




Article

Nonlinear Hammerstein System Identification: A Novel Application of Marine Predator Optimization Using the Key Term Separation Technique

Khizer Mehmood ¹, Naveed Ishtiaq Chaudhary ^{2,*}, Zeshan Aslam Khan ¹, Khalid Mehmood Cheema ³, Muhammad Asif Zahoor Raja ², Ahmad H. Milyani ⁴ and Abdullah Ahmed Azhari ⁵

¹ Department of Electrical and Computer Engineering, International Islamic University Islamabad (IIUI), Islamabad 44000, Pakistan

² Future Technology Research Center, National Yunlin University of Science and Technology, 123 University Road, Section 3, Douliou, Yunlin 64002, Taiwan

³ Department of Electronic Engineering, Fatima Jinnah Women University, Rawalpindi 46000, Pakistan

⁴ Department of Electrical and Computer Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia

⁵ The Applied College, King Abdulaziz University, Jeddah 21589, Saudi Arabia

* Correspondence: chaudni@yuntech.edu.tw

Abstract: The mathematical modelling and optimization of nonlinear problems arising in diversified engineering applications is an area of great interest. The Hammerstein structure is widely used in the modelling of various nonlinear processes found in a range of applications. This study investigates the parameter optimization of the nonlinear Hammerstein model using the abilities of the marine predator algorithm (MPA) and the key term separation technique. MPA is a population-based metaheuristic inspired by the behavior of predators for catching prey, and utilizes Brownian/Levy movement for predicting the optimal interaction between predator and prey. A detailed analysis of MPA is conducted to verify the accurate and robust behavior of the optimization scheme for nonlinear Hammerstein model identification.

Keywords: nonlinear systems; parameter estimation; swarm optimization; marine predator algorithm

MSC: 93C10; 68T20



Citation: Mehmood, K.; Chaudhary, N.I.; Khan, Z.A.; Cheema, K.M.; Raja, M.A.Z.; Milyani, A.H.; Azhari, A.A. Nonlinear Hammerstein System Identification: A Novel Application of Marine Predator Optimization Using the Key Term Separation Technique. *Mathematics* **2022**, *10*, 4217. <https://doi.org/10.3390/math10224217>

Academic Editors: Juan Francisco Sánchez-Pérez, Gonzalo García Ros and Manuel Conesa

Received: 3 October 2022

Accepted: 8 November 2022

Published: 11 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The identification of nonlinear systems is considered a challenging task because of the inherent stiffness and complex system representation [1,2]. A nonlinear system modelled through the block-oriented Hammerstein structure is relatively a simple approach, where a nonlinear block is followed by a linear dynamical subsystem [3,4]. The identification of Hammerstein models is of the utmost importance owing to their ability to model various nonlinear processes [5–7]. The pioneering work for nonlinear Hammerstein systems was presented by Narendra et al., in the 1960s, by proposing an iterative identification algorithm [8]. Chang et al. proposed a non-iterative scheme [9]. Vörös et al. presented a key term separation principle for the identification of a Hammerstein model [10,11]. Ding et al. introduced hierarchical least squares [12,13] and hierarchical gradient descent algorithms [14] for the Hammerstein model. Chaudhary et al. introduced the concept of fractional gradient algorithms for Hammerstein systems by proposing normalized fractional least mean square (FLMS) [15], sign FLMS [16], multi-innovation FLMS [17], hierarchical quasi fractional gradient descent [18], and fractional hierarchical gradient descent [19] algorithms.

Different system identification scenarios have multimodal error surfaces, and traditional gradient descent-based approaches may converge to a sub-optimal solution [20,21]. The system identification may be expressed as an optimization problem that can be

solved through a stochastic search methodology such as evolutionary and swarm heuristics [20]. The research community proposed these population-based heuristics for Hammerstein model identification. For example, Raja et al. exploited genetic algorithms [22]; Mehmood et al. presented differential evolution [23], weighted differential evolution [24], and backtracking search heuristics [25]; and Altaf et al. presented adaptive evolutionary heuristics [26]. The swarm-based optimization heuristics for Hammerstein system identification include particle swarm optimization [27], cuckoo search algorithm [28], snake optimizer algorithm [29], and fractional swarming optimization heuristics [30], etc.

Metaheuristics have been applied in different engineering optimization problems. Heuristics are mainly categorized as (a) swarm intelligence, (b) physics-based, or (c) evolutionary algorithms. Swarm intelligence mimics the behavior of herds present in nature. Physics-based methods are inspired by the laws of physics, while evolutionary methods are inspired by biological processes in nature. The significant methods proposed in all of these domains are summarized in Table 1.

Table 1. Classification of metaheuristics.

Domain	Technique
Swarm Intelligence	Particle swarm optimization (PSO) [31,32] Dwarf Mongoose optimization (DMO) [33,34] Ant Colony optimization (ACO) [35,36] Cuckoo search [37,38] Aquila Optimizer (AO) [39] Spider monkey optimization [40]
Physics based	Simulated Annealing [41,42] Gravitational search algorithm [43,44] Circle search algorithm [45,46] Colliding bodies optimizer [47] Transient search optimizer [48] Big bang big crunch [49]
Evolutionary	Differential Evolution [50,51] Genetic algorithm [52,53] Tree growth algorithm [54] Arithmetic optimization algorithm [55,56] Genetic programming [57] Evolutionary strategy [58]

Among the various swarm-intelligence-based techniques, the Marine Predator Algorithm (MPA) [59] has recently been proposed and applied in various applications such as optimal power flow [60], economic load dispatch [61], wireless sensor network [62], and path planning [63]. Wadood et al. [64] applied MPA to minimize breakdown in an electrical power system. Lu et al. [65] applied MPA along with the convolutional neural network for the optimal detection of lung cancer. Hoang et al. [66] utilized MPA in the optimization of hyperparameters of support vector machines for remote sensing. Yang et al. [67] applied multi strategy MPA for semi supervised extreme machine learning for classification.

In this study, we explored MPA for effective parameter estimation of the Hammerstein structure. The detailed performance evaluation of the proposed scheme for Hammerstein identification was conducted for different noise conditions. The reliability of the proposed approach in comparison with the other recently introduced metaheuristics was established through detailed analyses based on multiple independent executions and statistical tests.

The remainder of the paper is outlined as follows: Section 2 describes the Hammerstein model structure. Section 3 presents the MPA methodology with a flow chart description. Section 4 provides the results of detailed simulations in terms of graphical and tabular representation. Finally, Section 5 concludes the study by presenting the main findings of the current investigation.

2. System Model

Considering the Hammerstein output error (HOE) model, whose output is written as

$$y_m(t) = y(t) + \varepsilon(t), \tag{1}$$

where $y(t)$ is the noise free output and $\varepsilon(t)$ is the additive white noise. The noise free output of HOE model is presented in (2)

$$y(t) + \sum_{s=1}^{n_g} g_s y(t-s) = \sum_{s=0}^{n_h} h_s \bar{\mu}(t-s), \tag{2}$$

where $\bar{\mu}(t)$ is the output of nonlinear block with basis (k_1, k_2, \dots, k_p) and o_1, o_2, \dots, o_p are the corresponding parameters represented as

$$\bar{\mu}(t) = \sum_{l=1}^p o_l k_l(\mu(t)).$$

By choosing $\bar{\mu}(t)$ as the keyterm and applying the keyterm separation principle [18,19], and letting $h_0 = 1$, then (2) can be expressed as

$$\begin{aligned} y(t) &= -\sum_{s=1}^{n_g} g_s y(t-s) + \sum_{s=0}^{n_h} h_s \bar{\mu}(t-s), \\ y(t) &= -\sum_{s=1}^{n_g} g_s y(t-s) + h_0 \bar{\mu}(t) + \sum_{s=1}^{n_h} h_s \bar{\mu}(t-s), \\ y(t) &= -\sum_{s=1}^{n_g} g_s y(t-s) + \sum_{s=1}^{n_h} h_s \bar{\mu}(t-s) + \sum_{l=1}^p o_l k_l(\mu(t)). \end{aligned} \tag{3}$$

The parameter vectors $\mathbf{g}, \mathbf{h}, \mathbf{o}$ and information vector $\delta_g(t), \delta_h(t), \mathbf{k}(t)$ are presented in (4) and (5), respectively

$$\mathbf{g} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_{n_g} \end{bmatrix} \in \mathbb{R}^{n_g}, \mathbf{h} = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_{n_h} \end{bmatrix} \in \mathbb{R}^{n_h}, \mathbf{o} = \begin{bmatrix} o_1 \\ o_2 \\ \vdots \\ o_p \end{bmatrix} \in \mathbb{R}^p, \tag{4}$$

$$\delta_g(t) = \begin{bmatrix} -y(t-1) \\ -y(t-2) \\ \vdots \\ -y(t-n_g) \end{bmatrix} \in \mathbb{R}^{n_g}, \delta_h(t) = \begin{bmatrix} \bar{\mu}(t-1) \\ \bar{\mu}(t-2) \\ \vdots \\ \bar{\mu}(t-n_h) \end{bmatrix} \in \mathbb{R}^{n_h}, \mathbf{k}(t) = \begin{bmatrix} k_1(\mu(t)) \\ k_2(\mu(t)) \\ \vdots \\ k_p(\mu(t)) \end{bmatrix} \in \mathbb{R}^p. \tag{5}$$

Then, (3) can be expressed as presented in (6)

$$y(t) = \delta_g^T(t)\mathbf{g} + \delta_h^T(t)\mathbf{h} + \mathbf{k}^T(t)\mathbf{o}. \tag{6}$$

3. Methodology

In this section, the MPA-based methodology for the parameter estimation of the Hammerstein model is presented.

3.1. Marine Predator Algorithm

The MPA is a population-based metaheuristic inspired from the behavior of predators when catching prey [59]. It utilizes Brownian and Levy movement for the optimal interaction between predator and prey. Its mathematical model, pseudocode, and algorithm flowchart are presented below.

3.1.1. Formulation

MPA starts with the initialization of the population uniformly distributed over the search space, as presented in (7).

$$J_0 = J_{\min} + \text{rand} (J_{\max} - J_{\min}). \tag{7}$$

Two matrices were also constructed, the Elite matrix (EM) and Prey matrix (PM), which consist of the position vector with the best fitness and the proposed random positions during initialization, respectively, as presented in (8) and (9).

$$EM = \begin{bmatrix} J_{1,1}^I & \cdots & J_{1,b}^I \\ \vdots & \ddots & \vdots \\ J_{N_p,1}^I & \cdots & J_{N_p,b}^I \end{bmatrix}_{N_p \times b} \tag{8}$$

$$PM = \begin{bmatrix} J_{1,1} & \cdots & J_{1,b} \\ \vdots & \ddots & \vdots \\ J_{N_p,1} & \cdots & J_{N_p,b} \end{bmatrix}_{N_p \times b}, \tag{9}$$

where J^I represents the top predator and $J_{a,b}$ is the dimension of the prey.

3.1.2. Optimization

The optimization of MPA consists of three phases, as presented below.

Phase I

In this phase, the predator moves faster than the prey by using Brownian movement for first third of the iterations. The prey matrices are updated as presented in (10) and (11)

$$\vec{SS}_i = \vec{R}_B \otimes \left(\vec{EM}_i - (\vec{R}_B \otimes \vec{PM}_i) \right), i = 1, 2, \dots, N_p, \tag{10}$$

$$\vec{PM}_i = \vec{PM}_i + (C \cdot \vec{R} \otimes \vec{SS}_i), \tag{11}$$

where R_B is the normal distributed vector representing the Brownian movement, $C = 0.5$, and R is a vector $\in [0.1]$.

Phase II

In this phase, both the prey and predator are moving at the same pace. The prey moves on Levy movement, while the predator moves using Brownian movement between one third and two third of the maximum iterations. The updated matrices are presented in (12)–(15)

$$\vec{SS}_i = \vec{R}_L \otimes \left(\vec{EM}_i - (\vec{R}_L \otimes \vec{PM}_i) \right), i = 1, 2, \dots, N_p/2, \tag{12}$$

$$\vec{PM}_i = \vec{PM}_i + (C \cdot \vec{R} \otimes \vec{SS}_i), \tag{13}$$

$$\vec{SS}_i = \vec{R}_B \otimes \left(\vec{R}_B \otimes (\vec{EM}_i - \vec{PM}_i) \right), i = 1, 2, \dots, N_p, \tag{14}$$

$$\vec{PM}_i = \vec{EM}_i + (C \cdot DF \otimes \vec{SS}_i), \tag{15}$$

where $DF = (1 - \frac{It}{T})^{(2\frac{It}{T})}$ is adaptive factor to control stepsize of the predator movement.

