Identification of Input Nonlinear Output Error System through Evolutionary Mating Optimization Algorithm

Muhammad Raza

Registration No.: 176-FET/MSEE/F22

Supervisor:

Dr. Khizer Mehmood

Co Supervisor:

Dr. Khalid Mehmood Cheema

Department of Electrical & Computer Engineering

Faculty of Engineering and Technology

International Islamic University Islamabad

2025

DISSERTATION

A dissertation submitted to the Department of Electrical Engineering, International Islamic University Islamabad as a partial fulfillment of the requirements for the award of the degree.

Department of Electrical & Computer Engineering

Faculty of Engineering and Technology

International Islamic University Islamabad

2025

DEDICATION

With the grace of Almighty ALLAH (S.W.T), I was able to complete the research work. This effort is dedicated to my respected teachers, colleagues and family. Their support and love helped me to compete this task.

CERTIFICATE OF APPROVAL

Title of Thesis: Identification of Input Nonlinear Output Error System through Evolutionary Mating Optimization Algorithm.

Name of Student: Muhammad Raza

Registration No: 176-MSEE/FET/F22

Accepted by the Faculty of Engineering and Technology, International Islamic University Islamabad in partial fulfillment of the requirement of the MS Degree in Electrical Engineering.

Viva voice committee:

Dean FET, IIU Islamabad

Assistant Professor, Chairman DEE, FET, IIUI

External Examiner

Internal Examiner

Supervisor

Dr. Khizer Mehmood

DECLARATION

I certify that research work titled “Identification of Input Nonlinear Output Error System through Evolutionary Mating Optimization Algorithm” has been completed by me and it has done before presented anywhere for evaluation. Furthermore, I have properly acknowledged the material taken from related sources.

ACKNOWLEDGEMENT

I am thankful to all my family, teachers and colleagues for their valuable contribution and support me to fulfill this task.

ABSTRACT

Optimization of Nonlinear system parameters faces various challenges in the research community due to uncertainty and correlated parameters. In this research, key term separation method is used for mathematical modeling of IN-OE system and identification is accomplished by using evolutionary-based Evolutionary Mating Algorithm (EMA) and chaotic evolutionary mating algorithm (CEMA) in exploration process of EMA. The fitness function used to identify IN-OE system parameters implements mean-square error (MSE) between desired and estimated values. Simulations results demonstrate that EMA with a chaotic sinusoidal map (CEMA9) executes better results than the EMA, its other chaotic variants, as well as other recently introduced metaheuristics for diverse variations in the system model. MSE based analysis and results of statistical test illustrate the performance of CEMA9 for the identification of the IN-OE system.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Overview

Nonlinear systems are widely used in engineering applications as well as social, economic, physical and life sciences fields [1]. Nonlinear identification models represent the dynamics of nonlinear systems especially when linear models cannot depict the accurate system parameters [2]. Hammerstein and Weiner Models provide insight knowledge about the nonlinear systems dynamics [3]. Neural state space identification are used to represent deep learning based nonlinear state space models [4]. Nonlinear grey box models can also estimate the physical parameters of nonlinear systems [5]. Application of nonlinear systems are industry 4.0 [6], visual object tracking [7], mobile robot network [8], triangulation of GPS [9], civil engineering [10], smart grids [11], auxiliary model identification [12], and many other research applications. Figure 1.1 represents nonlinear system applications that are commonly used in engineering.

Input Nonlinear output-error (IN-OE) is a block-oriented Hammerstein identification model of input nonlinear system. The parameters of IN-OE model are estimated by applying parameter identification techniques and metaheuristic algorithms.

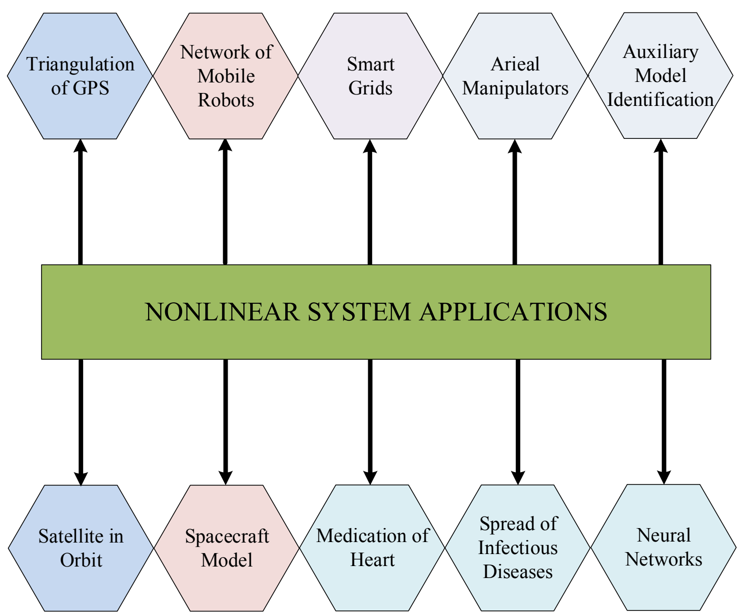


Figure 1.1: Nonlinear System Applications

1.2 Problem Statement

Optimization of IN-OE system induces an essential role in various domains of engineering problems. Traditional methods struggle in finding global optima due to various challenges such as complexity, scalability, convergence and robustness. These challenges makes identification of IN-OE system parameters is a difficult task which can be achieved by using metaheuristic algorithms. During brief literature review, it has been observed that complexity of the problem increases especially the dimensions of the problem by using the different optimization techniques. However metaheuristic algorithms still not applied on the IN-OE model. Moreover, the optimal parameters of metaheuristic algorithms vary for different problems. This research explores the identification of IN-OE system parameters through metaheuristic algorithm which is very useful in designing the controller of nonlinear systems.

1.3 Contributions

The major contributions of this research work are:

- Enhanced variants of the EMA namely CEMA1, CEMA2, CEMA3, CEMA4, CEMA5, CEMA6, CEMA7, CEMA8, CEMA9, and CEMA10 are proposed by incorporating ten chaotic maps for engineering optimization problems.
- The evaluation of CEMA1, CEMA2, CEMA3, CEMA4, CEMA5, CEMA6, CEMA7, CEMA8, CEMA9, and CEMA10 is done on mathematical functions having both uni and multimodal features.
- The robustness of CEMA1, CEMA2, CEMA3, CEMA4, CEMA5, CEMA6, CEMA7, CEMA8, CEMA9, CEMA10, is also assessed for IN-OE model against COA, EMA, HHO, and PSO indicating its realism in real-world applications.

1.4 Thesis Organization

Chapter 1 presents an overview, problem statement and objectives of research. Chapter 2 covers the literature review of harmonics estimation and mathematical model of parameter estimation of power system harmonics. Chapter 3 describes the proposed methodology based on DE and MLADE. Chapter 4 discusses the results and stimulations. Chapter 5 presents the conclusion and future work of research.

CHAPTER 2

LITERATURE REVIEW

This chapter presents the literature review of IN-OE model parameters optimization by using metaheuristic optimization algorithms.

2.1 IN-OE Model

Hammerstein and Weiner models are block oriented system identification models used to find out the solution of linear and nonlinear systems. Both models have same elements but integrated in reverse order. The nonlinear system selected for this research proposal is based on Hammerstein model known as nonlinear input output-error (IN-OE) system. It comprises of two main subsystems i.e. static nonlinear subsystem and the other one is linear dynamical subsystem [13]. Keeping in view the features of linear dynamic subsystem, IN-OE system can be classified as output-error and equation-error systems. Output error models are frequently used in the field of stochastic process specifically for the identification of nonlinear system parameters [14].

2.2 Identification Methods

Several identification techniques have been proposed in literature for IN-OE model parameters. Especially over-parameterization [15], multi-innovative identification [16], key-term separation [17], hierarchical identification [18], and auxiliary model [19] are more prominent techniques applying on nonlinear system applications.

The over-parameterization techniques can be applied on complex nonlinear systems in order to re-evaluate the system variables so that output behaves linearly. Furthermore optimization algorithms can be used to find out the best values of system key parameters. This technique has also been used with other methods to find the nonlinear system parameters. F. Ding and X. Zhang applied this technique in 2021 to identify the IN-OE system parameters. L. Xu and H. Ma proposed the hierarchical identification algorithm to identify the nonlinear IN-OE systems with high dimensions and complex structures in 2021. Feng Ding applied the key-term separation method on IN-OE model which helps to avoid excessive calculation required in over parameterization technique. Recently gradient iterative method and least square iterative technique were proposed for the identification of IN-OE system parameters. Table 2.1 represents the limitations of iterative methods and techniques applied on IN-OE model

Table 2.1: Identification methods of IN-OE

Title	Algorithm Used/Tuning Methods	Limitations	Year of Publication
Nonlinear IN-OE system using key term separation method[9]	Gradient based iterative method & least square method	Problem in finding global minima especially with noise induction	2021
Nonlinear IN-OE system using auxiliary model [10]	AM-GI, O-AM-LSI & AM-MIGI algorithms	Complexity increases with increase in no of iterations and leads to increase computational cost	2021
Nonlinear IN-OE system using over parameterization method [11]	O-AM-HLSI & O-AM-HGI algorithms	Overfitting problem and slow convergence	2023

2.3 Metaheuristic Optimization Algorithms

Metaheuristic optimization algorithms are also very effective to estimate the IN-OE system parameters. These algorithms are very useful to solve complex computational problems efficiently through optimization of system key parameters [20]. A lot of researchers proposed new metaheuristic algorithms to solve real world problems [21]. Optimal solution in energy sectors especially in the field of smart grids was proposed using harmony search metaheuristic algorithm [22]. In medical field, classification of heart related disease were also identified through combination of different metaheuristic algorithms [23]. In control system problems, metaheuristic algorithms provides optimal solution to control the attitude and altitude of unmanned aerial vehicle [24]. Metaheuristic algorithms also provide optimal solution in the networks of wireless sensors [25]. They are applied for estimation of harmonics in power systems [26]. Metaheuristic algorithms can be categorized as evolutionary process based algorithms [27], physics based algorithms [28], human inspired algorithms [29] and swarm intelligence algorithms [30]. Figure 2.1 represents the categories of metaheuristic algorithms.

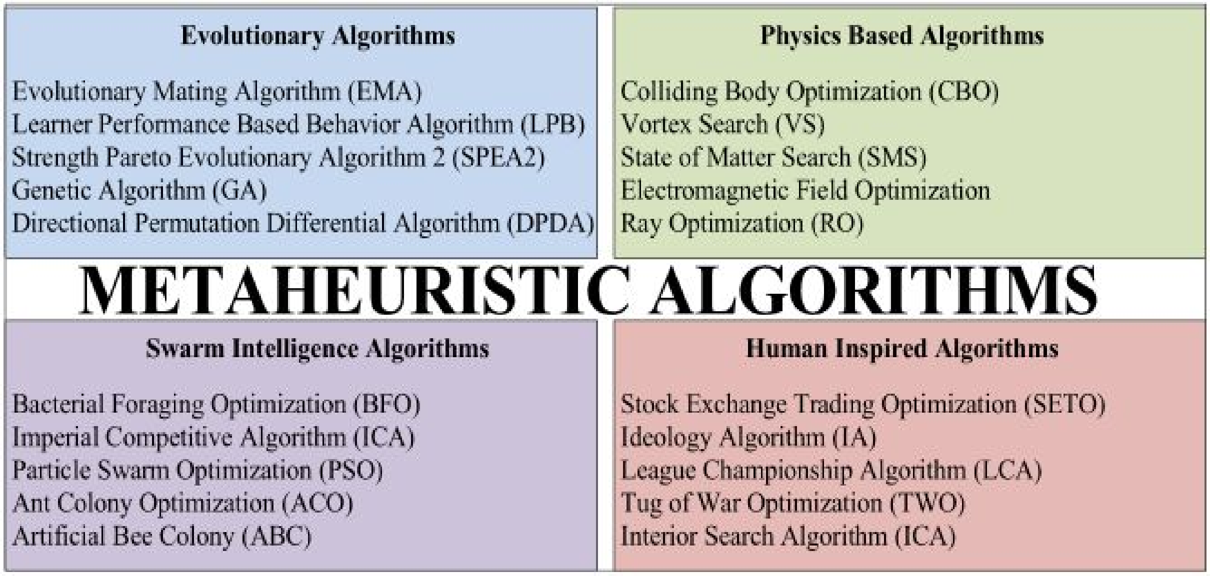


Figure 2.1: Categories of Metaheuristic Algorithms

Evolutionary Algorithms relies on Darwinian Theory. This class of metaheuristic algorithms includes Quantum based avian navigation optimizer algorithm [31], Strength Pareto Evolutionary Algorithm 2 [32], Genetic Algorithm [33], and Directional Permutation Differential Evolution Algorithm [34] and Learner Performance based Behavior algorithm [35]. Quantum based avian navigation optimizer algorithm (QANA) was proposed in 2021. This algorithm explores the idea of precise navigation of migratory birds while travelling to long-distance aerial paths. This approach divides the population into multiple groups in order to find out best parameters. This algorithm is effectively applied on partial landscape analysis. Another evolutionary strategy algorithm named as Strength Pareto Evolutionary Algorithm was proposed in 2023. SPEA is an improved version of Pareto Archived Evolutionary Strategy. SPEA2 improves local search ability to get effective results. It is successfully implemented in UAV cargo delivery services. Genetic Algorithm (GA) was proposed in 1992. GA has the ability to solve the real world problems of any engineering field. Based on fitness function GA arrange tournaments to develop new population for finding the optimal solution. In 2021, a new metaheuristic algorithm Directional Permutation Differential (DPDE) Algorithm was proposed by Shangce Gao to find out the solution of Photovoltaic Generation System. In this algorithm, strong global exploration ability helps to estimate the system parameters and avoiding from local optima. A Learner Performance based Behavior algorithm (LPB) was proposed in 2021. LBP is based on accepting graduate students in different departments at university and defining the procedures to improve the study level of students through GPA in different stages. The parameters used in LBP are crossover and mutation. This algorithm is successfully implemented on travelling salesman problem.

