The method provides enough convergence in most of the cases and the rate of convergence depends on the nonlinear operator $A(v)$. The detail description of the HPM and its applications can be seen in [47–50].

Using HPM, the solution of PE-I (1) starting from $u_0(t) = t + (1/6)t^3$ can be given as

$$u = u(0) + tu'(0) + \frac{1}{6}t^3 + 6 \iint_0^t u^2 dt dt. \quad (16)$$

using the initial conditions (1), the homotopy constructed by the method can be given as

$$H(u, p) = u - t - \frac{1}{6}t^3 + 6p \iint_0^t u^2 dt dt = 0. \quad (17)$$

Inserting $\sum_{i=0}^{\infty} p^i u_i$ instead of u and comparing the coefficients of 1, 2, 3, 4, and 5 powers of p , the approximate solutions by HPM for PE-I are, respectively, given as

$$\begin{aligned} u_1(t) &= t + \frac{1}{6}t^3 + \frac{1}{2}t^4 + \frac{1}{15}t^6 + \frac{1}{336}t^8, \\ u_2(t) &= t + \frac{1}{6}t^3 + \frac{1}{2}t^4 + \frac{1}{15}t^6 + \frac{1}{7}t^7 \\ &\quad + \dots + \frac{1}{40}t^9 + \frac{17}{46200}t^{11} + \frac{1}{26208}t^{13}, \\ u_3(t) &= t + \frac{1}{6}t^3 + \frac{1}{2}t^4 + \frac{1}{15}t^6 + \frac{1}{7}t^7 \\ &\quad + \dots + \frac{5219}{8408400}t^{14} + \frac{3551}{144144000}t^{16} + \frac{95}{224550144}t^{18}, \\ u_4(t) &= t + \frac{1}{6}t^3 + \frac{1}{2}t^4 + \frac{1}{15}t^6 + \frac{1}{7}t^7 + \dots + \frac{163469}{14378364000}t^{19} \\ &\quad + \frac{163451}{491203440000}t^{21} + \frac{145806371}{3950151364800000}t^{23}, \\ u_5(t) &= t + \frac{1}{6}t^3 + \frac{1}{2}t^4 + \frac{1}{15}t^6 + \frac{1}{7}t^7 + \dots \\ &\quad + \frac{4047099827}{23087434930560000}t^{24} + \frac{145806371}{39501513648000000}t^{26} \\ &\quad + \frac{74849}{1737058836344832}t^{28}. \end{aligned} \quad (18)$$

Along with the analytical solutions given by VIM and HPM, the results are also calculated with MATHEMATICA $y_M(t)$. These results are used for comparison with the solution provided by the proposed design scheme.

Mathematical model of the equation is made with DE-NN networks given by (2) by taking 10 numbers of neurons, resulting in 30 unknown weights in DE-NN networks. The weights are taken as bounded real numbers between -50 to 50 . The initial swarm consists of 160 particles, each with 30 numbers of elements, which is equivalent to the number of weights for DE-NN networks. Input of the training set is taken as $t \in (0, 1)$ with a step of $h = 0.1$. The fitness function

provided in (4) can be formulated as

$$e = \frac{1}{11} \sum_{m=0}^{10} (\hat{y}_m'' - 6\hat{y}_m^2 - t_m)^2 + \frac{1}{2} (\hat{y}_0^2 + (\hat{y}'_0 - 1)^2) \Big|_j, \quad (19)$$

$$j = 1, 2, 3, \dots, 2000.$$

The PSO algorithm runs iteratively in order to compute the minimum of fitness function (19) and the best particle of PSO technique is passed to ASA algorithm as a start point for rapid local search. The parameter settings for the execution of algorithms are provided in Table 1.