Phase III

In this phase, the predator is moving faster than the prey. The predator moves by using Levy movement and the prey matrix is updated as presented in (16) and (17)

$$\vec{SS}_i = \vec{R}_L \otimes \left(\vec{R}_L \otimes (\vec{EM}_i - \vec{PM}_i) \right), i = 1, 2, \dots, N_p, \tag{16}$$

$$\vec{PM}_i = \vec{EM}_i + (C.DF \otimes \vec{SS}_i). \tag{17}$$

3.1.3. Fish Aggregating Devices' (FAD's) Effect

The fish aggregating devices' (FAD's) effect is added in MPA to simulate the natural behavior of fish, as they spend 80% of their time in immediate locations and 20% of the time searching for other spaces [59], as presented in (18) and (19)

$$\vec{PM}_i = \begin{cases} \vec{PM}_i + DF[J_{\min} + \vec{R} \otimes (J_{\max} - J_{\min})] \otimes \vec{U}, & \text{if } q \leq \text{FAD's} \\ \vec{PM}_i + [\text{FAD's}(1 - q) + q](\vec{PM}_{q_1} - \vec{PM}_{q_2}), & \text{if } q > \text{FAD's} \end{cases} \tag{18}$$

$$\tag{19}$$

where $\text{FAD's} = 0.2$, \vec{U} is a binary vector, q is a random number $\in [0, 1]$, J_{\max} and J_{\min} are the upper and lower bounds, respectively, q_1 and q_2 are subscripts of the prey matrix. The flowchart for MPA is shown in Figure 1.

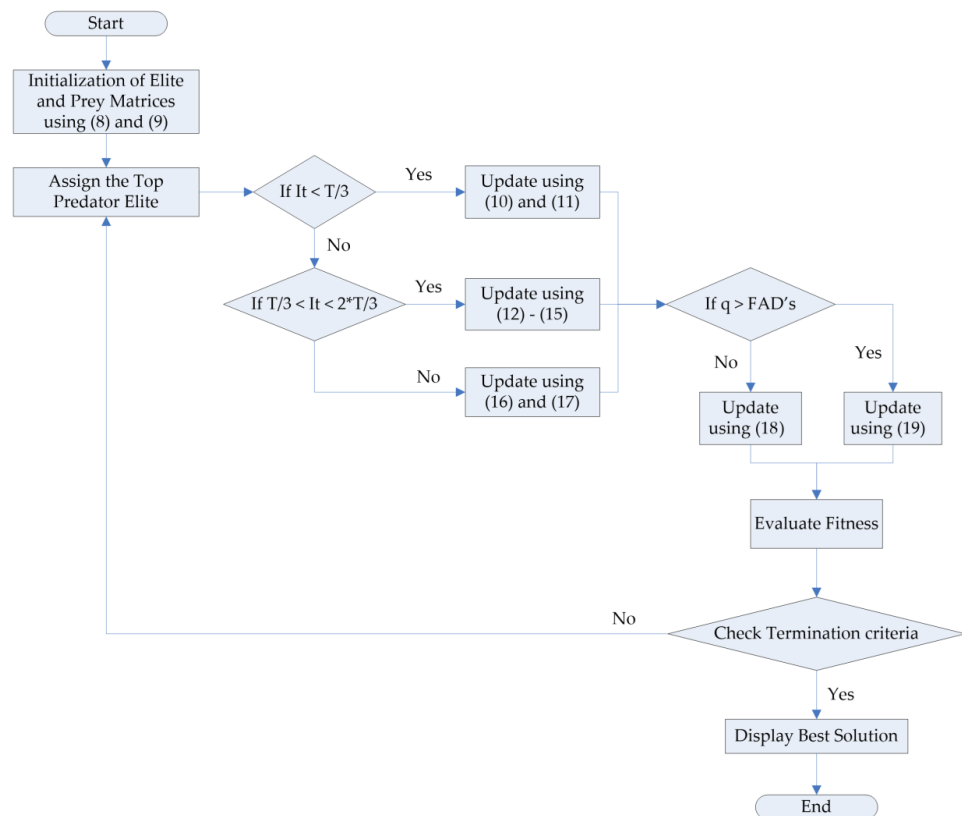


Figure 1. MPA flowchart.

4. Performance Analysis

The detailed performance analysis of MPA for the HOE model is presented in this section. HOE model identification was conducted on numerous noise levels, several generations, and population sizes. The proposed scheme is examined in terms of accuracy, and is measured by the fitness function presented in (20)

$$\text{Fitness Function} = \frac{1}{J} \sum_{w=1}^J [y_m(t_w) - \widetilde{y}_m(t_w)]^2, \tag{20}$$

here, \widetilde{y}_m is the estimated response through MPA, and y_m is the actual response of the HOE model.

4.1. Case Study 1

For the simulation study, we considered the following HOE model

$$y_m(t) = y(t) + \varepsilon(t), \tag{21}$$

$$y(t) + g_1y(t-1) + g_2y(t-2) = h_0\bar{\mu}(t) + h_1\bar{\mu}(t-1) + h_2\bar{\mu}(t-2), h_0 = 1,$$

$$\bar{\mu}(t) = o_1k_1(\mu(t)) + o_2k_2(\mu(t)) + o_3k_3(\mu(t)) = o_1\mu(t) + o_2\mu^2(t) + o_3\mu^3(t) \tag{22}$$

$$\mathbf{g} = [g_1, g_2]^T = [-1.100, 0.900]^T \tag{23}$$

$$\mathbf{h} = [h_1, h_2]^T = [-0.800, -0.600]^T \tag{24}$$

$$\mathbf{o} = [o_1, o_2, o_3]^T = [-0.900, 0.600, 0.200]^T \tag{25}$$

$$\boldsymbol{\omega} = [g_1, g_2, h_1, h_2, o_1, o_2, o_3]^T = [-1.100, 0.900, -0.800, -0.600, -0.900, 0.600, 0.200]^T \tag{26}$$

The input is taken as zero mean unit variance random signal with the data length as 25 and the number of generations taken as 500 and 1000. The performance of MPA is assessed by introducing AWGN noise with five variations, i.e., [60 dB, 50 dB, 40 dB, 30 dB, 10 dB] to the HOE model for two variations of generation (T) [500, 1000] and populations (Np) [50, 100], and the fitness plots are demonstrated in Figure 2.

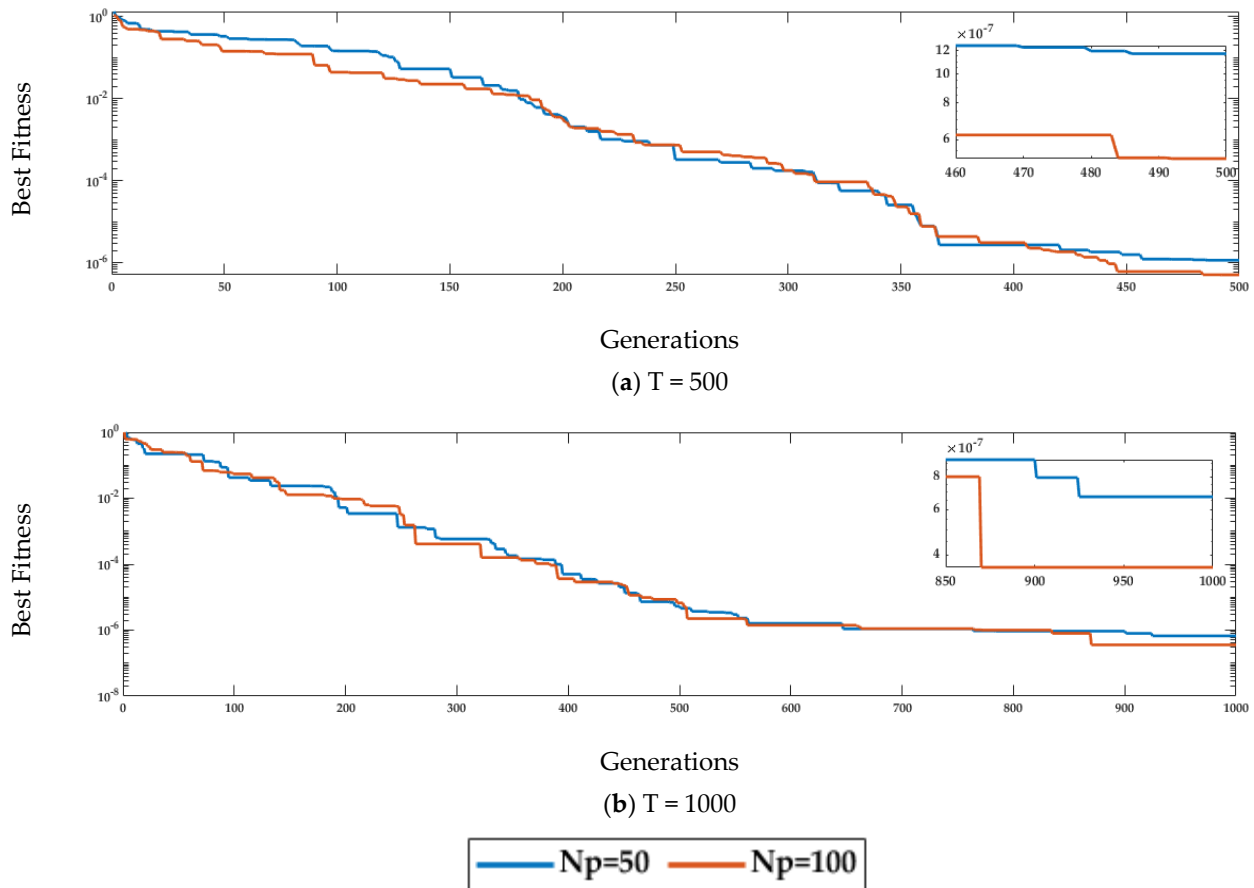


Figure 2. Fitness plots for MPA w.r.t population sizes for case study 1.

The fitness curves in Figure 2a correspond to the best fitness of the MPA algorithm for generation $T = 500$, while Figure 2b correspond to same variation for generation $T = 1000$. It is noted from Figure 2a,b that the fitness of MPA for the two generation variations, i.e., [500, 1000], reduces with the increase in population sizes.

The performance of MPA is also evaluated in terms of the generation variations as demonstrated by the fitness curves in Figure 3.

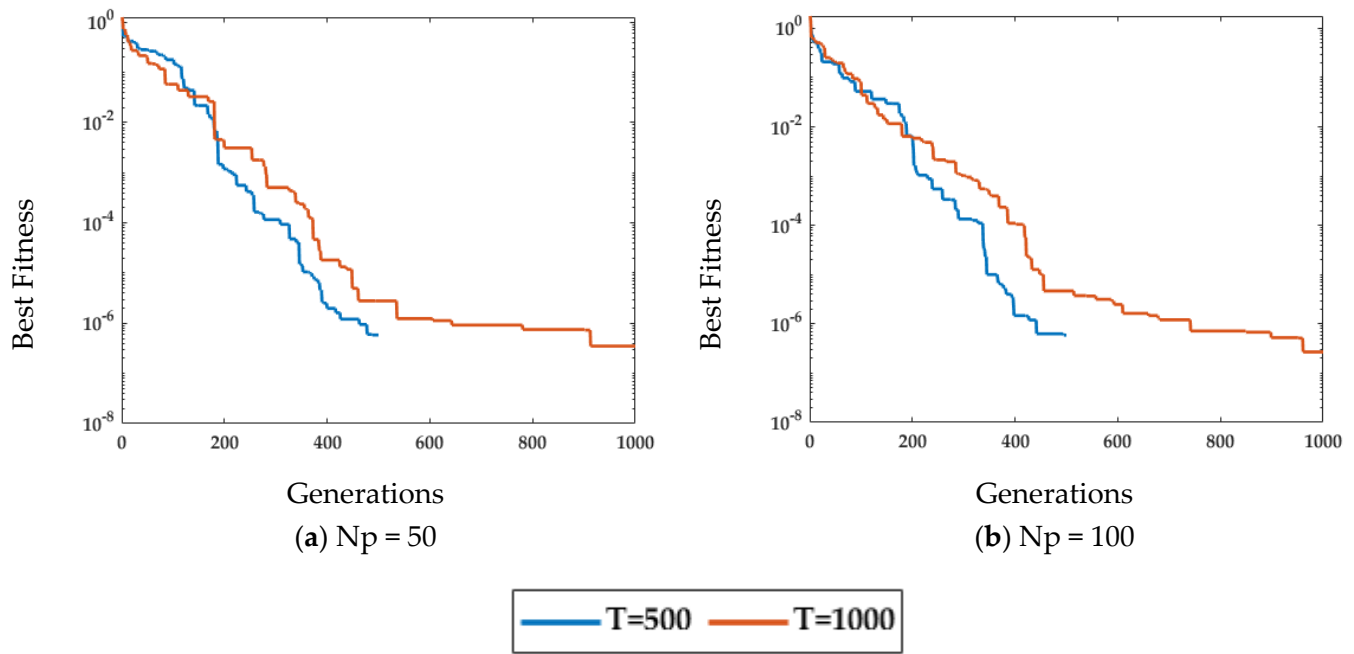


Figure 3. Fitness plots for MPA w.r.t generation variations for case study 1.

The fitness curves in Figure 3a correspond to the best fitness of the MPA algorithm for population size $N_p = 50$, whereas Figure 3b is related to the same variation for population $N_p = 100$. It is noted from Figure 3a,b that the fitness of MPA for the two population sizes, i.e., [50, 100], is reduced significantly with the increase in generations.