Metaheuristic algorithms based on physics laws are also developed in literature such as colliding body optimization [36], vortex search algorithm [37], matter search optimization algorithm [38] , electromagnetic field optimization algorithm [39], and ray optimization Algorithm [40]. Colliding Body Optimization was established in 2019. It works on colliding bodies' principle in which collision of two bodies in one direction is acceptable. Two groups are developed from population one from best side and the other group from middle. The first group is stationary while the middle group is moving towards the best solution. New mass and velocity values are obtained after the collision process. Then termination conditions are checked. It is successfully applied on systems composed of continuous variables as well as discrete variables. Vortex Search (VS) Algorithm was proposed in 2015. Its working principle is formulated on vortex-like occurrence in non-rotational incompressible fluids. It is effective in training of feed forward neural networks. The State of Matter Search optimization algorithm was proposed in 2013. The basic idea used in SMS algorithm is dependency of best solution upon the states of matter. It is successfully implemented in template matching optimization problems. Electromagnetic Field Optimization Algorithm was proposed in 2015. It is formulated on attraction and repulsion forces of electromagnet. It is successfully implemented on optimal coordination of directional over current relays. Ray Optimization Algorithm (RO) is established on Snell's Law of refraction. The direction of light changes when passes through different medium. The best solution depends upon ray scattering, ray movement and ray convergence parameters. It is successfully applied on truss structures design.

Metaheuristic Algorithms are also Swarm Intelligence (SI) based in which behavior of species i.e. birds, fish, ants, is used to provide optimal solution such as bacterial foraging algorithm [41], salp swarm algorithm [42], particle swarm optimization [43], ant colony optimization [44], synergistic swarm optimization algorithm [45] and Imperialist Competitive Algorithm [46]. Bacterial Foraging Algorithm was published in 2007. It is articulated on coli bacteria foraging behavior. The chemotactic reflexes of bacteria provide the optimal solution of real world problems. It is successfully implemented in solar PV parameters optimization problem. Salp

Swarm Algorithm (SSA) is also used to solve optimization problems. This algorithm uses swarming behavior of salps when routing and hunting in oceans. It is successfully implemented in marine propeller design parameter optimization. PSO was proposed in 1995. It is based on the movement of birds. The speed and velocities of the birds are decided to find the solution. It is successfully implemented in portfolio optimization problem. ACO was established in 1992. It relies on the collective behavior of ants to find out the solution. It is successfully used in image detection problems. Synergistic Swarm Optimization Algorithm (SSOA) integrates swarm intelligence with synergistic cooperation in order to search the efficient optimal solution. Imperialist Competitive Algorithm (ICA) was proposed in 2002. ICA is based on imperialist concept in which each agent or colony tries to make empire by capturing the small colonies. The competition among empires finds out the best solution of the problem. It is successfully applied on optimal design problem of skeletal structures.

Several human-inspired algorithms are also established to solve optimization problems such as stock exchange trading optimization [47], ideology algorithm [48], league championship algorithm [49], tug of war optimization [50], and interior search algorithm [51]. They are formulated on the human behaviors and interactions. Stock exchange trading optimization (SETO) algorithm is formulated on traders' behavior when prices fluctuate in stock market. It is successfully implemented in global optimization problem. Ideology Algorithm (IA) is based on behavior of political party's individuals who tries to improve their ranking and position in party. It is successfully applied in unconstraint optimization problems. League Championship Algorithm (LCA) was offered in 2014 used for optimization problems. In this algorithm artificial teams are developed to play championship. Progress of each team and players performance are analyzed to find out best solution of optimization problem. Tug of war optimization (TWO) was developed in 2021. The working principle of this is taken from the game tug of war. Each candidate is treated as a team participated in a rope pulling competitions. Team performance is determined by pulling force exerted on each other. TWO algorithm is very useful in multimodal and non-convex function. Interior search algorithm (ISA) was proposed in 2014. The working principle is based on interior decoration and design. Systematic methodology is used to create space for interior design and decoration strategy that fulfils customer satisfaction. The customer satisfaction is directly proportional to the solution quality. ISA shows satisfactory results on optimal welded beam design problem. Table 2.2 represents the advantages and limitations of recently proposed metaheuristic algorithms.

Table 2.2: Advantages and limitations of recently proposed Metaheuristic Algorithms

Algorithms	**Advantages**	**Limitations**	**Proposed Year**
LPB	Avoid local optima in computation problems	slow converge for complex problems	2021
SPEA2	Very efficient in local search to find dominant solution in each iteration	Not good for problems having objectives more than 5	2023
GA	they can solve problems from various domains, from engineering and medicine to finance and logistics	slow due to their structural complexity	1992
DPDE	Strong global exploration ability	computationally expensive	2021
CBO	Independent from internal parameters	low accuracy	2019
VS	Fast execution	Created number of local minimum points	2015
SMS	better performance in global optimization problems	Not suitable for complex problems	2013
EFO	Better approach to avoid the local optimal point and find global optimal	Limited to constrained optimization problems	2015
RO	Require few parameters to tune	poor local search ability	2012
BFO	effectively applied to solve real world continuous optimization problem	Convergence is very slow.	2002
ICA	Effective in solving large-scale scheduling problems	the empires are fixed until they are swapped for colonies and transformed into them	2007
PSO	computationally efficient	Not efficient for large datasets	1995

ACO	good performance in solving discrete problems	Convergence speed for large data sets	1992
ABC	Does not require external parameters like crossover ratio and mutation ratio	Not able to handle population diversity and slow global convergence	2005
SETO	Very simple to implement	gradually convergence towards optimal point	2021
IA	few parameters to adjust	Very less research work done on this algorithm	2017
LCA	solve scheduling problem easily	Limited to scheduling problems	2019
TWO	Show good results on non-convex functions	Less research done on complex problems	2016
ISA	Require tuning of only one parameter	slow convergence speed	2014

2.4 Evolutionary Mating Algorithm

Evolutionary Mating Algorithm (EMA) was proposed in 2023 [52]. It is based on Hardy Weinberg equilibrium principle to produce new offspring. Environmental factor like predator is also included in this algorithm. The main advantage of EMA comparing with other evolutionary algorithms is the capability of fast searching because it divides the whole population into two groups. EMA evaluates the performance of produced offspring and compare with performance of parents. If the performance of produced offspring is better than parents then it is directly replaced with its parents and controls the population expansion. The evaluation procedure of EMA is directly used without defining sorting process which makes EMA evaluation process faster than other optimization algorithms. The computational complexity of EMA is low because only two parameters crossover probability and predator probability need to be identified. EMA has been successfully tested on unimodal, multimodal and composite benchmark functions. EMA efficiently approaches global optima solution by avoiding local optima in multimodal and composite benchmark functions. The initialization process of EMA comprises of population generation in the form of two matrices. The selection of search agents in the matrices is based on sexual identity i e. male or female. The mating process is defined by using Hardy's principle, in which search agents are selected randomly from both matrices to produce one or more than one new offspring. The performance of new offspring is evaluated with its parents also and will be replaced if performance parameters of new offspring is better than its parent. The predetermine values to identify the crossover probability and predator probability are obtained from tuning of EMA parameters. EMA is very efficient to solve constraint optimization problems. It is applied on various areas such as optimization of pneumatic servo systems [53], battery charge estimation of electric vehicles [54], management of energy in smart buildings [55], solar power generation [56], and energy management systems [57]. However it is still not applied in the system identification problem to find out the optimal parameters of IN-OE model. This research work explores the diversity of EMA by applying on IN-OE model and identify accurate parameters using key term separation technique.

2.5 Chaos Theory

Chaos theory is a scientific approach to solve the complex system problems. According to this theory, dynamical systems are highly dependent on its initial conditions, consist of primary patterns (known as chaotic maps) and follow deterministic laws under specific time scale which depends upon the system dynamics. The theory explains that how a small change in the initial conditions can produce uncertainty in the dynamical system. This theory also deals with non-linear dynamics that illustrate the expected results in high-dimensional systems. It predicts the system response in the short term without repeating themselves, and exhibits necessary qualitative effects by introducing small changes within the process. The chaos theory enhances the performance of metaheuristic algorithms by avoiding the local optima and improves convergence speed. Applications of Chaos theory exist in several engineering fields such as chaotic generator in communication system [58], image encryption [59], internet of things [60], and random bit generators [61]. By applying chaotic maps in several metaheuristic algorithms, convergence and efficiency of the system are enhanced in search space such as chaotic Archimedes optimization algorithm [62], chaotic PSO [63], bird swarm optimization algorithm with chaotic mapping [64], chaotic young double slit experiment optimizer [65], and chaotic marine predator algorithm [66]. In this research work, chaotic variants of EMA are developed to identify the optimal parameters of IN-OE model. The comparison of EMA chaotic variants with EMA and other metaheuristic algorithms are also performed to evaluate the performance. The description of chaotic maps is given in Table 2.3.

Table 2.3: Chaotic Maps

Map No.	**Map Name**	**Map Equation**
CEMA1	Chebyshev map [67]	$x_{r+1} = \cos(r\cos^{-1}(x_r))$
CEMA2	Circle map [68]	$x_{r+1} = \mathrm{mod}(x_r + 0.2 - (\frac{0.5}{2x})\sin(2\pi x_r),1)$
CEMA3	Gauss/mouse map [69]	$x_{r+1} = \begin{cases} 1, x_r = 0 \\ \frac{1}{\mathrm{mod}(x_r,1)} \end{cases} otherwise$
CEMA4	Iterative map [70]	$x_{r+1} = \sin(\frac{0.7x}{x_r})$
CEMA5	Logistic map [71]	$x_{r+1} = 4x_r(1 - x_r)$
CEMA6	Piecewise map [72]	$x_{r+1} = \begin{cases} \frac{x_r}{0.4}, 0 \le x_r < 0.4 \\ \frac{x_r - 0.4}{0.1}, 0.4 \le x_r < 0.5 \\ \frac{0.6 - x_r}{0.1}, 0.5 \le x_r < 0.6 \\ \frac{1 - x_r}{0.4}, 0.6 \le x_r < 1 \end{cases}$
CEMA7	Sine map [73]	$x_{r+1} = \sin(2\pi x_r)$
CEMA8	Singer map [74]	$x_{r+1} = 1.07(7.8x_r + 23.31x_r^2 + 28.75x_r^3 - 13.30x_r^4)$
CEMA9	Sinusoidal map [75]	$x_{r+1} = 2.3x_r^2\sin(2\pi x_r)$
CEMA10	Tent map [76]	$x_{r+1} = \begin{cases} \frac{x_r}{0.7}, & x_r < 0.7 \\ \frac{10}{3}(1 - x_r), & x_r \ge 0.7 \end{cases}$

Conventional methods used to identify IN-OE model parameters primarily focus on local exploration and have limitations to handle local minima. This will affect the accurate identification of IN-OE model parameters in terms of solution quality and robustness. On the other hand, metaheuristic algorithms are specifically designed to explore the search spaces by using stochastic methods inspired with natural phenomena (evolution, swarm intelligence, physics laws and human based) that are strongly capable to escape from local minima and find better global solution. It can be seen from literature review that Evolutionary Mating Algorithm (EMA) is not applied for the identification of IN-OE model parameters. The motivation of this research is to explore the EMA metaheuristic algorithm for the accurate identification of IN-OE model parameters. The objective of this research to also investigate the EMA thoroughly and propose improvements to get better results for IN-OE identification problem. Finally compare the results of EMA with other metaheuristic algorithms.

CHAPTER 3

Methodology

In this chapter mathematical models of IN-OE, EMA and chaotic EMA are presented. Pseudo code and flow charts of EMA and Chaotic EMA are also discussed.

3.1 Mathematical Model of IN-OE

Consider the input nonlinear system represented in Figure. 3.1.

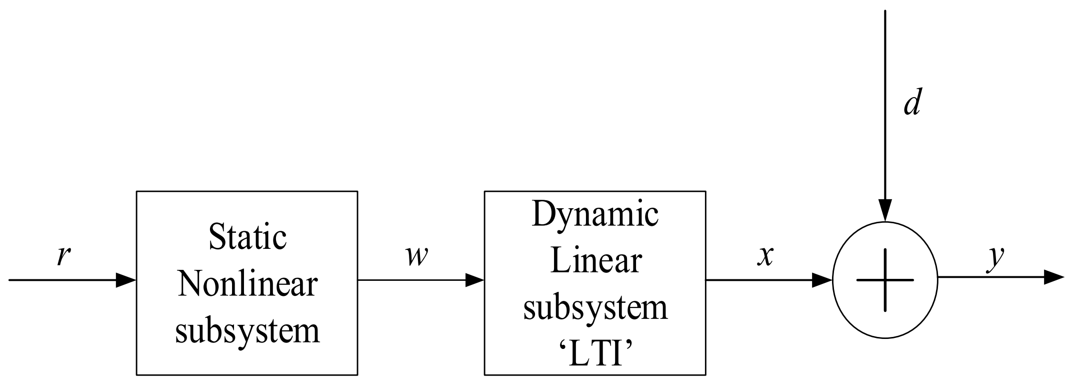


Figure 3.1: IN-OE System Model

Where 'r' is the input of static nonlinear block,

'w' is the output of nonlinear subsystem,

'x' is the output of linear time invariant system,

'd' is disturbance or noise induce in the system

and 'y' is the output of IN-OE model. The output of IN-OE model is given by:

$$y(\tau) = x(\tau) + d(\tau), \tag{3.1}$$

The output eq. of linear subsystem is given by

$$x(\tau) = \frac{C}{D} w(\tau), \tag{3.2}$$

Where C and D are the polynomials with q^{-1} operator and represented as follows:

$$C = 1 + c_1 q^{-1} + c_2 q^{-2} + \ldots + c_{n_c} q^{-n_c}, \quad (3.3)$$

$$D = 1 + d_1 q^{-1} + d_2 q^{-2} + \ldots + d_{n_d} q^{-n_d}, \quad (3.4)$$

$w(\tau)$ belongs to real number $r(\tau)$ along with set of known basis functions $f_k(r(\tau))$ with parameters β_k, therefore output of nonlinear system is represented as

$$w(\tau) = f(r(\tau)) = \sum_{k=1}^{m} \beta_k f_k(r(\tau)) = \beta_1 f_1(r(\tau)) + \beta_2 f_2(r(\tau)) + \ldots + \beta_m f_m(r(\tau)), \quad (3.5)$$

It is seen from the above equations that output of nonlinear system is in series combination with the transfer function of the LTI subsystem. So for any non-zero value of β_k gives identifiable relation between input and output of IN-OE model.