Similarly, hundred independent runs of PSO and PSO hybrid with ASA (PSO-ASA) algorithms have been performed for finding the optimal weights. The weights obtained by the algorithms are used in (2) to obtain the proposed solutions $\hat{y}(x)$ of PE-I. One set of weights for which the value of fitness function 1.3699×10^{-04} and 9.1210×10^{-08} for PSO and PSO-ASA, respectively, are provided in Table 2. Results calculated by these weights are provided in Tables 3 and 4 for inputs between 0 and 1 with a step of 0.1. The solutions of 3rd order VIM, 5th order HPM for the same inputs are also given in Table 3 while detailed information has been provided in Table 4. It can be seen that the value of mean absolute error (MAE) for 3rd order VIM, 5th order HPM, PSO, and PSO-ASA techniques are 4.6098×10^{-04} , 7.6946×10^{-05} , 9.8594×10^{-04} , 1.1198×10^{-05} , respectively.

The values of minimum (MIN), maximum (MAX), mean, and standard deviation (STD), for absolute error (AE) $|y_M(t) - \hat{y}(t)|$, are calculated on the basis of hundred independent runs for PSO and PSO-ASA methods. Results are provided in Table 5 for inputs 0 and 1 with step of 0.1 for PE-I. It can be seen that values of AE on the basis of MIN are of the order 10^{-03} to 10^{-04} and 10^{-06} to 10^{-08} for PSO and PSO-ASA algorithms, while the values of AE based on mean lies in the range 10^{-01} to 10^{-02} and 10^{-04} to 10^{-05} , respectively, for said solvers. The lowest values of statistical parameters are found by hybrid technique PSO-ASA than that of PSO algorithm.

The accuracy and convergence of the algorithms are analyzed further by calculating the values of MAE and fitness achieved (FA) for 100 independent runs. Results are shown graphically on semilog scale in order to elaborate the difference in the values. The value of FA and MAE against number of independent runs are shown graphically in Figure 3(a) for PSO and PSO-ASA methods. The numbers of independent runs are arranged on the basis of ascending order of the fitness achieved, as well as values of MAE and are plotted in Figure 3(b). On the basis of acceptability criteria of $MAE \leq 10^{-03}$, the convergence for PSO and PSO-ASA is 1% and 100%, respectively.

5. Comparative Analysis of Results

In this section, the comparative study for results is presented based on values of MAE, execution time (ET), mean fitness achieved (MFA), global mean absolute error (GMAE),

TABLE 2: A set of weights trained for DE-NN networks.

PSO		PSO-ASA		PSO		PSO-ASA	
w_1	11.003223968159700	-7.607757117408950	w_6	2.212628719834520	2.474582479870870		
w_2	1.215553756976230	-1.490826886546450	w_7	-3.435984020017740	1.576108544205200		
w_3	-0.105724097992031	-0.545895962495148	w_8	3.162425319231580	-3.958973407625250		
w_4	-0.631338164295177	7.264468183878910	w_9	0.783489331068287	1.998216516692480		
w_5	5.142485647745030	-8.830221260895670	w_{10}	-1.369123625736370	0.021432666153674		
α_1	2.212942902261790	1.227208825263930	α_6	1.400872158822050	1.316114527930020		
α_2	8.952236157349910	-3.632183934646160	α_7	-0.420235540017917	1.323004693243890		
α_3	0.211094325564814	4.680032352849500	α_8	0.771318701716261	-2.560496480052050		
α_4	-0.918096490077305	9.995894668412700	α_9	-1.394533732778610	1.004134898951920		
α_5	34.739272824112700	-0.083663885403684	α_{10}	-0.343093197542097	0.887019181878860		
β_1	-14.756692394364600	8.897738213117890	β_6	1.201391820838980	1.100068363593600		
β_2	-3.441636802273880	2.675860058162530	β_7	3.935692917406700	0.587872446048014		
β_3	2.708026281856620	-0.220415897788109	β_8	-3.847728945588120	5.075428578185270		
β_4	1.986093010073140	-9.999999632285810	β_9	-1.648848649331190	-2.429219273541780		
β_5	-9.056219460761080	8.881048710700930	β_{10}	-0.531535793135294	2.174973763274050		

TABLE 3: Comparison of the results for solution of Painlevé I.