The behavior of MPA for different noise variations is evaluated by fixing the population size [50, 100] and changing the generation size [500, 1000] for five values of noise levels [60 dB, 50 dB, 40 dB, 30 dB, 10 dB], and the fitness plots are presented in Figure 4.

It is noted from the fitness plots demonstrated in Figure 4a–d that for a fixed population size and number of generations, the fitness achieved by MPA for a low level of noise, i.e., [60 dB], is quite a lot less compared with the fitness for a high noise level, i.e., [50 dB, 40 dB, 30 dB, 10 dB]. Yet, MPA achieves the minimum value of fitness for the smallest value of noise, i.e., [60 dB] for $N_p = 100$ and $T = 1000$. Therefore, it is confirmed from the curves in Figure 4 that the performance of MPA is lower for higher noise values.

To further investigate MPA, it is compared with other metaheuristics, namely the prairie dog optimization (PDO) [68], sine cosine algorithm (SCA) [69], and whale optimization algorithm (WOA) [70], for $N_p = 100$ and $T = 1000$, and noise levels of [60 dB, 50 dB, 40 dB, 30 dB, 10 dB] are presented in Figure 5.

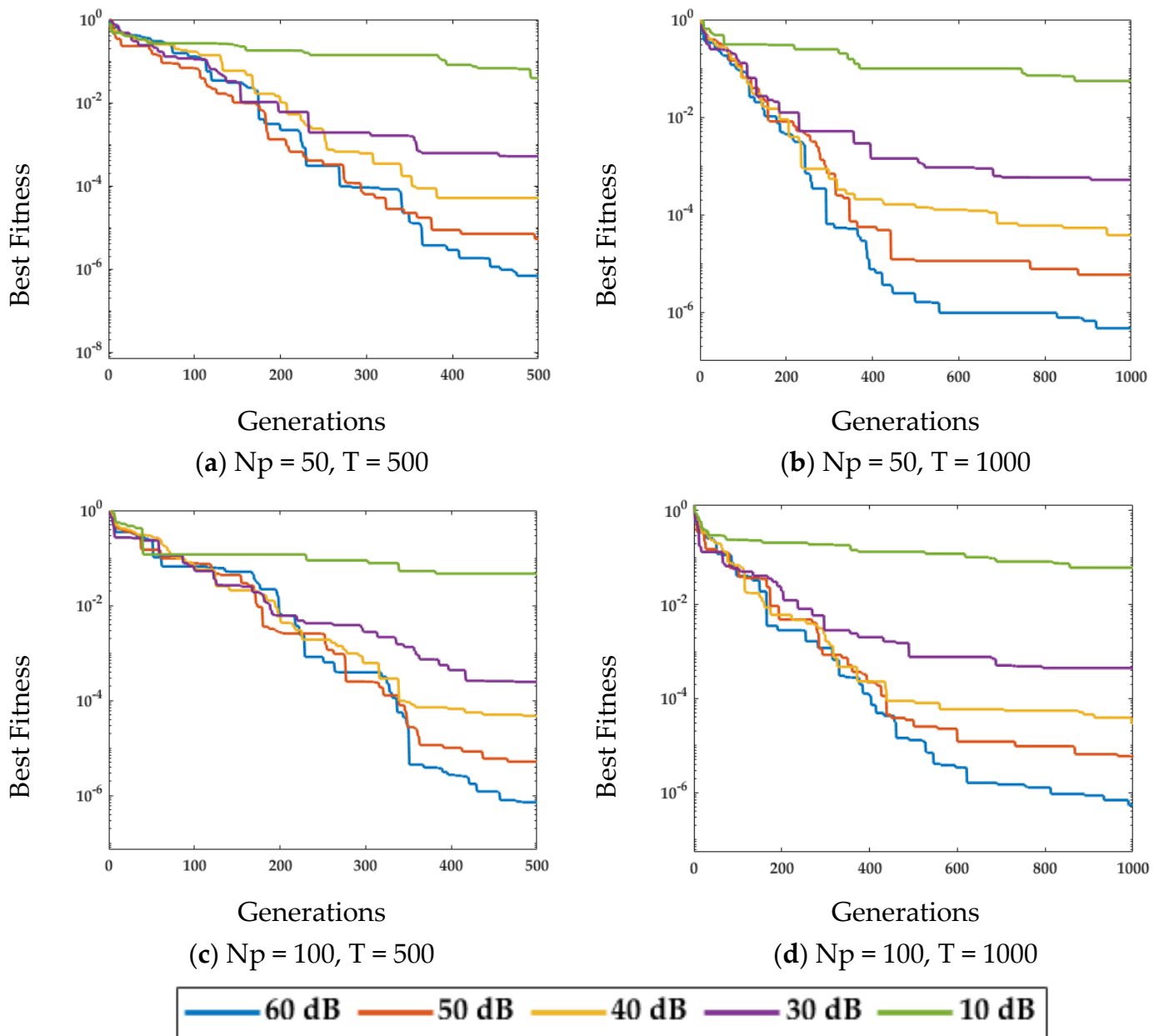


Figure 4. Fitness plots for MPA w.r.t noise levels for case study 1.

Figure 5a–e demonstrates the convergence of MPA with SCA, WOA, and PDO at $N_p = 100$ and $T = 1000$ over all of the noise levels, i.e., [60 dB, 50 dB, 40 dB, 30 dB, 10 dB]. It is noted from Figure 5a–e that with the increase in noise level, the fitness value also increases. However, MPA achieves the lowest fitness compared with SCA, WOA, and PDO for all of the noise levels.

MPA is further evaluated statistically against PDO, WOA, and SCA at $N_p = 100$ and $T = 1000$ for 20 independent runs. Figure 6 shows the fitness value of MPA, SCA, WOA, and PDO on run#1, run#10, and run#20 for all of the noise levels.

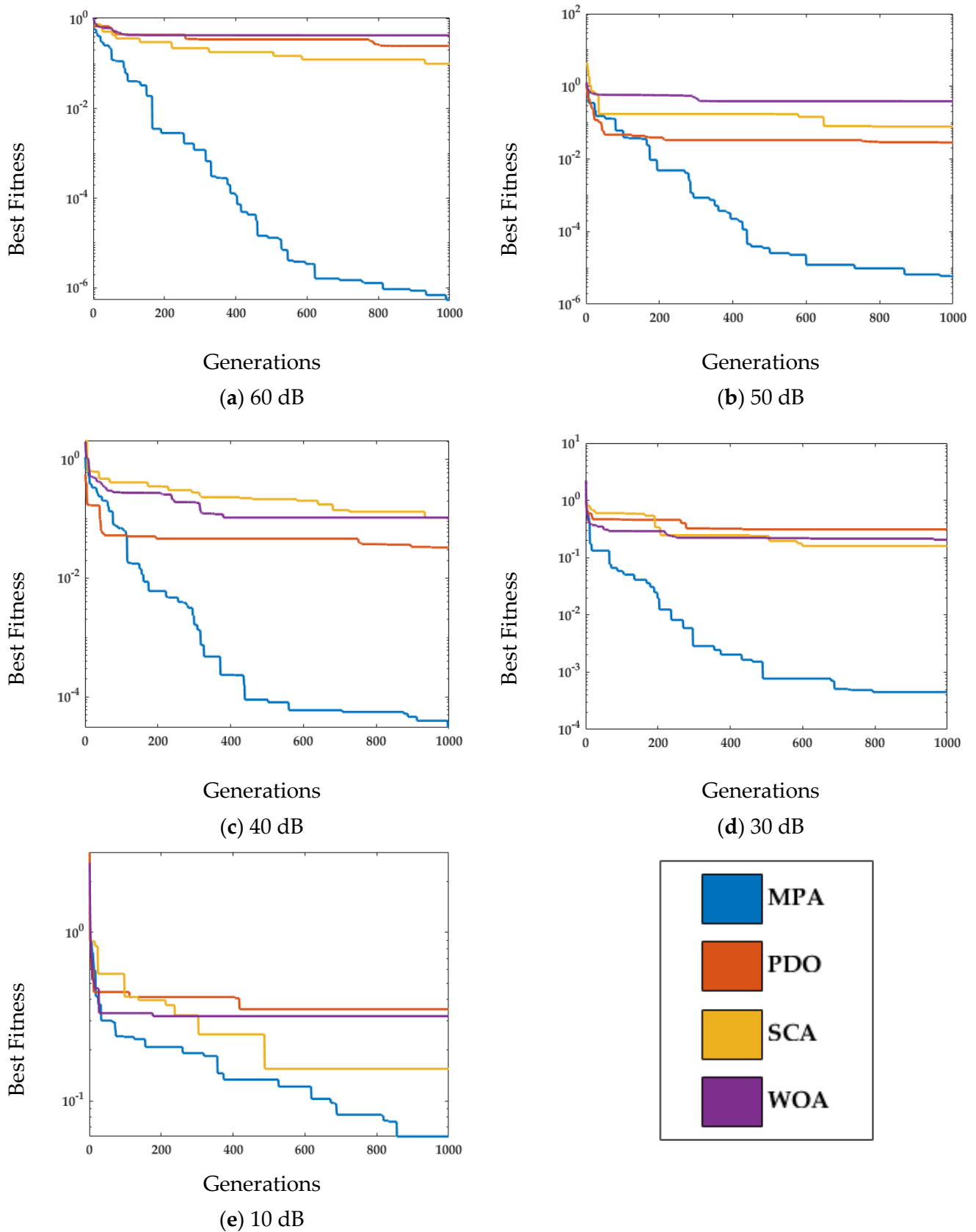


Figure 5. Fitness plot comparison of MPA with PDO, SCA, and WOA at $N_p = 100$ and $T = 1000$ for case study 1.

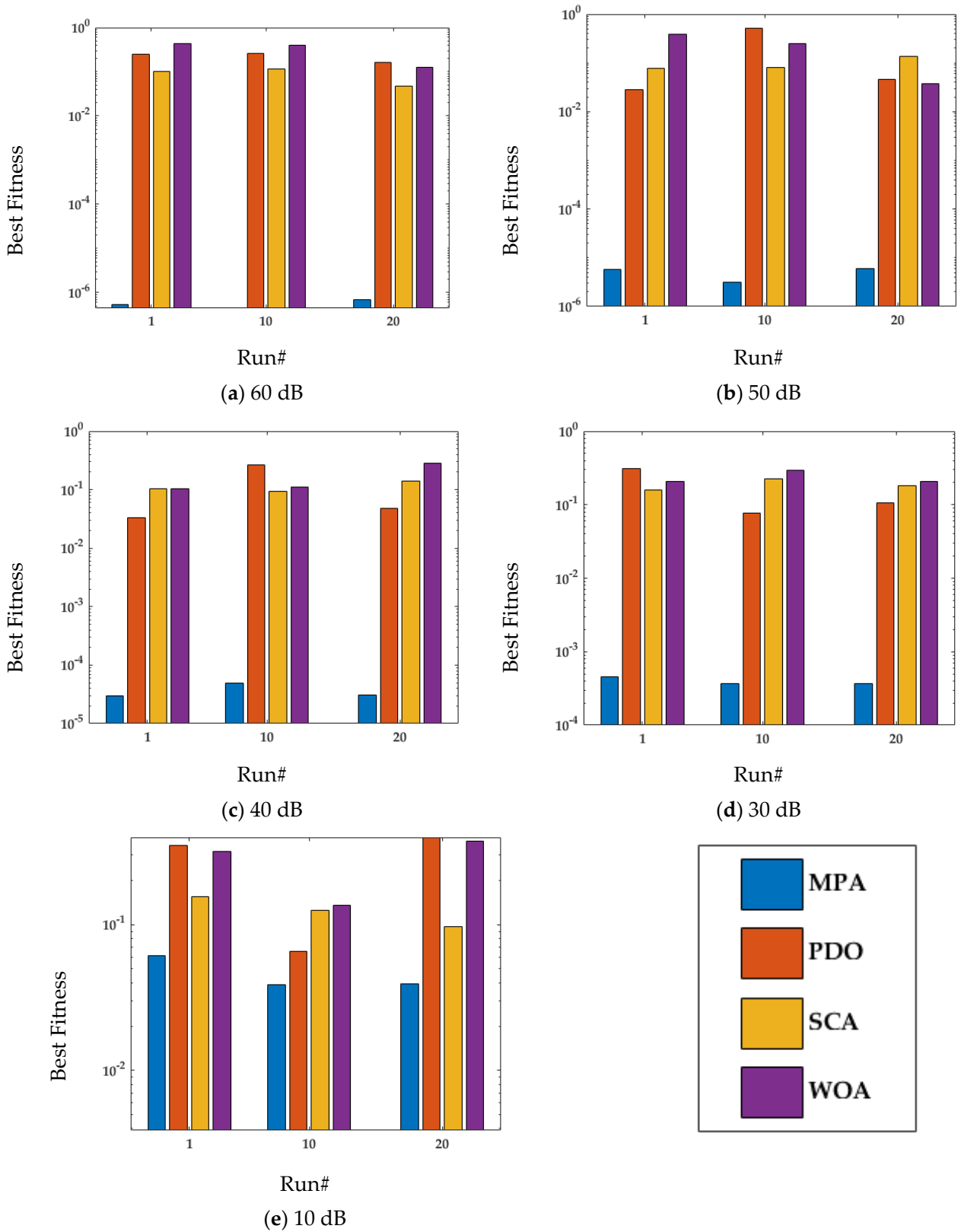


Figure 6. Run# fitness comparison of MPA with PDO, SCA and WOA at $N_p = 100$ and $T = 1000$ for case study 1.

