

Nature-inspired computational intelligence integration with Nelder–Mead method to solve nonlinear benchmark models

Muhammad Asif Zahoor Raja¹  · Aneela Zameer² · Adiq Kausar Kiani³ · Azam Shehzad⁴ · Muhammad Abdul Rehman Khan¹

Received: 17 January 2016 / Accepted: 8 August 2016 / Published online: 17 August 2016
© The Natural Computing Applications Forum 2016

Abstract In the present study, nature-inspired computing technique has been designed for the solution of nonlinear systems by exploiting the strength of particle swarm optimization (PSO) hybrid with Nelder–Mead method (NMM). Fitness function based on least square approximation theory is developed for the systems, while optimization of the design variables is performed with PSO, an efficient global search method, refined with NMM for rapid local convergence. Sixteen variants of the proposed hybrid scheme PSO-NMM have been evaluated on five benchmark nonlinear systems, namely interval arithmetic benchmark model, kinematic application model, neurophysiology problem, combustion model and chemical

equilibrium system. Reliability and effectiveness of the proposed solver have been validated after comparison with the results of statistical analysis based on massive data generated for sufficiently large number of independent executions.

Keywords Nonlinear system of equations · Hybrid computing · Particle swarm optimization · Nelder–Mead method · Nature-inspired computing · Benchmark models

Mathematics Subject Classification 65H10 · 65H20 · 65C20

✉ Muhammad Asif Zahoor Raja
rasifzahoor@yahoo.com;
Muhammad.asif@ciit-attock.edu.pk

Aneela Zameer
aneelas@gmail.com; aneelaz@pieas.edu.pk

Adiq Kausar Kiani
adiqakian@gmail.com

Azam Shehzad
azamshehzad@yahoo.com

Muhammad Abdul Rehman Khan
ma.rehman@ciit-attock.edu.pk

¹ Department of Electrical Engineering, COMSATs Institute of Information Technology, Attock Campus, Attock, Pakistan

² Department of Computer and Information Sciences, Pakistan Institute of Engineering and Applied Sciences (PIEAS), Nilore, Islamabad 45650, Pakistan

³ Department of Economics, Federal Urdu University of Arts Science and Technology, Islamabad, Pakistan

⁴ Department of Mathematics, Preston University, Kohat, Islamabad Campus, Islamabad, Pakistan

1 Introduction

Strength of artificial intelligence algorithms (AIAs) has extensively been exploited to optimize the difficult nonlinear problems arising in a variety of fields [1–3]. Especially in computational and applied mathematics, AIAs based on unsupervised feed-forward artificial neural networks (ANNs) trained with global and local search methodologies have been applied to solve many linear and nonlinear systems. Few potential applications of AIA using ANNs, evolutionary and swarming intelligence, are highly stiff oscillatory systems based on Van der Pol-type nonlinear differential equations [4], mathematical model for fuel ignition model in combustion theory through one-dimensional Bratu-type equations [5], transformed third-order BVP of nonlinear two-dimensional Bratu-type equations [6], the maximum flow problem [7], similarity transformed system for thin film flow of third-grade fluids [8], nonlinear Lane–Emden-type equations [9], mathematical model for confinement of plasma represented with Troesch-type problems [10], magnetohydrodynamics

(MHD) problem of fluid flow governed with Jeffery–Hamel-type equations [11], nonlinear Emden–Fowler type singular system [12], nonlinear first Painlevé-type equations [13, 14], nonlinear systems based on initial value problems [15], functional ordinary differential equations of pantograph types [16], fractional differential equations of nonlinear Riccati types [17], nanotechnology problems based on multi-wall carbon nanotubes [18] and fluid mechanic problem based on steady thin film flow of Johnson–Segalman fluid on vertical cylinder [19].

A number of numerical solvers have been proposed by the research community for solving problems based on nonlinear system of equations [20–22], but one of the simplest, oldest and, most commonly used numerical procedures is Newton–Raphson method (NRM) [23, 24], but its performance is sensitive to the initial weights as it is highly prone to getting stuck in the local minimum. Therefore, in order to utilize NRM algorithm effectively for the solution of system of nonlinear equations, it is supplied with biased initial guess by another global search solver. Besides NRM, many other iterative and recursive solvers have been introduced and used in applied mathematics, literature due to their unique advantages, limitations and applicability. For instance, Kelley, Campbell and Broyden’s provided numerical schemes for the solution of nonlinear equations and their systems [25–27]. Jacobian-free Newton–Krylov method has been used for deriving the lower triangular half of the sparse Jacobian matrix through automatic differentiation [28, 29]. Recently, Jafari and Gejji, Abbasbandy, Sharma and Vahidi have provided new solver [30–36] to effectively determine the solution of problems based on nonlinear equations. Most of the solutions available in the existing literature are based on deterministic procedure using iterative and recursive solvers [37–39], but well-established strength of the stochastic solvers, based on AIAs, has not been extensively explored for the solution of nonlinear equations. The aim of this study is to investigate the potential of AIA algorithms based on NIC technique for providing an alternate, accurate, convergent and efficient system for solving system of nonlinear equations.

In this study, NIC technique based on swarm intelligence aided with NMM is applied for solving nonlinear systems. Sixteen variants of particle swarm optimization (PSO) algorithm have been developed and applied to optimize the variables for five well-known nonlinear systems. The design parameters of these systems are further tuned with NMM for rapid local convergence. The salient features of the proposed methodologies are given briefly as:

- Novel design of nature-inspired heuristic algorithms based on exploration and exploitation in standard PSO method hybrid with NMM to solve nonlinear systems.

- Performance of the proposed schemes is evaluated on five benchmark models based on interval arithmetic benchmark, kinematic application, neurophysiology, combustion and chemical equilibrium models.
- Reliability and effectiveness of the schemes are analyzed through the results of statistical analyses based on sufficiently large number of independent runs.
- Computational complexity indices are used to measure the comparative efficiency.
- Simple concept, effortlessness in implementations, consistent accuracy, steady convergence and wider applicability domain are the significant illustrative perks of the schemes.

The rest of the paper is organized as follows: In section two, design methodology for solving nonlinear equations and their systems is presented, i.e., formulation of the objective function and learning methodologies used for training. In section three, results of simulation are presented and compared with the studies based on statistical performance indices for accuracy and complexity. Conclusions and potential future research directions are submitted in the last section.

2 Methodology

Design scheme presented here for finding the solution of systems of nonlinear equations consists of two parts: In the first part, formulation of objective function based on the residual error in absolute or the mean square sense is formulated, while in the second part, details of PSO and its variants are outlined. The graphical abstract of the proposed methodology is shown in Fig. 1.

2.1 Fitness function for systems nonlinear of equations

General form of system of nonlinear equations is written as:

$$\mathbf{X}(\mathbf{y}) = \mathbf{0}. \quad (1)$$

Equation (1) can be written in the expanded form as:

$$\mathbf{X}(\mathbf{y}) = \mathbf{0}, \Rightarrow \begin{bmatrix} x_1(\mathbf{y}) \\ x_2(\mathbf{y}) \\ \vdots \\ x_n(\mathbf{y}) \end{bmatrix} = \mathbf{0}, \quad (2)$$

$$\Rightarrow \begin{cases} x_1(y_1, y_2, y_3, \dots, y_n) = 0, \\ x_2(y_1, y_2, y_3, \dots, y_n) = 0, \\ \vdots \\ x_n(y_1, y_2, y_3, \dots, y_n) = 0. \end{cases}$$

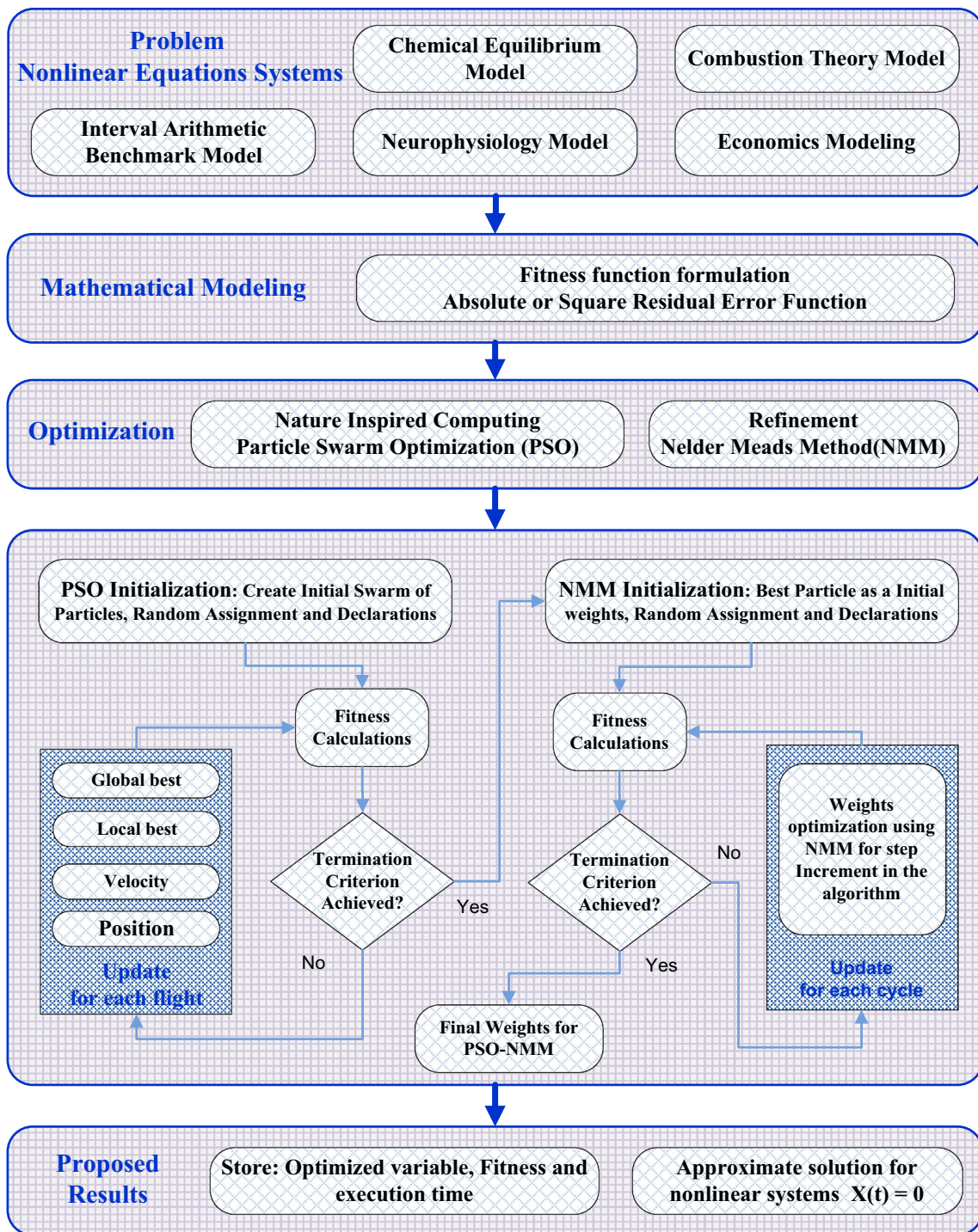


Fig. 1 Graphical view of proposed design methodology

To determine the solution for a system (1) with n number of unknowns and n number of equations, objective function based on the residual error in mean absolute or square sense can be written as [37, 38]:

$$O_{ir} = \begin{cases} \frac{1}{n} \sum_{i=1}^n |r_i|, & \text{or} \\ \frac{1}{n} \sum_{i=1}^n r_i^2. \end{cases} \quad (3)$$

where (r_1, r_2, \dots, r_n) are the residual error functions associated with (x_1, x_1, \dots, x_n) equations, respectively, and can be written as:

$$\mathbf{r} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} = \begin{bmatrix} x_1(y_1, y_2, y_3, \dots, y_n) \\ x_2(y_1, y_2, y_3, \dots, y_n) \\ \vdots \\ x_n(y_1, y_2, y_3, \dots, y_n) \end{bmatrix}. \quad (4)$$

Now objective is to determine the variables of system (1), such that the residual error function (3) $\text{Oir} \rightarrow 0$, which is only possible when each of $\mathbf{r} = [r_1, r_2, \dots, r_n] \rightarrow 0$, consequently $\mathbf{X}(\mathbf{y}) \rightarrow \mathbf{0}$, i.e., values of the vector $\mathbf{y} = [y_1, y_2, y_3, \dots, y_n]$ will be the approximate solution for the system.

2.2 Optimization of variables

The procedure to use NIC technique based on variants of PSO for the optimization of residual error is elaborated in this section. PSO algorithms, introduced by Kennedy and Eberhart [40], belong to the class of NIC and have been developed using mathematical model of bird flocking and fish schooling. Reliable and efficient exploration and exploitation of candidate solutions is a well-established strength of PSO algorithm, and its convergence and stability are well-proven facts on number of benchmark problems; please see [41–45] and references therein. Numerous discrete and continuous types of the PSO algorithms have been used for optimization in many applications such as adaptive IIR system identification [46], text feature selection [47], product quality estimation [48], nonlinear parameter estimation [49] and optimization of fractional order PID controller [50].

Following the standard procedure of PSO algorithm, a candidate solution of the problem-specific objective function, known as a particle, is globally searched in the entire space. The population of these candidate solutions or particles is named as a ‘swarm’. All particles of the swarm have their own role to find a better solution of the problem-specific objective function. The particles are initialized randomly, and then, they participate in the process of optimization as a set of particle, i.e., population or swarm. Update in the position and velocity of every particle is made during each flight or iteration taking into account of their previous local $\mathbf{P}_{\text{Lbest}}^{t-1}$ and global $\mathbf{P}_{\text{Gbest}}^{t-1}$ best positions. The mathematical structure of standard PSO in terms of velocity and updated positions’ equations is written as:

$$\mathbf{v}_i^t = w\mathbf{v}_i^{t-1} + a_1\mathbf{rand}_1(\mathbf{P}_{\text{Lbest}}^{t-1} - \mathbf{z}_i^{t-1}) + a_2\mathbf{rand}_2(\mathbf{P}_{\text{Gbest}}^{t-1} - \mathbf{z}_i^{t-1}), \quad (5)$$

$$\mathbf{z}_i^t = \mathbf{z}_i^{t-1} + \mathbf{v}_i^t, \quad (6)$$

where the vector \mathbf{z}_i is the i th particle of the swarm \mathbf{Z} , i.e., $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_m]$, and m integer denotes the total number of particles. Velocity vector (\mathbf{v}_i) is associated with respective i th \mathbf{z}_i particle, w is the inertia weight, a_1 is a local acceleration factor, a_2 is the global acceleration constant, while \mathbf{rand}_1 and \mathbf{rand}_2 are random vectors with elements bounded between 0 and 1. The values of constants w , a_1 and a_2 are linearly increasing or decreasing over the course of the search between 0 and 1. The elements of velocity are limited to its minimum (MIN) and maximum (MAX) values, i.e., $v_i \in [-v_{\text{max}}, v_{\text{max}}]$.