T	$y_M(t)$	Results for $u(t)$		Proposed Results $\hat{y}(t)$	
		VIM	HPM	PSO	PSO-ASA
0.1	0.1002167469	0.1002167477	0.1002167477	0.101267314	0.100211733
0.2	0.2021394539	0.2021394527	0.2021394527	0.202944631	0.202137722
0.3	0.3086307548	0.3086307489	0.3086307492	0.309301843	0.308629917
0.4	0.4239862999	0.4239862788	0.4239862895	0.424625376	0.423985302
0.5	0.5543401416	0.5543399112	0.5543401182	0.555018673	0.554341815
0.6	0.7084621313	0.7084596600	0.7084620603	0.709234588	0.708467300
0.7	0.8992500131	0.8992296942	0.8992493803	0.900159983	0.899254787
0.8	1.1465318491	1.1463982509	1.1465240263	1.147598102	1.146541253
0.9	1.4825246345	1.4817789520	1.4824439956	1.483815424	1.482531252
1.0	1.9631285609	1.9594210423	1.9624483064	1.965104083	1.963204324

TABLE 4: Comparison of the results for solution of Painlevé I.

T	$u_k(t)$ for VIM			$u_k(x)$ for HPM			DE-NN		
	$k = 1$	$k = 2$	$k = 3$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	PSO	PSO-ASA
0.1	$1.35E - 08$	$8.00E - 10$	$8.00E - 10$	$7.96E - 10$	$8.00E - 10$	$8.00E - 10$	$8.00E - 10$	$1.05E - 03$	$5.01E - 06$
0.2	$1.85E - 06$	$1.21E - 09$	$1.19E - 09$	$4.88E - 09$	$1.19E - 09$	$1.19E - 09$	$1.19E - 09$	$8.05E - 04$	$1.73E - 06$
0.3	$3.20E - 05$	$9.34E - 09$	$5.85E - 09$	$2.22E - 07$	$6.96E - 09$	$5.62E - 09$	$5.61E - 09$	$6.71E - 04$	$8.38E - 07$
0.4	$2.45E - 04$	$1.44E - 07$	$2.11E - 08$	$3.94E - 06$	$6.87E - 08$	$1.12E - 08$	$1.04E - 08$	$6.39E - 04$	$9.98E - 07$
0.5	$1.20E - 03$	$2.23E - 06$	$2.30E - 07$	$3.79E - 05$	$1.12E - 06$	$5.31E - 08$	$2.34E - 08$	$6.79E - 04$	$1.67E - 06$
0.6	$4.50E - 03$	$2.23E - 05$	$2.47E - 06$	$2.45E - 04$	$1.24E - 05$	$6.38E - 07$	$7.10E - 08$	$7.72E - 04$	$5.17E - 06$
0.7	$1.40E - 02$	$1.62E - 04$	$2.03E - 05$	$1.21E - 03$	$9.75E - 05$	$7.55E - 06$	$6.33E - 07$	$9.10E - 04$	$4.77E - 06$
0.8	$3.84E - 02$	$9.20E - 04$	$1.34E - 04$	$4.97E - 03$	$5.98E - 04$	$6.89E - 05$	$7.82E - 06$	$1.07E - 03$	$9.40E - 06$
0.9	$9.63E - 02$	$4.39E - 03$	$7.46E - 04$	$1.78E - 02$	$3.05E - 03$	$5.02E - 04$	$8.06E - 05$	$1.29E - 03$	$6.62E - 06$
1.0	$2.27E - 01$	$1.84E - 02$	$3.71E - 03$	$5.74E - 02$	$1.36E - 02$	$3.07E - 03$	$6.80E - 04$	$1.98E - 03$	$7.58E - 05$
Mean	$3.82E - 02$	$2.39E - 03$	$4.61E - 04$	$8.16E - 03$	$1.73E - 03$	$3.65E - 04$	$7.69E - 05$	$9.86E - 04$	$1.12E - 05$

TABLE 5: Statistical analysis of results based on values of absolute error.