It is noted from Figure 6a–e that MPA achieves the lowest fitness compared with SCA, WOA, and PDO for different independent runs for all of the noise levels. Moreover, it is

also observed that fitness increases with the increase in noise levels for all of the methods. Further statistical analysis on all of the independent runs is demonstrated in Figure 7.

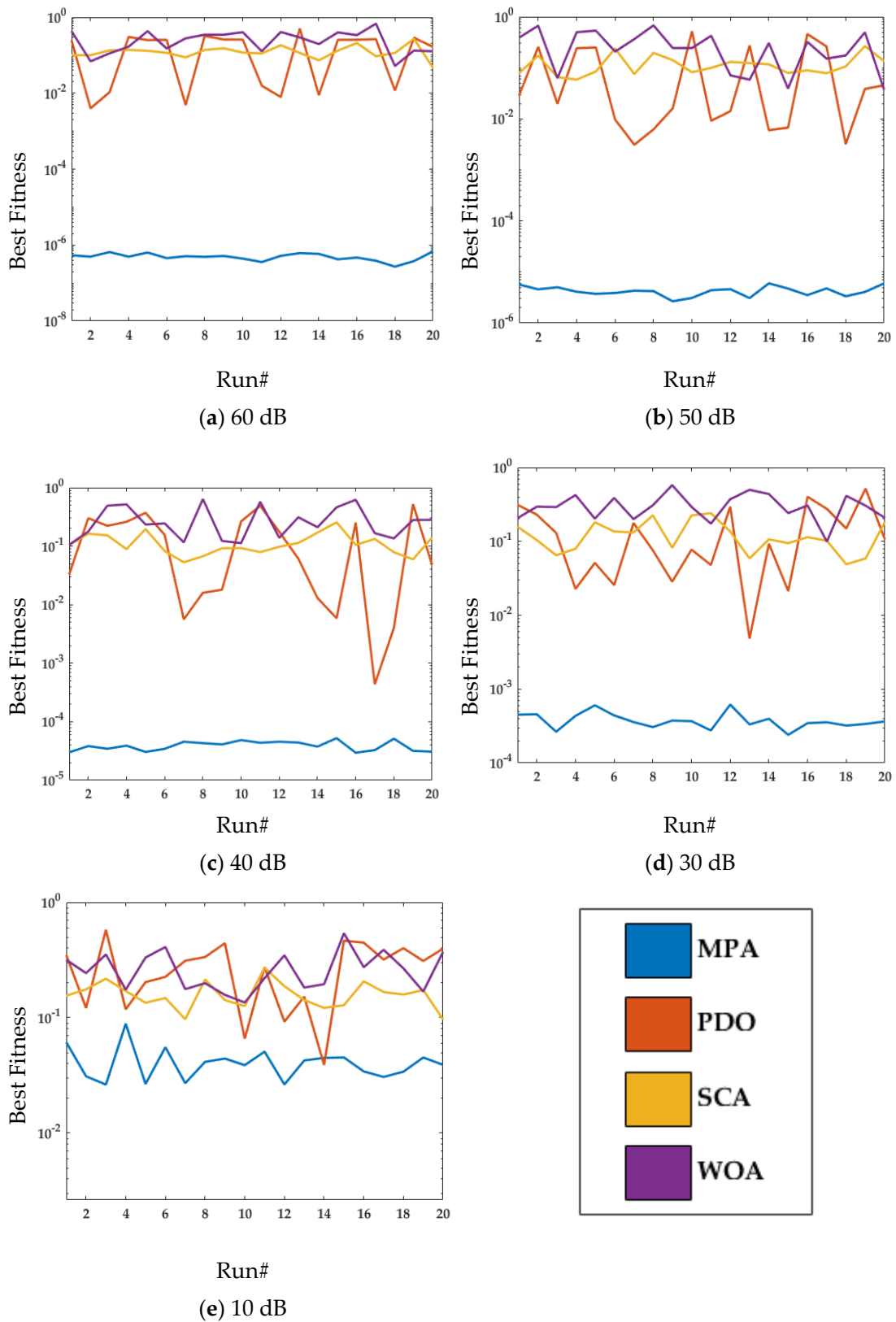


Figure 7. Statistical comparison of MPA with PDO, SCA, and WOA at $N_p = 100$ and $T = 1000$ for case study 1.

It is noted from Figure 7a–e that MPA outperforms SCA, WOA, and PDO for noise levels of [60 dB, 50 dB, 40 dB, 30 dB, 10 dB] in all of the independent runs. Moreover, it is also observed that there is an increase in fitness values for high noise levels for all of the methods.

Tables 2–6 show the performance of all of the algorithms in terms of the average fitness, best fitness, and estimated weights for the [60 dB, 50 dB, 40 dB, 30 dB, 10 dB] noise levels. It is noted that for lower noise levels, i.e., 60 dB, MPA gives better results compared with higher noise levels. Moreover, at a low noise level, the estimated weights are closer to the true value. It is also observed from Table 2 that the best fitness achieved for noise level = 60 dB is 2.6×10^{-7} . Similarly, the best fitness value for noise levels = [50 dB, 40 dB, 30 dB, 10 dB], provided in Tables 3–6, are [2.6×10^{-6} , 2.9×10^{-5} , 2.4×10^{-4} , 2.6×10^{-2}], respectively. Therefore, it is validated from Tables 2–6 that the fitness of MPA increases with the increase in noise levels and it decreases by increasing the population size and number of generations.

Table 2. Estimated weight analysis w.r.t generation and population sizes at 60 dB noise for case study 1.

Methods	T	Np	Design Parameters							Best Fitness	Avg Fitness
			g ₁	g ₂	h ₁	h ₂	o ₁	o ₂	o ₃		
MPA	500	50	−1.0999	0.9003	−0.7999	−0.6007	−0.8994	0.5998	0.1998	3.6×10^{-7}	6.6×10^{-7}
		100	−1.1004	0.9004	−0.8003	−0.5999	−0.8983	0.6010	0.1999	4.3×10^{-7}	5.5×10^{-7}
	1000	50	−1.0997	0.8997	−0.7998	−0.5993	−0.9028	0.5992	0.2003	3.5×10^{-7}	5.2×10^{-7}
		100	−1.0999	0.9000	−0.7997	−0.5998	−0.9003	0.6002	0.2002	2.6×10^{-7}	4.9×10^{-7}
PDO	500	50	−1.0518	0.8524	−0.6261	−0.7220	−0.9588	0.5851	0.2329	0.0097	0.1631
		100	−1.0600	0.8566	−0.7128	−0.6272	−1.0817	0.5417	0.2249	0.0059	0.1618
	1000	50	−1.0701	0.8637	−0.6899	−0.8059	−0.8724	0.5222	0.1791	0.0119	0.1677
		100	−1.0730	0.8670	−0.7100	−0.6694	−0.9522	0.5646	0.2155	0.0039	0.1848
SCA	500	50	−1.1554	0.8944	−1.2713	−0.3910	−1.3633	0.2760	0.0024	0.0785	0.1676
		100	−1.0469	0.8100	−0.9144	−0.3546	−1.8413	0.2371	0.2565	0.0719	0.1567
	1000	50	−0.9063	0.6642	−0.8466	−0.6712	−1.7836	−0.0064	0.0990	0.1167	0.1977
		100	−1.0292	0.8255	−1.0403	−0.3248	−1.2033	0.5034	0.1878	0.0464	0.1285
WOA	500	50	−1.1307	0.9794	−1.0294	−0.7458	−0.0103	0.8304	0.2124	0.1346	0.3724
		100	−1.1050	0.9013	−0.8204	−0.7087	−0.8466	0.5344	0.1570	0.0042	0.3019
	1000	50	−1.0270	0.8621	−0.4477	−0.6913	−0.7818	0.8436	0.3795	0.0294	0.3219
		100	−0.9356	0.7506	−0.5725	−0.6927	−1.4624	0.4017	0.2705	0.0521	0.2748
True Values			−1.1000	0.9000	−0.8000	−0.6000	−0.9000	0.6000	0.2000	0	0

Table 3. Estimated weights analysis w.r.t generation and population sizes at 50 dB noise for case study 1.

Methods	T	Np	Design Parameters							Best Fitness	Avg Fitness
			g ₁	g ₂	h ₁	h ₂	o ₁	o ₂	o ₃		
MPA	500	50	−1.0990	0.8998	−0.7971	−0.6027	−0.9000	0.6006	0.2000	3.5×10^{-6}	5.4×10^{-6}
		100	−1.1007	0.9007	−0.8010	−0.6005	−0.8976	0.5999	0.1995	3.2×10^{-6}	4.7×10^{-6}
	1000	50	−1.1007	0.9007	−0.8016	−0.6004	−0.9021	0.5988	0.1989	3.1×10^{-6}	4.6×10^{-6}
		100	−1.0999	0.8996	−0.7972	−0.6010	−0.9034	0.5994	0.2006	2.6×10^{-6}	4.2×10^{-6}
PDO	500	50	−1.0989	0.8856	−0.6730	−0.8708	−0.8491	0.4867	0.1613	0.0122	0.2468
		100	−1.0950	0.8850	−0.7609	−0.6493	−1.0193	0.5226	0.1898	0.0026	0.1971
	1000	50	−1.0568	0.8489	−0.6975	−0.6522	−1.0537	0.5183	0.2255	0.0070	0.2406
		100	−1.0821	0.8792	−0.6982	−0.7261	−0.8873	0.5679	0.2011	0.0031	0.1237

Table 3. Cont.

Methods	T	Np	Design Parameters					Best Fitness	Avg Fitness		
			g_1	g_2	h_1	h_2	o_1			o_2	o_3
SCA	500	50	-1.0086	0.7452	-0.4993	-0.7635	-1.4064	0.4192	0.2437	0.0902	0.1928
		100	-1.1220	1.0084	-0.3534	-1.1780	-0.1565	0.8028	0.2544	0.0801	0.1949
	1000	50	-1.1596	0.8947	-1.0397	-0.4481	-1.0611	0.3793	0.1573	0.0295	0.1382
		100	-0.9708	0.7561	-0.4713	-0.7527	-1.2920	0.4789	0.2752	0.0588	0.1212
WOA	500	50	-1.0691	0.8544	-0.9376	-0.4316	-1.1141	0.5184	0.2023	0.0104	0.3667
		100	-0.9926	0.8132	-0.6119	-0.6398	-1.0712	0.6237	0.2896	0.0216	0.3111
	1000	50	-1.0723	0.9288	-0.2641	-0.9755	-0.2221	0.9602	0.3590	0.0657	0.4094
		100	-1.0765	0.8998	-0.3190	-1.1725	-0.5884	0.6273	0.2237	0.0370	0.2990
True Values			-1.1000	0.9000	-0.8000	-0.6000	-0.9000	0.6000	0.2000	0	0

Table 4. Estimated weight analysis w.r.t generation and population sizes at 40 dB noise for case study 1.

Methods	T	Np	Design Parameters					Best Fitness	Avg Fitness		
			g_1	g_2	h_1	h_2	o_1			o_2	o_3
MPA	500	50	-1.0976	0.9002	-0.8108	-0.5990	-0.9099	0.5912	0.1952	3.8×10^{-5}	5.5×10^{-5}
		100	-1.0992	0.8985	-0.7965	-0.6014	-0.9045	0.5973	0.2016	3.2×10^{-5}	5.0×10^{-5}
	1000	50	-1.0978	0.8969	-0.7999	-0.6023	-0.9044	0.5961	0.1984	3.1×10^{-5}	4.2×10^{-5}
		100	-1.0978	0.8965	-0.7981	-0.5994	-0.9046	0.5976	0.2016	2.9×10^{-5}	3.9×10^{-5}
PDO	500	50	-1.1077	0.8619	-0.7954	-0.8527	-0.6526	0.5071	0.1500	0.0260	0.2769
		100	-1.0527	0.8379	-0.7737	-0.6079	-0.9045	0.5709	0.2188	0.0071	0.1967
	1000	50	-1.0243	0.8187	-0.8046	-0.5028	-1.2278	0.5109	0.2319	0.0110	0.1863
		100	-1.1020	0.8933	-0.8539	-0.5378	-0.9603	0.5789	0.1965	4.3×10^{-4}	0.1612
SCA	500	50	-1.2458	1.0114	-1.2823	-0.5958	-0.4440	0.6779	0.0084	0.1362	0.2290
		100	-1.1005	0.9112	-0.7042	-0.7149	-0.7310	0.6718	0.2431	0.0195	0.1565
	1000	50	-1.0389	0.8189	-0.8901	-0.5615	-1.1855	0.4103	0.2097	0.0556	0.1500
		100	-1.1279	0.8664	-1.1919	-0.3155	-1.4402	0.2008	0.1338	0.0530	0.1164
WOA	500	50	-1.1736	0.9365	-1.3780	-0.3366	-0.5491	0.6183	0.0930	0.0515	0.4267
		100	-0.4672	0.3825	-0.4320	-1.1173	-1.9501	-0.1694	0.1607	0.1660	0.4192
	1000	50	-1.0051	0.8141	-0.2655	-0.9009	-1.0431	0.5755	0.3375	0.0660	0.4354
		100	-0.7311	0.6054	-0.3689	-1.0833	-1.0464	0.3766	0.2605	0.1053	0.2973
True Values			-1.1000	0.9000	-0.8000	-0.6000	-0.9000	0.6000	0.2000	0	0

Table 5. Estimated weight analysis w.r.t generation and population sizes at 30 dB noise for case study 1.