Power of updating weights with PSO decreases considerably with the increase in the number of flights of the swarm; therefore, the real strength of this global optimizer can be seen by the process of hybridization with the local search algorithm. Nelder–Mead method (NMM) is an efficient local search algorithm incorporated for further refinement of the results by taking the global best particles of PSO as the starting point of the NMM algorithm. The flow diagram of the proposed mechanism based on PSO–NMM for optimization of residual error function is shown in Fig. 1, while the details about the intermediate steps are as follows:

Step 1: Initialization of PSO Initialize the particle with randomly assigned bounded real entries equal to the number of unknown variables of nonlinear systems. Mathematical notation of the particle is given as:

$$\mathbf{z} = [y_1, y_2, y_3, \dots, y_n]$$

where member of the particle represents the solution for the system. The set of particles m , i.e., swarm \mathbf{Z} , is written as:

$$\mathbf{Z} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \vdots \\ \mathbf{z}_m \end{bmatrix} = \begin{bmatrix} y_1, y_2, y_3, \dots, y_n \\ y_1, y_2, y_3, \dots, y_n \\ \vdots \\ y_1, y_2, y_3, \dots, y_n \end{bmatrix}.$$

Sixteen variants of PSO algorithm are formulated for different values assigned to m , i.e., number of particles in the swarm and the number of flights a swarm executed. The variants of PSO algorithms are listed in Table 1. The fixed parameters initialized to each variant of PSO algorithm are listed in Table 2. These parameters are set carefully after extensive numerical experimentation to avoid premature convergence.

Step 2: Fitness evaluation Calculate the value of residual error for each particle by using the relationship given in Eq. (3).

Table 1 List of PSO variants hybrid with NMMs

Method	Parameters values		Method	Parameters values	
	Flights	Particles		Flights	Particles
PSO-NMM-1	50	25	PSO-NMM-9	250	25
PSO-NMM-2	50	50	PSO-NMM-10	250	50
PSO-NMM-3	50	100	PSO-NMM-11	250	100
PSO-NMM-4	50	200	PSO-NMM-12	250	200
PSO-NMM-5	100	25	PSO-NMM-13	500	25
PSO-NMM-6	100	50	PSO-NMM-14	500	50
PSO-NMM-7	100	100	PSO-NMM-15	500	100
PSO-NMM-8	100	200	PSO-NMM-16	500	200

Table 2 Parameters’ settings for the optimization algorithms

Method	Parameters	Settings	Parameters	Settings
PSO	Particle size: optimization variables	10, 5, 6, 10 and 5 for P-1 to P-5, respectively	Local acceleration c_1	Linear decreasing (2.5–0.5)
	V_{max}	02	Global acceleration c_2	Linear increasing (0.5–2.5)
	Function tolerance ‘Tolfun’	0	Inertia ω	Linearly decreasing (0.9–0.4)
NMM	Initial weights	Best particles of PSO	X-tolerance ‘TolX’	0
	‘MaxIter’	10,000	Velocity Span	(–2, 2)
	‘MaxFunEvals’	200,000	Other	Defaults

Step 3: *Ranking* Rank each particle through the MIN values of the residual error function, and the particle with smaller values is ranked higher and vice versa.

Step 4: *Termination* Terminate the executions of PSO variants for the following conditions:

- Predefined fitness value is achieved, e.g., $O_{ir} \leq 10^{-15}$
- A MAX number of cycles or flights are executed. If above mentioned criteria are fulfilled, then proceed to step 6; otherwise, continue.

Step 5: *Renewal* Renew the velocity and position of the particles using Eqs. (5) and (6), respectively. Proceed to step 2 for the next flight of the swarm Z .

Step 6: *Fine tuning*: MATLAB built-in routine for optimization through ‘fminsearch’ function is utilized for simplex approach based on Nelder–Meads method. The refinement of the parameters is attained by taking best particle of PSO variants as initial weights, while other values of parameters are set as per values given in Table 2.

The theory, underlying concepts, mathematical description and applications of NMM can be seen in [51, 52].

Step 7: *Storage* Values of the best global of PSO and PSO-NMM along with their corresponding residual error functions and time consumed are stored for the execution of the algorithm.

Step 8: *Statistical analysis* To generate a large set of data for reliable statistical analysis, repeat the steps 1–7 of the PSO-NMM procedure to form a sufficiently large number of independent executions.

3 Numerical experimentation

Results of sixteen variants of PSO and PSO-NMMs to find the solution of systems of nonlinear equations for five different problems are presented here. Comprehensive numerical experimentation for each case study is carried out to check the effectiveness of the proposed schemes in comparison with the results of statistical analysis.

3.1 Problem 1: interval arithmetic benchmark model

arithmetic benchmark (IAB) model. Mathematical expressions of nonlinear equations of the IAB model [53–56] can be written as:

In this study, the performance of the design solver is examined for nonlinear system of equations of interval

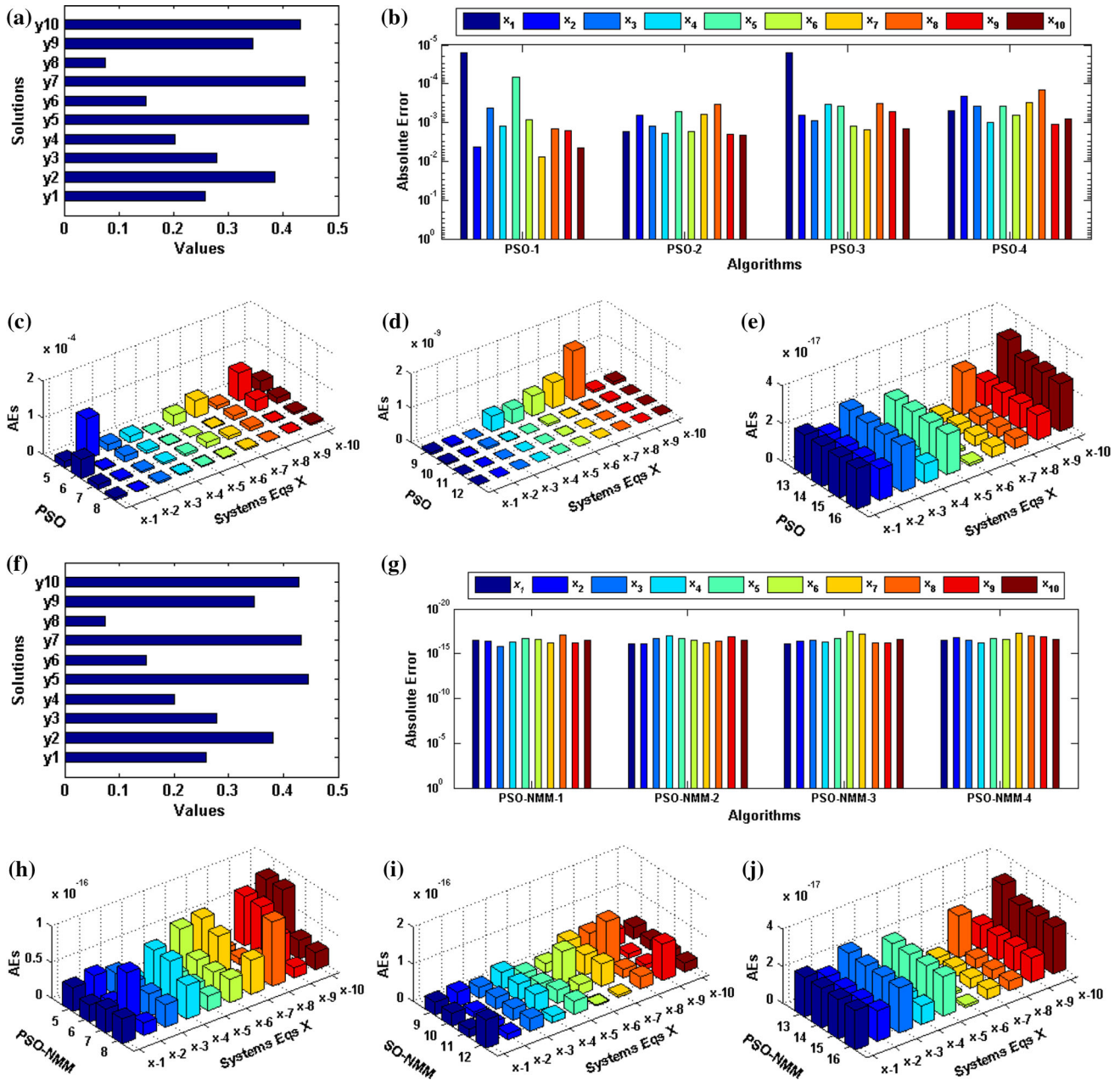


Fig. 2 Proposed results of PSO and PSO-NMMs for interval arithmetic benchmark model presented in problem 1. **a** Approximate solution for PSO-1. **b** Absolute errors (AEs) for constituents equations of the system. **c** AEs for PSO-5 to PSO-8. **d** AEs for PSO-9 to PSO-12. **e** AEs for PSO-13 to PSO-16. **f** Approximate solution for PSO-

NMM-1. **g** Absolute errors (AEs) for constituents equations of the system. **h** AEs for PSO-NMM-5 to PSO-NMM-8. **i** AEs for PSO-NMM-9 to PSO-NMM-12. **j** AEs for PSO-NMM-13 to PSO-NMM-16

$$\begin{bmatrix} x_1(\mathbf{y}) \\ x_2(\mathbf{y}) \\ x_3(\mathbf{y}) \\ x_4(\mathbf{y}) \\ x_5(\mathbf{y}) \\ x_6(\mathbf{y}) \\ x_7(\mathbf{y}) \\ x_8(\mathbf{y}) \\ x_9(\mathbf{y}) \\ x_{10}(\mathbf{y}) \end{bmatrix} = \mathbf{0} \Rightarrow \begin{cases} y_1 - 0.25428722 - 0.18324757y_4y_3y_9 = 0, \\ y_2 - 0.37842197 - 0.16275449y_1y_{10}y_6 = 0, \\ y_3 - 0.27162577 - 0.16955071y_1y_2y_{10} = 0, \\ y_4 - 0.19807914 - 0.15585316y_7y_1y_6 = 0, \\ y_5 - 0.44166728 - 0.19950920y_7y_6y_3 = 0, \\ y_6 - 0.14654113 - 0.18922793y_8y_5y_{10} = 0, \\ y_7 - 0.42937161 - 0.21180486y_2y_5y_8 = 0, \\ y_8 - 0.07056438 - 0.17081208y_1y_7y_6 = 0, \\ y_9 - 0.34504906 - 0.19612740y_{10}y_6y_8 = 0, \\ y_{10} - 0.42651102 - 0.21466544y_4y_8y_1 = 0. \end{cases} \tag{7}$$

Here the system model (7) consists of 10 equations and the unknown vector $\mathbf{y} = [y_1, y_2, y_3, \dots, y_n]$ comprises of $n = 10$ elements. The proposed hybrid schemes are applied to solve a nonlinear equation as per the procedure outlined in the last section, while the residual error as fitness function for the system is constructed as:

deviation (STD) values and are listed in Table 3 for both PSO and PSO-NMMs. It can be seen that average residual error for PSO-1 to PSO-4, PSO-5 to PSO-8, PSO-9 to PSO-12 and PSO-13 to PSO-16 lies in the range 10^{-05} – 10^{-07} , 10^{-07} – 10^{-10} , 10^{-15} – 10^{-22} , and 10^{-12} – 10^{-34} , respectively, while for PSO-NMMs, these results are around 10^{-31} – 10^{-34} . It is observed that the increase in number of flights from 50 to 500 or swarm size from 25 to 200 particles contributes toward the improvement of results, which is more conspicuous for PSO-1 to PSO-12. Generally, the results obtained by hybrid approaches PSO-NMM have higher precision than that of PSO variants, while comparing the hybrid approaches, no significant difference is seen in the results.

3.2 Problem 2: chemical equilibrium application model (CEAM)

To observe the performance of proposed solver, a second case of nonlinear equation systems has been taken from

$$\begin{aligned} \text{Oir} &= \frac{1}{10} \sum_{i=1}^{10} r_i^2 \\ &= \frac{1}{10} \sum_{i=1}^{10} \left((y_1 - 0.25428722 - 0.18324757y_4y_3y_9)^2 + (y_2 - 0.37842197 - 0.16275449y_1y_{10}y_6)^2 + \right. \\ &\quad \left. (y_3 - 0.27162577 - 0.16955071y_1y_2y_{10})^2 + (y_4 - 0.19807914 - 0.15585316y_7y_1y_6)^2 + \right. \\ &\quad \left. (y_5 - 0.44166728 - 0.19950920y_7y_6y_3)^2 + (y_6 - 0.14654113 - 0.18922793y_8y_5y_{10})^2 + \right. \\ &\quad \left. (y_7 - 0.42937161 - 0.21180486y_2y_5y_8)^2 + (y_8 - 0.07056438 - 0.17081208y_1y_7y_6)^2 + \right. \\ &\quad \left. (y_9 - 0.34504906 - 0.19612740y_{10}y_6y_8)^2 + (y_{10} - 0.42651102 - 0.21466544y_4y_8y_1)^2 \right). \end{aligned} \tag{8}$$

Optimization of residual error function (8) is conducted with sixteen variants of PSO and PSO-NMMs for 100 independent executions. The approximate solutions and

CEAM [53, 54, 57]. Mathematical relations for CEAM can be written as follows:

$$\begin{bmatrix} x_1(\mathbf{y}) \\ x_1(\mathbf{y}) \\ x_1(\mathbf{y}) \\ x_1(\mathbf{y}) \\ x_1(\mathbf{y}) \end{bmatrix} = \mathbf{0}, \Rightarrow \begin{cases} y_1y_2 + y_1 - 3y_5 = 0, \\ 2y_1y_2 + y_1 + y_2y_3^2 + c_8y_2 - cy_5 + 2c_{10}y_2^2 + c_7y_2y_3 + c_9t_2t_4 = 0, \\ 2y_2y_3^2 + 2c_5y_3^2 - 8y_5 + c_6y_3 + c_7y_2y_3 = 0, \\ c_9y_2y_4 + 2y_4^2 - 4cy_5 = 0, \\ y_1(y_2 + 1) + c_{10}y_2^2 + y_2y_3^2 + c_8y_2 + c_5y_3^2 + y_4^2 - 1 + c_6y_3 + c_7t_2t_3 + c_9t_2t_4 = 0. \end{cases} \tag{9}$$

their absolute errors (AEs) calculated for each equation of nonlinear system (7) are shown graphically in Fig. 2 for both algorithms. Results are also calculated for statistical performance indices in terms of mean and standard

here the values of the constants are $c = 10, c_5 = 0.193, c_6 = 0.000410, c_7 = 0.000545, c_8 = 0.000000449, c_9 = 0.0000340,$ and $c_{10} = 0.000000961$. The unknown vector, $\mathbf{y} = [y_1, y_2, y_3, y_4, y_5]$, comprises of $n = 5$ elements.