By applying key term separation methodology, IN-OE system is defined as

$$\nu = [c^{\mathrm{T}}, d^{\mathrm{T}}, \beta^{\mathrm{T}}]^{\mathrm{T}} \in R^{s}, \quad (3.6)$$

$$\lambda_2(\tau) = [\lambda_c^{\mathrm{T}}(\tau), \lambda_d^{\mathrm{T}}(\tau), f^{\mathrm{T}}(\tau)]^{\mathrm{T}} \in R^{s}, \quad (3.7)$$

$$y(\tau) = \lambda_2^{\mathrm{T}}(\tau)\nu + v(\tau), \quad (3.8)$$

Equation (8) represents the identification model obtained from key term separation method, where ν represents the parameter vector needs to be identified and it includes all system parameters. λ_2 denotes the information vector and it corresponds nonlinear subsystem relationship with LTI dynamic subsystem.

3.2 Mathematical Model of EMA

In EMA, male and female candidates used in solution are represented as follows:

$$A_m = \begin{pmatrix} a_1^1 & \cdots & a_1^d \\ \vdots & \ddots & \vdots \\ a_{n/2}^1 & \cdots & a_{n/2}^d \end{pmatrix}, \quad (3.9)$$

$$A_f = \begin{pmatrix} a^1_{\frac{n}{2}+1} & \cdots & a^d_{\frac{n}{2}+1} \\ \vdots & \ddots & \vdots \\ a^1_n & \cdots & a^d_n \end{pmatrix}, \tag{3.10}$$

$$H_{mates} = H_{mating\_ratio} + [{}^*H_{mates(t)} - {}^*H_{mates(k)}], \tag{3.11}$$

$$H_{mates} = 1 + [\mathrm{var}(A^T_{m,*}) - \mathrm{var}(A^T_{f,*})], \tag{3.12}$$

Hardy–Weinberg principle will be applied to get new offspring, the

$$A^T_{child} = \begin{Bmatrix} o.*X^T_{m,*} + r.a^T_{f,*} & for & H_{mates} \geq 0 \\ o.*X^T_{f,*} + r.a^T_{m,*} & for & H_{mates} < 0 \end{Bmatrix}, \tag{3.13}$$

$$o = randn(1, d), \tag{3.14}$$

where

$$r = (1 - o), \tag{3.15}$$

new offspring is represented as follows:

$$A^{T+1}_{child} = U.*A^T_{childi} + A^{best}_i.*(1-U), i = 1, 2, ..., d \tag{3.16}$$

$$U = rand(1, d) < Wr \tag{3.17}$$

Therefore,

$$A^{T+1}_{child} = rand(1, d).*A^{best}_i, for\ s <\in [0,1] \tag{3.18}$$

It is noted that only two parameters are required to tune crossover probability '*Wr*' and predator probability's'. The pseudo code of EMA is shown below where as its flowchart is shown in Figure 3.2.

Algorithm 1: EMA
Initialization population matrices A_m and A_f by using equations (3.9) and (3.10) Set W_r and s values

Evaluate the fitness of each candidate

Choose the best candidate A_i^{best}

while (P<maximum iteration)

    for (n=1 until half of population)

        Calculate H_{mates} using equations (3.11) and (3.12)

        Create the new offspring using expressions (3.13) and (3.14)

        New offspring with the effect of best solution so far using equations (3.16) and (3.17)

        Boundary Check

        Calculate fitness of new offspring

        Compare the fitness with father, mother and current best solution

        If better then replace and update in father/mother pools and current best sol.

        Else die

            If r<pre-set value [probability of encountering the predator]

            Compute solution by applying equation (3.18)

            Analyze the new candidate fitness

            If better then replace and update in current best

            Else die

            end

    end

  end

 P=P+1

end

Return A_i^{best}

Figure 3.2: EMA Flow Chart

3.3 Mathematical model of Chaotic EMA

In this article, ten improved variants of the EMA were proposed by incorporating ten eminent chaotic maps in its exploration mechanism. The mathematical model of chaotic variants of EMA for the initialization of population and evaluation of candidate's fitness is same as mentioned in eq. (3.9) to eq. (3.17). The exploration process of chaotic variants of EMA are described as follows:

$$A_{child}^{T+1} = C_s(1,d).*A_i^{best}, for\ s <\in [0,1] \tag{3.19}$$

Therefore C_s for chaotic variants of EMA are describes as follow:

CEMA1: $$C_s = \cos(i\cos^{-1}(x_i)) \tag{3.20}$$

CEMA2: $$C_s = \mathrm{mod}(x_i + 0.2 - (\frac{0.5}{2x})\sin(2\pi x_i),1) \tag{3.21}$$

CEMA3: $$C_s = \begin{cases} 1, x_i = 0 \\ \frac{1}{\mathrm{mod}(x_i,1)} \end{cases} otherwise \tag{3.22}$$

CEMA4: $$C_s = \sin(\frac{0.7x}{x_i}) \tag{3.23}$$

CEMA5: $$C_s = 4x_i(1-x_i) \tag{3.24}$$

CEMA6: $$C_s = \begin{cases} \frac{x_i}{0.4}, 0 \le x_i < 0.4 \\ \frac{x_i - 0.4}{0.1}, 0.4 \le x_i < 0.5 \\ \frac{0.6 - x_i}{0.1}, 0.5 \le x_i < 0.6 \\ \frac{1 - x_i}{0.4}, 0.6 \le x_i < 1 \end{cases} \tag{3.25}$$

CEMA7: $$C_s = \sin(2\pi x_i) \tag{3.26}$$

CEMA8 $$C_s = 1.07(7.8x_i + 23.31x_i^2 + 28.75x_i^3 - 13.30x_i^4) \tag{3.27}$$

CEMA: $$C_s = 2.3x_i^2 \sin(2\pi x_i) \tag{3.28}$$

CEMA10: $$C_s = \begin{cases} \frac{x_i}{0.7}, & x_i < 0.7 \\ \frac{10}{3}(1-x_i), & x_i \geq 0.7 \end{cases} \tag{3.29}$$

The pseudo code of chaotic EMA is shown below where as its flowchart is shown in Figure 3.3.

Algorithm 2: CEMA

Initialization population matrices A_m and A_f by using equations (3.9) and (3.10)
Set $W_{r.}$ and s values
Evaluate the fitness of each candidate
Choose the best candidate A_i^{best}
while (P<maximum iteration)
    for (n=1 until half of population)
        Calculate H_{mates} using equations (3.11) and (3.12)
        Create the new offspring using expressions (3.13) and (3.14)
        New offspring with the effect of best solution so far using equations (3.16) and (3.17)
        Boundary Check
        Calculate fitness of new offspring
        Compare the fitness with father, mother and current best solution
        If better then replace and update in father/mother pools and current best sol.
        Else die
            If C_s<pre-set value [probability of encountering the predator] equations (3.19) to (3.29)
            Compute solution by applying equation (3.18)
            Analyze the new candidate fitness
            If better then replace and update in current best
            Else die
            end
    end
  end

P=P+1

end

Return A_i^{best}

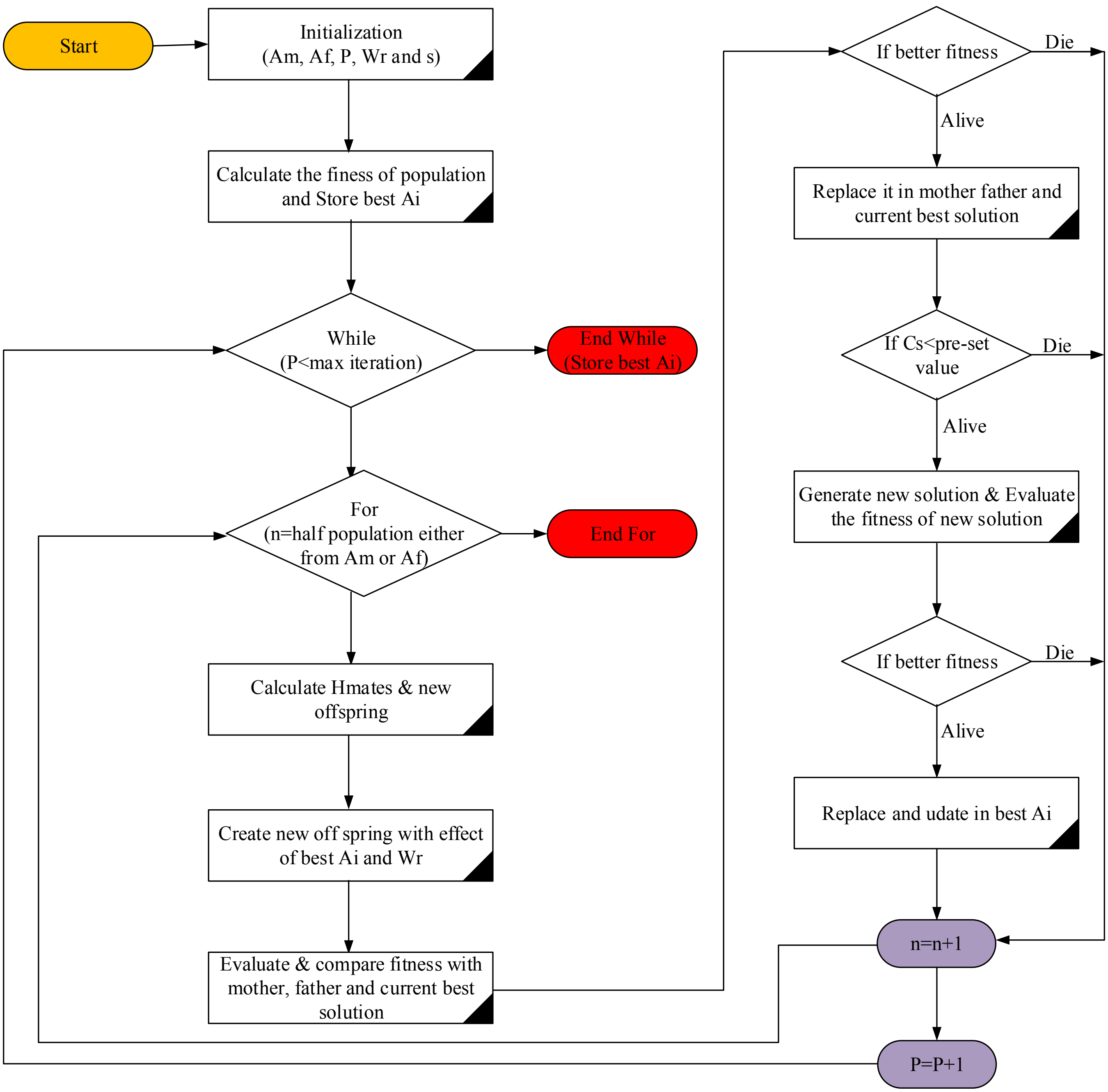


Figure 3.3: Chaotic EMA Flow Chart

CHAPTER 4

Simulations and Analysis

In this chapter, simulation results of EMA, Chaotic variants of EMA and other metaheuristic algorithms (COA, HHO and PSO) for mathematical functions and IN-OE model are presented.

4.1 Mathematical Functions

Tables 4.1-4.22 shows the analysis of mathematical functions at Population (Pop) =60, iterations =3000 for 50 independent runs in respect of STD, best fitness, worst fitness and average fitness.

Table 4.1 represents the results of EMA and its chaotic variants on unimodal mathematical function FUN_I. It is observed from Table 4.1 that EMA shows better performance in terms of average fitness, best fitness and worst fitness while all algorithms have zero STD on unimodal mathematical function FUN_I.

Table 4.1: Analysis of Proposed methodology on FUN_I function

FUN_I				
Methods	**A Fitness**	**B Fitness**	**W Fitness**	**STD**
EMA	1.02E-289	2.86E-308	4.13E-288	0
CEMA1	8.66E-214	2.04E-231	2.62E-212	0
CEMA2	4.51E-192	5.63E-208	1.89E-190	0
CEMA3	1.07E-291	0.00E+00	5.34E-290	0
CEMA4	3.87E-205	5.69E-221	1.88E-203	0
CEMA5	4.13E-210	1.20E-226	2.06E-208	0
CEMA6	1.68E-185	7.01E-208	8.40E-184	0
CEMA7	1.46E-223	8.92E-253	5.25E-222	0
CEMA8	6.53E-172	8.52E-187	3.26E-170	0
CEMA9	1.07E-138	9.71E-146	4.63E-137	0
CEMA10	6.38E-186	7.56E-200	2.40E-184	0

Table 4.2 represents the results of EMA and its chaotic variants on unimodal mathematical function FUN_II. It is observed from Table 4.2 that CEMA3 shows better performance in terms of average fitness, best fitness and worst fitness while CEMA7 shows better results in respect of STD on unimodal mathematical function FUN_II.

Table 4.2: Analysis of Proposed methodology on FUN_II function

FUN_II				
Methods	**A Fitness**	**B Fitness**	**W Fitness**	**STD**
EMA	1.54E-188	1.73E-198	4.49E-187	0.00E+00
CEMA1	1.31E-150	6.41E-165	2.98E-149	5.13E-150
CEMA2	1.33E-134	1.85E-141	3.66E-133	6.05E-134
CEMA3	5.47E-199	1.59E-254	2.73E-197	0.00E+00
CEMA4	3.34E-144	5.04E-165	1.44E-142	2.04E-143
CEMA5	3.95E-147	7.42E-168	1.32E-145	2.01E-146
CEMA6	5.51E-133	6.19E-143	1.17E-131	2.32E-132
CEMA7	3.58E-153	2.78E-169	9.16E-152	1.70E-152
CEMA8	3.26E-118	1.66E-127	1.62E-116	2.30E-117
CEMA9	2.16E-92	3.98E-95	3.47E-91	5.68E-92
CEMA10	7.68E-131	2.86E-145	1.42E-129	2.62E-130

Table 4.3 represents the results of EMA and its chaotic variants on unimodal mathematical function FUN_III. It is observed from Table 4.3 that CEMA3 shows better performance on unimodal mathematical function FUN_III in terms of STD, average fitness, best fitness and worst fitness.