Methods	T	Np	Design Parameters					Best Fitness	Avg Fitness		
			g_1	g_2	h_1	h_2	o_1			o_2	o_3
MPA	500	50	-1.0993	0.9027	-0.8065	-0.5937	-0.9485	0.5739	0.2001	3.0×10^{-4}	5.2×10^{-4}
		100	-1.0884	0.8840	-0.7943	-0.6330	-0.8651	0.5970	0.1908	2.4×10^{-4}	4.5×10^{-4}
	1000	50	-1.1002	0.8957	-0.7948	-0.5956	-0.9017	0.5975	0.2046	2.7×10^{-4}	4.1×10^{-4}
		100	-1.1034	0.9003	-0.8125	-0.5908	-0.9229	0.5861	0.1983	2.4×10^{-4}	3.8×10^{-4}
PDO	500	50	-0.9118	0.6720	-1.0179	-0.4754	-1.3536	0.2024	0.1578	0.0461	0.2711
		100	-1.0106	0.8154	-0.6439	-0.8667	-0.9536	0.4742	0.1776	0.0095	0.2513
	1000	50	-0.8903	0.7105	-0.6975	-0.6399	-1.3032	0.3473	0.2361	0.0230	0.2707
		100	-1.0686	0.8515	-0.9238	-0.4909	-0.9426	0.5455	0.1848	0.0048	0.1520

Table 5. Cont.

Methods	T	Np	Design Parameters						Best Fitness	Avg Fitness	
			g ₁	g ₂	h ₁	h ₂	o ₁	o ₂			o ₃
SCA	500	50	-1.0744	0.8886	-0.8612	-0.4488	-1.3053	0.3867	0.1863	0.0784	0.2060
		100	-1.1646	0.8605	-1.3450	-0.1424	-1.3534	0.1530	0.1114	0.0709	0.1638
	1000	50	-1.0843	0.9227	-0.6941	-0.8631	-1.0894	0.4127	0.1170	0.0719	0.1513
		100	-1.0649	0.8689	-0.8942	-0.4560	-0.9881	0.6281	0.2674	0.0489	0.1265
WOA	500	50	-1.1271	0.9493	0.5670	-1.9626	0.0465	0.8384	0.3408	0.1945	0.4343
		100	-0.9911	0.8016	0.0386	-1.9189	-0.5169	0.4345	0.1694	0.0826	0.3993
	1000	50	-1.2537	0.8268	-1.9499	0.4810	-1.8826	-0.1519	0.0787	0.1643	0.3945
		100	-1.1938	0.9840	-0.8044	-0.4657	0.0998	1.1197	0.3217	0.0991	0.3120
True Values			-1.1000	0.9000	-0.8000	-0.6000	-0.9000	0.6000	0.2000	0	0

Table 6. Estimated weight analysis w.r.t generation and population sizes at 10 dB noise for case study 1.

Methods	T	Np	Design Parameters						Best Fitness	Avg Fitness	
			g ₁	g ₂	h ₁	h ₂	o ₁	o ₂			o ₃
MPA	500	50	-1.0044	0.9248	-0.6585	-0.5989	-1.2175	0.5851	0.2879	0.0266	0.0651
		100	-0.4721	0.3117	0.0368	-1.0912	-1.7806	0.1460	0.4641	0.0350	0.0585
	1000	50	-1.0797	0.8075	-1.0865	-0.3263	-0.8897	0.5562	0.2113	0.0257	0.0537
		100	-1.0986	0.9158	-0.6987	-0.6994	-0.9153	0.5574	0.1941	0.0263	0.0417
PDO	500	50	-0.4146	0.2163	-0.3158	-1.3845	-1.1701	0.0619	0.1626	0.0591	0.3364
		100	-1.0651	0.6972	-1.6685	-0.2576	-0.5222	0.4760	0.0516	0.0810	0.2740
	1000	50	-0.7514	0.5082	-0.2147	-0.6914	-1.2349	0.8102	0.5757	0.0707	0.3043
		100	-0.7529	0.5434	-0.3480	-1.1650	-0.9728	0.3101	0.1813	0.0390	0.2830
SCA	500	50	-0.6897	0.4400	-0.8164	-0.9964	-1.5210	-0.0562	0.1077	0.1750	0.2764
		100	-1.1314	0.8923	-1.5527	-0.7299	-0.8486	0.2096	0.0101	0.0933	0.1738
	1000	50	-0.0228	0.0203	-0.0002	-2.0000	-1.7703	-0.2665	0.1090	0.1005	0.1781
		100	-0.2253	-0.0028	-0.1909	-1.6631	-1.0498	0.1043	0.2336	0.0966	0.1620
WOA	500	50	-0.1134	-0.0868	-0.3416	-0.7243	-0.4913	0.7683	0.4553	0.1016	0.3102
		100	-1.3169	0.8183	-1.6177	0.1795	-1.4909	0.0159	0.1090	0.0760	0.2376
	1000	50	-0.7933	0.6662	-1.2328	-0.5992	-1.9345	-0.2411	0.0103	0.0824	0.2924
		100	0.0301	0.1867	-1.9827	-1.9827	-1.0564	-0.1705	0.0200	0.1355	0.2732
True Values			-1.1000	0.9000	-0.8000	-0.6000	-0.9000	0.6000	0.2000	0	0

4.2. Case Study 2

The performance of the MPA is also considered for another Hammerstein model based on the autoregressive exogenous input structure, i.e., H-ARX [13,14], with the same model parameters as considered in case study 1. The H-ARX model is mathematically defined as

$$y(t) = \frac{H(q)}{G(q)}\bar{\mu}(t) + \frac{1}{G(q)}\varepsilon(t), \tag{27}$$

where G(q) and H(q) are polynomials in the shift operator, and rearranging (27) gives the following

$$G(q)y(t) = H(q)\bar{\mu}(t) + \varepsilon(t) \tag{28}$$

$$y(t) = -g_1y(t-1) - g_2y(t-2) + h_0\bar{\mu}(t) + h_1\bar{\mu}(t-1) + h_2\bar{\mu}(t-2) + \varepsilon(t), h_0 = 1,$$

$$\bar{\mu}(t) = o_1k_1(\mu(t)) + o_2k_2(\mu(t)) + o_3k_3(\mu(t)) = o_1\mu(t) + o_2\mu^2(t) + o_3\mu^3(t)$$

$$\omega = [g_1, g_2, h_1, h_2, o_1, o_2, o_3]^T = [-1.100, 0.900, -0.800, -0.600, -0.900, 0.600, 0.200]^T.$$

The input data are generated in the same way as mentioned in case study 1, while the noise in the ARX model is taken as white Gaussian with constant variance, i.e., [0.01, 0.05, 0.10]. Two variations of generation (T) [500, 1000] and population Np = 50 are considered. The fitness for case study 2 is calculated through (20) by using y instead of y_m, and the fitness plots are demonstrated in Figure 8.

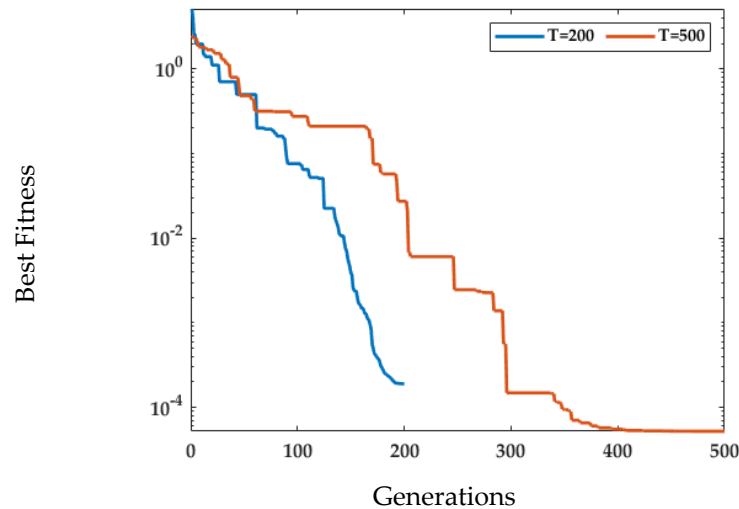


Figure 8. Fitness plots for MPA w.r.t generation variations for case study 2.

The fitness curves in Figure 8 correspond to the best fitness of the MPA algorithm for population size Np = 50. It is noted from Figure 8 that the fitness of MPA reduces significantly with the increase in generations.

The behavior of MPA for different noise variations is evaluated at population size Np = 50 and changing the generation size [200, 500] for three values of noise levels [0.01,0.05,0.10], and the fitness plots are presented in Figure 9.

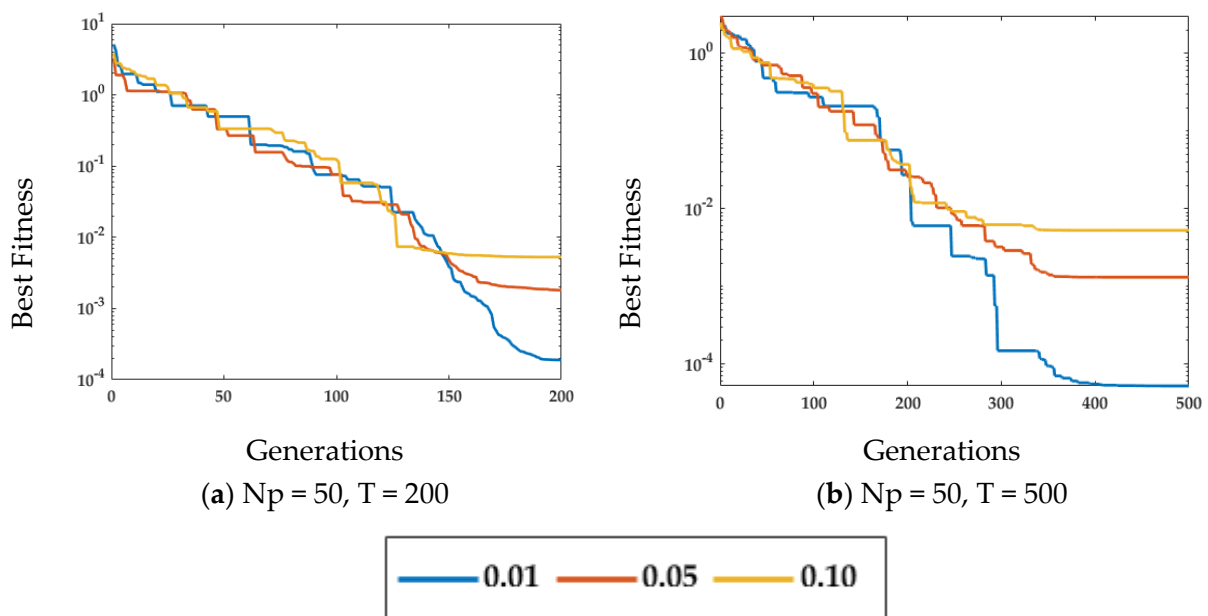


Figure 9. Fitness plots for MPA w.r.t noise levels for case study 2.

It is noted from the fitness plots demonstrated in Figure 9a,b that for a fixed population size and number of generations, the fitness achieved by MPA for a low level of noise,

i.e., [0.01], is quite a lot less compared with the fitness for a high noise level, i.e., [0.05,0.10]. Yet, MPA achieves the minimum value of fitness for the smallest value of noise, i.e., [0.01] for $N_p = 50$ and $T = 500$. Therefore, it is confirmed from the curves in Figure 9 that the performance of MPA is lower for higher noise values.

Similar to example 1, it is further compared with the prairie dog optimization (PDO) [68], sine cosine algorithm (SCA) [69], and whale optimization algorithm (WOA) [70] for $N_p = 50$ and $T = 500$, and noise levels of [0.01,0.05,0.10] are presented in Figure 10.