Table 3 Results of statistical performance indices for 100 independent runs of PSO and PSO-NMMs in case of problem 1

Methods	Fitness ‘OIR values’			Methods	Fitness ‘OIR values’		
	MIN	Mean	STD		MIN	Mean	STD
PSO-1	1.05E−05	1.26E−04	2.71E−04	PSO-NMM-1	3.46E−33	6.20E−32	9.22E−32
PSO-2	2.10E−06	2.62E−05	2.22E−05	PSO-NMM-2	1.87E−33	8.87E−32	1.94E−31
PSO-3	7.81E−07	7.81E−06	5.83E−06	PSO-NMM-3	2.01E−33	1.29E−31	4.95E−31
PSO-4	3.93E−07	2.91E−06	1.79E−06	PSO-NMM-4	8.62E−34	8.33E−32	1.63E−31
PSO-5	2.28E−09	1.58E−07	5.46E−07	PSO-NMM-5	2.20E−33	8.20E−32	2.13E−31
PSO-6	4.46E−10	7.82E−09	9.67E−09	PSO-NMM-6	2.13E−33	7.00E−32	9.58E−32
PSO-7	6.68E−11	1.12E−09	1.10E−09	PSO-NMM-7	1.57E−33	1.09E−31	2.28E−31
PSO-8	1.26E−11	2.00E−10	2.33E−10	PSO-NMM-8	1.70E−33	5.95E−32	1.47E−31
PSO-9	3.36E−19	3.82E−15	1.68E−14	PSO-NMM-9	1.69E−33	9.41E−32	3.18E−31
PSO-10	1.18E−21	5.57E−19	9.83E−19	PSO-NMM-10	3.09E−33	1.39E−31	6.11E−31
PSO-11	3.54E−23	1.04E−20	2.79E−20	PSO-NMM-11	1.07E−33	6.93E−32	1.10E−31
PSO-12	1.47E−24	1.11E−22	1.53E−22	PSO-NMM-12	1.99E−33	8.50E−32	2.51E−31
PSO-13	3.53E−34	1.31E−12	1.31E−11	PSO-NMM-13	3.53E−34	1.99E−32	4.07E−32
PSO-14	2.72E−34	4.26E−34	8.89E−34	PSO-NMM-14	2.72E−34	3.57E−34	2.44E−34
PSO-15	2.72E−34	2.72E−34	0.00E+00	PSO-NMM-15	2.72E−34	2.72E−34	0.00E+00
PSO-16	2.72E−34	2.72E−34	0.00E+00	PSO-NMM-16	2.72E−34	2.72E−34	0.00E+00

Similar approach, adopted in the last problem, is followed to solve the equations of the system (9), but the residual error function for this case is developed as given below:

3.3 Problem 3: neurophysiology application model

The performance of the proposed scheme is examined for nonlinear systems representing the neurophysiology

$$\begin{aligned}
 \text{Oir} &= \frac{1}{5} \sum_{i=1}^5 r_i^2 \\
 &= \frac{1}{5} \sum_{i=1}^5 \left(\begin{aligned} &(y_1y_2 + y_1 - 3y_5)^2 + (2y_1y_2 + y_1 + y_2y_3^2 + c_8y_2 - cy_5 + 2c_{10}y_2^2 + c_7y_2y_3 + c_9t_2t_4)^2 \\ &+ (2y_2y_3^2 + 2c_5y_3^2 - 8y_5 + c_6y_3 + c_7y_2y_3)^2 + (c_9y_2y_4 + 2y_4^2 - 4cy_5)^2 \\ &+ (y_1(y_2 + 1) + c_{10}y_2^2 + y_2y_3^2 + c_8y_2 + c_5y_3^2 + y_4^2 - 1 + c_6y_3 + c_7t_2t_3 + c_9t_2t_4)^2 \end{aligned} \right). \tag{10}
 \end{aligned}$$

Approximate solutions and their AEs calculated for each equation of the system (9) are given in Fig. 2 for both variants of PSO and PSO-NMMs, while the results for a statistical indicator of mean and STD values are listed in Table 4. It is observed that MIN values of residual error for PSO-1 to PSO-16 range from 10^{-04} to 10^{-07} , while these values for PSO-NMMs range from 10^{-31} to 10^{-34} . The residual error on the basis of mean values of both the algorithm is around 10^{-03} ; this is due to the fact that a single bad run can drastically affect the mean values. It is also observed that the increase in number of flights or particles in PSO does not contribute toward a decrease in the residual error for this problem (Fig. 3).

application model (NPAM). The NPAM is written in terms of simultaneous equations as follows: [53, 58]

$$\begin{bmatrix} x_1(\mathbf{y}) \\ x_2(\mathbf{y}) \\ x_3(\mathbf{y}) \\ x_4(\mathbf{y}) \\ x_5(\mathbf{y}) \\ x_6(\mathbf{y}) \end{bmatrix} = \mathbf{0}, \Rightarrow \begin{cases} y_1^2 + y_3^2 - 1 = 0, \\ y_2^2 + y_4^2 - 1 = 0, \\ y_5y_3^2 + y_6y_4^2 - c_1 = 0, \\ y_5y_1^3 + y_6y_2^3 - c_2 = 0, \\ y_5y_1y_3^2 + y_6y_4^2y_2 - c_3 = 0, \\ y_5y_1^2y_3 + y_6y_2^2y_4 - c_4 = 0. \end{cases} \tag{11}$$

Here the values of constant c_1, c_2, c_3 and c_4 are zeros. System (10) consists of six unknown and six numbers of equations; therefore, the residual error function is formulated as:

$$\begin{aligned}
 \text{Oir} &= \frac{1}{6} \sum_{i=1}^6 r_i^2 \\
 &= \frac{1}{5} \sum_{i=1}^5 \left((y_1^2 + y_3^2 - 1)^2 + (y_2^2 + y_4^2 - 1)^2 + (y_5 y_3^2 + y_6 y_4^2 - c_1)^2 + (y_5 y_1^3 + y_6 y_2^3 - c_2)^2 \right. \\
 &\quad \left. + (y_5 y_1 y_3^2 + y_6 y_4^2 y_2 - c_3)^2 + (y_5 y_1^2 y_3 + y_6 y_2^2 y_4 - c_4)^2 \right).
 \end{aligned}
 \tag{12}$$

Optimization of error function (11) is performed with PSO and PSO-NMMs, and results are listed in Table 5 in terms of statistical operator of the mean and standard deviation, while the approximate solutions and their AEs for the constituting equations are plotted in Fig. 4. It is seen that average residual error for PSO-1 to PSO-4, PSO-5 to PSO-8, PSO-9 to PSO-12 and PSO-13 to PSO-16 lies in the range 10^{-06} – 10^{-07} , 10^{-07} – 10^{-09} , 10^{-09} – 10^{-11} and 10^{-10} – 10^{-11} , respectively, while for PSO-NMMs, these results are around 10^{-21} – 10^{-32} . Additionally, desired minimum values of residual errors are achieved for each PSO-NMM, while these results are only obtained from PSO-13 to PSO-16. It is observed that highly accurate results are determined consistently with hybrid algorithms for this problem.

representing combustion process with temperature around 3000 °C [53, 59]. Governing mathematical expressions for this problem are given below:

$$\begin{aligned}
 \begin{bmatrix} x_1(\mathbf{y}) \\ x_2(\mathbf{y}) \\ x_3(\mathbf{y}) \\ x_4(\mathbf{y}) \\ x_5(\mathbf{y}) \\ x_6(\mathbf{y}) \\ x_7(\mathbf{y}) \\ x_8(\mathbf{y}) \\ x_9(\mathbf{y}) \\ x_{10}(\mathbf{y}) \end{bmatrix} = \mathbf{0} \Rightarrow \begin{cases} y_2 + 2y_6 + y_9 + 2y_{10} - 10^{-5} = 0, \\ y_3 + y_8 - 3 \times 10^{-5} = 0, \\ y_1 + y_3 + 2y_5 + 2y_8 + y_9 + y_{10} - 5 \times 10^{-5} = 0, \\ y_4 + 2y_7 - 10^{-5} = 0, \\ 0.5140437 \times 10^{-7} y_5 - y_1^2 = 0, \\ 0.1006932 \times 10^{-6} y_6 - 2y_2^2 = 0, \\ 0.7816278 \times 10^{-15} y_7 - y_4^2 = 0, \\ 0.1496236 \times 10^{-6} y_8 - y_1 y_3 = 0, \\ 0.6194411 \times 10^{-7} y_9 - y_1 y_2 = 0, \\ 0.2089296 \times 10^{-14} y_{10} - y_1 y_2^2 = 0. \end{cases}
 \end{aligned}
 \tag{13}$$

3.4 Problem 4: combustion theory application model

In this case, performance of the proposed solver is evaluated for the well-known system of nonlinear equations

Similar approach, as followed in the previous cases, is adopted to solve system (13) by constructing a residual error function for the number of unknown $n = 10$ as:

Table 4 Results of statistical performance indices for 100 independent runs of PSO and PSO-NMMs in case of problem 2

Methods	Fitness ‘OIR values’			Methods	Fitness ‘OIR values’		
	MIN	Mean	STD		MIN	Mean	STD
PSO-1	5.80E-05	5.41E-03	7.62E-03	PSO-NMM-1	3.74E-34	3.37E-03	7.00E-03
PSO-2	1.99E-04	4.61E-03	7.58E-03	PSO-NMM-2	2.44E-33	3.37E-03	7.00E-03
PSO-3	2.40E-05	2.67E-03	6.01E-03	PSO-NMM-3	1.31E-33	1.78E-03	5.35E-03
PSO-4	4.33E-05	2.15E-03	5.49E-03	PSO-NMM-4	4.18E-34	1.60E-03	5.11E-03
PSO-5	1.28E-04	3.00E-03	6.65E-03	PSO-NMM-5	2.38E-33	2.49E-03	6.19E-03
PSO-6	4.82E-05	2.11E-03	5.75E-03	PSO-NMM-6	3.58E-34	1.78E-03	5.35E-03
PSO-7	3.34E-05	1.85E-03	5.43E-03	PSO-NMM-7	6.56E-35	1.42E-03	4.84E-03
PSO-8	4.49E-05	6.39E-04	2.78E-03	PSO-NMM-8	4.30E-34	3.55E-04	2.50E-03
PSO-9	3.89E-05	4.68E-03	8.14E-03	PSO-NMM-9	2.44E-33	3.91E-03	7.39E-03
PSO-10	2.54E-05	1.90E-03	5.53E-03	PSO-NMM-10	2.22E-34	1.42E-03	4.84E-03
PSO-11	2.11E-05	1.15E-03	4.23E-03	PSO-NMM-11	1.21E-33	1.07E-03	4.24E-03
PSO-12	9.84E-06	5.79E-04	3.04E-03	PSO-NMM-12	2.58E-34	5.33E-04	3.04E-03
PSO-13	2.75E-05	3.44E-03	7.04E-03	PSO-NMM-13	1.11E-34	3.37E-03	7.00E-03
PSO-14	1.39E-05	2.81E-03	6.44E-03	PSO-NMM-14	1.10E-33	2.66E-03	6.37E-03
PSO-15	3.55E-06	1.61E-03	5.10E-03	PSO-NMM-15	1.22E-34	1.60E-03	5.11E-03
PSO-16	4.16E-07	5.73E-04	3.05E-03	PSO-NMM-16	5.93E-34	5.33E-04	3.04E-03

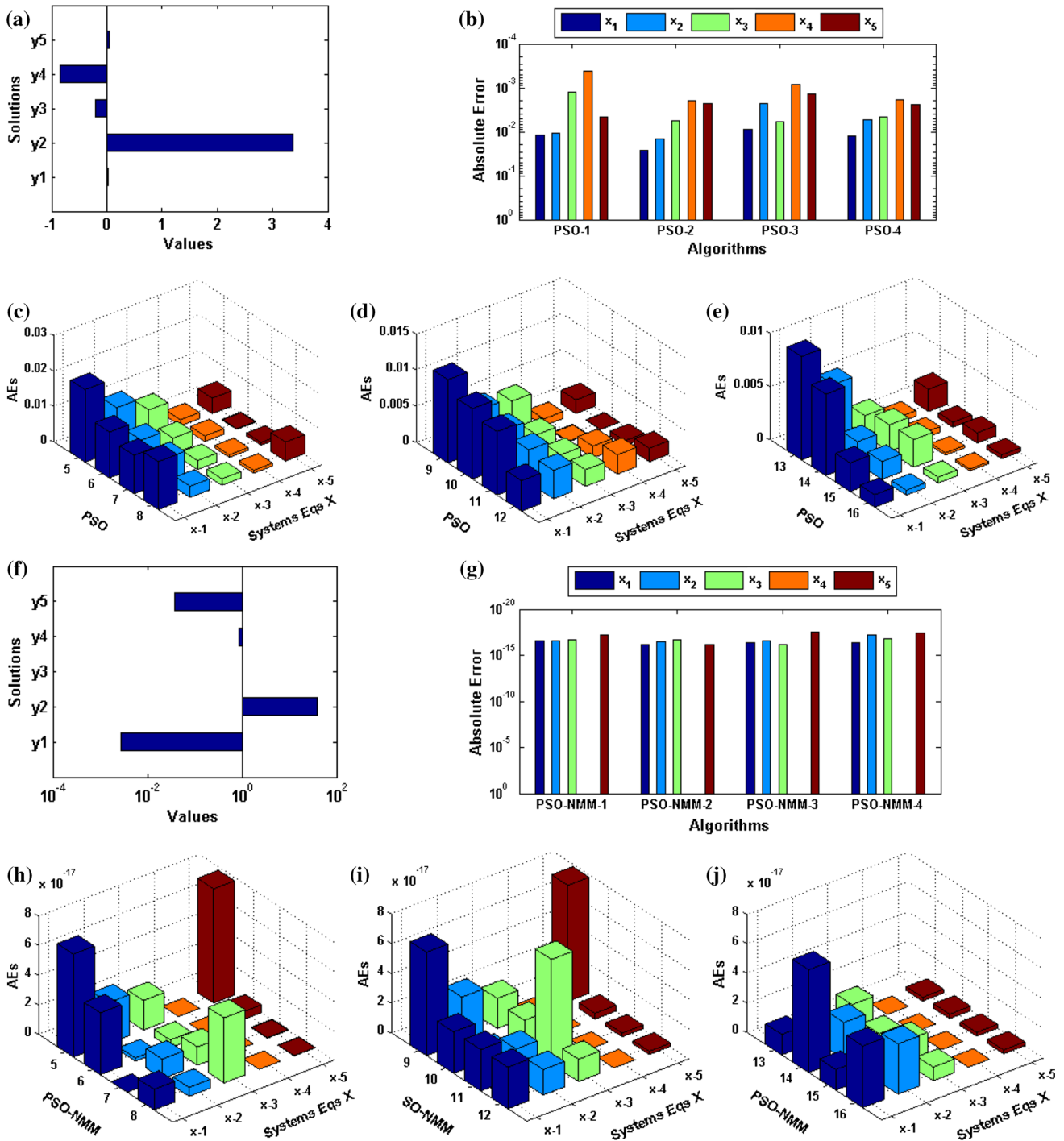


Fig. 3 Proposed results of PSO and PSO-NMMs for chemical equilibrium application model presented in problem 2. **a** Approximate solution for PSO-1. **b** Absolute errors (AEs) for constituents equations of the system. **c** AEs for PSO-5 to PSO-8. **d** AEs for PSO-9 to PSO-12. **e** AEs for PSO-13 to PSO-16. **f** Approximate solution for PSO-