t	PSO				PSO-ASA			
	MIN	MAX	MEAN	STD	MIN	MAX	MEAN	STD
0.1	1.0506E-03	2.2340E-01	9.2274E-02	5.1432E-02	3.1813E-08	2.1601E-03	9.6880E-05	2.7723E-04
0.2	8.0518E-04	2.4845E-01	1.0033E-01	5.5966E-02	3.4372E-08	2.2102E-03	9.7567E-05	2.7860E-04
0.3	6.7109E-04	2.7628E-01	1.1029E-01	6.2147E-02	8.3812E-07	2.1940E-03	1.0290E-04	2.8024E-04
0.4	6.3908E-04	3.2281E-01	1.2356E-01	7.0441E-02	1.2269E-07	2.3203E-03	1.1739E-04	3.0450E-04
0.5	6.7853E-04	4.2304E-01	1.4202E-01	8.1644E-02	3.0549E-07	2.7264E-03	1.3725E-04	3.5327E-04
0.6	7.7246E-04	5.4096E-01	1.6843E-01	9.7069E-02	9.4072E-07	3.3844E-03	1.6471E-04	4.2624E-04
0.7	9.0997E-04	6.8783E-01	2.0732E-01	1.1879E-01	4.7794E-08	4.2143E-03	2.1027E-04	5.3421E-04
0.8	1.0663E-03	8.8150E-01	2.6607E-01	1.5005E-01	5.0503E-07	5.4046E-03	2.7398E-04	6.8757E-04
0.9	1.2908E-03	1.1516E+00	3.5688E-01	1.9625E-01	4.0723E-06	7.5318E-03	3.8273E-04	9.5844E-04
1.0	1.9755E-03	1.5506E+00	5.0247E-01	2.6814E-01	2.9828E-06	1.0874E-02	5.6390E-04	1.3640E-03

TABLE 6: Comparative analysis of the results.

Parameters	$h = 0.1$		$h = 0.05$	
	PSO	PSO-ASA	PSO	PSO-ASA
GMAE	1.9576E-01	2.0372E-04	1.8444E-01	6.3211E-04
MF	1.1837E-02	4.6577E-06	1.0804E-02	7.1294E-06
MAE $\leq 10^{-02}$	21%	100%	22%	100%
MAE $\leq 10^{-03}$	01%	100%	00%	099%
MAE $\leq 10^{-04}$	00%	095%	00%	083%
FA $\leq 10^{-03}$	50%	100%	55%	100%
FA $\leq 10^{-04}$	05%	100%	05%	100%
FA $\leq 10^{-05}$	01%	099%	00%	098%
MIN-ET	075.23 s	079.29 s	132.42 s	157.14 s
MAX-ET	103.15 s	113.74 s	145.67 s	191.32 s
M-ET	076.93 s	092.17 s	142.43 s	183.57 s
STD-ET	002.93 s	007.32 s	002.02 s	007.62 s

number of grid points, and convergence rate for PSO and PSO-ASA proposed solvers.

The effects on results by changing the number of grid points used in fitness function (19) are analyzed. By taking the number of grid points 21 instead of 11, the new fitness function is formulated using the input of the training set $t \in (0, 1)$ with a step of $h = 0.05$ and given as

$$e = \frac{1}{21} \sum_{m=0}^{20} \left(\hat{y}_m'' - 6\hat{y}_m^2 - t_m \right)^2 + \frac{1}{2} \left(\hat{y}_0^2 + (\hat{y}_0' - 1)^2 \right) \Bigg|_j, \quad j = 1, 2, 3, \dots, 2000. \quad (20)$$

Hundred independents runs of PSO and PSO-ASA method are executed for finding the appropriate weights and subsequently the solution of IVP of PE-1. The MFA is defined as average values of fitness achieved taken over all independent runs of the algorithm. Similarly, GMAE is average value of mean absolute error for all independents runs of solvers. The values of MFA and GMAE are calculated

for PSO and PSO-ASA techniques using both 11 and 21 grids points and results are tabulated in Table 6. It is found that the values of MFA are of the order 10^{-02} and 10^{-06} for PSO and PSO-ASA, respectively. The values of MAE are matched with MATHEMATICA solution $y_M(t)$ upto 2 to 3 and 5 to 8 decimal places for PSO and PSO-ASA techniques. The value of GMAE for 11 and 21 grids points are determined and given in Table 6. It is found that values of GMAE are of the order 10^{-04} for both the cases for inputs between 0 and 1 with a step of $h = 0.1$. So, by increasing number of grid points the same level of the accuracy in results is maintained for both PSO and PSO-ASA techniques. The increase in number of grids points, that is, decrease in the mesh size h , does not guaranty improvement of results for stochastic solvers and same is observed in our simulations by changing the value of step size $h = 0.1$ to $h = 0.05$.