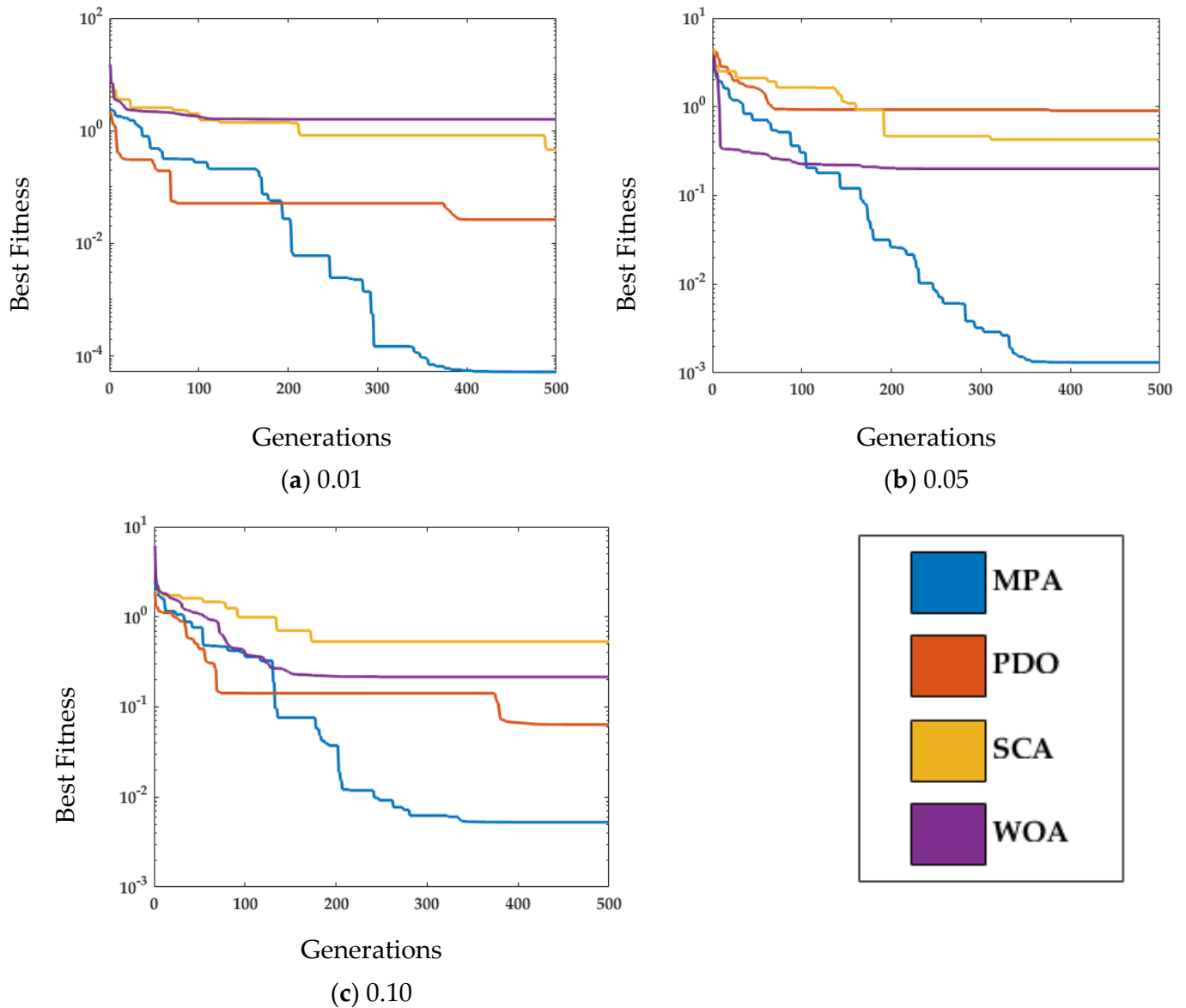


Figure 10. Fitness plots comparison of MPA with PDO, SCA, and WOA at $N_p = 50$ and $T = 500$ for case study 2.

Figure 10a–c demonstrates the convergence of MPA with SCA, WOA, and PDO at $N_p = 50$ and $T = 500$ over all the noise levels, i.e., [0.01,0.05,0.10]. It is noted from Figure 10a–c that with the increase in noise level, the fitness value also increases. However, MPA achieves the lowest fitness compared with SCA, WOA, and PDO for all of the noise levels.

MPA is further evaluated statistically against PDO, WOA, and SCA at $N_p = 50$ and $T = 500$ for 20 independent runs. Figure 11 shows the fitness value of MPA, SCA, WOA, and PDO on run#1, run#10, and run#20 for all of the noise levels.

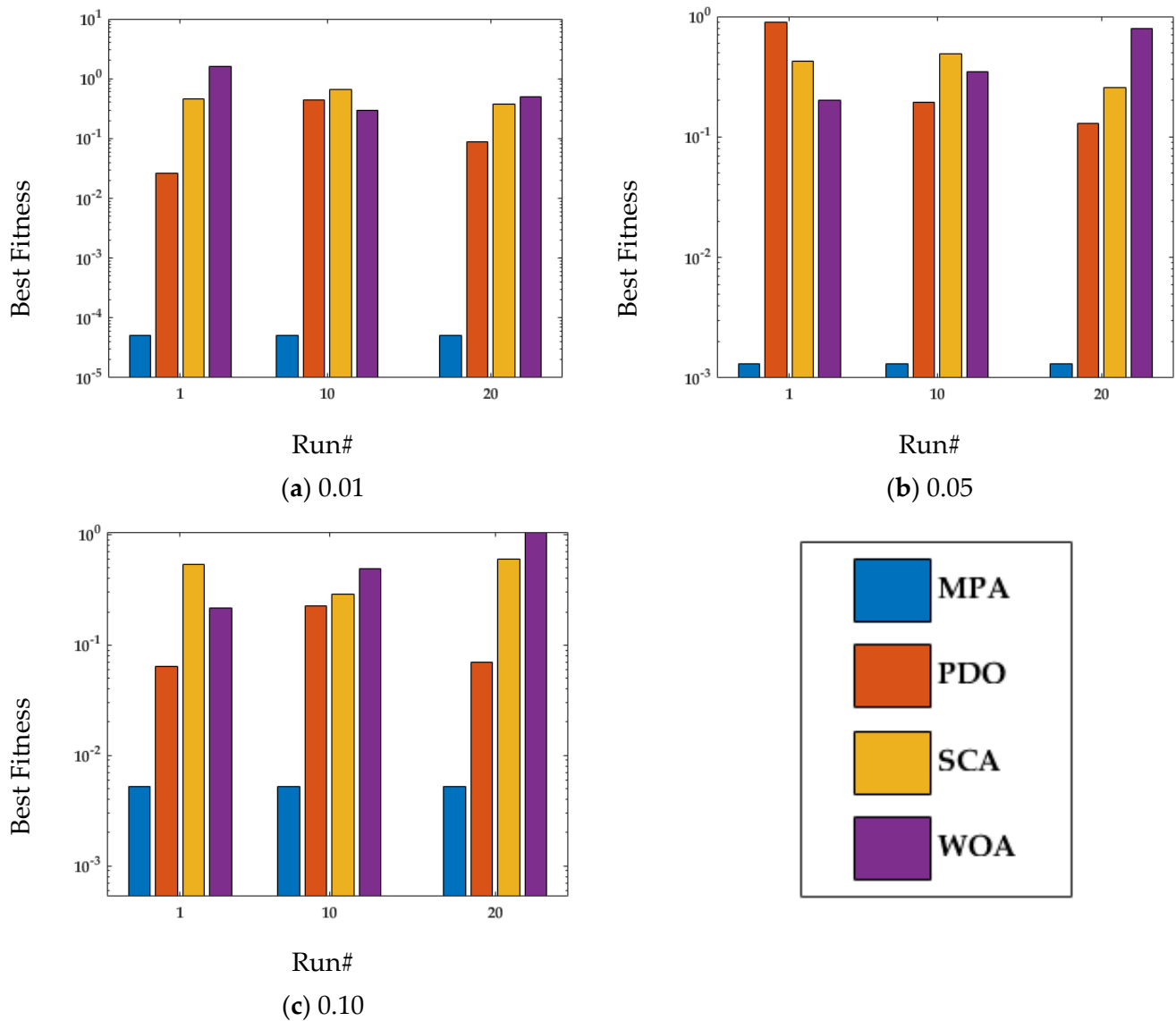


Figure 11. Run# fitness comparison of MPA with PDO, SCA, and WOA at $N_p = 50$ and $T = 500$ for case study 2.

It is noted from Figure 11a–c that MPA achieves the lowest fitness compared with SCA, WOA, and PDO for different independent runs for all noise levels. Moreover, it is also observed that the fitness increases with the increase in noise levels for all of the methods. Further statistical analysis on all independent runs is demonstrated in Figure 12.

It is noted from Figure 12a–c that MPA outperforms SCA, WOA, and PDO for noise levels of [0.01,0.05,0.10] in all of the independent runs. Moreover, it is also observed that there is an increase in fitness value for high noise levels for all of the methods.

Tables 7–9 show the performance of all of the algorithms in terms of the average fitness, best fitness, and estimated weights for [0.01,0.05,0.10] noise levels at $N_p = 50$. It is noted that for lower noise levels i.e., 0.01, MPA gives better results compared with the higher noise levels. Moreover, at a low noise level, the estimated weights are closer to the true value. It is also observed from Table 7 that the best fitness achieved for noise level = 0.01 is 5.2×10^{-5} . Similarly, the best fitness value for noise levels = [0.05,0.10] provided in Tables 8 and 9 are [1.3×10^{-3} , 5.3×10^{-3}], respectively. Therefore, it is validated from Tables 7–9 that the fitness of MPA increases with the increase in noise levels and it decreases by increasing the number of generations.

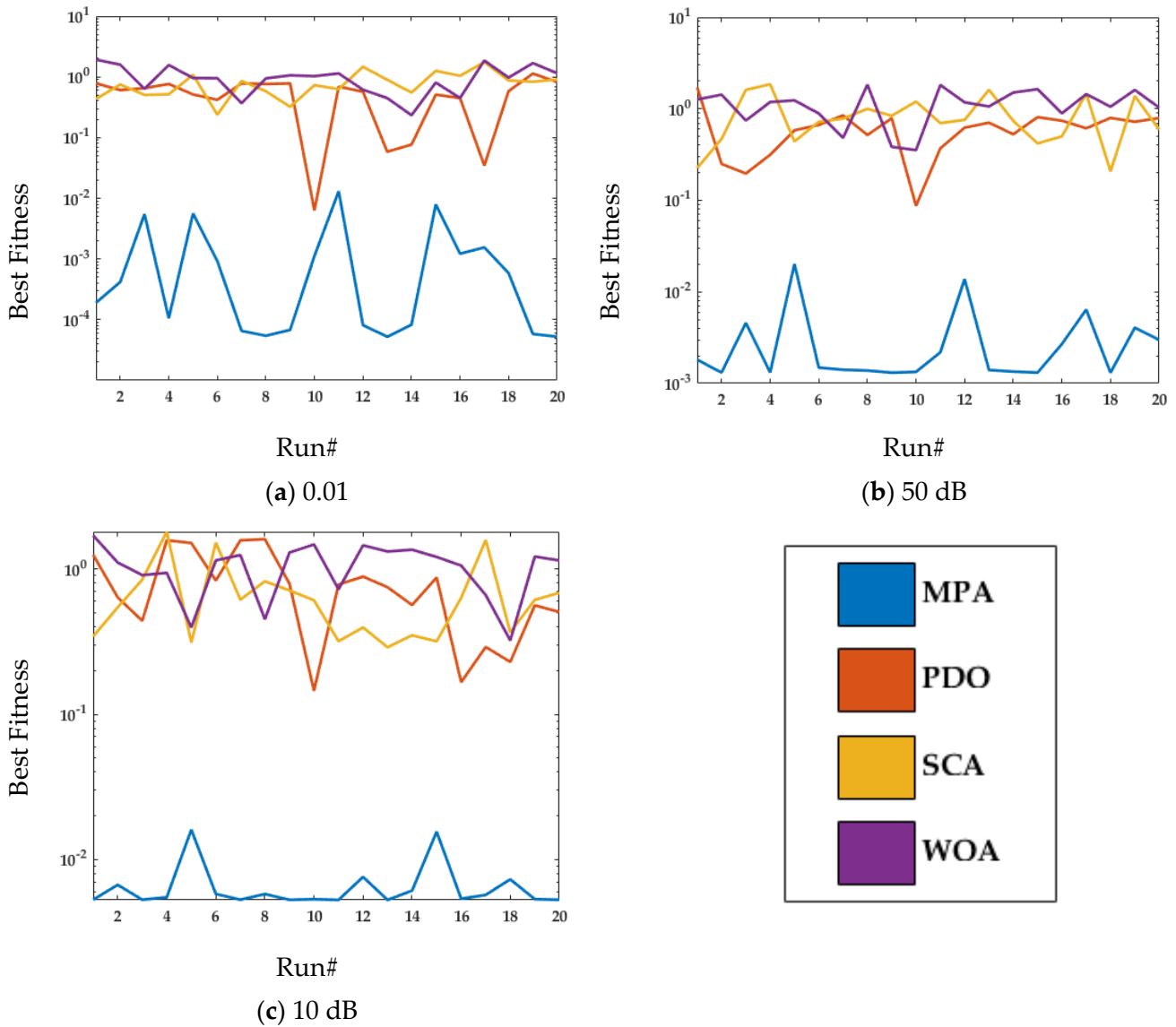


Figure 12. Statistical comparison of MPA with PDO, SCA, and WOA at $N_p = 50$ and $T = 200$ for case study 2.