NMM-1. **g** Absolute errors (AEs) for constituents equations of the system. **h** AEs for PSO-NMM-5 to PSO-NMM-8. **i** AEs for PSO-NMM-9 to PSO-NMM-12. **j** AEs for PSO-NMM-13 to PSO-NMM-16

$$Oir = \frac{1}{10} \sum_{i=1}^{10} r_i^2 = \frac{1}{10} \sum_{i=1}^{10} \left(\begin{aligned} &(y_2 + 2y_6 + y_9 + 2y_{10} - 10^{-5})^2 + (y_3 + y_8 - 3 \times 10^{-5})^2 \\ &+ (y_1 + y_3 + 2y_5 + 2y_8 + y_9 + y_{10} - 5 \times 10^{-5})^2 + (y_4 + 2y_7 - 10^{-5})^2 \\ &+ (0.5140437 \times 10^{-7}y_5 - y_1^2)^2 + (0.1006932 \times 10^{-6}y_6 - 2y_2^2)^2 \\ &+ (0.7816278 \times 10^{-15}y_7 - y_4^2)^2 + (0.1496236 \times 10^{-6}y_8 - y_1y_3)^2 \\ &+ (0.6194411 \times 10^{-7}y_9 - y_1y_2)^2 + (0.2089296 \times 10^{-14}y_{10} - y_1y_2^2)^2 \end{aligned} \right). \tag{14}$$

Optimization of residual error function (13) is conducted, and approximate solutions are plotted graphically in Fig. 5 for variants of PSO and PSO-NMMs. The values of mean and STD based on 100 independent runs of the algorithms are listed in Table 6. It can be seen that average residual error for PSO-1 to PSO-4, PSO-5 to PSO-8, PSO-9 to PSO-12 and PSO-13 to PSO-16 ranges from 10^{-04} to 10^{-05} , 10^{-05} – 10^{-06} , 10^{-07} – 10^{-08} and 10^{-07} – 10^{-09} , respectively, while for PSO-NMMs these results are around 10^{-17} . Minimum values of PSO and PSO-NMMs range from 10^{-05} to 10^{-12} and 10^{-33} – 10^{-35} , respectively. The dominance of hybrid schemes is generally observed for this case study as well.

3.5 Problem 5: economics modeling system

The econometric model of arbitrary dimensions is studied in this case [53, 59]. The problem in term of mathematical equations is stated as:

$$\begin{cases} \left(y_k + \sum_{i=1}^{n-k-1} y_i y_i + k \right) y_n - c_k = 0; & 1 \leq k \leq n - 1 \\ \sum_{i=1}^{n-1} y_i + 1 = 0. \end{cases} \tag{15}$$

Here the value of constant c_k is chosen randomly. Proposed schemes are examined in a special case of the problem (15) by taking the values of $n = 5$ and $k = [1-4]$ as:

$$\begin{bmatrix} x_1(\mathbf{y}) \\ x_2(\mathbf{y}) \\ x_3(\mathbf{y}) \\ x_4(\mathbf{y}) \\ x_5(\mathbf{y}) \end{bmatrix} = \mathbf{0}, \Rightarrow \begin{cases} y_1 + y_2 + y_3 + y_4 + 1 = 0, \\ y_1y_5 + y_1y_2y_5 + y_2y_3y_5 + y_3y_4y_5 - 1 = 0, \\ y_2y_5 + y_1y_3y_5 + y_2y_4y_5 - 1 = 0, \\ y_3y_5 + y_1y_4y_5 - 1 = 0, \\ y_4y_5 - 1 = 0. \end{cases} \tag{16}$$

Table 5 Results of statistical performance indices for 100 independent runs of PSO and PSO-NMMs in case of problem 3

Methods	Fitness ‘OIR values’			Methods	Fitness ‘OIR values’		
	MIN	Mean	STD		MIN	Mean	STD
PSO-1	2.11E-08	7.07E-06	1.56E-05	PSO-NMM-1	0.00E+00	3.17E-15	3.17E-14
PSO-2	1.03E-08	1.91E-06	4.19E-06	PSO-NMM-2	0.00E+00	1.02E-13	1.02E-12
PSO-3	6.11E-09	1.87E-06	8.24E-06	PSO-NMM-3	0.00E+00	1.60E-15	1.60E-14
PSO-4	5.66E-09	9.42E-07	3.27E-06	PSO-NMM-4	0.00E+00	6.07E-32	2.48E-31
PSO-5	5.37E-13	1.01E-07	4.83E-07	PSO-NMM-5	0.00E+00	1.13E-32	3.78E-32
PSO-6	8.73E-14	3.05E-08	1.41E-07	PSO-NMM-6	0.00E+00	5.41E-32	2.44E-31
PSO-7	1.89E-14	1.57E-08	7.60E-08	PSO-NMM-7	0.00E+00	1.32E-30	1.28E-29
PSO-8	1.12E-14	1.18E-09	5.09E-09	PSO-NMM-8	0.00E+00	3.03E-21	3.03E-20
PSO-9	3.30E-27	3.96E-09	3.08E-08	PSO-NMM-9	0.00E+00	1.01E-30	7.03E-30
PSO-10	3.32E-28	5.33E-11	2.82E-10	PSO-NMM-10	0.00E+00	2.84E-29	1.63E-28
PSO-11	1.35E-28	1.38E-09	1.33E-08	PSO-NMM-11	0.00E+00	4.33E-30	2.61E-29
PSO-12	1.01E-29	1.41E-11	1.11E-10	PSO-NMM-12	0.00E+00	6.14E-30	3.71E-29
PSO-13	0.00E+00	5.50E-11	3.74E-10	PSO-NMM-13	0.00E+00	4.26E-32	2.30E-31
PSO-14	0.00E+00	1.02E-11	5.97E-11	PSO-NMM-14	0.00E+00	2.99E-32	1.66E-31
PSO-15	0.00E+00	1.48E-11	1.39E-10	PSO-NMM-15	0.00E+00	3.09E-32	1.27E-31
PSO-16	0.00E+00	2.37E-10	2.34E-09	PSO-NMM-16	0.00E+00	2.36E-32	1.16E-31

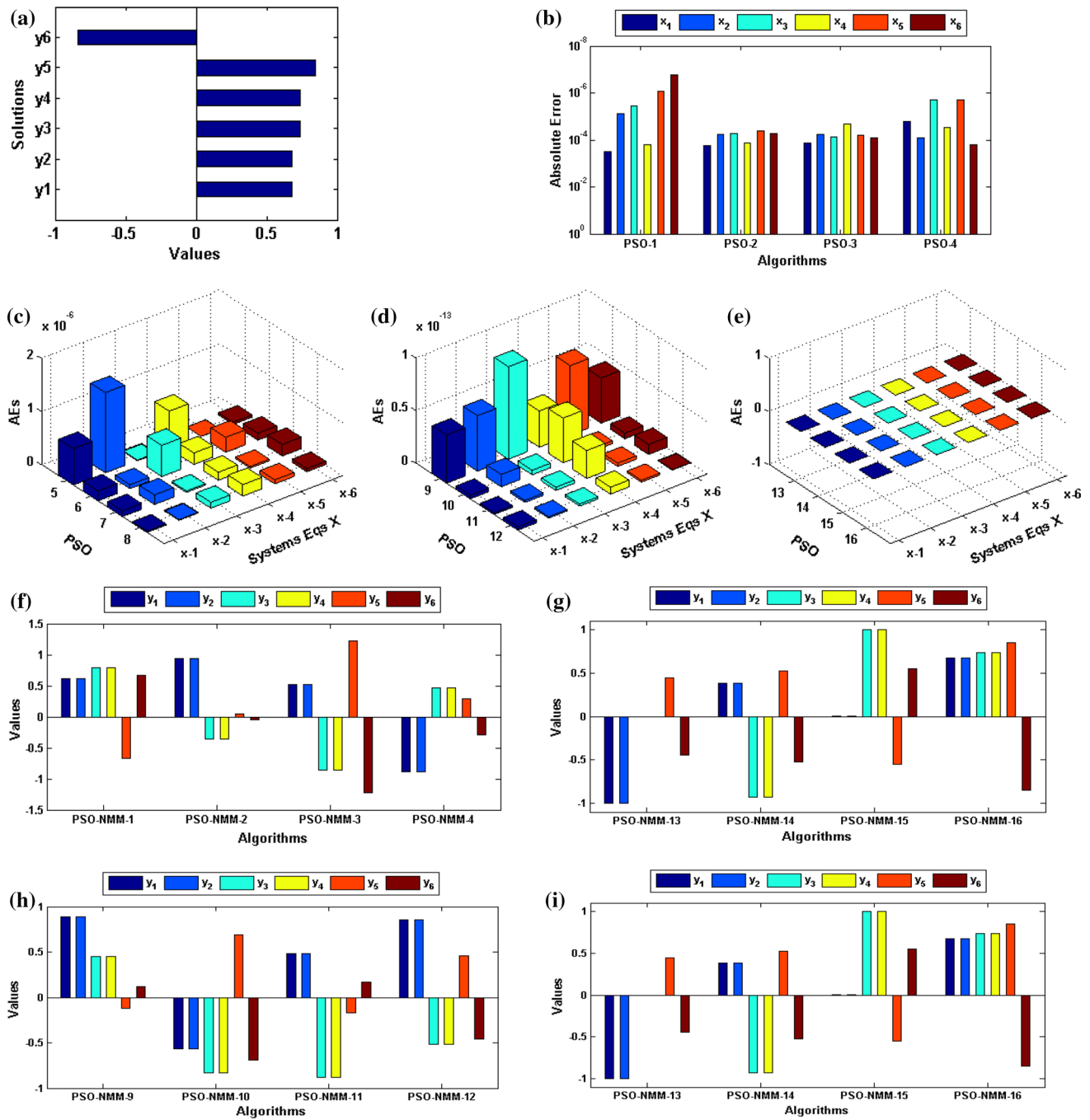


Fig. 4 Proposed results of PSO and PSO-NMMs for neurophysiology application model presented in problem 3. **a** Approximate solution for PSO-1. **b** Absolute errors (AEs) for constituents equations of the system. **c** AEs for PSO-5 to PSO-8. **d** AEs for PSO-9 to PSO-12.

e AEs for PSO-13 to PSO-16. **f** Approximate solution for PSO-NMM-1 to PSO-NMM-4. **g** Approximate solution for PSO-NMM-5 to PSO-NMM-8. **h** Approximate solution for PSO-NMM-9 to PSO-NMM-12. **i** Approximate solution for PSO-NMM-13 to PSO-NMM-16

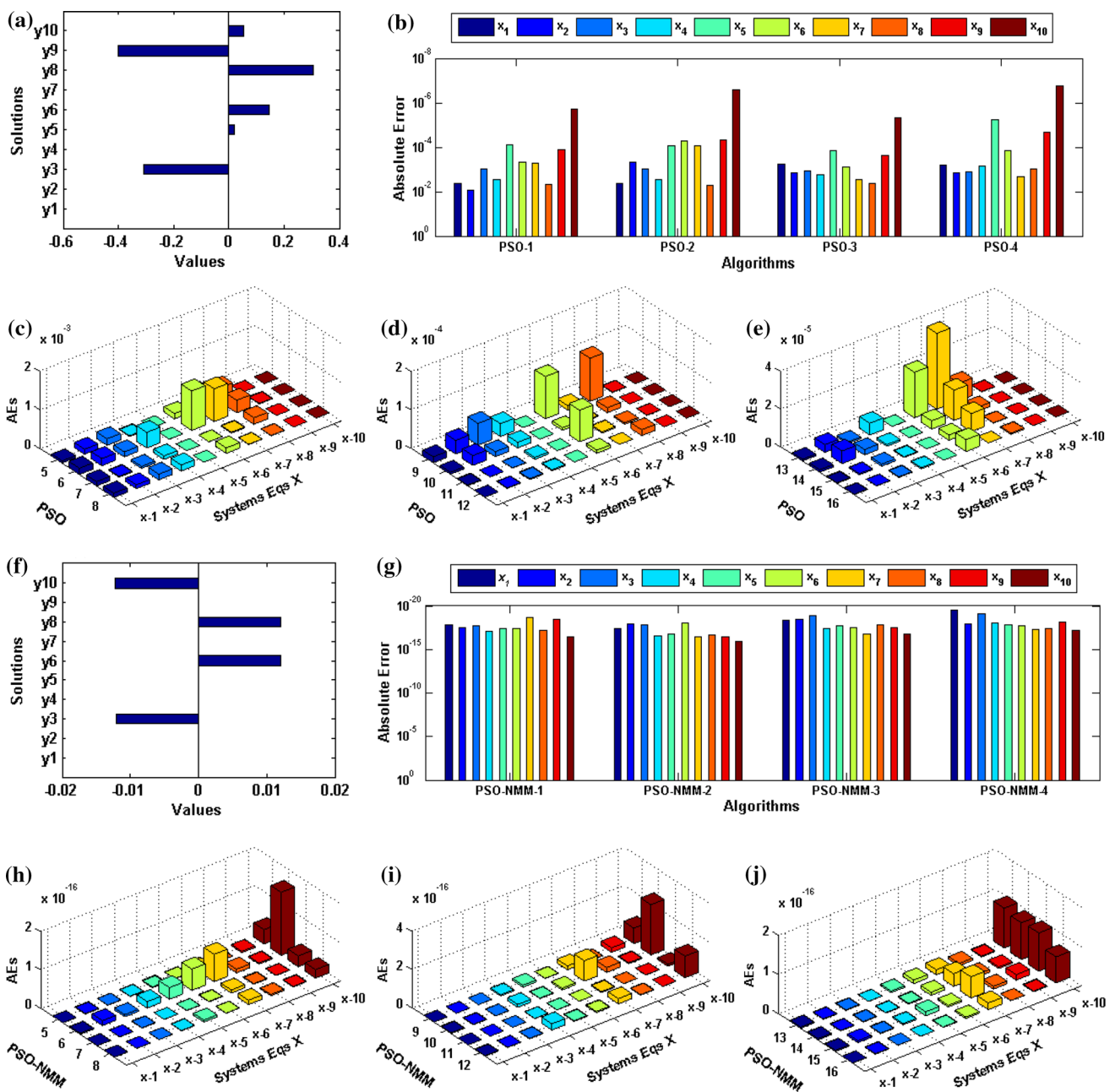


Fig. 5 Proposed results of PSO and PSO-NMMs for combustion theory application model presented in problem 4. **a** Approximate solution for PSO-1. **b** Absolute errors (AEs) for constituents equations of the system. **c** AEs for PSO-5 to PSO-8. **d** AEs for PSO-9 to PSO-12. **e** AEs for PSO-13 to PSO-16. **f** Approximate solution for PSO-