The convergence rates of the proposed stochastic solvers are determined based on the value of MAE $\leq 10^{-02}$, 10^{-03} , and 10^{-04} , as well as the value of MFA $e \leq 10^{-03}$, 10^{-04} , and 10^{-05} . Results are tabulated in Table 6 for both PSO and PSO-ASA techniques. It can be seen that, with the acceptability

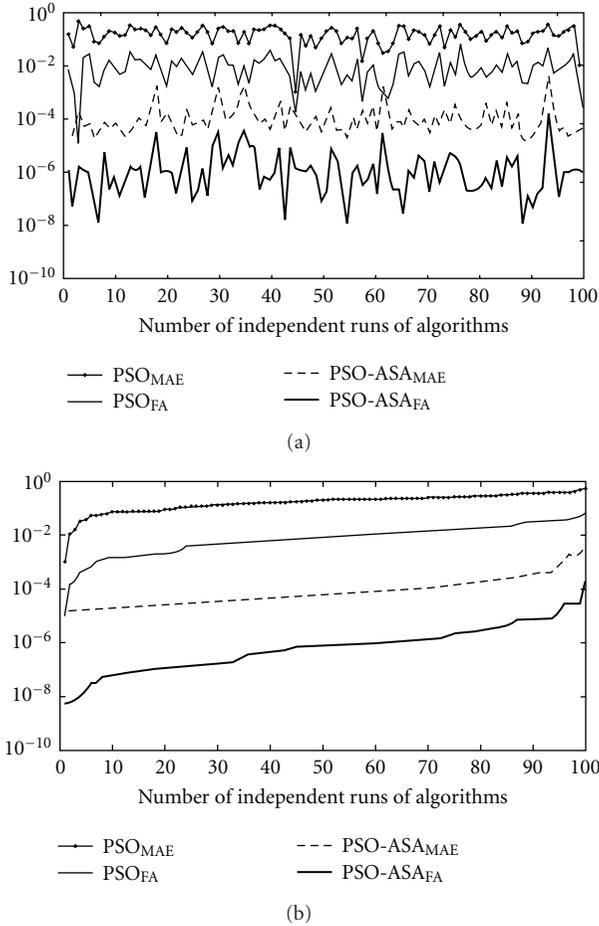


FIGURE 3: Comparison of results on the basis of fitness achieved (FA) and mean absolute error (MAE).

of the results based on value of MFA $e \leq 10^{-05}$, the hybrid approach PSO-ASA achieved this criteria for 99% and 98% of the independents runs for 11 and 21 numbers of grid points, respectively. In addition to that, results are consistently convergent for larger number of independents runs for PSO-ASA method than that of PSO algorithms.

Moreover, the computational complexity is analyzed by calculating the time of execution of each algorithm for 100 numbers of independent runs for both $h = 0.1$ and $h = 0.05$. The values for minimum execution time (MIN-ET), maximum execution time (MAX-ET), mean execution time (M-ET), and standard deviation of execution time (STD-ET) are provided in the Table 6. It can be seen that values of M-ET for PSO and PSO-ASA techniques are 76.93 seconds and 92.17 seconds, respectively, for $h = 0.1$, and almost double values of M-ET in case for $h = 0.05$. The time taken for the execution for hybrid approach PSO-ASA is relatively greater than that of PSO algorithm, but this can be overshadow by its invariable dominance in accuracy and convergence. The time analysis provided in this article is carried out using Dell Latitude D630 laptop with Intel(R) Core(TM) 2 Duo CPU T9300, 2.50 GHz, and running with MATLAB version R2009b.