Table 4.2: Analysis of Proposed methodology on FUN_II function

FUN_III				
Methods	**A Fitness**	**B Fitness**	**W Fitness**	**STD**
EMA	2.06E-125	1.46E-143	1.01E-123	1.42E-124
CEMA1	8.55E-72	4.50E-94	4.10E-70	5.79E-71
CEMA2	7.51E-55	6.79E-68	2.36E-53	3.55E-54

CEMA3	2.18E-133	1.35E-178	1.09E-131	1.54E-132
CEMA4	6.19E-64	1.67E-80	2.91E-62	4.12E-63
CEMA5	1.96E-70	3.15E-89	9.13E-69	1.29E-69
CEMA6	2.48E-57	2.28E-75	4.90E-56	9.85E-57
CEMA7	7.55E-78	2.07E-101	3.38E-76	4.78E-77
CEMA8	2.36E-43	3.01E-57	1.05E-41	1.49E-42
CEMA9	1.41E-30	4.12E-36	2.56E-29	4.88E-30
CEMA10	6.43E-56	5.04E-72	1.69E-54	2.74E-55

Table 4.4 represents the results of EMA and its chaotic variants on unimodal mathematical function FUN_IV. It is observed from Table 4.4 that EMA shows better performance in terms of STD, average fitness and worst fitness while CEMA3 shows better results in respect of best fitness on unimodal mathematical function FUN_IV.

Table 4.3: Analysis of Proposed methodology on FUN_IV function

FUN_IV				
Methods	**A Fitness**	**B Fitness**	**W Fitness**	**STD**
EMA	1.30E-98	1.11E-105	3.90E-97	6.10E-98
CEMA1	1.28E-60	2.64E-68	2.76E-59	4.21E-60
CEMA2	2.83E-49	1.58E-54	5.84E-48	9.65E-49
CEMA3	4.14E-97	1.03E-138	8.07E-96	1.36E-96
CEMA4	1.44E-54	7.40E-64	6.28E-53	8.90E-54
CEMA5	1.12E-58	6.15E-66	4.68E-57	6.61E-58
CEMA6	1.43E-49	1.59E-56	2.56E-48	4.67E-49
CEMA7	2.75E-62	2.00E-69	1.18E-60	1.67E-61
CEMA8	1.97E-41	6.92E-46	6.50E-40	9.26E-41
CEMA9	1.35E-29	1.62E-34	4.24E-28	6.11E-29
CEMA10	5.07E-48	1.20E-54	2.13E-46	3.02E-47

Table 4.5 represents the results of EMA and its chaotic variants on unimodal mathematical function FUN_V. It is observed from Table 4.5 that EMA shows better performance in terms of STD, average fitness and worst fitness while CEMA8 shows better results in respect of best fitness on unimodal mathematical function FUN_V.

Table 4.4: Analysis of Proposed methodology on FUN_V function

FUN_V				
Methods	**A Fitness**	**B Fitness**	**W Fitness**	**STD**
EMA	4.37E-01	3.45E-01	5.07E-01	3.76E-02
CEMA1	2.22E+00	2.76E-01	8.63E+01	1.21E+01
CEMA2	5.15E-01	3.51E-01	4.08E+00	5.17E-01
CEMA3	4.41E-01	3.66E-01	5.43E-01	4.26E-02
CEMA4	4.45E-01	3.60E-01	5.80E-01	4.48E-02
CEMA5	4.38E-01	2.88E-01	5.43E-01	5.46E-02
CEMA6	6.21E-01	2.41E-01	5.47E+00	8.65E-01
CEMA7	5.04E-01	3.67E-01	3.17E+00	3.87E-01
CEMA8	6.27E-01	1.04E-03	4.25E+00	7.94E-01
CEMA9	1.77E+00	3.03E-03	9.76E+00	2.39E+00
CEMA10	6.55E-01	1.61E-01	5.89E+00	8.53E-01

Table 4.6 represents the results of EMA and its chaotic variants on unimodal mathematical function FUN_VI. It is observed from Table 4.6 that all algorithms have zero STD, average fitness, best fitness and worst fitness on unimodal mathematical function FUN_VI.

Table 4.5: Analysis of Proposed methodology on FUN_VI function

FUN_VI				
Methods	**A Fitness**	**B Fitness**	**W Fitness**	**STD**
EMA	0	0	0	0
CEMA1	0	0	0	0
CEMA2	0	0	0	0
CEMA3	0	0	0	0

CEMA4	0	0	0	0
CEMA5	0	0	0	0
CEMA6	0	0	0	0
CEMA7	0	0	0	0
CEMA8	0	0	0	0
CEMA9	0	0	0	0
CEMA10	0	0	0	0

Table 4.7 represents the results of EMA and its chaotic variants on unimodal mathematical function FUN_VII. It is observed from Table 4.7 that CEMA9 shows better performance in terms of average fitness, CEMA2 shows better results in terms of best fitness and worst fitness. While, CEMA7 shows better results in respect of STD on unimodal mathematical function FUN_VII.

Table 4.6: Analysis of Proposed methodology on FUN_VII function

FUN_VII				
Methods	**A Fitness**	**B Fitness**	**W Fitness**	**STD**
EMA	1.35E-05	6.54E-07	5.93E-05	1.20E-05
CEMA1	1.29E-05	3.57E-07	4.63E-05	1.12E-05
CEMA2	1.10E-05	3.49E-08	4.23E-05	1.17E-05
CEMA3	1.29E-05	1.26E-07	6.09E-05	1.26E-05
CEMA4	1.15E-05	1.61E-07	7.13E-05	1.39E-05
CEMA5	1.49E-05	1.00E-07	9.30E-05	1.82E-05
CEMA6	1.13E-05	4.87E-08	5.11E-05	1.02E-05
CEMA7	1.12E-05	7.57E-08	4.64E-05	9.90E-06
CEMA8	1.40E-05	6.63E-07	5.47E-05	1.25E-05
CEMA9	1.08E-05	7.38E-08	6.51E-05	1.32E-05
CEMA10	1.40E-05	2.12E-07	9.00E-05	1.61E-05

Table 4.8 represents the results of EMA and its chaotic variants on multimodal mathematical function FUN_VIII. It is observed from Table 4.8 that EMA shows better performance in terms of average fitness. EMA, CEMA1, CEMA2, CEMA3, CEMA4 and CEMA8 shows better results in terms of best fitness. While, EMA shows better results in respect of STD and worst fitness on multimodal mathematical function FUN_VIII.

Table 4.7: Analysis of Proposed methodology on FUN_VIII function

FUN_VIII				
Methods	**A Fitness**	**B Fitness**	**W Fitness**	**STD**
EMA	-3.62E+03	-4.19E+03	-3.00E+03	2.69E+02
CEMA1	-3.55E+03	-4.19E+03	-2.52E+03	3.46E+02
CEMA2	-3.63E+03	-4.19E+03	-2.88E+03	3.15E+02
CEMA3	-3.54E+03	-4.19E+03	-2.76E+03	3.22E+02
CEMA4	-3.57E+03	-4.19E+03	-2.76E+03	3.21E+02
CEMA5	-3.49E+03	-4.07E+03	-2.76E+03	3.18E+02
CEMA6	-3.45E+03	-4.07E+03	-2.76E+03	2.82E+02
CEMA7	-3.54E+03	-4.07E+03	-2.64E+03	3.57E+02
CEMA8	-3.53E+03	-4.19E+03	-2.52E+03	3.36E+02
CEMA9	-3.57E+03	-4.07E+03	-2.88E+03	2.83E+02
CEMA10	-3.54E+03	-3.95E+03	-2.88E+03	2.98E+02

Table 4.9 represents the results of EMA and its chaotic variants on multimodal mathematical function FUN_IX. It is observed from Table 4.9 that CEMA2 shows better performance in terms of average fitness. CEMA1 shows better results in terms of best fitness. CEMA3, CEMA4, CEMA5 and CEMA7 show better results in respect of worst fitness while CEMA10 shows better performance on multimodal mathematical function FUN_IX in terms of STD.

Table 4.8: Analysis of Proposed methodology on FUN_IX function

FUN_IX				
Methods	**A Fitness**	**B Fitness**	**W Fitness**	**STD**
EMA	1.79E-01	0.00E+00	8.95E+00	1.27E+00

CEMA1	8.95E-01	-4.19E+03	1.09E+01	2.79E+00
CEMA2	9.95E-02	0.00E+00	4.97E+00	7.04E-01
CEMA3	0	0	0	0
CEMA4	0	0	0	0
CEMA5	0	0	0	0
CEMA6	4.58E-01	0.00E+00	9.95E+00	1.88E+00
CEMA7	0	0	0	0
CEMA8	8.76E-01	0.00E+00	9.95E+00	2.47E+00
CEMA9	6.21E+00	9.95E-01	1.19E+01	2.55E+00
CEMA10	1.99E-01	0.00E+00	4.97E+00	9.85E-01

Table 4.10 represents the results of EMA and its chaotic variants on multimodal mathematical function FUN_X. It is observed from Table 4.10 that all algorithms show almost similar results on multimodal mathematical function FUN_X in terms of STD, average fitness, best fitness and worst fitness.

Table 4.9: Analysis of Proposed methodology on FUN_X function

FUN_X				
Methods	**A Fitness**	**B Fitness**	**W Fitness**	**STD**
EMA	8.88E-16	8.88E-16	8.88E-16	0
CEMA1	8.88E-16	8.88E-16	8.88E-16	0
CEMA2	8.88E-16	8.88E-16	8.88E-16	0
CEMA3	8.88E-16	8.88E-16	8.88E-16	0
CEMA4	8.88E-16	8.88E-16	8.88E-16	0
CEMA5	8.88E-16	8.88E-16	8.88E-16	0
CEMA6	8.88E-16	8.88E-16	8.88E-16	0
CEMA7	8.88E-16	8.88E-16	8.88E-16	0
CEMA8	8.88E-16	8.88E-16	8.88E-16	0
CEMA9	4.16E-15	8.88E-16	4.44E-15	0
CEMA10	8.88E-16	8.88E-16	8.88E-16	0

Table 4.11 represents the results of EMA and its chaotic variants on multimodal mathematical function FUN_XI. It is observed from Table 4.11 that EMA shows better performance in terms of average fitness. All algorithms show similar results in terms of best fitness. CEMA4 shows better results in respect of worst fitness while CEMA3 shows better performance on multimodal mathematical function FUN_XI in terms of STD.

Table 4.10: Analysis of Proposed methodology on FUN_XI function

FUN_XI				
Methods	**A Fitness**	**B Fitness**	**W Fitness**	**STD**
EMA	9.85E-03	0	1.01E-01	2.38E-02
CEMA1	1.86E-02	0	1.11E-01	3.37E-02
CEMA2	3.23E-02	0	1.11E-01	3.96E-02
CEMA3	5.51E-03	0	9.10E-02	1.95E-02
CEMA4	8.02E-03	0	7.87E-02	2.11E-02
CEMA5	1.83E-02	0	1.65E-01	4.01E-02
CEMA6	2.21E-02	0	1.60E-01	3.57E-02
CEMA7	1.66E-02	0	1.35E-01	3.24E-02
CEMA8	2.30E-02	0	1.99E-01	4.23E-02
CEMA9	6.30E-02	0	1.45E-01	3.39E-02
CEMA10	4.87E-02	0	1.72E-01	4.47E-02

Table 4.12 represents the results of EMA and its chaotic variants on multimodal mathematical function FUN_XII. It is observed from Table 4.12 that all algorithms show almost similar results in terms of STD, best fitness, average fitness and worst fitness.

Table 4.11: Analysis of Proposed methodology on FUN_XII function

FUN_XII				
Methods	**A Fitness**	**B Fitness**	**W Fitness**	**STD**
EMA	4.71E-32	4.71E-32	4.71E-32	1.66E-47
CEMA1	4.71E-32	4.71E-32	4.71E-32	1.66E-47
CEMA2	6.22E-03	4.71E-32	3.11E-01	4.40E-02

CEMA3	4.71E-32	4.71E-32	4.71E-32	1.66E-47
CEMA4	1.24E-02	4.71E-32	3.11E-01	6.16E-02
CEMA5	4.71E-32	4.71E-32	4.71E-32	1.66E-47
CEMA6	4.71E-32	4.71E-32	4.71E-32	1.66E-47
CEMA7	4.71E-32	4.71E-32	4.71E-32	1.66E-47
CEMA8	4.71E-32	4.71E-32	4.71E-32	1.66E-47
CEMA9	6.22E-03	4.71E-32	3.11E-01	4.40E-02
CEMA10	4.71E-32	4.71E-32	4.71E-32	1.66E-47

Table 4.13 represents the results of EMA and its chaotic variants on multimodal mathematical function FUN_XIII. It is observed from Table 4.13 that EMA, CEMA8 and CEMA9 show better performance in terms of STD and average fitness. All algorithms show similar results on multimodal mathematical function FUN_XIII in terms of best fitness and worst fitness.

Table 4.12: Analysis of Proposed methodology on FUN_XIII function

FUN_XIII				
Methods	**A Fitness**	**B Fitness**	**W Fitness**	**STD**
EMA	1.32E-03	1.35E-32	1.10E-02	3.61E-03
CEMA1	2.64E-03	1.35E-32	1.10E-02	4.74E-03
CEMA2	2.20E-03	1.35E-32	1.10E-02	4.44E-03
CEMA3	1.76E-03	1.35E-32	1.10E-02	4.07E-03
CEMA4	2.20E-03	1.35E-32	1.10E-02	4.44E-03
CEMA5	4.58E-03	1.35E-32	9.74E-02	1.42E-02
CEMA6	2.20E-03	1.35E-32	1.10E-02	4.44E-03
CEMA7	1.98E-03	1.35E-32	1.10E-02	4.26E-03
CEMA8	1.32E-03	1.35E-32	1.10E-02	3.61E-03
CEMA9	1.32E-03	1.35E-32	1.10E-02	3.61E-03
CEMA10	1.98E-03	1.35E-32	1.10E-02	4.26E-03

Table 4.14 represents the results of EMA and its chaotic variants on multimodal mathematical function FUN_XIV. It is observed from Table 4.14 that CEMA3 shows better performance in terms of average fitness. All algorithms show similar results on multimodal mathematical function FUN_XIV in terms of best fitness and worst fitness. CEMA1 shows better performance in terms of STD.