NMM-1. **g** Absolute errors (AEs) for constituents equations of the system. **h** AEs for PSO-NMM-5 to PSO-NMM-8. **i** AEs for PSO-NMM-9 to PSO-NMM-12. **j** AEs for PSO-NMM-13 to PSO-NMM-16

Table 6 Results of statistical performance indices for 100 independent runs of PSO and PSO-NMMs in case of problem 4

Methods	Fitness ‘OIR values’			Methods	Fitness ‘OIR values’		
	MIN	Mean	STD		MIN	Mean	STD
PSO-1	1.24E−05	3.13E−04	2.95E−04	PSO-NMM-1	1.53E−34	9.34E−17	1.90E−16
PSO-2	5.39E−06	2.20E−04	3.09E−04	PSO-NMM-2	1.58E−33	7.67E−17	1.44E−16
PSO-3	3.39E−06	9.05E−05	1.02E−04	PSO-NMM-3	6.03E−35	1.03E−16	2.84E−16
PSO-4	9.90E−07	3.31E−05	2.99E−05	PSO-NMM-4	8.88E−36	6.65E−17	1.47E−16
PSO-5	3.64E−08	2.32E−05	5.30E−05	PSO-NMM-5	1.39E−34	5.87E−17	1.11E−16
PSO-6	2.09E−07	6.66E−06	6.85E−06	PSO-NMM-6	3.62E−33	4.95E−17	1.22E−16
PSO-7	5.29E−09	3.50E−06	5.69E−06	PSO-NMM-7	8.61E−35	5.58E−17	1.36E−16
PSO-8	8.73E−09	1.51E−06	2.28E−06	PSO-NMM-8	6.54E−35	7.83E−17	2.13E−16
PSO-9	3.03E−09	6.64E−07	1.97E−06	PSO-NMM-9	7.58E−34	7.96E−17	2.44E−16
PSO-10	1.26E−10	1.62E−07	1.96E−07	PSO-NMM-10	8.10E−33	5.85E−17	1.45E−16
PSO-11	6.74E−10	5.55E−08	6.99E−08	PSO-NMM-11	1.35E−35	7.69E−17	1.77E−16
PSO-12	3.08E−11	2.76E−08	4.34E−08	PSO-NMM-12	1.45E−33	4.69E−17	1.04E−16
PSO-13	2.32E−10	1.76E−07	7.69E−07	PSO-NMM-13	1.06E−33	3.20E−17	7.31E−17
PSO-14	3.32E−11	1.32E−08	2.50E−08	PSO-NMM-14	1.00E−33	3.60E−17	6.99E−17
PSO-15	1.01E−11	4.88E−09	8.68E−09	PSO-NMM-15	1.28E−33	6.91E−17	2.37E−16
PSO-16	3.99E−12	2.22E−09	4.21E−09	PSO-NMM-16	4.69E−34	5.05E−17	1.09E−16

The residual error function for the system (16) is developed as:

$$\begin{aligned} \text{Oir} &= \frac{1}{5} \sum_{i=1}^5 r_i^2 \\ &= \frac{1}{5} \sum_{i=1}^5 \left((y_1 + y_2 + y_3 + y_4 + 1)^2 + (y_1 y_5 + y_1 y_2 y_5 + y_2 y_3 y_5 + y_3 y_4 y_5 - 1)^2 + \right. \\ &\quad \left. (y_2 y_5 + y_1 y_3 y_5 + y_2 y_4 y_5 - 1)^2 + (y_3 y_5 + y_1 y_4 y_5 - 1)^2 + (y_4 y_5 - 1)^2 \right). \end{aligned} \quad (17)$$

Results of 16 variants of the proposed adaptive schemes are listed in Table 7 for the values of mean and STD calculated for 100 independent runs of the algorithms. Additionally, approximate solutions of the system (16) are shown in Fig. 6. It can be seen that average residual error for PSO-1 to PSO-4, PSO-5 to PSO-8, PSO-9 to PSO-12 and PSO-13 to PSO-16 ranges from 10^{-04} to 10^{-07} , 10^{-07} – 10^{-12} , 10^{-17} – 10^{-27} and 10^{-29} – 10^{-33} , respectively, while for PSO-NMMs, these results are around 10^{-32} .

4 Comparative studies

In this section, comparison between variants of PSO and PSO-NMM is presented through results of statistical analysis based on 100 independent runs for all five non-linear systems.

The accuracy of the proposed schemes is examined for 100 independent runs of algorithms. An independent run of the algorithms is defined as a run with a randomly gener-

ated initial swarm of the particles with different seeds. Results on the basis of value of fitness, *i.e.*, the values of Oir, for a number of independent runs are plotted in case of PSO and PSO-NMMs in Fig. 7a–h, respectively, for problem 1. These plots are given on semilog scale and fitness, sorted in ascending manner, in order to clearly highlight small variations in the results. On the similar pattern, results for the problems 2, 3, 4 and 5 are shown in Figs. 8, 9, 10 and 11, respectively, for each variation. It can be seen that generally for all five problems, with the increase in the number of particles in the swarm from 25 to 200 for the fixed cycles of the algorithm, the results of the PSO variants improve, which is more evident in case of PSO-1 to PSO-4 for 50 cycles, PSO-5 to PSO-8 for 100 cycles and PSO-9 to PSO-12 for 150 cycles. However, for the cases PSO-13 to PSO-16 for 500 cycles, no noticeable change is observed in the results. Whereas in cases of

Table 7 Results of statistical performance indices for 100 independent runs of PSO and PSO-NMMs in case of problem 5

Methods	Fitness ‘OIR values’			Methods	Fitness ‘OIR values’		
	MIN	Mean	STD		MIN	Mean	STD
PSO-1	6.42E-07	2.09E-04	4.30E-04	PSO-NMM-1	2.47E-33	6.76E-32	7.46E-32
PSO-2	6.12E-07	1.98E-05	2.15E-05	PSO-NMM-2	2.47E-33	7.27E-32	7.40E-32
PSO-3	9.57E-08	2.80E-06	2.62E-06	PSO-NMM-3	2.47E-33	9.47E-32	1.05E-31
PSO-4	1.12E-08	4.84E-07	3.66E-07	PSO-NMM-4	2.47E-33	5.37E-32	4.96E-32
PSO-5	2.93E-11	7.81E-07	3.81E-06	PSO-NMM-5	2.47E-33	5.22E-32	4.06E-32
PSO-6	1.74E-12	6.13E-10	9.16E-10	PSO-NMM-6	2.47E-33	8.78E-32	1.02E-31
PSO-7	4.02E-13	3.71E-11	4.43E-11	PSO-NMM-7	9.86E-33	6.13E-32	6.91E-32
PSO-8	7.02E-14	3.67E-12	4.10E-12	PSO-NMM-8	9.86E-33	6.37E-32	6.14E-32
PSO-9	2.21E-23	2.35E-17	8.19E-17	PSO-NMM-9	2.47E-33	8.00E-32	1.06E-31
PSO-10	4.22E-25	1.72E-21	5.73E-21	PSO-NMM-10	2.47E-33	7.11E-32	6.61E-32
PSO-11	5.79E-28	4.41E-25	7.45E-25	PSO-NMM-11	2.47E-33	6.29E-32	6.22E-32
PSO-12	2.77E-30	2.52E-27	4.08E-27	PSO-NMM-12	4.93E-33	4.91E-32	4.13E-32
PSO-13	2.47E-33	1.29E-29	1.04E-28	PSO-NMM-13	2.47E-33	1.43E-32	1.05E-32
PSO-14	2.47E-33	1.16E-32	7.37E-33	PSO-NMM-14	2.47E-33	1.08E-32	6.05E-33
PSO-15	2.47E-33	1.03E-32	4.56E-33	PSO-NMM-15	2.47E-33	1.02E-32	4.46E-33
PSO-16	2.47E-33	9.69E-33	4.32E-33	PSO-NMM-16	2.47E-33	9.60E-33	4.24E-33

hybrid PSO-NMM approach by taking the best particle of PSO as the starting point of NMM, precision enhanced considerably. Generally, results of PSO-NMMs are improved as compared to respective PSO variants and this improvement is more apparent in the case of PSO-NMM-1 to PSO-NMM-12 and for the cases for which PSO algorithm achieved fitness around 10^{-30} ; however, no significant contribution of hybridization with local search is observed.

Next accuracy and convergence are examined for 100 independent runs of PSO and PSO-NMMs achieving different levels of accuracy. Results of convergence analyses, based on different criterion, are listed in Table 6 for variants of PSO for all five problems, while for variants of PSO-NMMs, these results are given in Table 7. It can be seen that criteria based on fitness values around 10^{-07} – 10^{-09} , almost 100 % of the runs for all hybrid approaches PSO-NMM-1 to PSO-NMM-16 and algorithms PSO-13 to PSO-16, have fulfilled the criteria, while PSO-1 to PSO-12 gave relatively degraded convergence rate. Generally, in all five cases, the rate of convergence of the hybrid PSO-NMMs approaches is relatively better than that of PSO variants. Additionally, it can be seen that the performance of the proposed algorithms based on both PSO and PSO-NMMs is relatively superior for the third problem as compared to the other four problems and more accurate results are obtained for second problem as compared to the remaining problems (Tables 8, 9).

Complexity analysis of the variants of PSO and PSO-NMMs is performed on the basis of average time consumed by the algorithm to find the approximate solution of a given

nonlinear system. The values of mean execution time along with STD for 100 independent runs of each solver for all five problems are listed in Table 10. It is seen from the results presented that with the increase in the number of particles of the swarm or number of flights, the values of mean execution time also increase. The values of complexity operator for PSO variants are relatively smaller than that of PSO-NMMs in case of all five case studies, i.e., hybrid approaches take 3–4 times more time as compared to the execution time of PSO variants. The optimization process of the proposed algorithms relatively takes longer execution time in case of problem 2 as compared to other problems. Simulations are carried out in the present study on ACER Laptop Model V3-471G with Core i5-3210 M 2.6 GHz processor with turbo boost up to 3.1 GHz, 4.00 GB DDR3 memory and running MATLAB software package version 2012a in a Microsoft Windows 7 operating environment.

The true comparison among between stochastic and deterministic numerical methods is not possible due to their inherent different optimization procedures; however, in order to establish the effectiveness of proposed design algorithms, a generic comparison among standard solvers is given here. The solution of all five benchmark studies are determined by using ‘fsolve’ routine, a built-in function of MATLAB optimization toolbox for solving nonlinear system of equations with known initial start point based on trust-region-dogleg, trust-region-reflective and Levenberg–Marquardt algorithms. Default setting of the parameters of all three algorithms is used for finding the solution. It is found that the results obtained by all three algorithms lie in the range of 10^{-03} – 10^{-09} for arithmetic

benchmark model, chemical equilibrium, neurophysiology, combustion theory and economics models, whereas the proposed methods based on PSO hybrid with NMM achieved the accuracy of order 10^{-30} . The complexity of the deterministic solver is superior from stochastic

methods, which is understandable due to the stochastic solver based on complex global search procedures. However, this aspect of stochastic numerical scheme is overshadowed due to consistently accurate results without known initial bias guess.