6. Conclusion

On the basis of the simulations and their comparative studies provided in the last sections, it can be concluded that the IVP of PE-I can be solved effectively by the designed computational intelligence algorithms using neural networks supported with PSO and ASA techniques. Results obtained using the weights for DE-NN networks of PE-I trained by PSO-ASA algorithm are found to be better than from PSO techniques. The values of absolute error from MATHEMATICA solution $y_M(t)$ of PSO-ASA lies in the range 10^{-04} to 10^{-07} and match upto 5 to 7 decimal places with state of art analytical techniques of VIM and HPM.

On the basis of 100 independent runs for each algorithm to solve PE-I using 11 and 21 grid points, the PSO-ASA technique is found to be invariable superior than PSO algorithm using the criteria based on the values of mean absolute error, global mean absolute error, mean fitness achieved, and convergence rate. However, as far as computational complexity is concerned that PSO-ASA technique is slightly more computational exhaustive, but it can be overlook due to its overall dominance in the performance. This study can also be extended to use other biological inspired computational algorithms for solving the PE-I, as well as other Painlevé transcendentals.

Acknowledgment

The authors gratefully acknowledge Professor Dr. Adiq Kausar Kiani for providing her valuable suggestions to improve the presentation of the paper.

References

- [1] A. P. Bassom, P. A. Clarkson, and A. C. Hicks, "Numerical studies of the fourth Painlevé equation," *IMA Journal of Applied Mathematics*, vol. 50, no. 2, pp. 167–139, 1993.
- [2] J. He, "Variational iteration method—a kind of non-linear analytical technique: some examples," *International Journal of Non-Linear Mechanics*, vol. 34, no. 4, pp. 699–708, 1999.
- [3] J. H. He, "Some asymptotic methods for strongly nonlinear equations," *International Journal of Modern Physics B*, vol. 20, no. 10, pp. 1141–1199, 2006.
- [4] M. J. Ablowitz and H. Segur, *Solitons and the Inverse Scattering Transform*, Studies in Applied Mathematics 4, SIAM, Philadelphia, Pa, USA, 1981.
- [5] M. Tajiri and S. Kawamoto, "Reduction of KdV and cylindrical KdV equations to Painlevé equation," *Journal of the Physical Society of Japan*, vol. 51, no. 5, pp. 1678–1681, 1982.
- [6] R. Haberman, "Slowly varying jump and transition phenomena associated with algebraic bifurcation problems," *SIAM Journal on Applied Mathematics*, vol. 37, no. 1, pp. 69–106, 1979.
- [7] A. S. Fokas, A. R. Its, and A. V. Kitaev, "Discrete Painlevé equations and their appearance in quantum gravity," *Communications in Mathematical Physics*, vol. 142, no. 2, pp. 313–344, 1991.
- [8] A. S. Fokas, A. R. Its, and A. V. Kitaev, "Matrix models of two-dimensional quantum gravity and isomonodromic solutions of "Discrete Painlevé" equations," *Journal of Mathematical Sciences*, vol. 73, no. 4, pp. 415–429, 1995.