Table 4.13: Analysis of Proposed methodology on FUN_XIV function

FUN_XIV				
Methods	**A Fitness**	**B Fitness**	**W Fitness**	**STD**
EMA	1.59E+00	9.98E-01	1.08E+01	1.78E+00
CEMA1	1.24E+00	9.98E-01	5.93E+00	7.87E-01
CEMA2	1.63E+00	9.98E-01	1.08E+01	1.78E+00
CEMA3	1.23E+00	9.98E-01	5.93E+00	9.88E-01
CEMA4	1.55E+00	9.98E-01	1.17E+01	1.79E+00
CEMA5	1.69E+00	9.98E-01	1.08E+01	1.79E+00
CEMA6	1.28E+00	9.98E-01	5.93E+00	8.24E-01
CEMA7	1.49E+00	9.98E-01	5.93E+00	1.37E+00
CEMA8	1.61E+00	9.98E-01	1.27E+01	1.95E+00
CEMA9	1.55E+00	9.98E-01	1.08E+01	1.80E+00
CEMA10	1.59E+00	9.98E-01	7.87E+00	1.52E+00

Table 4.15 represents the results of EMA and its chaotic variants on multimodal mathematical function FUN_XV. It is observed from Table 4.15 that CEMA2 shows better performance in terms of average fitness and STD. All algorithms show similar results on multimodal mathematical function FUN_XV in terms of best fitness and worst fitness.

Table 4.14: Analysis of Proposed methodology on FUN_XV function

FUN_XV				
Methods	**A Fitness**	**B Fitness**	**W Fitness**	**STD**

EMA	9.10E-04	3.07E-04	2.04E-02	2.83E-03
CEMA1	1.29E-03	3.07E-04	2.04E-02	3.95E-03
CEMA2	3.81E-04	3.07E-04	1.22E-03	2.51E-04
CEMA3	7.82E-04	3.07E-04	2.04E-02	2.84E-03
CEMA4	9.47E-04	3.07E-04	2.04E-02	2.83E-03
CEMA5	4.91E-04	3.07E-04	1.22E-03	3.70E-04
CEMA6	1.15E-03	3.07E-04	2.04E-02	3.97E-03
CEMA7	8.75E-04	3.07E-04	2.04E-02	2.84E-03
CEMA8	9.28E-04	3.07E-04	2.04E-02	2.83E-03
CEMA9	9.28E-04	3.07E-04	2.04E-02	2.83E-03
CEMA10	1.58E-03	3.07E-04	2.04E-02	4.80E-03

Table 4.16 represents the results of EMA and its chaotic variants on multimodal mathematical function FUN_XVI. It is observed from Table 4.16 that all algorithms show similar results on multimodal mathematical function FUN_XVI in terms of average fitness, best fitness and worst fitness. While all chaotic variants of EMA show better performance in terms of STD.

Table 4.15: Analysis of Proposed methodology on FUN_XVI function

FUN_XVI				
Methods	**A Fitness**	**B Fitness**	**W Fitness**	**STD**
EMA	-1.03E+00	-1.03E+00	-1.03E+00	2.12E-11
CEMA1	-1.03E+00	-1.03E+00	-1.03E+00	3.46E-16
CEMA2	-1.03E+00	-1.03E+00	-1.03E+00	3.33E-16
CEMA3	-1.03E+00	-1.03E+00	-1.03E+00	3.63E-16
CEMA4	-1.03E+00	-1.03E+00	-1.03E+00	3.33E-16
CEMA5	-1.03E+00	-1.03E+00	-1.03E+00	3.31E-16
CEMA6	-1.03E+00	-1.03E+00	-1.03E+00	3.33E-16
CEMA7	-1.03E+00	-1.03E+00	-1.03E+00	3.42E-16
CEMA8	-1.03E+00	-1.03E+00	-1.03E+00	3.37E-16
CEMA9	-1.03E+00	-1.03E+00	-1.03E+00	3.59E-16

CEMA10	-1.03E+00	-1.03E+00	-1.03E+00	3.46E-16

Table 4.17 represents the results of EMA and its chaotic variants on multimodal mathematical function FUN_XVII. It is observed from Table 4.17 that all algorithms show similar results on multimodal mathematical function FUN_XVII in terms of average fitness, best fitness and worst fitness. While, CEMA8 shows better performance in terms of STD.

Table 4.16: Analysis of Proposed methodology on FUN_XVII function

FUN_XVII				
Methods	**A Fitness**	**B Fitness**	**W Fitness**	**STD**
EMA	3.00E+00	3.00E+00	3.00E+00	2.66E-15
CEMA1	3.00E+00	3.00E+00	3.00E+00	1.06E-10
CEMA2	3.00E+00	3.00E+00	3.00E+00	7.23E-15
CEMA3	3.00E+00	3.00E+00	3.00E+00	2.67E-15
CEMA4	3.00E+00	3.00E+00	3.00E+00	3.28E-15
CEMA5	3.00E+00	3.00E+00	3.00E+00	3.72E-15
CEMA6	3.54E+00	3.00E+00	3.00E+01	3.82E+00
CEMA7	3.00E+00	3.00E+00	3.00E+00	2.53E-15
CEMA8	3.00E+00	3.00E+00	3.00E+00	2.22E-15
CEMA9	4.62E+00	3.00E+00	3.00E+01	6.48E+00
CEMA10	3.00E+00	3.00E+00	3.00E+00	2.86E-15

Table 4.18 represents the results of EMA and its chaotic variants on multimodal mathematical function FUN_XVIII. It is observed from Table 4.18 that all algorithms show similar results on multimodal mathematical function FUN_XVIII in terms of STD, average fitness, best fitness and worst fitness.

Table 4.17: Analysis of Proposed methodology on FUN_XVIII function

FUN_XVIII				
Methods	**A Fitness**	**B Fitness**	**W Fitness**	**STD**
EMA	-3.86E+00	-3.86E+00	-3.86E+00	3.14E-15
CEMA1	-3.86E+00	-3.86E+00	-3.86E+00	3.14E-15

CEMA2	-3.86E+00	-3.86E+00	-3.86E+00	3.14E-15
CEMA3	-3.86E+00	-3.86E+00	-3.86E+00	3.14E-15
CEMA4	-3.86E+00	-3.86E+00	-3.86E+00	3.14E-15
CEMA5	-3.86E+00	-3.86E+00	-3.86E+00	3.14E-15
CEMA6	-3.86E+00	-3.86E+00	-3.86E+00	3.14E-15
CEMA7	-3.86E+00	-3.86E+00	-3.86E+00	3.14E-15
CEMA8	-3.86E+00	-3.86E+00	-3.86E+00	3.14E-15
CEMA9	-3.86E+00	-3.86E+00	-3.86E+00	3.14E-15
CEMA10	-3.86E+00	-3.86E+00	-3.86E+00	3.14E-15

Table 4.19 represents the results of EMA and its chaotic variants on multimodal mathematical function FUN_XIX. It is observed from Table 4.19 that all algorithms show almost similar results on multimodal mathematical function FUN_XIX in terms of average fitness, best fitness and worst fitness. While CEMA9 performs better results on multimodal mathematical function FUN_XIX in terms of STD.

Table 4.18: Analysis of Proposed methodology on FUN_XIX function

FXIX				
Methods	**A Fitness**	**B Fitness**	**W Fitness**	**STD**
EMA	-3.26E+00	-3.32E+00	-3.20E+00	6.01E-02
CEMA1	-3.25E+00	-3.32E+00	-3.14E+00	6.06E-02
CEMA2	-3.25E+00	-3.32E+00	-3.20E+00	5.84E-02
CEMA3	-3.25E+00	-3.32E+00	-3.20E+00	5.89E-02
CEMA4	-3.26E+00	-3.32E+00	-3.20E+00	6.01E-02
CEMA5	-3.25E+00	-3.32E+00	-3.14E+00	6.16E-02
CEMA6	-3.26E+00	-3.32E+00	-3.20E+00	5.99E-02
CEMA7	-3.25E+00	-3.32E+00	-3.14E+00	6.11E-02
CEMA8	-3.26E+00	-3.32E+00	-3.20E+00	6.00E-02
CEMA9	-3.25E+00	-3.32E+00	-3.20E+00	5.77E-02
CEMA10	-3.24E+00	-3.32E+00	-3.14E+00	6.33E-02

Table 4.20 represents the results of EMA and its chaotic variants on multimodal mathematical function FUN_XX. It is observed from Table 4.20 that all chaotic variants show better results on multimodal mathematical function FUN_XX in terms of STD, average fitness, best fitness and worst fitness.

Table 4.19: Analysis of Proposed methodology on FUN_XX function

FUN_XX				
Methods	**A Fitness**	**B Fitness**	**W Fitness**	**STD**
EMA	-5.60E+00	-1.02E+01	-2.63E+00	2.92E+00
CEMA1	-6.38E+00	-1.02E+01	-2.63E+00	3.01E+00
CEMA2	-6.23E+00	-1.02E+01	-2.63E+00	3.13E+00
CEMA3	-5.96E+00	-1.02E+01	-2.63E+00	2.66E+00
CEMA4	-6.39E+00	-1.02E+01	-2.63E+00	3.36E+00
CEMA5	-6.55E+00	-1.02E+01	-2.63E+00	3.23E+00
CEMA6	-5.79E+00	-1.02E+01	-2.63E+00	3.06E+00
CEMA7	-5.65E+00	-1.02E+01	-2.63E+00	3.14E+00
CEMA8	-6.55E+00	-1.02E+01	-2.63E+00	3.58E+00
CEMA9	-5.60E+00	-1.02E+01	-2.63E+00	3.41E+00
CEMA10	-5.73E+00	-1.02E+01	-2.63E+00	3.10E+00

Table 4.21 represents the results of EMA and its chaotic variants on multimodal mathematical function FUN_XXI. It is observed from Table 4.21 that CEMA1 shows better results on multimodal mathematical function FUN_XXI in terms of average fitness. While, all other algorithms show similar results on multimodal mathematical function FUN_XXI in terms of STD, best fitness and worst fitness.

Table 4.20: Analysis of Proposed methodology on FUN_XXI function

FUN_XXI				
Methods	**A Fitness**	**B Fitness**	**W Fitness**	**STD**
EMA	-6.53E+00	-1.04E+01	-2.75E+00	3.42E+00

CEMA1	-7.06E+00	-1.04E+01	-2.75E+00	3.33E+00
CEMA2	-6.20E+00	-1.04E+01	-2.75E+00	3.42E+00
CEMA3	-6.04E+00	-1.04E+01	-2.75E+00	3.12E+00
CEMA4	-6.87E+00	-1.04E+01	-2.75E+00	3.37E+00
CEMA5	-6.85E+00	-1.04E+01	-1.84E+00	3.39E+00
CEMA6	-6.86E+00	-1.04E+01	-1.84E+00	3.50E+00
CEMA7	-6.48E+00	-1.04E+01	-2.75E+00	3.21E+00
CEMA8	-6.74E+00	-1.04E+01	-1.84E+00	3.50E+00
CEMA9	-6.59E+00	-1.04E+01	-2.75E+00	3.36E+00
CEMA10	-5.90E+00	-1.04E+01	-1.84E+00	3.49E+00

Table 4.22 represents the results of EMA and its chaotic variants on multimodal mathematical function FUN_XXII. It is observed from Table 4.22 that CEMA7 shows better results on multimodal mathematical function FUN_XXII in terms of average fitness. While, all other algorithms show similar results on multimodal mathematical function FUN_XXII in terms of STD, best fitness and worst fitness.

Table 4.21: Analysis of Proposed methodology on FUN_XXII function

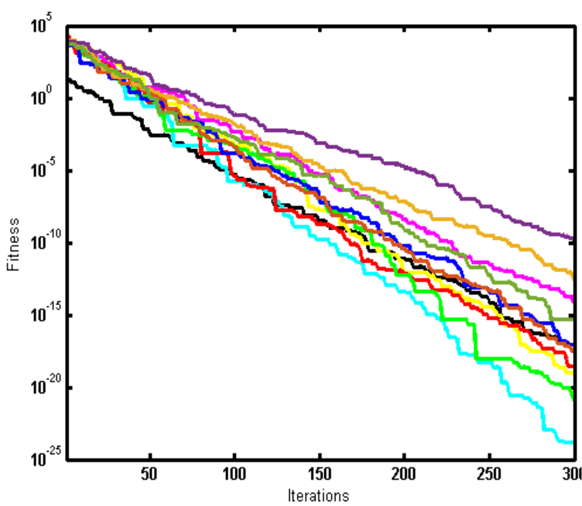
FUN_XXII				
Methods	**A Fitness**	**B Fitness**	**W Fitness**	**STD**
EMA	-6.90E+00	-1.05E+01	-2.42E+00	3.48E+00
CEMA1	-6.81E+00	-1.05E+01	-2.42E+00	3.43E+00
CEMA2	-6.80E+00	-1.05E+01	-2.42E+00	3.57E+00
CEMA3	-7.22E+00	-1.05E+01	-2.43E+00	3.43E+00
CEMA4	-7.06E+00	-1.05E+01	-2.42E+00	3.46E+00
CEMA5	-7.77E+00	-1.05E+01	-2.43E+00	3.48E+00
CEMA6	-6.68E+00	-1.05E+01	-2.42E+00	3.42E+00
CEMA7	-6.39E+00	-1.05E+01	-2.42E+00	3.25E+00
CEMA8	-7.26E+00	-1.05E+01	-2.43E+00	3.39E+00
CEMA9	-7.29E+00	-1.05E+01	-2.42E+00	3.62E+00

CEMA10	-6.80E+00	-1.05E+01	-2.42E+00	3.43E+00

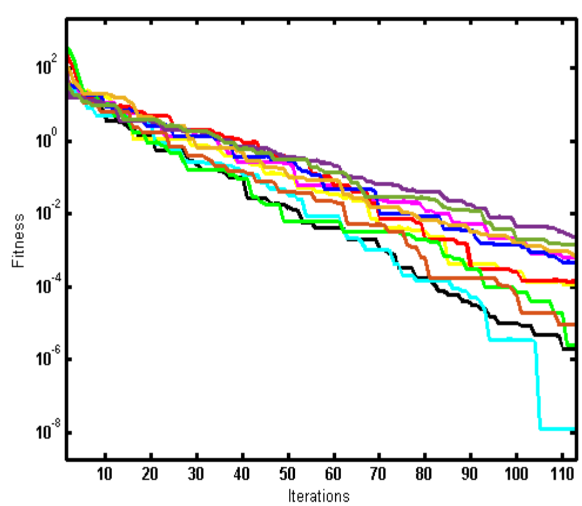
After detail analysis of EMA and its chaotic variants upon unimodal and multimodal mathematical functions, it can be seen from tables 4.1-4.22 that the proposed variants of EMA performs better than EMA for functions FUN_II, FUN_III, FUN_VII, FUN_IX, FUN_XI, FUN_XIV, FUN_XV, FUN_XVI, FUN_XVII, FUN_XIX, FUN_XX, FUN_XXI and FUN_XXII. EMA performs better in functions FUN_I, FUN_IV, FUN_V and FUN_VIII than other EMA variants while performance of EMA and its chaotic variants have similar performance in functions FUN_VI, FUN_X, FUN_XII, FUN_XIII and FUN_XVIII.