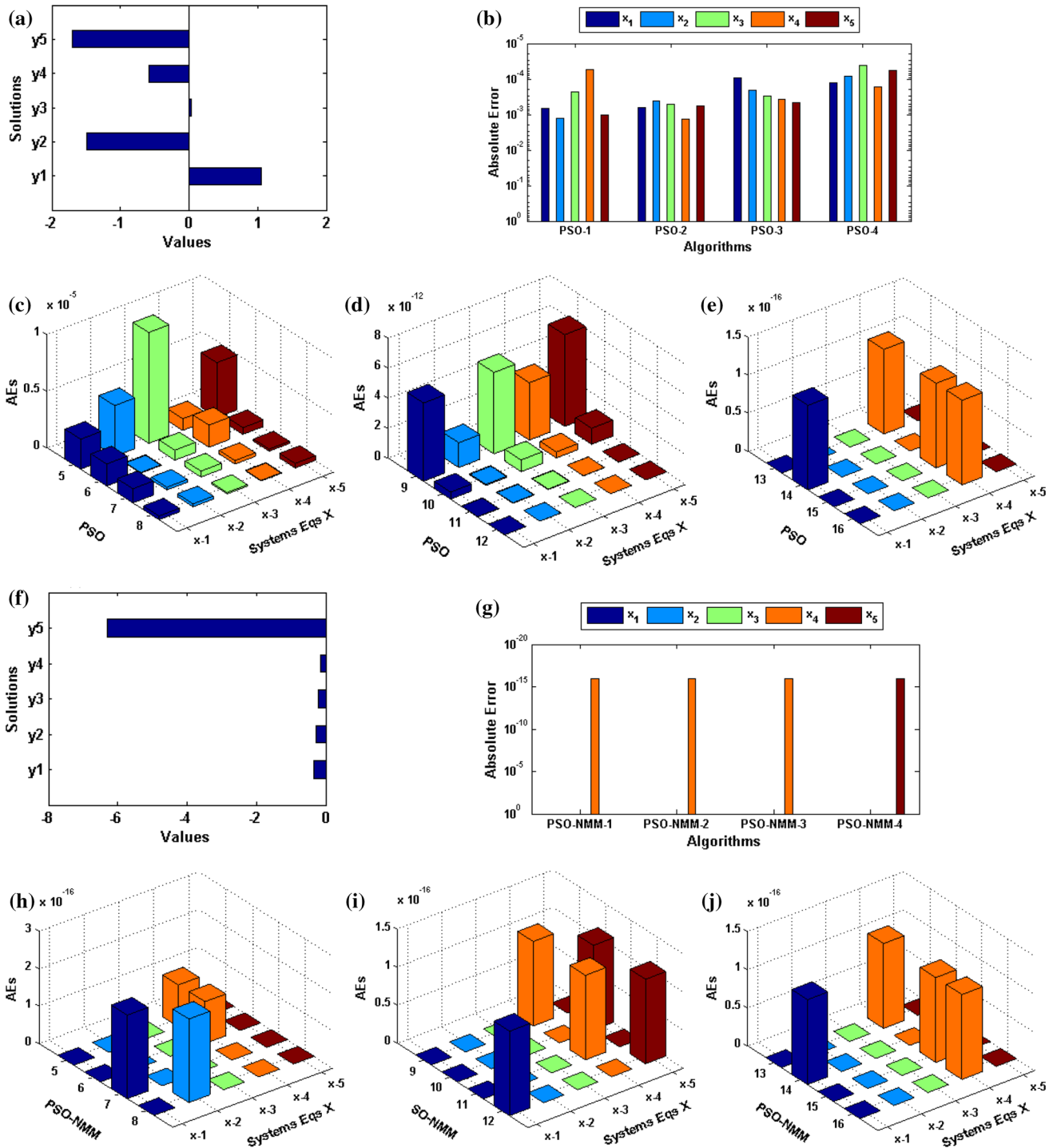


Fig. 6 Proposed results of PSO and PSO-NMMs for economics modeling system presented in problem 5. **a** Approximate solution for PSO-1. **b** Absolute errors (AEs) for constituents equations of the system. **c** AEs for PSO-5 to PSO-8. **d** AEs for PSO-9 to PSO-12.

e AEs for PSO-13 to PSO-16. **f** Approximate solution for PSO-NMM-1. **g** Absolute errors (AEs) for constituents equations of the system. **h** AEs for PSO-NMM-5 to PSO-NMM-8. **i** AEs for PSO-NMM-9 to PSO-NMM-12. **j** AEs for PSO-NMM-13 to PSO-NMM-16

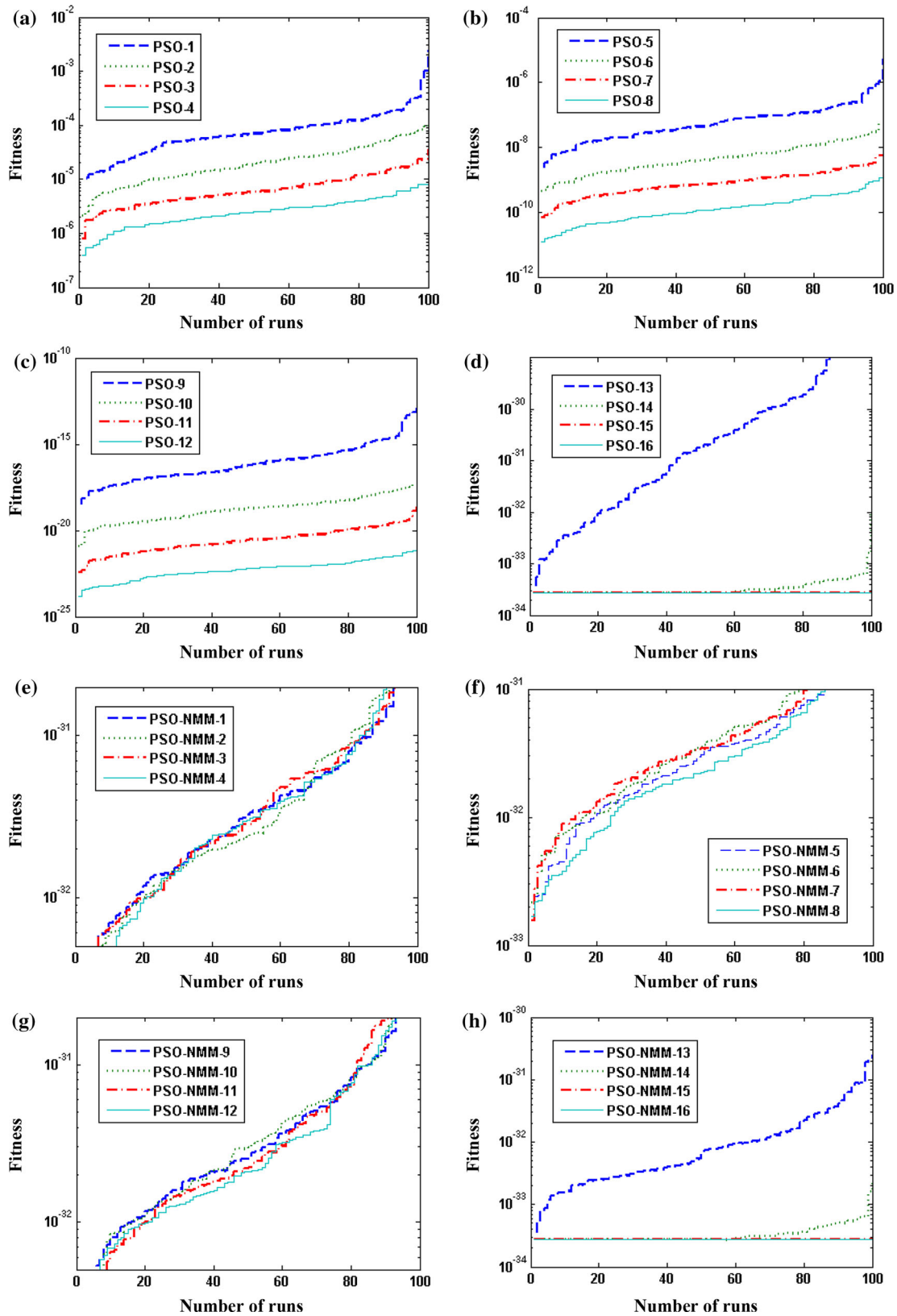


Fig. 7 Plot of sorted fitness, O_{ir} , values for 100 independent runs of algorithms in the case of the interval arithmetic benchmark model presented in problem 1. **a–h** Are for PSO and PSO-NMMs, respectively

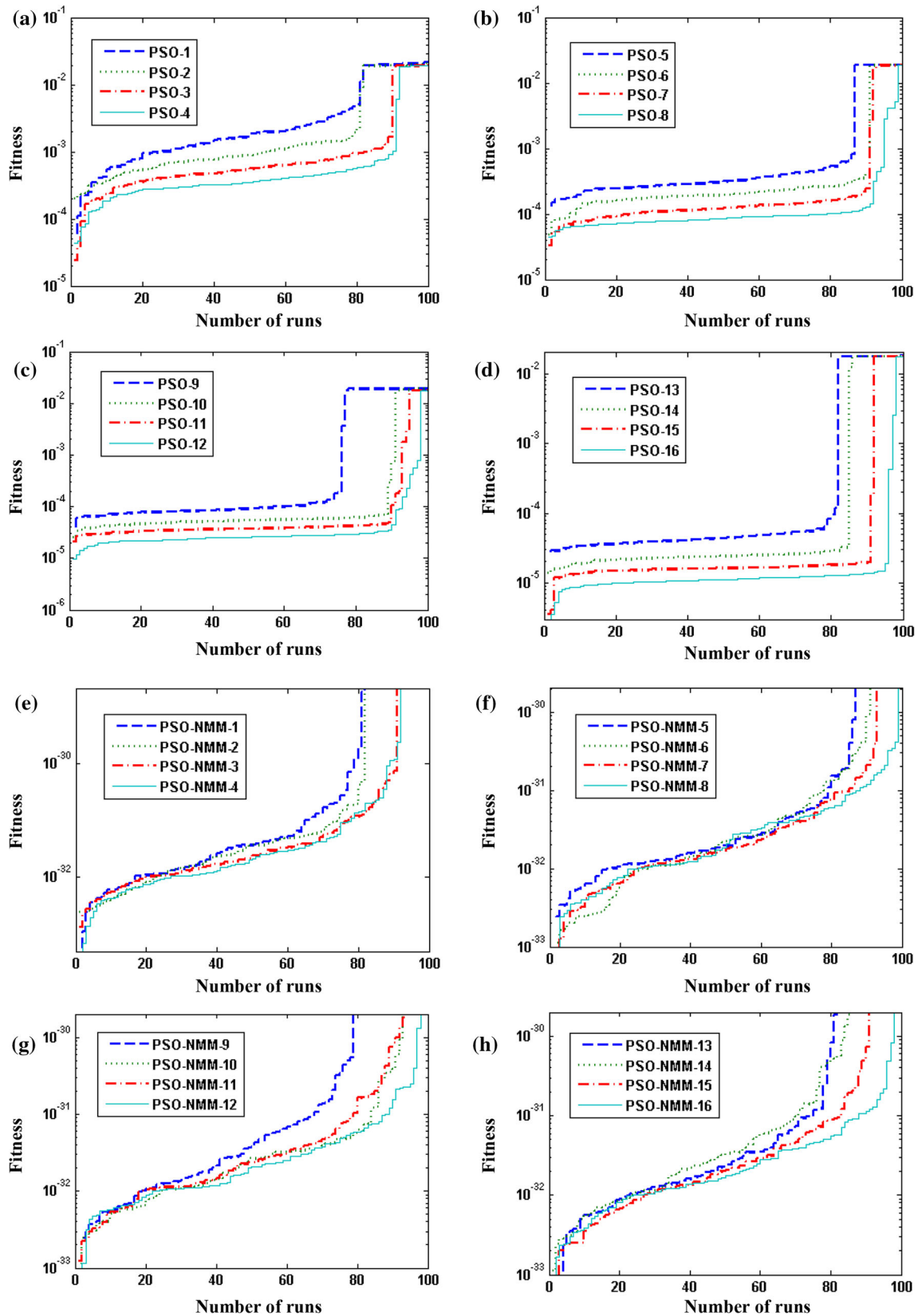


Fig. 8 Plot of sorted fitness, O_{ir} , values for 100 independent runs of algorithms in the case of the chemical equilibrium application model presented in problem 2. **a–h** Are for PSO and PSO-NMMs, respectively

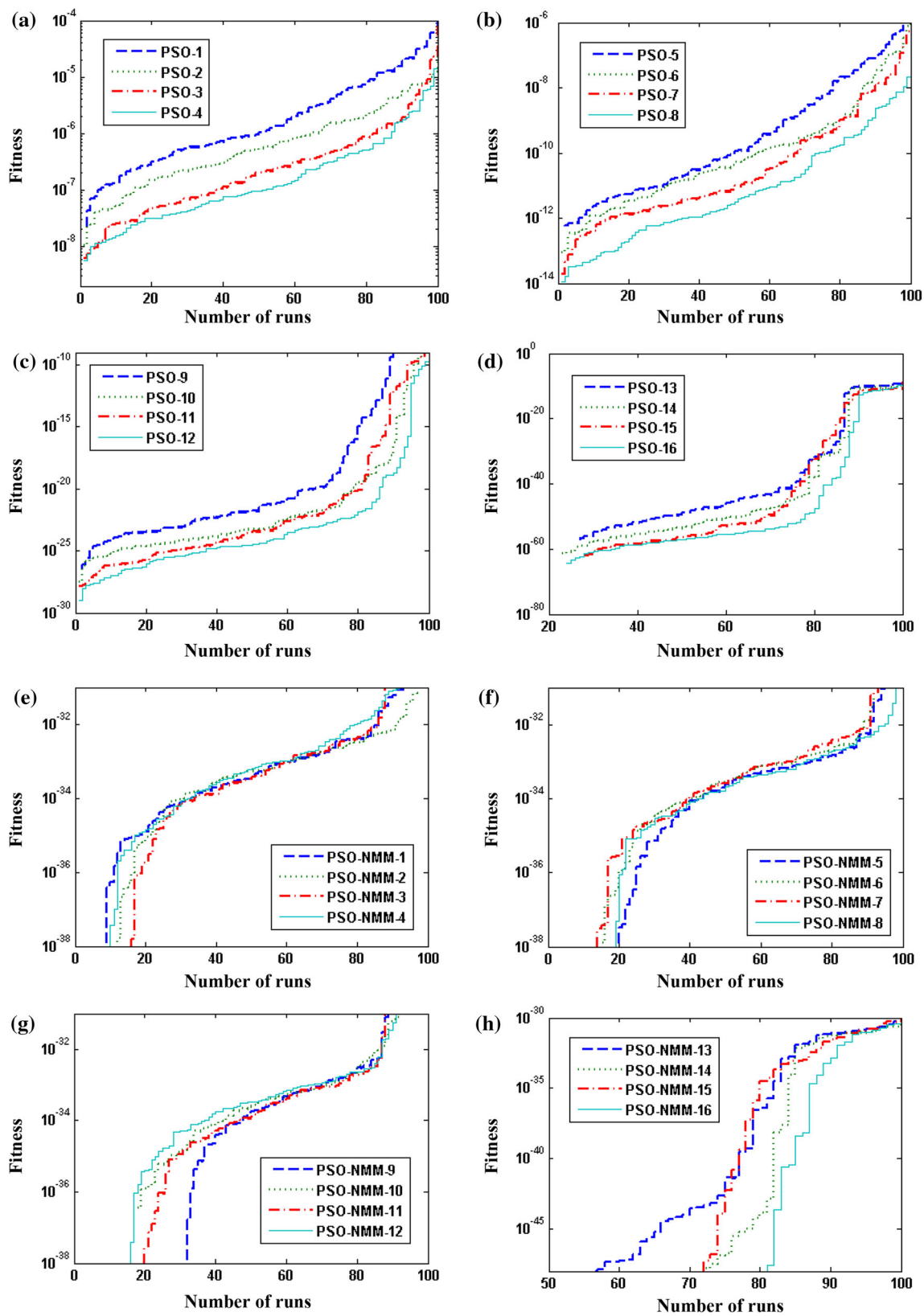


Fig. 9 Plot of sorted fitness, Oir, values for 100 independent runs of algorithms in the case of the neurophysiology application model presented in problem 3. **a–h** Are for PSO and PSO-NMMs, respectively