- [9] M. J. Ablowitz and P. A. Clarkson, *Solitons, Nonlinear Evolution Equation and Inverse Scattering*, Cambridge University Press, 1991.
- [10] V. I. Gromak, I. Laine, and S. Shimomura, *Painleve Differential Equations in the Complex Plane*, Walter de Gruyter, New York, NY, USA, 2002.
- [11] D. Dai and L. Zhang, "On tronquée solutions of the first Painlevé hierarchy," *Journal of Mathematical Analysis and Applications*, vol. 368, no. 2, pp. 393–399, 2010.
- [12] A. B. O. Daalhuis, "Hyperasymptotics for nonlinear odes ii. The first Painlevé equation and a second-order riccati equation," *Proceedings of the Royal Society A*, vol. 461, no. 2062, pp. 3005–3021, 2005.
- [13] S. Y. Lee, R. Teodorescu, and P. Wiegmann, "Viscous shocks in hele-shaw flow and stokes phenomena of the Painlevé I transcendent," *Physica D*, vol. 240, no. 13, pp. 1080–1091, 2011.
- [14] N. A. Kudryashov, "The first and second Painlevé equations of higher order and some relations between them," *Physics Letters, Section A*, vol. 224, no. 6, pp. 353–360, 1997.
- [15] N. A. Kudryashov, "Amalgamations of the Painlevé equations," *Journal of Mathematical Physics*, vol. 44, no. 12, pp. 6160–6178, 2003.
- [16] G. Cresswell and N. Joshi, "The discrete first, second and thirty-fourth Painlevé hierarchies," *Journal of Physics A*, vol. 32, no. 4, pp. 655–669, 1999.
- [17] P. R. Gordoa and A. Pickering, "Nonisospectral scattering problems: a key to integrable hierarchies," *Journal of Mathematical Physics*, vol. 40, no. 11, pp. 5749–5786, 1999.
- [18] U. Muğan and F. Jrad, "Painlevé test and the first Painlevé hierarchy," *Journal of Physics A*, vol. 32, no. 45, pp. 7933–7952, 1999.
- [19] U. Mugan and F. Jrad, "Painlevé test and higher order differential equations," *Journal of Nonlinear Mathematical Physics*, vol. 9, no. 3, pp. 282–310, 2002.
- [20] B. Fornberg and J. A. C. Weideman, "A Numerical methodology for the Painlevé equations," *Journal of Computational Physics*, vol. 230, no. 15, pp. 5957–5973, 2011.
- [21] N. A. Kudryashov, "Fourth-order analogies to the Painlevé equations," *Journal of Physics A*, vol. 35, no. 21, pp. 4617–4632, 2002.
- [22] N. A. Kudryashov, "One method for finding exact solutions of nonlinear differential equations," *Communications in Nonlinear Science and Numerical Simulations*, vol. 17, no. 6, pp. 2248–2253, 2012.
- [23] H. Saidi, N. Khelil, S. Hassouni, and A. Zerarka, "Energy spectra of the Schrödinger equation and the differential quadrature method: improvement of the solution using particle swarm optimization," *Applied Mathematics and Computation*, vol. 182, no. 1, pp. 559–566, 2006.
- [24] Z. Y. Lee, "Method of bilaterally bounded to solution blasius equation using particle swarm optimization," *Applied Mathematics and Computation*, vol. 179, no. 2, pp. 779–786, 2006.
- [25] M. A. Z. Raja, I. M. Qureshi, and J. A. Khan, "Swarm Intelligent optimized neural networks for solving fractional differential equations," *International Journal of Innovative Computing*, vol. 7, no. 11, pp. 6301–6318, 2011.
- [26] M. A. Z. Raja, J. A. Khan, and I. M. Qureshi, "Heuristic computational approach using swarm intelligence in solving fractional differential equations," in *Proceedings of the 12th Annual Genetic and Evolutionary Computation Conference (GECCO'10)*, pp. 2023–2026, Portland, Oregon, July 2010.
- [27] M. A. Z. Raja, J. A. Khan, and I. M. Qureshi, "A new stochastic approach for solution of riccati differential equation of fractional order," *Annals of Mathematics and Artificial Intelligence*, vol. 60, no. 3, pp. 229–250, 2010.
- [28] M. A. Z. Raja, J. A. Khan, and I. M. Qureshi, "Swarm intelligence optimized neural network for solving fractional order system of bagely-tervik equation," *Engineering Intelligent System*, vol. 19, no. 1, pp. 41–51, 2011.
- [29] E. Hesameddini and A. Peyrovi, "The use of variational iteration method and homotopy perturbation method for Painlevé equation I," *Applied Mathematical Sciences*, vol. 3, no. 37–40, pp. 1861–1871, 2009.
- [30] S. S. Behzadi, "Convergence of iterative methods for solving Painlevé equation," *Applied Mathematical Sciences*, vol. 4, no. 30, pp. 1489–1507, 2010.
- [31] D. R. Parisi, M. C. Mariani, and M. A. Laborde, "Solving differential equations with unsupervised neural networks," *Chemical Engineering and Processing: Process Intensification*, vol. 42, no. 8-9, pp. 715–721, 2003.
- [32] Y. Shirvany, M. Hayati, and R. Moradian, "Multilayer perceptron neural networks with novel unsupervised training method for numerical solution of the partial differential equations," *Applied Soft Computing Journal*, vol. 9, no. 1, pp. 20–29, 2009.
- [33] J. A. Khan, M. A. Z. Raja, and I. M. Qureshi, "Numerical treatment of nonlinear emden-fowler equation using stochastic technique," *Annals of Mathematics and Artificial Intelligence*, vol. 63, no. 2, pp. 185–207, 2011.
- [34] J. A. Khan, M. A. Z. Raja, and I. M. Qureshi, "Novel approach for van der pol oscillator on the continuous time domain," *Chinese Physics Letters*, vol. 28, no. 11, Article ID 110205, 2011.
- [35] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, Perth, Australia, 1995.
- [36] S. N. Sivanandam and P. Visalakshi, "Multiprocessor scheduling using hybrid particle swarm optimization with dynamically varying inertia," *International Journal of Computer Science & Applications*, vol. 4, no. 3, pp. 95–106, 2007.
- [37] G. D. Li, S. Masuda, D. Yamaguchi, and M. Nagai, "The optimal GNN-PID control system using particle swarm optimization algorithm," *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 10, pp. 3457–3469, 2009.
- [38] R. D. A. Araújo, "Swarm-based translation-invariant morphological prediction method for financial time series forecasting," *Information Sciences*, vol. 180, no. 24, pp. 4784–4805, 2010.
- [39] X. Li and J. Wang, "A steganographic method based upon JPEG and particle swarm optimization algorithm," *Information Sciences*, vol. 177, no. 15, pp. 3099–3109, 2007.
- [40] R. Ellahi, S. Abbasbandy, T. Hayat, and A. Zeeshan, "On comparison of series and numerical solutions for second Painlevé equation," *Numerical Methods for Partial Differential Equations*, vol. 26, no. 5, pp. 1070–1078, 2010.
- [41] M. Dehghan and F. Shakeri, "The numerical solution of the second Painlevé equation," *Numerical Methods for Partial Differential Equations*, vol. 25, no. 5, pp. 1238–1259, 2009.
- [42] E. Hesameddini, "Homotopy perturbation method for second Painlevé equation and comparisons with analytic continuation extension and chebishev series method," *International Mathematical Forum*, vol. 5, no. 13, pp. 629–637, 2010.
- [43] J. H. He, "A short remark on fractional variational iteration method," *Physics Letters*, vol. 375, no. 38, pp. 3362–3364, 2011.
- [44] M. Dehghan, S. A. Yousefi, and A. Lotfi, "The use of He's variational iteration method for solving the telegraph and fractional telegraph equations," *International Journal for*