The convergence plots of EMA, CEMA1, CEMA2, CEMA3, CEMA4, CEMA5, CEMA6, CEMA7, CEMA8, CEMA9 and CEMA10 are presenting in figures 4.1(a-e) to 4.5(a-b).

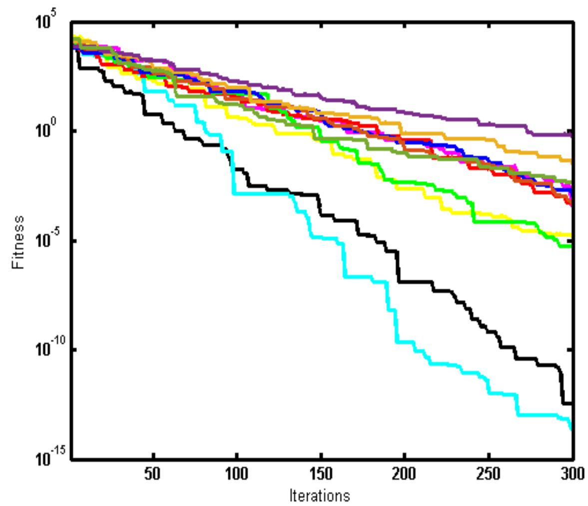
Figure 4.1(a-e) describes the convergence plots of FUN_I, FUN_II, FUN_III, FUN_IV and FUN_V functions. It is observed that CEMA3 shows better convergence than all other algorithms in FUN_I, FUN_II, FUN_III and FUN_IV mathematical functions. While CEMA4 shows better convergence in FUN_V mathematical function.



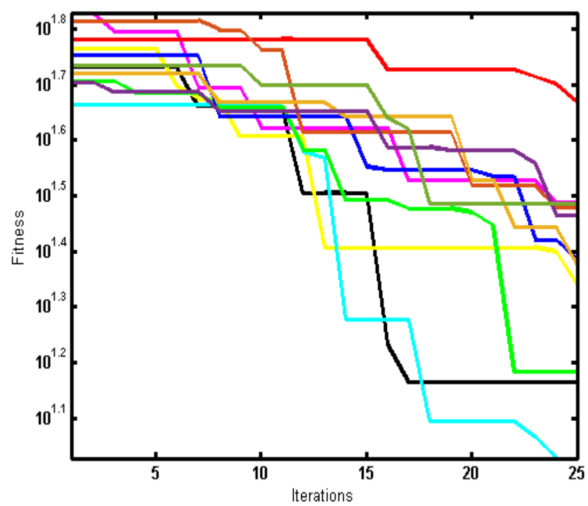
(a) FUN_I



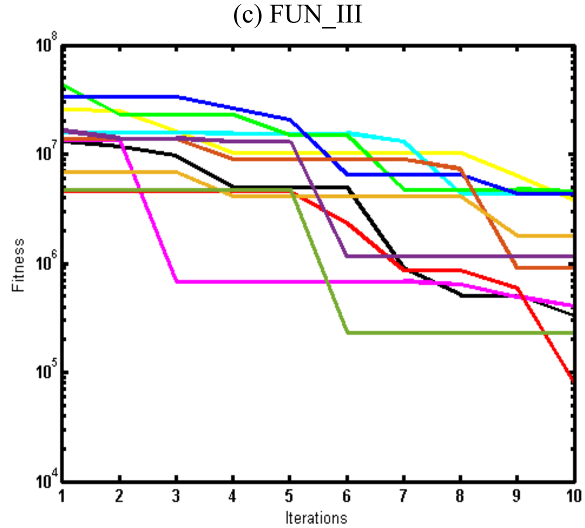
(b) FUN_II



(c) FUN_III



(d) FUN_IV



(e) FUN_V

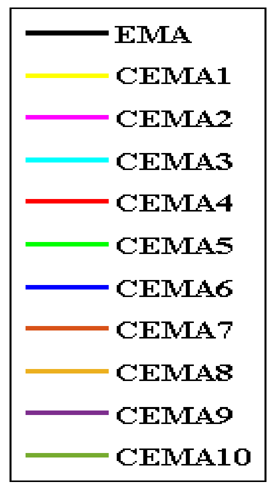


Figure 4.1: Analysis on FUN_I, FUN_II, FUN_III, FUN_IV and FUN_V

Figure 4.2(a-e) shows the convergence of FUN_VI, FUN_VII, FUN_VIII, FUN_IX and FUN_X functions. It is observed that CEMA5 shows better convergence than all other algorithms in FUN_VI mathematical function. CEMA7 shows better convergence than all other algorithms in FUN_VII and FUN_X mathematical functions. CEMA4 shows better convergence than all other algorithms in FUN_VIII mathematical function. While CEMA3 shows better convergence in FUN_IX mathematical function.

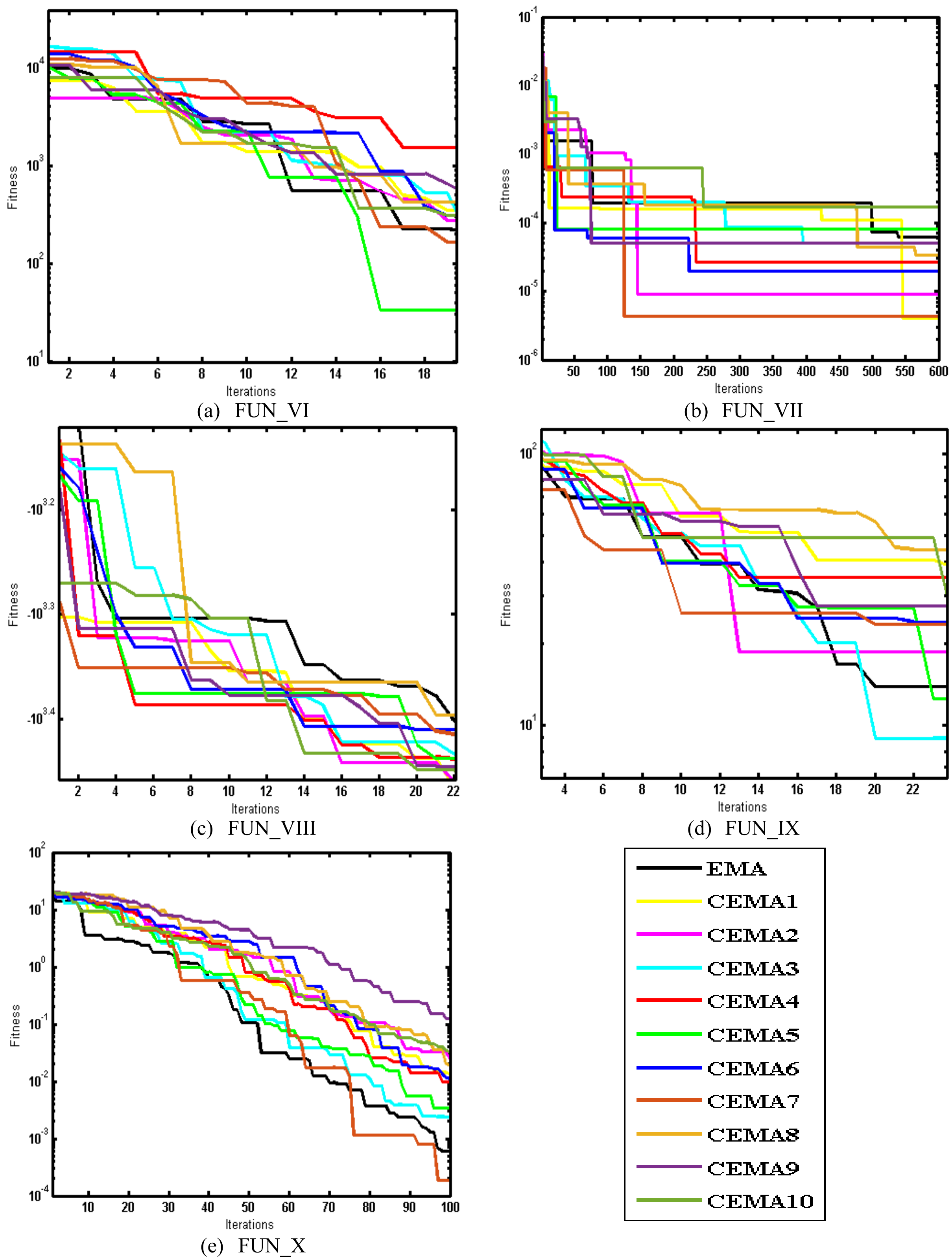
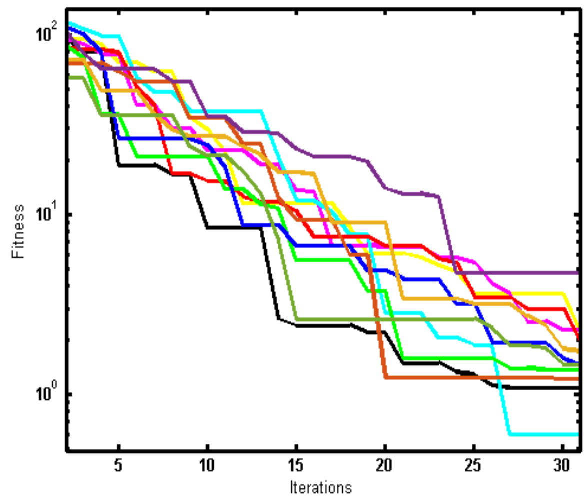
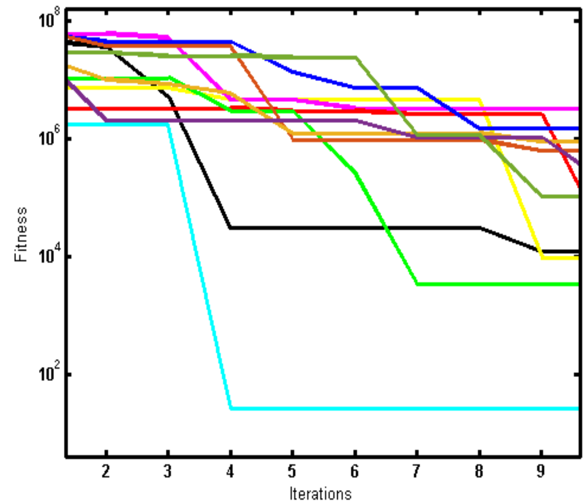


Figure 4.1: Analysis on FUN_VI, FUN_VII, FUN_VIII, FUN_IX and FUN_X

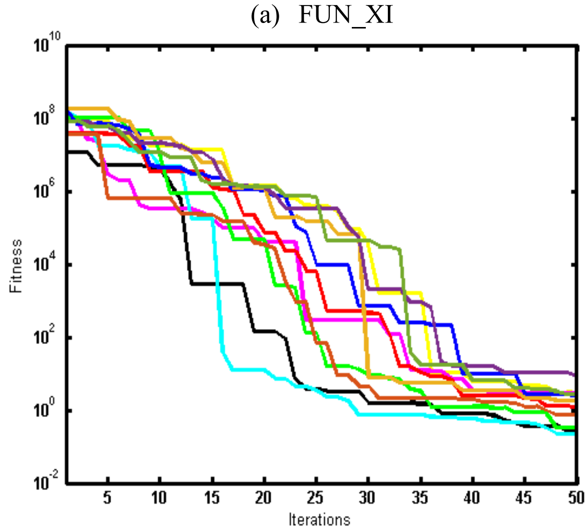
Figure 4.3 (a-e) shows the convergence of FUN_XI, FUN_XII, FUN_XIII, FUN_XIV and FUN_XV. It is observed that CEMA3 shows better convergence than all other algorithms in FUN_XI, FUN_XII and FUN_XIII mathematical functions. CEMA9 shows better convergence than all other algorithms in FUN_XIV mathematical function. While CEMA6 shows better convergence in FUN_XV mathematical function.