Table 7. Estimated weight analysis w.r.t generations at a 0.01 noise level for case study 2.

Methods	T	Design Parameters							Best Fitness	Avg Fitness
		g_1	g_2	h_1	h_2	o_1	o_2	o_3		
MPA	200	-1.1017	0.8978	-0.8044	-0.5999	-0.8932	0.6001	0.1980	5.2×10^{-5}	1.9×10^{-3}
	500	-1.1018	0.8979	-0.8044	-0.5996	-0.8920	0.6010	0.1982	5.2×10^{-5}	5.2×10^{-5}
PDO	200	-1.0836	0.8766	-0.7975	-0.6473	-1.0023	0.5269	0.1740	6.3×10^{-3}	0.5494
	500	-1.0899	0.8948	-0.5766	-0.6963	-0.7177	0.7521	0.2727	0.0146	0.3834
SCA	200	-1.1650	0.9793	-1.8630	-1.2628	0.0036	0.6294	0.0072	0.2384	0.8104
	500	-1.1364	0.8896	-1.6057	-0.2327	-0.7748	0.5465	0.0233	0.0867	0.4969
WOA	200	-0.9963	0.7957	-0.3799	-0.7507	-1.3856	0.6501	0.3586	0.2331	1.0182
	500	-1.1450	0.9706	-0.2738	-0.6472	0.3553	1.4730	0.4975	0.1753	0.6517
True Values		-1.1000	0.9000	-0.8000	-0.6000	-0.9000	0.6000	0.2000	0	0

Table 8. Estimated weight analysis w.r.t generations at a 0.05 noise level for case study 2.

Methods	T	Design Parameters							Best Fitness	Avg Fitness
		g_1	g_2	h_1	h_2	o_1	o_2	o_3		
MPA	200	-1.1109	0.8910	-0.8271	-0.5952	-0.8589	0.6044	0.1903	1.3×10^{-3}	3.7×10^{-3}
	500	-1.1109	0.8910	-0.8262	-0.5954	-0.8583	0.6049	0.1906	1.3×10^{-3}	1.3×10^{-3}
PDO	200	-1.0406	0.8332	-0.4206	-1.3664	-0.7828	0.4256	0.1194	0.0872	0.6333
	500	-1.0866	0.8555	-0.8423	-0.6316	-1.0984	0.4647	0.1520	0.0172	0.4124
SCA	200	-1.1050	0.8054	-1.4742	-0.1922	-1.0218	0.5351	0.0089	0.2073	0.8758
	500	-1.0695	0.8453	-1.1957	-1.4701	-0.1459	0.5871	0.0042	0.1845	0.4258
WOA	200	-0.9064	0.7102	-0.3235	-1.1091	-1.2099	0.4528	0.2038	0.3539	1.1520
	500	-1.0549	0.8315	-1.0059	-0.9560	-1.2754	0.1507	-0.0263	0.1994	0.5828
True Values		-1.1000	0.9000	-0.8000	-0.6000	-0.9000	0.6000	0.2000	0	0

Table 9. Estimated weight analysis w.r.t generations at a 0.10 noise level for case study 2.

Methods	T	Design Parameters							Best Fitness	Avg Fitness
		g_1	g_2	h_1	h_2	o_1	o_2	o_3		
MPA	200	-1.1267	0.8856	-0.8717	-0.5808	-0.8127	0.6069	0.1776	5.3×10^{-3}	6.8×10^{-3}
	500	-1.1265	0.8857	-0.8653	-0.5835	-0.8107	0.6093	0.1794	5.3×10^{-3}	5.3×10^{-3}
PDO	200	-1.0194	0.7655	-0.6219	-0.9296	-1.1517	0.3711	0.1296	0.1452	0.7990
	500	-1.0938	0.8706	-0.6854	-0.7346	-0.8723	0.5692	0.2007	0.0172	0.3089
SCA	200	-1.0574	0.9286	-0.0290	-1.3075	-0.6739	0.5813	0.3124	0.2892	0.6840
	500	-1.1517	0.8870	-1.4557	-0.1035	-1.4434	0.2615	0.0165	0.1655	0.4139
WOA	200	-1.1503	0.8178	-1.9646	0.2299	-1.9588	-0.0416	-0.0822	0.3225	1.0572
	500	-1.0863	0.8635	-0.3088	-0.8407	-0.6800	0.8381	0.3392	0.0734	0.6578
True Values		-1.1000	0.9000	-0.8000	-0.6000	-0.9000	0.6000	0.2000	0	0

The results of the detailed simulations imply that the MPA-based swarming optimization heuristics efficiently approximates the parameters of the Hammerstein systems.

5. Conclusions

The main findings of the investigation are as follows:

The parameter optimization of the nonlinear mathematical model represented through the Hammerstein structure is studied using the marine predator algorithm, MPA, and the key term separation principle. The mathematical model and working principle of the MPA are inspired from the behavior of predators when catching prey. The optimal interaction between the predator and prey is modelled using Brownian and Levy movement. The key term separation is a technique used to avoid redundancy in the estimation of Hammerstein model parameters, thus making the MPA more efficient. The accuracy and robustness of the MPA based optimization scheme with key term separation technique is established for HOE and H-ARX estimation through the simulation results for different noise scenarios.

Future work may extend the application of MPA to optimize the muscle fatigue systems [71,72] and solve fractional order problems [73–75].

Author Contributions: Methodology, K.M.; visualization, Z.A.K.; formal analysis, Z.A.K., N.I.C. and M.A.Z.R.; writing—original draft preparation, K.M.; writing—review and editing, N.I.C., Z.A.K. and M.A.Z.R.; project administration, K.M.C., A.A.A. and A.H.M.; funding acquisition, K.M.C., A.A.A. and A.H.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: M.A.Z. Raja like to acknowledge the support of the National Science and Technology Council (NSTC), Taiwan under grant no. NSTC 111-2221-E-224-043.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Schoukens, J.; Ljung, L. Nonlinear System Identification: A User-Oriented Road Map. *IEEE Control Syst.* **2019**, *39*, 28–99. [[CrossRef](#)]
2. Yukai, S.; Chao, Y.; Zhigang, W.; Liuyue, B. Nonlinear System Identification of an All Movable Fin with Rotational Freeplay by Subspace-Based Method. *Appl. Sci.* **2020**, *10*, 1205. [[CrossRef](#)]
3. Giri, F.; Bai, E.W. (Eds.) *Block-Oriented Nonlinear System Identification*; Springer: Berlin/Heidelberg, Germany, 2010; Volume 1, p. 0278-0046.
4. Billings, S.A. *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*; John Wiley & Sons: New York, NY, USA, 2013.
5. Tissaoui, K. Forecasting implied volatility risk indexes: International evidence using Hammerstein-ARX approach. *Int. Rev. Financial Anal.* **2019**, *64*, 232–249. [[CrossRef](#)]
6. Bai, Y.-T.; Wang, X.-Y.; Jin, X.-B.; Zhao, Z.-Y.; Zhang, B.-H. A Neuron-Based Kalman Filter with Nonlinear Autoregressive Model. *Sensors* **2020**, *20*, 299. [[CrossRef](#)]
7. Ljung, L. *System Identification: Theory for the User*, 2nd ed.; Prentice Hall: Hoboken, NJ, USA, 1987.
8. Narendra, K.; Gallman, P. An iterative method for the identification of nonlinear systems using a Hammerstein model. *IEEE Trans. Autom. Control* **1966**, *11*, 546–550. [[CrossRef](#)]
9. Chang, F.; Luus, R. A noniterative method for identification using Hammerstein model. *IEEE Trans. Autom. Control* **1971**, *16*, 464–468. [[CrossRef](#)]
10. Vörös, J. Parameter identification of discontinuous hammerstein systems. *Automatica* **1997**, *33*, 1141–1146. [[CrossRef](#)]
11. Voros, J. Recursive identification of hammerstein systems with discontinuous nonlinearities containing dead-zones. *IEEE Trans. Autom. Control* **2003**, *48*, 2203–2206. [[CrossRef](#)]
12. Chen, H.; Ding, F. Hierarchical Least Squares Identification for Hammerstein Nonlinear Controlled Autoregressive Systems. *Circuits Syst. Signal Process.* **2014**, *34*, 61–75. [[CrossRef](#)]
13. Ding, F.; Chen, H.; Xu, L.; Dai, J.; Li, Q.; Hayat, T. A hierarchical least squares identification algorithm for Hammerstein nonlinear systems using the key term separation. *J. Frankl. Inst.* **2018**, *355*, 3737–3752. [[CrossRef](#)]
14. Chen, H.; Xiao, Y.; Ding, F. Hierarchical gradient parameter estimation algorithm for Hammerstein nonlinear systems using the key term separation principle. *Appl. Math. Comput.* **2014**, *247*, 1202–1210. [[CrossRef](#)]
15. Chaudhary, N.I.; Khan, Z.A.; Zubair, S.; Raja, M.A.Z.; Dedovic, N. Normalized fractional adaptive methods for nonlinear control autoregressive systems. *Appl. Math. Model.* **2018**, *66*, 457–471. [[CrossRef](#)]
16. Chaudhary, N.I.; Aslam, M.S.; Baleanu, D.; Raja, M.A.Z. Design of sign fractional optimization paradigms for parameter estimation of nonlinear Hammerstein systems. *Neural Comput. Appl.* **2019**, *32*, 8381–8399. [[CrossRef](#)]
17. Chaudhary, N.I.; Raja MA, Z.; He, Y.; Khan, Z.A.; Machado, J.T. Design of multi innovation fractional LMS algorithm for parameter estimation of input nonlinear control autoregressive systems. *Appl. Math. Model.* **2021**, *93*, 412–425. [[CrossRef](#)]
18. Chaudhary, N.I.; Raja, M.A.Z.; Khan, Z.A.; Cheema, K.M.; Milyani, A.H. Hierarchical Quasi-Fractional Gradient Descent Method for Parameter Estimation of Nonlinear ARX Systems Using Key Term Separation Principle. *Mathematics* **2021**, *9*, 3302. [[CrossRef](#)]
19. Chaudhary, N.I.; Raja, M.A.Z.; Khan, Z.A.; Mehmood, A.; Shah, S.M. Design of fractional hierarchical gradient descent algorithm for parameter estimation of nonlinear control autoregressive systems. *Chaos Solitons Fractals* **2022**, *157*, 111913. [[CrossRef](#)]
20. Gotmare, A.; Bhattacharjee, S.S.; Patidar, R.; George, N.V. Swarm and evolutionary computing algorithms for system identification and filter design: A comprehensive review. *Swarm Evol. Comput.* **2017**, *32*, 68–84. [[CrossRef](#)]
21. Jui, J.J.; Ahmad, M.A. A hybrid metaheuristic algorithm for identification of continuous-time Hammerstein systems. *Appl. Math. Model.* **2021**, *95*, 339–360. [[CrossRef](#)]
22. Raja MA, Z.; Shah, A.A.; Mehmood, A.; Chaudhary, N.I.; Aslam, M.S. Bio-inspired computational heuristics for parameter estimation of nonlinear Hammerstein controlled autoregressive system. *Neural Comput. Appl.* **2018**, *29*, 1455–1474. [[CrossRef](#)]
23. Mehmood, A.; Aslam, M.S.; Chaudhary, N.I.; Zameer, A.; Raja, M.A.Z. Parameter estimation for Hammerstein control autoregressive systems using differential evolution. *Signal Image Video Process.* **2018**, *12*, 1603–1610. [[CrossRef](#)]
24. Mehmood, A.; Raja, M.A.Z.; Shi, P.; Chaudhary, N.I. Weighted differential evolution-based heuristic computing for identification of Hammerstein systems in electrically stimulated muscle modeling. *Soft Comput.* **2022**, *26*, 8929–8945. [[CrossRef](#)]
25. Mehmood, A.; Zameer, A.; Chaudhary, N.I.; Raja, M.A.Z. Backtracking search heuristics for identification of electrical muscle stimulation models using Hammerstein structure. *Appl. Soft Comput.* **2019**, *84*, 105705. [[CrossRef](#)]
26. Altaf, F.; Chang, C.-L.; Chaudhary, N.I.; Raja, M.A.Z.; Cheema, K.M.; Shu, C.-M.; Milyani, A.H. Adaptive Evolutionary Computation for Nonlinear Hammerstein Control Autoregressive Systems with Key Term Separation Principle. *Mathematics* **2022**, *10*, 1001. [[CrossRef](#)]
27. Nanda, S.J.; Panda, G.; Majhi, B. Improved identification of Hammerstein plants using new CPSO and IPSO algorithms. *Expert Syst. Appl.* **2010**, *37*, 6818–6831. [[CrossRef](#)]