- Numerical Methods in Biomedical Engineering*, vol. 27, no. 2, pp. 219–231, 2011.
- [45] A. M. Wazwaz and R. Rach, “Comparison of the adomian decomposition method and the variational iteration method for solving the lane-Emden equations of the first and second kinds,” *Kybernetes*, vol. 40, no. 9, pp. 1305–1318, 2011.
- [46] J. H. He, “Notes on the optimal variational iteration method,” *Applied Mathematics Letters*, vol. 25, no. 10, pp. 1579–1581, 2012.
- [47] J. H. He, “Homotopy perturbation method with an auxiliary term,” *Abstract and Applied Analysis*, vol. 2012, Article ID 857612, 7 pages, 2012.
- [48] H. E. Ji-Huan, “A note on the homotopy perturbation method,” *Thermal Science*, vol. 14, no. 2, pp. 565–568, 2010.
- [49] T. Öziş and A. Yildirim, “Comparison between Adomian’s method and He’s homotopy perturbation method,” *Computers and Mathematics with Applications*, vol. 56, no. 5, pp. 1216–1224, 2008.
- [50] X. Feng, Y. He, and J. Meng, “Application of homotopy perturbation method to the Bratu-type equations,” *Topological Methods in Nonlinear Analysis*, vol. 31, no. 2, pp. 243–252, 2008.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