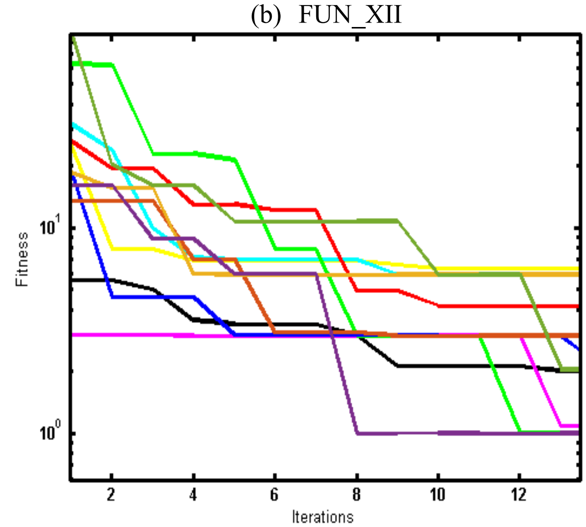
(a) FUN_XI



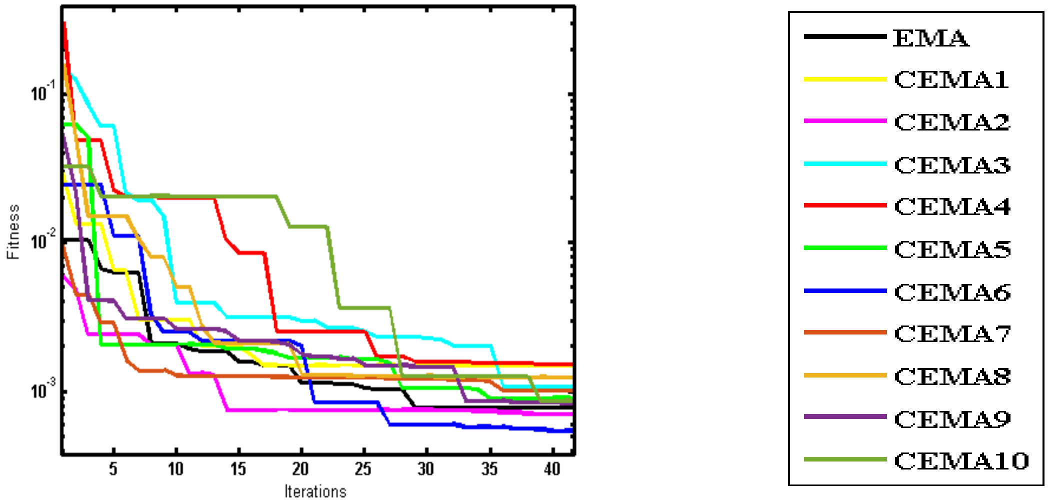
(b) FUN_XII



(c) FUN_XIII



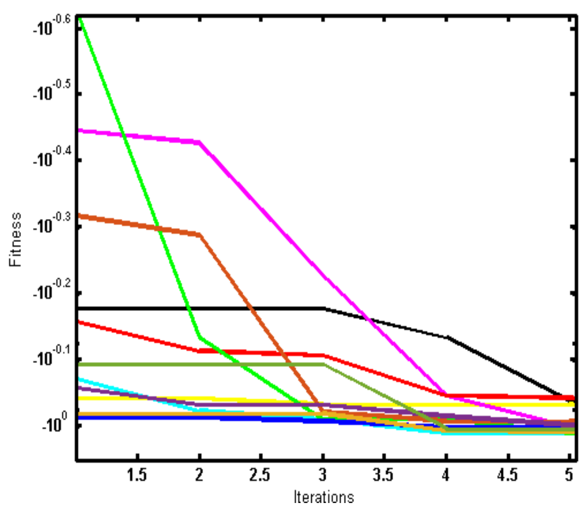
(d) FUN_XIV



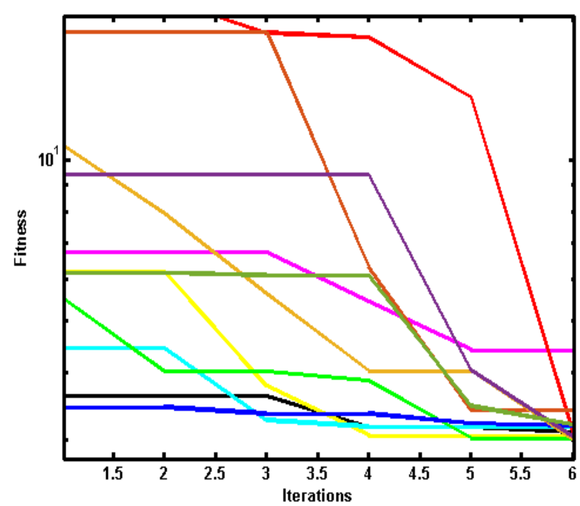
(e) FUN_XV

Figure 4.2: Analysis on FUN_XI, FUN_XII, FUN_XIII, FUN_XIV and FUN_XV

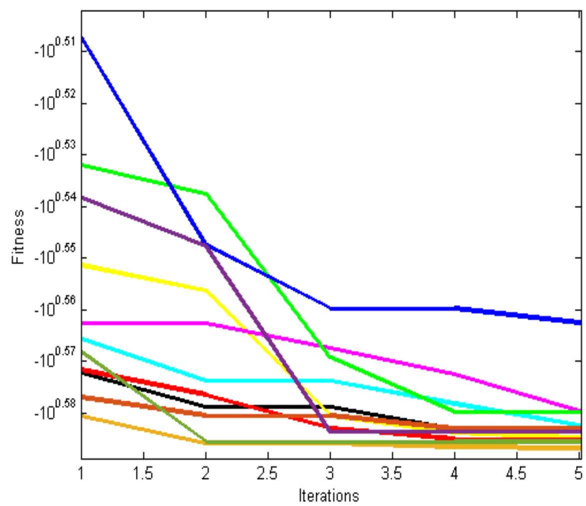
Figure 4.4 (a-e) shows the convergence of FUN_XVI, FUN_XVII, FUN_XVIII, FUN_XIX and FUN_XX. It is observed that CEMA5 shows better convergence than all other algorithms in FUN_XVI and FUN_XVII mathematical functions. CEMA7 shows better convergence than all other algorithms in FUN_XVIII mathematical function. CEMA6 shows better convergence than all other algorithms in FUN_XIX mathematical function. While CEMA8 shows better convergence in FUN_XX mathematical function.



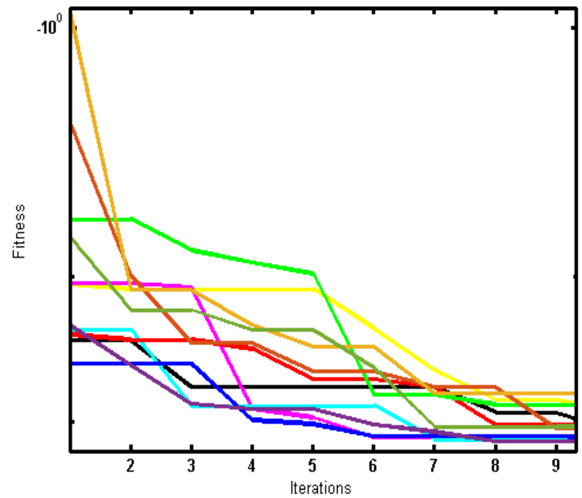
(a) FUN_XVI



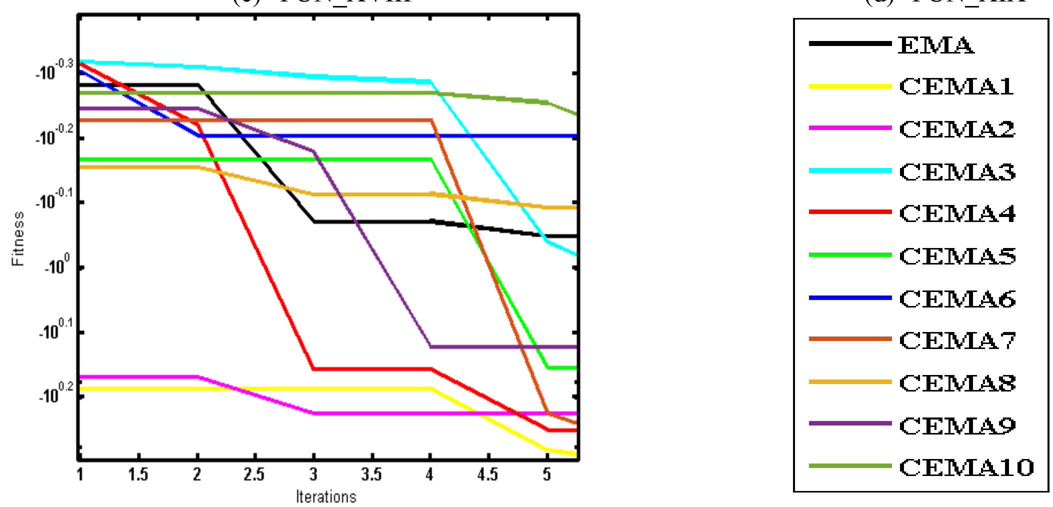
(b) FUN_XVII



(c) FUN_XVIII



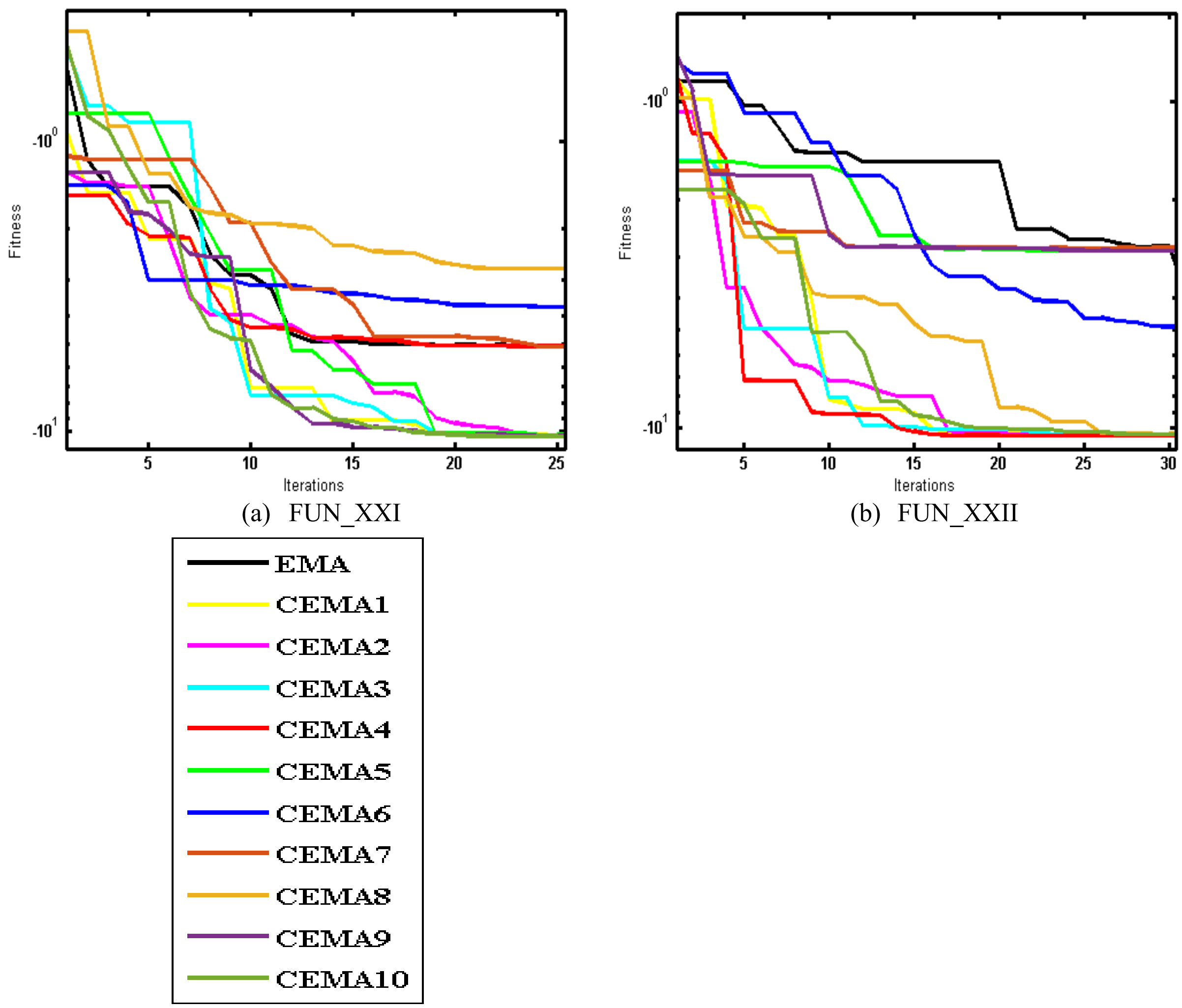
(d) FUN_XIX



(e) FUN_XX

Figure 4.3: Analysis on FUN_XVI, FUN_XVII, FUN_XVIII, FUN_XIX and FUN_XX

Figure 4.5 (a-b) demonstrates the convergence plots for FUN_XXI and FUN_XXII functions. It is observed that CEMA10 shows better convergence than all other algorithms in FUN_XXI mathematical function. While CEMA4 shows better convergence in FUN_XXII mathematical function.



(a) FUN_XXI (b) FUN_XXII

Figure 4.4: Analysis on FUN_XXI and FUN_XXII

After detail analysis of EMA and its chaotic variants upon unimodal and multimodal mathematical functions, it is seen from figures (4.1-4.5) that the chaotic variants of EMA show superior performance than EMA for mathematical functions in terms of convergence.

4.2 Parameter tuning of EMA on INOE Model

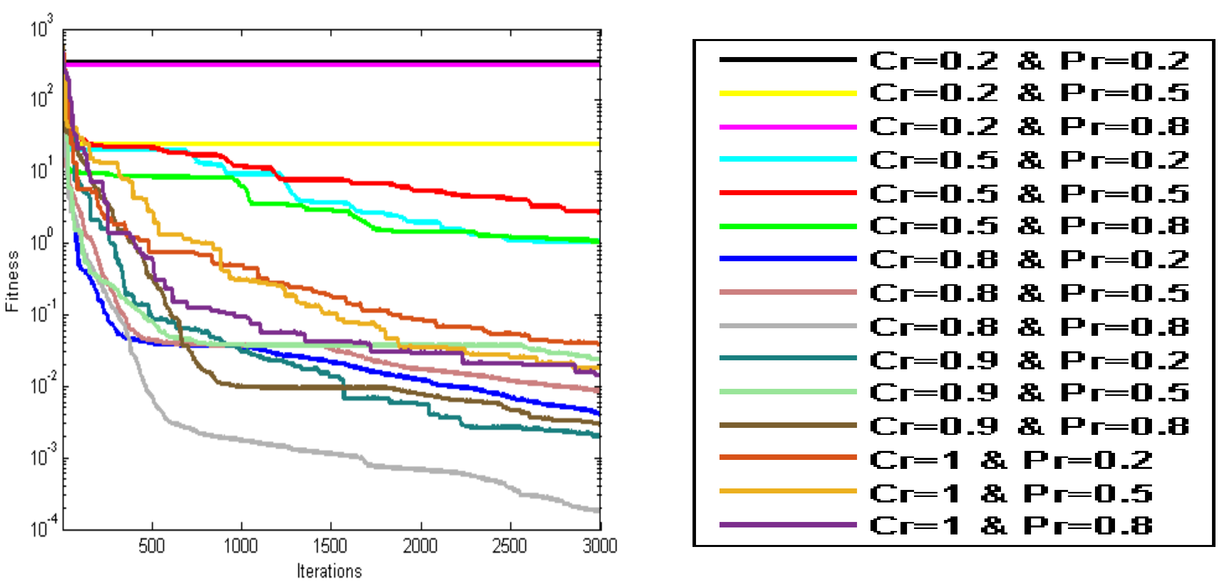
The tuning of Crossover probability and Predator probability parameters for EMA is being performed at population size (Pop) =50, iterations =3000 for independent 50 runs. It is seen form the Table 4.23 that best results are obtained when parameter values of both crossover probability and predator probability = 0.8.