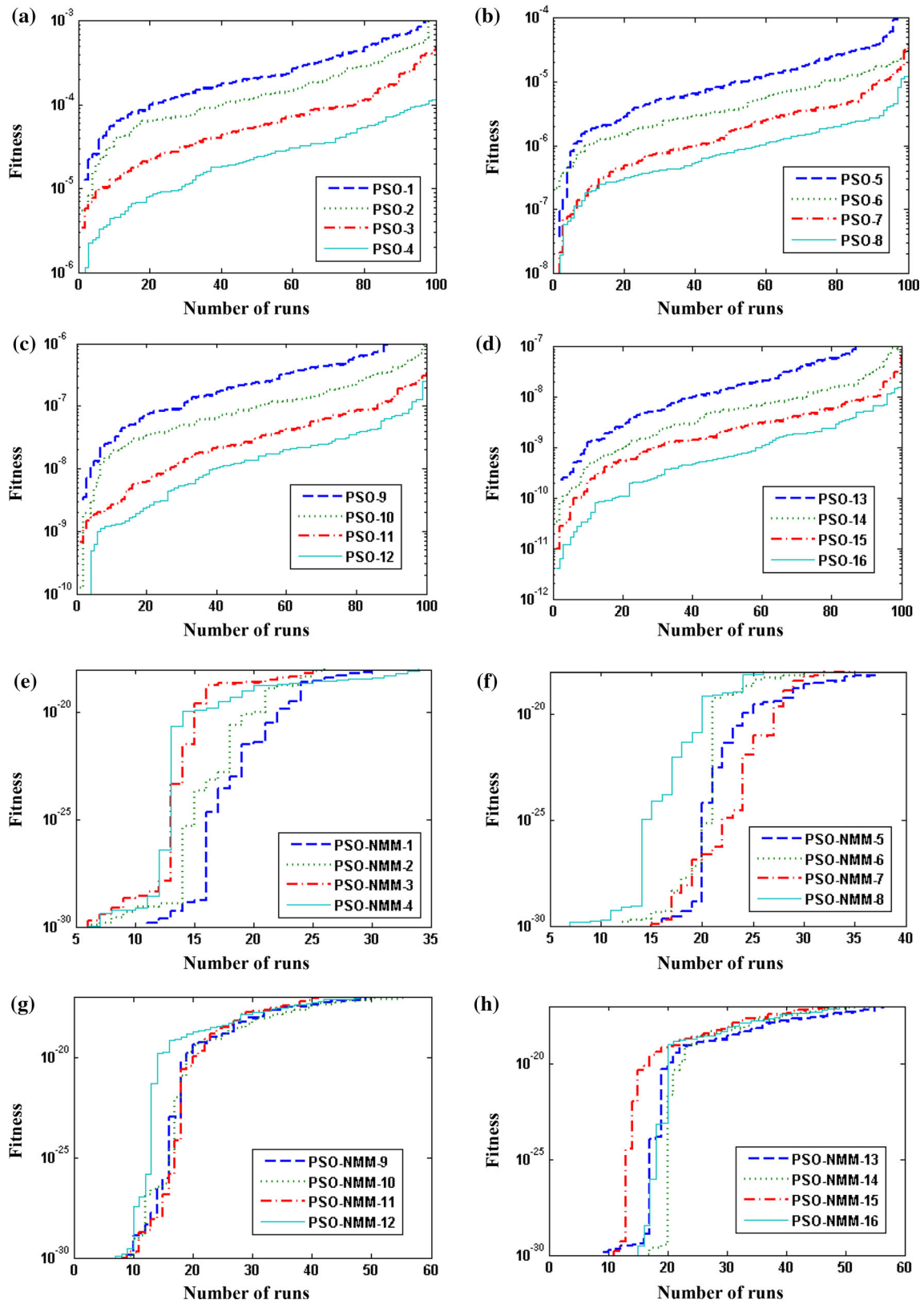


Fig. 10 Plot of sorted fitness, Oir, values for 100 independent runs of algorithms in the case of combustion theory application model presented in problem 4. **a–h** Are for PSO and PSO-NMMs, respectively

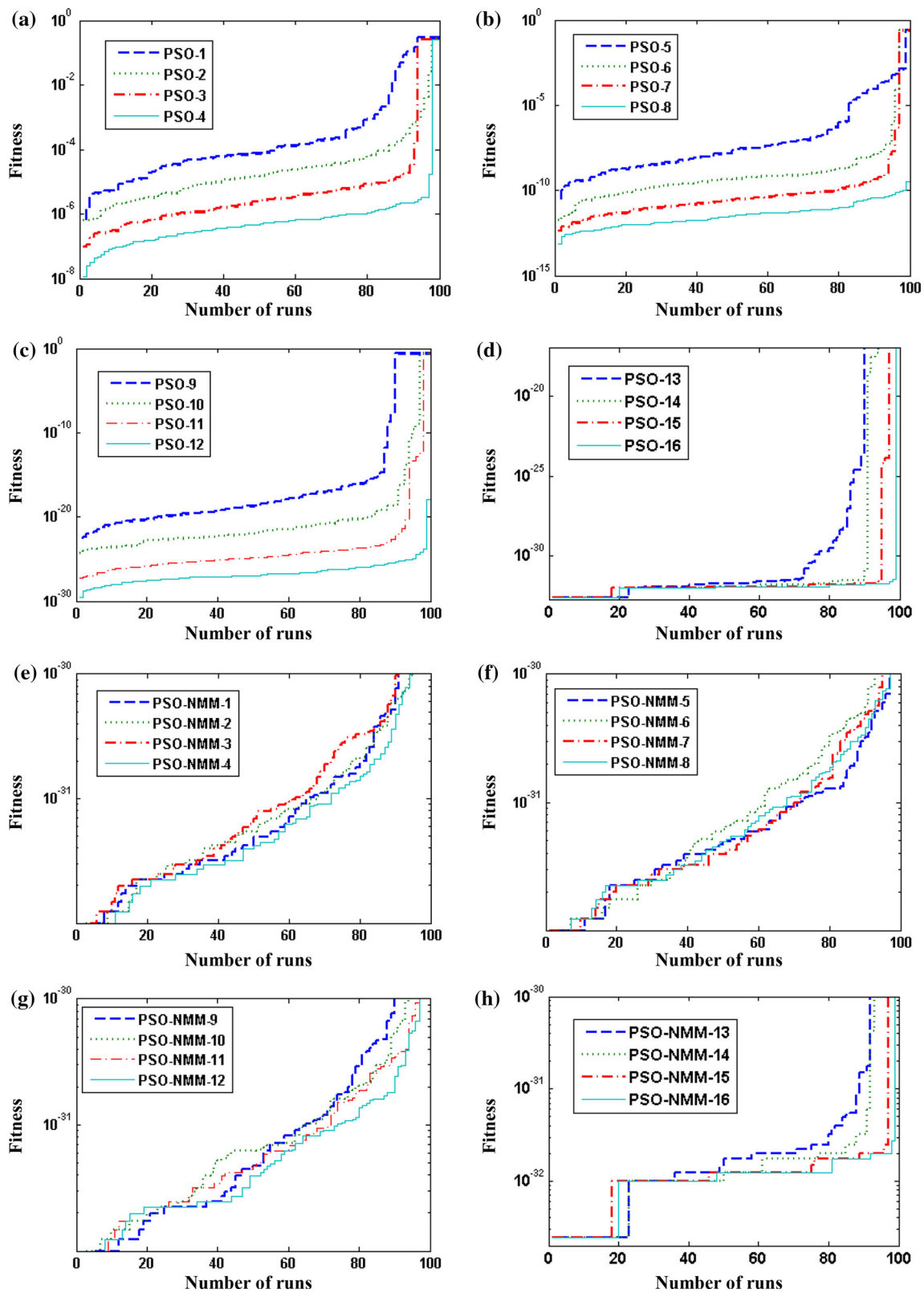


Fig. 11 Plot of sorted fitness, O_{ir} , values for 100 independent runs of algorithms in the case of economics modeling system presented in problem 5. **a–h** Are for PSO and PSO-NMMs, respectively

Table 8 Comparison of results on the basis of convergence analysis for 100 independent runs of PSO variants

Methods	Convergence criteria on fitness $\leq 10^{-xx}$														
	Problem 1			Problem 2			Problem 3			Problem 4			Problem 5		
	03	05	09	03	05	07	03	05	09	03	05	07	03	05	09
PSO-1	98	0	0	23	0	0	100	82	0	97	0	0	81	13	0
PSO-2	100	20	0	53	0	0	100	98	0	97	3	0	94	39	0
PSO-3	100	77	0	82	0	0	100	97	0	100	5	0	93	84	0
PSO-4	100	100	0	89	0	0	100	98	0	100	28	0	97	96	0
PSO-5	100	100	0	86	0	0	100	100	63	100	52	2	96	82	12
PSO-6	100	100	10	90	0	0	100	100	79	100	75	0	96	95	71
PSO-7	100	100	62	90	0	0	100	100	80	100	91	6	96	96	93
PSO-8	100	100	98	94	0	0	100	100	87	100	97	5	99	99	99
PSO-9	100	100	100	75	0	0	100	100	89	100	99	29	89	89	87
PSO-10	100	100	100	90	0	0	100	100	97	100	100	51	96	96	95
PSO-11	100	100	100	92	0	0	100	100	98	100	100	85	97	97	97
PSO-12	100	100	100	97	1	0	100	100	99	100	100	95	99	99	99
PSO-13	100	100	100	81	0	0	100	100	99	100	100	86	91	91	91
PSO-14	100	100	100	84	0	0	100	100	100	100	100	98	93	93	93
PSO-15	100	100	100	91	2	0	100	100	99	100	100	100	96	96	96
PSO-16	100	100	100	96	22	0	100	100	99	100	100	100	98	98	98

Table 9 Comparison of results on the basis of convergence analysis for 100 independent runs of PSO-NMMs

Methods	Convergence criteria on fitness $\leq 10^{-xx}$														
	Problem 1			Problem 2			Problem 3			Problem 4			Problem 5		
	03	05	09	03	05	07	03	05	09	03	05	07	03	05	09
PSO-NMM-1	100	100	100	81	81	78	99	99	97	21	15	10	93	93	90
PSO-NMM-2	100	100	98	81	81	81	99	99	99	19	14	5	97	97	94
PSO-NMM-3	100	100	98	90	90	90	99	99	96	14	12	5	93	93	91
PSO-NMM-4	100	100	99	91	91	88	100	100	98	13	12	5	97	97	94
PSO-NMM-5	100	100	99	86	86	86	100	100	100	23	19	15	98	98	96
PSO-NMM-6	100	100	100	90	90	90	100	100	97	20	20	11	96	96	92
PSO-NMM-7	100	100	98	92	92	92	100	100	98	26	21	14	96	96	94
PSO-NMM-8	100	100	99	98	98	98	99	99	99	19	13	6	99	99	96
PSO-NMM-9	100	100	99	78	78	78	100	100	98	18	15	8	89	89	89
PSO-NMM-10	100	100	98	92	92	92	100	100	94	18	15	8	96	96	93
PSO-NMM-11	100	100	100	94	94	91	100	100	94	19	16	8	97	97	96
PSO-NMM-12	100	100	99	97	97	96	100	100	91	13	12	6	99	99	96
PSO-NMM-13	100	100	100	81	81	80	100	100	99	19	16	8	91	91	91
PSO-NMM-14	100	100	100	85	85	83	100	100	99	21	19	15	93	93	92
PSO-NMM-15	100	100	100	91	91	90	100	100	100	16	12	10	96	96	96
PSO-NMM-16	100	100	100	97	97	97	100	100	99	19	17	14	98	98	98

5 Conclusions

On the basis of simulation performed in this study, following conclusions are drawn:

1. Sixteen variants PSO-1 to PSO-16 of swarm intelligence algorithms hybridized with Nelder–Mead

2. Validation of the performance of the proposed schemes based on 100 independent runs is established through consistently getting the small values of statistically method PSO-NMM-1 to PSO-NMM-16 are developed effectively to solve the nonlinear system of equations by taking different scenarios based on the number of particles in the swarm and the flights of the swarm.

Table 10 Complexity analysis based on average execution time consumed by the algorithms during optimization process for PSO and PSO-NMMs

Methods	Problem 1		Problem 2		Problem 3		Problem 4		Problem 5	
	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
PSO-1	0.0138	0.0091	0.0167	0.0359	0.0124	0.0012	0.0124	0.0025	0.0110	0.0014
PSO-2	0.0206	0.0017	0.0209	0.0019	0.0195	0.0022	0.0188	0.0016	0.0166	0.0016
PSO-3	0.0351	0.0021	0.0350	0.0022	0.0322	0.0021	0.0320	0.0024	0.0271	0.0013
PSO-4	0.0637	0.0028	0.0641	0.0026	0.0589	0.0029	0.0577	0.0028	0.0490	0.0024
PSO-5	0.0237	0.0013	0.0237	0.0015	0.0223	0.0013	0.0221	0.0015	0.0201	0.0013
PSO-6	0.0382	0.0024	0.0380	0.0017	0.0358	0.0025	0.0351	0.0022	0.0306	0.0013
PSO-7	0.0672	0.0035	0.0671	0.0034	0.0616	0.0033	0.0604	0.0028	0.0516	0.0022
PSO-8	0.1243	0.0048	0.1245	0.0044	0.1138	0.0052	0.1121	0.0050	0.0951	0.0037
PSO-9	0.0565	0.0024	0.0568	0.0029	0.0528	0.0020	0.0526	0.0027	0.0474	0.0030
PSO-10	0.0939	0.0050	0.0931	0.0036	0.0865	0.0037	0.0848	0.0033	0.0740	0.0034
PSO-11	0.1649	0.0052	0.1645	0.0066	0.1504	0.0067	0.1488	0.0066	0.1263	0.0041
PSO-12	0.3120	0.0238	0.3098	0.0146	0.2818	0.0146	0.2774	0.0132	0.2361	0.0140
PSO-13	0.1118	0.0046	0.1110	0.0023	0.1052	0.0042	0.1033	0.0038	0.0922	0.0026
PSO-14	0.1847	0.0053	0.1837	0.0054	0.1701	0.0049	0.1670	0.0041	0.1459	0.0048
PSO-15	0.3261	0.0056	0.3271	0.0083	0.3001	0.0084	0.2932	0.0060	0.2505	0.0045
PSO-16	0.6125	0.0105	0.6144	0.0127	0.5588	0.0135	0.5489	0.0120	0.4641	0.0081
PSO-NMM-1	1.4346	0.8203	0.9118	0.6160	0.8881	0.6835	0.4158	0.3828	1.0212	0.5467
PSO-NMM-2	1.5038	0.7740	1.0341	0.5256	0.8001	0.6959	0.4319	0.3754	1.0384	0.5519
PSO-NMM-3	1.5270	0.8330	1.0725	0.5295	0.8047	0.7157	0.3925	0.2990	1.0131	0.5835
PSO-NMM-4	1.4836	0.8058	1.0417	0.5350	0.8201	0.6945	0.4687	0.3383	1.0099	0.5722
PSO-NMM-5	1.4012	0.8035	0.9869	0.5033	0.7234	0.6552	0.3945	0.2821	0.9792	0.5236
PSO-NMM-6	1.6059	0.7158	1.0195	0.5331	0.6763	0.6733	0.4100	0.2921	0.8794	0.6115
PSO-NMM-7	1.4248	0.8204	1.0166	0.5337	0.7811	0.6893	0.4535	0.3059	1.0524	0.5245
PSO-NMM-8	1.5140	0.8031	1.1803	0.4667	0.7511	0.6868	0.4991	0.3366	1.0183	0.5644
PSO-NMM-9	1.4426	0.8017	0.9864	0.5694	0.7385	0.6836	0.3934	0.2499	0.9770	0.5238
PSO-NMM-10	1.4637	0.8489	1.0892	0.5304	0.7812	0.7114	0.4869	0.3127	1.1028	0.5200
PSO-NMM-11	1.5984	0.8566	1.1962	0.5532	0.9072	0.7248	0.5173	0.3056	1.1871	0.5163
PSO-NMM-12	1.7646	0.8976	1.3990	0.5597	0.9813	0.7289	0.6511	0.3197	1.2391	0.5992
PSO-NMM-13	1.6497	0.7882	1.0547	0.5734	1.1439	0.6868	0.4964	0.3184	1.2071	0.3816
PSO-NMM-14	2.1161	0.2817	1.0572	0.6186	1.1444	0.7249	0.5143	0.2656	1.3251	0.2912
PSO-NMM-15	2.5246	0.2101	1.4157	0.6135	1.2881	0.8334	0.7396	0.3969	1.5876	0.2480
PSO-NMM-16	2.6934	0.1900	1.6640	0.5764	1.5519	0.7673	0.9495	0.3055	1.7210	0.2546

operators in terms of mean and standard deviation in case of all five nonlinear systems.