28. Gotmare, A.; Patidar, R.; George, N.V. Nonlinear system identification using a cuckoo search optimized adaptive Hammerstein model. *Expert Syst. Appl.* **2015**, *42*, 2538–2546. [[CrossRef](#)]
29. Janjanam, L.; Saha, S.K.; Kar, R. Optimal Design of Hammerstein Cubic Spline Filter for Non-Linear System Modelling Based on Snake Optimiser. *IEEE Trans. Ind. Electron.* **2022**. [[CrossRef](#)]
30. Altaf, F.; Chang, C.-L.; Chaudhary, N.I.; Cheema, K.M.; Raja, M.A.Z.; Shu, C.-M.; Milyani, A.H. Novel Fractional Swarming with Key Term Separation for Input Nonlinear Control Autoregressive Systems. *Fractal Fract.* **2022**, *6*, 348. [[CrossRef](#)]
31. Zheng, J.; Li, K.; Zhang, X. Wi-Fi Fingerprint-Based Indoor Localization Method via Standard Particle Swarm Optimization. *Sensors* **2022**, *22*, 5051. [[CrossRef](#)]
32. Jiang, L.; Tajima, Y.; Wu, L. Application of Particle Swarm Optimization for Auto-Tuning of the Urban Flood Model. *Water* **2022**, *14*, 2819. [[CrossRef](#)]
33. Mehmood, K.; Chaudhary, N.I.; Khan, Z.A.; Cheema, K.M.; Raja, M.A.Z.; Milyani, A.H.; Azhari, A.A. Dwarf Mongoose Optimization Metaheuristics for Autoregressive Exogenous Model Identification. *Mathematics* **2022**, *10*, 3821. [[CrossRef](#)]
34. Alissa, K.A.; Elkamchouchi, D.H.; Tarmissi, K.; Yafoz, A.; Alsini, R.; Alghushairy, O.; Mohamed, A.; Al Duhayyim, M. Dwarf Mongoose Optimization with Machine-Learning-Driven Ransomware Detection in Internet of Things Environment. *Appl. Sci.* **2022**, *12*, 9513. [[CrossRef](#)]
35. Ji, H.; Hu, H.; Peng, X. Multi-Underwater Gliders Coverage Path Planning Based on Ant Colony Optimization. *Electronics* **2022**, *11*, 3021. [[CrossRef](#)]
36. Liu, Q.; Zhu, S.; Chen, M.; Liu, W. Detecting Dynamic Communities in Vehicle Movements Using Ant Colony Optimization. *Appl. Sci.* **2022**, *12*, 7608. [[CrossRef](#)]
37. Al-Shammaa, A.A.; MAbdurraqeeb, A.; Noman, A.M.; Alkuhayli, A.; Farh, H.M. Hardware-In-the-Loop Validation of Direct MPPT Based Cuckoo Search Optimization for Partially Shaded Photovoltaic System. *Electronics* **2022**, *11*, 1655. [[CrossRef](#)]
38. Hameed, K.; Khan, W.; Abdalla, Y.S.; Al-Harbi, F.F.; Armghan, A.; Asif, M.; Qamar, M.S.; Ali, F.; Miah, S.; Alibakhshikenari, M.; et al. Far-Field DOA Estimation of Uncorrelated RADAR Signals through Coprime Arrays in Low SNR Regime by Implementing Cuckoo Search Algorithm. *Electronics* **2022**, *11*, 558. [[CrossRef](#)]
39. Mehmood, K.; Chaudhary, N.I.; Khan, Z.A.; Raja, M.A.Z.; Cheema, K.M.; Milyani, A.H. Design of Aquila Optimization Heuristic for Identification of Control Autoregressive Systems. *Mathematics* **2022**, *10*, 1749. [[CrossRef](#)]
40. Lee, J.G.; Chim, S.; Park, H.H. Energy-efficient cluster-head selection for wireless sensor networks using sampling-based spider monkey optimization. *Sensors* **2019**, *19*, 5281. [[CrossRef](#)]
41. He, F.; Ye, Q. A Bearing Fault Diagnosis Method Based on Wavelet Packet Transform and Convolutional Neural Network Optimized by Simulated Annealing Algorithm. *Sensors* **2022**, *22*, 1410. [[CrossRef](#)]
42. Xiao, S.; Tan, X.; Wang, J. A Simulated Annealing Algorithm and Grid Map-Based UAV Coverage Path Planning Method for 3D Reconstruction. *Electronics* **2021**, *10*, 853. [[CrossRef](#)]
43. Thiagarajan, K.; Anandan, M.M.; Stateczny, A.; Divakarachari, P.B.; Lingappa, H.K. Satellite Image Classification Using a Hierarchical Ensemble Learning and Correlation Coefficient-Based Gravitational Search Algorithm. *Remote Sens.* **2021**, *13*, 4351. [[CrossRef](#)]
44. Owolabi, T.; Rahman, M.A. Energy Band Gap Modeling of Doped Bismuth Ferrite Multifunctional Material Using Gravitational Search Algorithm Optimized Support Vector Regression. *Crystals* **2021**, *11*, 246. [[CrossRef](#)]
45. Qais, M.H.; Hasanien, H.M.; Turkey, R.A.; Alghuwainem, S.; Tostado-Véliz, M.; Jurado, F. Circle Search Algorithm: A Geometry-Based Metaheuristic Optimization Algorithm. *Mathematics* **2022**, *10*, 1626. [[CrossRef](#)]
46. Qais, M.H.; Hasanien, H.M.; Turkey, R.A.; Alghuwainem, S.; Loo, K.-H.; Elgendy, M. Optimal PEM Fuel Cell Model Using a Novel Circle Search Algorithm. *Electronics* **2022**, *11*, 1808. [[CrossRef](#)]
47. Ehteram, M.; Ahmed, A.N.; Ling, L.; Fai, C.M.; Latif, S.D.; Afan, H.A.; Banadkooki, F.B.; El-Shafie, A. Pipeline scour rates prediction-based model utilizing a multilayer perceptron-colliding body algorithm. *Water* **2020**, *12*, 902. [[CrossRef](#)]
48. Yang, W.; Xia, K.; Li, T.; Xie, M.; Zhao, Y. An Improved Transient Search Optimization with Neighborhood Dimensional Learning for Global Optimization Problems. *Symmetry* **2021**, *13*, 244. [[CrossRef](#)]
49. Almabrok, A.; Psarakis, M.; Dounis, A. Fast Tuning of the PID Controller in An HVAC System Using the Big Bang–Big Crunch Algorithm and FPGA Technology. *Algorithms* **2018**, *11*, 146. [[CrossRef](#)]
50. Wu, T.; Li, X.; Zhou, D.; Li, N.; Shi, J. Differential Evolution Based Layer-Wise Weight Pruning for Compressing Deep Neural Networks. *Sensors* **2021**, *21*, 880. [[CrossRef](#)] [[PubMed](#)]
51. Li, M.; Zhang, D.; Lu, S.; Tang, X.; Phung, T. Differential Evolution-Based Overcurrent Protection for DC Microgrids. *Energies* **2021**, *14*, 5026. [[CrossRef](#)]
52. Drachal, K.; Pawłowski, M. A Review of the Applications of Genetic Algorithms to Forecasting Prices of Commodities. *Economies* **2021**, *9*, 6. [[CrossRef](#)]
53. Awan, W.A.; Zaidi, A.; Hussain, M.; Hussain, N.; Syed, I. The design of a wideband antenna with notching characteristics for small devices using a genetic algorithm. *Mathematics* **2021**, *9*, 2113. [[CrossRef](#)]
54. Strumberger, I.; Minovic, M.; Tuba, M.; Bacanin, N. Performance of Elephant Herding Optimization and Tree Growth Algorithm Adapted for Node Localization in Wireless Sensor Networks. *Sensors* **2019**, *19*, 2515. [[CrossRef](#)] [[PubMed](#)]
55. Abualigah, L.; Diabat, A.; Sumari, P.; Gandomi, A. A Novel Evolutionary Arithmetic Optimization Algorithm for Multilevel Thresholding Segmentation of COVID-19 CT Images. *Processes* **2021**, *9*, 1155. [[CrossRef](#)]

56. Sharma, A.; Khan, R.A.; Sharma, A.; Kashyap, D.; Rajput, S. A Novel Opposition-Based Arithmetic Optimization Algorithm for Parameter Extraction of PEM Fuel Cell. *Electronics* **2021**, *10*, 2834. [[CrossRef](#)]
57. Faradonbeh, R.S.; Monjezi, M.; Armaghani, D.J. Genetic programming and non-linear multiple regression techniques to predict backbreak in blasting operation. *Eng. Comput.* **2015**, *32*, 123–133. [[CrossRef](#)]
58. Masoumi, N.; Xiao, Y.; Rivaz, H. ARENA: Inter-modality affine registration using evolutionary strategy. *Int. J. Comput. Assist. Radiol. Surg.* **2018**, *14*, 441–450. [[CrossRef](#)]
59. Faramarzi, A.; Heidarinejad, M.; Mirjalili, S.; Gandomi, A.H. Marine Predators Algorithm: A nature-inspired metaheuristic. *Expert Syst. Appl.* **2020**, *152*, 113377. [[CrossRef](#)]
60. Shaheen, M.A.; Yousri, D.; Fathy, A.; Hasanien, H.M.; Alkuhayli, A.; Muyeen, S.M. A novel application of improved marine predators algorithm and particle swarm optimization for solving the ORPD problem. *Energies* **2020**, *13*, 5679. [[CrossRef](#)]
61. Ebeed, M.; Alhejji, A.; Kamel, S.; Jurado, F. Solving the optimal reactive power dispatch using marine predators algorithm considering the uncertainties in load and wind-solar generation systems. *Energies* **2020**, *13*, 4316. [[CrossRef](#)]
62. He, Q.; Lan, Z.; Zhang, D.; Yang, L.; Luo, S. Improved Marine Predator Algorithm for Wireless Sensor Network Coverage Optimization Problem. *Sustainability* **2022**, *14*, 9944. [[CrossRef](#)]
63. Yang, L.; He, Q.; Yang, L.; Luo, S. A Fusion Multi-Strategy Marine Predator Algorithm for Mobile Robot Path Planning. *Appl. Sci.* **2022**, *12*, 9170. [[CrossRef](#)]
64. Wadood, A.; Khan, S.; Khan, B.M.; Ali, H.; Rehman, Z. Application of Marine Predator Algorithm in Solving the Problem of Directional Overcurrent Relay in Electrical Power System. *Eng. Proc.* **2021**, *12*, 9.
65. Lu, X.; Nanehkaran, Y.A.; Fard, M.K. A Method for Optimal Detection of Lung Cancer Based on Deep Learning Optimized by Marine Predators Algorithm. *Comput. Intell. Neurosci.* **2021**, *2021*, 3694723. [[CrossRef](#)] [[PubMed](#)]
66. Hoang, N.-D.; Tran, X.-L. Remote Sensing-Based Urban Green Space Detection Using Marine Predators Algorithm Optimized Machine Learning Approach. *Math. Probl. Eng.* **2021**, *2021*, 5586913. [[CrossRef](#)]
67. Yang, W.; Xia, K.; Li, T.; Xie, M.; Song, F. A Multi-Strategy Marine Predator Algorithm and Its Application in Joint Regularization Semi-Supervised ELM. *Mathematics* **2021**, *9*, 291. [[CrossRef](#)]
68. Ezugwu, A.E.; Agushaka, J.O.; Abualigah, L.; Mirjalili, S.; Gandomi, A.H. Prairie Dog Optimization Algorithm. *Neural Comput. Appl.* **2022**, *34*, 20017–20065. [[CrossRef](#)]
69. Mirjalili, S. SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowl.-Based Syst.* **2016**, *96*, 120–133. [[CrossRef](#)]
70. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [[CrossRef](#)]
71. Xiao, Y.; Liu, J.; Alkhatlan, A. Informatisation of educational reform based on fractional differential equations. *Appl. Math. Nonlinear Sci.* **2021**. [[CrossRef](#)]
72. Zhang, X.; Alahmadi, D. Study on the maximum value of flight distance based on the fractional differential equation for calculating the best path of shot put. *Appl. Math. Nonlinear Sci.* **2021**. [[CrossRef](#)]
73. Che, Y.; Keir, M.Y.A. Study on the training model of football movement trajectory drop point based on fractional differential equation. *Appl. Math. Nonlinear Sci.* **2021**, *7*, 425–430. [[CrossRef](#)]
74. Chandra, S.; Hayashibe, M.; Thondiyath, A. Muscle Fatigue Induced Hand Tremor Clustering in Dynamic Laparoscopic Manipulation. *IEEE Trans. Syst. Man, Cybern. Syst.* **2018**, *50*, 5420–5431. [[CrossRef](#)]
75. Xu, W.; Chu, B.; Rogers, E. Iterative learning control for robotic-assisted upper limb stroke rehabilitation in the presence of muscle fatigue. *Control Eng. Pract.* **2014**, *31*, 63–72. [[CrossRef](#)]