- The accuracy and convergence of the given schemes are further evaluated through calculation of percentage convergent runs based on different precision levels, and the results show that hybrid approaches are almost 100 % convergent on the basis of fitness $\leq 10^{-03}$ in case of all five nonlinear systems. For stiff criteria, i.e., fitness $\leq 10^{-09}$, problems 1, 3 and 5 are still around 100 % convergent, while for problems 2 and 5, convergence rates degraded but remained in close vicinity of 85 %.

- Computational complexity in terms of mean and standard deviation values shows that variants of PSO with more number of particles in the swarm or number of flights of the swarm have long execution time and vice versa, whereas the complexity of the hybrid computing approaches PSO-NMMs is always on the higher side than that of PSO variants, but this limitation can be afforded due to their better performance on accuracy and convergence from the rest.

One may explore to apply the design variants of PSO and PSO-NMMs to solve stiff nonlinear systems,

differential–algebraic system, ordinary, partial, fractional differential equations and their systems to different physical science application which still remained unsolved by conventional or classical techniques. Recently introduced fractional variants of PSO [60–64] look as a promising area to be exploited to study the dynamics of nonlinear systems. Additionally, the better performance in terms of accuracy and convergence may achieved by exploring neural network model for solving these benchmark systems.

References

- Nazemi AR (2013) Solving general convex nonlinear optimization problems by an efficient neurodynamic model. *Eng Appl Artif Intell* 26:685–696
- Nazemi AR (2014) A neural network model for solving convex quadratic programming problems with some applications. *Eng Appl Artif Intell* 32:54–62
- Nazemi AR, Nazemi M (2014) A gradient-based neural network method for solving strictly convex quadratic programming problems. *Cogn Comput* 6:484–495
- Khan JA, Raja MAZ, Syam MI, Tanoli SAK, Awan SE (2015) Design and application of nature inspired computing approach for non-linear stiff oscillatory problems. *Neural Comput Appl* 26(7):1763–1780. doi:10.1007/s00521-015-1841-z
- Raja MAZ (2014) Solution of the one-dimensional Bratu equation arising in the fuel ignition model using ANN optimised with PSO and SQP. *Connect Sci* 26(3):195–214. doi:10.1080/09540091.2014.907555
- Raja MAZ, Ahmad SUI, Samar R (2014) Solution of the 2-dimensional Bratu problem using neural network, swarm intelligence and sequential quadratic programming. *Neural Comput Appl* 25:1723–1739. doi:10.1007/s00521-014-1664-3
- Nazemi AR, Omid F (2012) A capable neural network model for solving the maximum flow problem. *J Comput Appl Math* 236:3498–3513
- Raja MAZ, Khan JA, Haroon T (2014) Stochastic numerical treatment for thin film flow of third grade fluid using unsupervised neural networks. *J Chem Inst Taiwan* 48:26–39. doi:10.1016/j.jtice.2014.10.018
- Mall S, Chakraverty S (2014) Chebyshev neural network based model for solving Lane–Emden type equations. *Appl Math Comput* 247:100–114
- Raja MAZ (2014) Stochastic numerical techniques for solving Troesch's Problem. *Inf Sci* 279:860–873. doi:10.1016/j.ins.2014.04.036
- Raja MAZ, Samar R, Haroon T, Shah SM (2015) Unsupervised neural network model optimized with evolutionary computations for solving variants of nonlinear MHD Jeffery–Hamel problem. *Appl Math Mech* 36(12):1611–1638
- Mall Susmita, Chakraverty S (2015) Numerical solution of nonlinear singular initial value problems of Emden–Fowler type using Chebyshev Neural Network method. *Neurocomputing* 149:975–982
- Raja MAZ, Khan JA, Behloul D, Haroon T, Siddiqui AM, Samar R (2015) Exactly satisfying initial conditions neural network models for numerical treatment of first Painlevé equation. *Appl Soft Comput* 26:244–256. doi:10.1016/j.asoc.2014.10.009
- Raja MAZ, Khan JA, Shah SM, Bhahoal D, Samar R (2015) Painlevé Transcendent. *Neural Comput Appl* 26(5):1055–1071. doi:10.1007/s00521-014-1774-y
- Mall S, Chakraverty S (2015) Multi layer versus functional link single layer neural network for solving nonlinear singular initial value problems. In: *Proceedings of the third international symposium on women in computing and informatics* (pp 678–683). ACM
- Raja MAZ (2014) Numerical treatment for boundary value problems of pantograph functional differential equation using computational intelligence algorithms. *Appl Soft Comput* 24:806–821. doi:10.1016/j.asoc.2014.08.055
- Raja MAZ, Manzar MA, Samar R (2015) An efficient computational intelligence approach for solving fractional order Riccati equations using ANN and SQP". *Appl Math Model* 39(10):3075–3093. doi:10.1016/j.apm.2014.11.024
- Raja MAZ, Farooq U, Chaudhary NI, Wazwaz AM (2016) Stochastic numerical solver for nanofluidic problems containing multi-walled carbon nanotubes. *Appl Soft Comput* 38:561–586
- Raja MAZ, Shah FH, Khan AA, Khan NA (2016) Design of bio-inspired computational intelligence technique for solving steady thin film flow of Johnson–Segalman fluid on vertical cylinder for drainage problems. *J Taiwan Inst Chem Eng* 60:59–75
- Li Min (2014) A Polak–Ribière–Polyak method for solving large-scale nonlinear systems of equations and its global convergence. *Appl Math Comput* 248:314–322
- Amat S, Hernández-Verón MA, Rubio MJ (2014) Improving the applicability of the secant method to solve nonlinear systems of equations. *Appl Math Comput* 247:741–752
- Sharma JR, Arora H, Petković MS (2014) An efficient derivative free family of fourth order methods for solving systems of nonlinear equations. *Appl Math Comput* 235:383–393
- Ortega JM, Rheinboldt WC (1970) Iterative solution of nonlinear equations in several variables, vol 30. Siam, Philadelphia
- Kelley CT (2003) Solving nonlinear equations with Newton's method, vol 1. Siam, Philadelphia
- Kelley CT (1999) Iterative methods for optimization, vol 18. Siam, Philadelphia
- Campbell SL et al (1996) GMRES and the minimal polynomial. *BIT Numer Math* 36(4):664–675
- Darvishi MT, Barati A (2007) A fourth order method from quadrature formulae to solve the system of nonlinear equations. *Appl Math Comput* 188:257–261
- Knoll DA, Keyes DE (2004) Jacobian free Newton–Krylov methods: a survey of approaches and applications. *J Comput Phys* 193:357–397
- Saad Y, van der Vorst HA (2000) Iterative solution of non linear system in the 20th century. *J Comput Appl Math* 123:1–33
- Jafari H, Gejji VD (2006) Revised Adomian decomposition method for solving a system of nonlinear equation. *J Appl Math Comput* 175:1–7
- Abbasbandy S (2005) Extended Newton method for a system on non linear equations by modified domain decomposition method. *J Appl Math Comput* 170:648–656
- Vahidi AR, Javadi S, Khorasani SM (2012) Solving system of nonlinear equations by restarted Adomian's method. *J Appl Math Comput* 6:509–516
- Sharma JR, Guha RK, Gupta P (2013) Improved King's methods with optimal order of convergence based on rational approximations. *Appl Math Lett* 26(4):473–480
- Sharma JR, Gupta P (2014) An efficient fifth order method for solving systems of nonlinear equations. *Comput Math Appl* 67(3):591–601
- Sharma JR, Gupta P (2013) On some efficient techniques for solving systems of nonlinear equations. *Adv Numer Anal* 2013:11, Art ID 252798. doi:10.1155/2013/252798

36. Sharma JR, Arora H (2014) An efficient family of weighted-Newton methods with optimal eighth order convergence. *Appl Math Lett* 29:1–6
37. Arqub OA, Al-Smadi M, Momani S, Hayat T (2015) Numerical solutions of fuzzy differential equations using reproducing kernel Hilbert space method. *Soft Comput.* doi:10.1007/s00500-015-1707-4
38. Abu O (2015) Arqub, adaptation of reproducing kernel algorithm for solving fuzzy Fredholm–Volterra integrodifferential equations. *Neural Comput Appl.* doi:10.1007/s00521-015-2110-x
39. Arqub OA, Abo-Hammour Z (2014) Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm. *Inf Sci* 279:396–415
40. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *Proceedings of IEEE international conference on neural networks*, Perth, Australia, vol 4. IEEE Service Center, Piscataway, pp 1942–1948
41. Trelea IC (2003) The particle swarm optimization algorithm: convergence analysis and parameter selection. *Inf Process Lett* 85(6):317–325
42. Ping Tian D (2013) A review of convergence analysis of particle swarm optimization. *Int J Grid Distrib Comput* 6(6):117–128
43. Schmitt M, Wanka R (2015) Particle swarm optimization almost surely finds local optima. *Theor Comput Sci* 561:57–72
44. García-Gonzalo E, Fernández-Martínez JL (2012) A brief historical review of particle swarm optimization (PSO). *J Bioinform Intell Control* 1(1):3–16
45. Zhang Y, Wang S, Ji G (2015) A comprehensive survey on particle swarm optimization algorithm and its applications. *Math Probl Eng* 2015:38, Art ID 931256. doi:10.1155/2015/931256
46. Jiang S, Wang Y, Ji Z (2015) A new design method for adaptive IIR system identification using hybrid particle swarm optimization and gravitational search algorithm. *Nonlinear Dyn* 79(4):2553–2576
47. Lu Y et al (2015) Improved particle swarm optimization algorithm and its application in text feature selection. *Appl Soft Comput* 35:629–636
48. Saeid S, Sadeghi MT, Marvas MA (2014) High reliability estimation of product quality using support vector regression and hybrid meta-heuristic algorithms. *J Taiwan Inst Chem Eng* 45:2225–2232
49. Prawin J, Rao ARM, Lakshmi K (2016) Nonlinear parametric identification strategy combining reverse path and hybrid dynamic quantum particle swarm optimization. *Nonlinear Dyn* 84(2):797–815
50. Liu X (2016) Optimization design on fractional order PID controller based on adaptive particle swarm optimization algorithm. *Nonlinear Dyn* 84(1):379–386
51. Nelder JA, Mead R (1965) A simplex method for function minimization. *Comput J* 7(4):308–313
52. Dennis JE, Woods DJ (1987) Optimization on microcomputers: the Nelder-Mead simplex algorithm. In: Wouk A (ed) *New computing environments: microcomputers in large-scale computing*. SIAM, Philadelphia, pp 116–122
53. Grosan Crina, Abraham Ajith (2008) A new approach for solving nonlinear equations systems. *IEEE Trans Syst Man Cybern Part A Syst Hum* 38(3):698–714
54. Grosan Crina, Abraham Ajith (2008) Multiple solutions for a system of nonlinear equations. *Int J Innov Comput Inf Control* 4(9):2161–2170
55. Van Hentenryck P, McAllester D, Kapur D (1997) Solving polynomial systems using a branch and prune approach. *SIAM J Numer Anal* 34(2):797–827
56. Hong H, Stahl V (1994) Safe starting regions by fixed points and tightening. *Computing* 53(3/4):323–335
57. Meintjes K, Morgan AP (1990) Chemical equilibrium systems as numerical test problems. *ACM Trans Math Softw* 16(2):143–151
58. Verschelde J, Verlinden P, Cools R (1994) Homotopies exploiting Newton polytopes for solving sparse polynomial systems. *SIAM J Numer Anal* 31(3):915–930
59. Morgan AP (1987) *Solving polynomial system using continuation for scientific and engineering problems*. Prentice-Hall, Englewood Cliffs
60. Pires ES, Machado JT, de Moura Oliveira PB, Cunha JB, Mendes L (2010) Particle swarm optimization with fractional-order velocity. *Nonlinear Dyn* 61(1–2):295–301
61. Couceiro MS, Rocha RP, Ferreira NF, Machado JT (2012) Introducing the fractional-order Darwinian PSO. *Signal Image Video Process* 6(3):343–350
62. Couceiro MS, Martins FM, Rocha RP, Ferreira NM, Sivasundaram S (2012) Introducing the fractional order robotic Darwinian PSO. In: *AIP conference proceedings*, American institute of physics, vol 1493, no 1, p 242
63. Pires EJS, Machado JAT, de Moura Oliveira PB (2013) Entropy diversity in multi-objective particle swarm optimization. *Entropy* 15(12):5475–5491
64. Couceiro M, Ghamisi P (2016) Fractional-order Darwinian PSO. In: *Merkle D (ed) Fractional order Darwinian particle swarm optimization*. Springer, Berlin, pp 11–20