Table 4.23: EMA Parameter tuning for IN-OE model

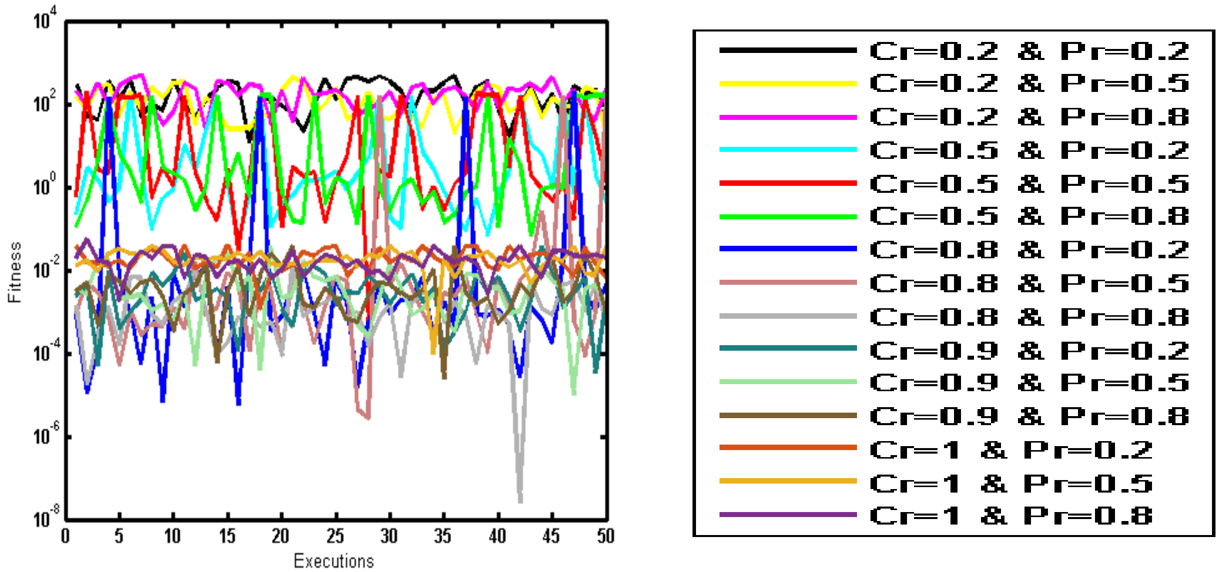
Parameter	**A Fitness**	**STD**

Crossover=0.2,predator=0.2	1.896E+02	4.761E+01
Crossover=0.2,predator=0.5	1.348E+02	1.527E+03
Crossover=0.2,predator=0.8	1.924E+02	2.430E+03
Crossover=0.5,predator=0.2	2.743E+01	7.021E+00
Crossover=0.5,predator=0.5	4.622E+01	8.639E+00
Crossover=0.5,predator=0.8	3.866E+01	1.048E+01
Crossover=0.8,predator=0.2	1.249E+01	8.313E+02
Crossover=0.8,predator=0.5	9.473E+00	2.216E+03
Crossover=0.8,predator=0.8	1.800E-03	4.436E+01
Crossover=0.9,predator=0.2	5.000E-03	5.180E+00
Crossover=0.9,predator=0.5	5.600E-03	6.709E+00
Crossover=0.9,predator=0.8	6.300E-03	5.572E+03
Crossover=1,predator=0.2	2.020E-02	1.747E+02
Crossover=1,predator=0.5	1.890E-02	4.854E+03
Crossover=1,predator=0.8	1.830E-02	9.329E+00

The convergence and statistical plots of tuned parameters of EMA on IN-OE model are presenting in figures 4.6(a-b). It is observed from figures that best results for EMA tuned parameters i.e. crossover probability and predator probability are 0.8 value for IN-OE model.



(a) Convergence Plot



(b) Statistical Plot

Figure 4.6: Convergence and Statistical plots of EMA Tuning

4.3 Analysis on IN-OE Model

The analysis of EMA and its chaotic variants is executed on mathematical functions having both unimodal and multimodal features. These variants are applied further for the identification of IN-OE model for multiple variations of iteration, noises and populations. The IN-OE model parameters are taken from [76] as given in (4.1)-(4.3)

$$C(q)=1+0.84q^{-1}+0.31q^{-2}, \quad (4.1)$$

$$D(q)=1-0.57q^{-1}+0.86q^{-2}, \quad (4.2)$$

$$\bar{w}(\tau)=-1.50w(\tau)-2.60w^{2}(\tau)+3.20w^{3}(\tau), \quad (4.3)$$

The parameter vector v is given in (4.4)

$$v=[c1,c2,d1,d2,\beta 1,\beta 2,\beta 3]^{T}=[0.84,0.31,-0.57,0.86,-1.50,-2.60,3.20]^{T} \quad (4.4)$$

$$Er(\tau)=y_{act}(\tau)-y_{est}(\tau), \quad (4.5)$$

Tables 4.24-4.29 represent the analysis for parameter vector estimated by EMA, CEMA1, CEMA2, CEMA3, CEMA4, CEMA5, CEMA6, CEMA7, CEMA8, CEMA9, CEMA10, COA, HHO and PSO at maximum iteration P=3000 and Pop =18, 50 and noise levels $E(\tau)$=1.91E-03, 1.91E-02 and 1.91E-01 respectively.

Table 4.24 represents the analysis for parameter vector estimated by EMA, CEMA1, CEMA2, CEMA3, CEMA4, CEMA5, CEMA6, CEMA7, CEMA8, CEMA9, CEMA10, COA, HHO and PSO at maximum iteration P=3000 and Pop =18 and noise level $E(\tau)$=1.91E-03. It can be seen that CEMA8 shows better results in terms of best fitness. While CEMA10 estimated weights are very close to true weights of the IN-OE model.

Table 4.24: Analysis of IN-OE at 1.91E-03 noise level and population size =18

Methods	**B Fitness**	**Estimated Weights**						
EMA	2.89E-04	0.837	0.308	-0.571	0.862	-1.676	-2.877	3.087
CEMA1	5.20E-04	0.842	0.311	-0.572	0.860	-1.230	-2.185	3.362
CEMA2	7.14E-05	0.838	0.309	-0.572	0.862	-1.514	-2.615	3.194
CEMA3	1.80E-04	0.840	0.310	-0.572	0.861	-1.350	-2.369	3.289
CEMA4	7.36E-04	0.840	0.310	-0.574	0.862	-1.197	-2.126	3.385
CEMA5	6.66E-05	0.839	0.309	-0.571	0.861	-1.568	-2.720	3.147
CEMA6	4.36E-05	0.839	0.309	-0.572	0.861	-1.515	-2.646	3.173
CEMA7	1.81E-05	0.839	0.309	-0.572	0.861	-1.482	-2.576	3.206
CEMA8	1.47E-05	0.839	0.309	-0.571	0.861	-1.513	-2.617	3.193
CEMA9	1.17E-04	0.839	0.309	-0.570	0.861	-1.614	-2.779	3.128
CEMA10	4.20E-05	0.840	0.310	-0.572	0.861	-1.435	-2.500	3.238
COA	2.90E+01	1.483	0.633	-0.963	-0.682	1.323	0.455	1.960
HHO	1.46E-02	0.838	0.310	-0.568	0.877	-0.998	-1.215	3.901
PSO	4.63E-02	0.834	0.305	-0.612	0.891	-1.156	-3.206	2.574
True Weights		0.840	0.310	-0.570	0.860	-1.500	-2.600	3.200

Table 4.25 represents the analysis for parameter vector estimated by EMA, CEMA1, CEMA2, CEMA3, CEMA4, CEMA5, CEMA6, CEMA7, CEMA8, CEMA9, CEMA10, COA, HHO and PSO at maximum iteration P=3000 and Pop =50 and noise level $E(\tau)$=1.91E-03. It can be seen that CEMA5 and CEMA10 show better results in terms of best fitness. It is also noted that CEMA5 and CEMA10 estimated weights are very close to true weights of the IN-OE model.

Table 4.25: Analysis of IN-OE at 1.91E-03 noise level and population size = 50

Methods	B Fitness	Weights						
EMA	1.94E-06	0.840	0.310	-0.570	0.860	-1.492	-2.586	3.206
CEMA1	2.11E-06	0.840	0.310	-0.570	0.860	-1.505	-2.603	3.199
CEMA2	1.84E-06	0.840	0.310	-0.570	0.860	-1.501	-2.598	3.201
CEMA3	2.24E-06	0.840	0.310	-0.570	0.860	-1.506	-2.605	3.199
CEMA4	6.70E-06	0.840	0.310	-0.570	0.860	-1.471	-2.555	3.217
CEMA5	1.76E-06	0.840	0.310	-0.570	0.860	-1.498	-2.594	3.203
CEMA6	1.80E-06	0.840	0.310	-0.570	0.860	-1.495	-2.589	3.205
CEMA7	2.40E-06	0.840	0.310	-0.570	0.860	-1.488	-2.580	3.208
CEMA8	2.05E-06	0.840	0.310	-0.570	0.860	-1.504	-2.603	3.200
CEMA9	1.92E-06	0.840	0.310	-0.570	0.860	-1.492	-2.586	3.206
CEMA10	1.76E-06	0.840	0.310	-0.570	0.860	-1.498	-2.594	3.203
COA	1.12E+01	0.840	0.235	-1.403	1.001	1.288	-1.455	1.036
HHO	4.77E-02	0.826	0.300	-0.629	0.922	-0.016	-0.272	4.000
PSO	4.15E-01	0.717	0.236	-0.693	1.065	-2.820	-2.295	3.744
True Weights		0.840	0.310	-0.570	0.860	-1.500	-2.600	3.200

Table 4.26 represents the analysis for parameter vector estimated by EMA, CEMA1, CEMA2, CEMA3, CEMA4, CEMA5, CEMA6, CEMA7, CEMA8, CEMA9, CEMA10, COA, HHO and PSO at maximum iteration P=3000 and Pop =18 and noise level $E(\tau)$=1.91E-02. It can be seen that CEMA3 shows better results in terms of best fitness. It is also noted that CEMA3 estimated weights are very close to true weights of the IN-OE model.

Table 4.26: Analysis of IN-OE at 1.91E-02 noise level and population size =18

Methods	Best FF	Estimated Weights						
EMA	2.58E-04	0.839	0.310	-0.571	0.862	-1.575	-2.690	3.170
CEMA1	3.41E-04	0.841	0.311	-0.572	0.862	-1.327	-2.301	3.324
CEMA2	2.61E-04	0.838	0.308	-0.575	0.864	-1.459	-2.514	3.235
CEMA3	1.87E-04	0.839	0.310	-0.572	0.862	-1.508	-2.585	3.211

CEMA4	3.33E-04	0.838	0.308	-0.576	0.865	-1.389	-2.409	3.275
CEMA5	4.85E-04	0.838	0.309	-0.571	0.863	-1.676	-2.846	3.108
CEMA6	5.65E-04	0.835	0.307	-0.578	0.867	-1.451	-2.507	3.234
CEMA7	2.84E-04	0.837	0.308	-0.574	0.864	-1.543	-2.650	3.180
CEMA8	2.66E-04	0.840	0.310	-0.573	0.862	-1.362	-2.361	3.299
CEMA9	2.38E-04	0.840	0.310	-0.573	0.863	-1.385	-2.396	3.284
CEMA10	3.13E-04	0.838	0.309	-0.572	0.863	-1.602	-2.733	3.152
COA	1.56E+01	0.712	0.348	-2.275	2.996	2.652	0.174	0.979
HHO	1.28E-01	0.824	0.308	-0.651	0.955	-1.308	-3.221	2.522
PSO	1.54E+00	0.704	0.233	-0.776	1.378	-2.787	-2.145	3.293
True Weights		0.840	0.310	-0.570	0.860	-1.500	-2.600	3.200

Table 4.27 represents the analysis for parameter vector estimated by EMA, CEMA1, CEMA2, CEMA3, CEMA4, CEMA5, CEMA6, CEMA7, CEMA8, CEMA9, CEMA10, COA, HHO and PSO at maximum iteration P=3000 and Pop =50 and noise level $E(\tau)$=1.91E-02. It can be seen that CEMA7 shows better results in terms of best fitness. It is also noted that CEMA7 estimated weights are very close to true weights of the IN-OE model.

Table 4.27: Analysis of IN-OE at 1.91E-02 noise level and population size =50

Methods	**Best FF**	**Estimated Weights**						
EMA	1.76E-04	0.840	0.310	-0.572	0.862	-1.466	-2.521	3.236
CEMA1	1.80E-04	0.840	0.310	-0.572	0.862	-1.448	-2.494	3.247
CEMA2	1.78E-04	0.839	0.310	-0.572	0.862	-1.490	-2.559	3.221
CEMA3	1.76E-04	0.840	0.310	-0.572	0.862	-1.465	-2.519	3.237
CEMA4	1.76E-04	0.840	0.310	-0.572	0.862	-1.468	-2.524	3.235
CEMA5	1.76E-04	0.840	0.310	-0.572	0.862	-1.464	-2.518	3.237
CEMA6	1.77E-04	0.840	0.310	-0.572	0.862	-1.460	-2.512	3.240
CEMA7	1.75E-04	0.840	0.310	-0.572	0.862	-1.472	-2.531	3.232
CEMA8	1.76E-04	0.840	0.310	-0.572	0.862	-1.467	-2.523	3.236
CEMA9	1.76E-04	0.840	0.310	-0.572	0.862	-1.477	-2.538	3.229

CEMA10	1.77E-04	0.840	0.310	-0.572	0.862	-1.456	-2.508	3.241
COA	4.31E+00	0.421	0.044	-1.568	1.400	1.937	-1.274	1.985
HHO	4.79E-02	0.813	0.288	-0.650	0.909	0.132	-0.578	3.775
PSO	2.63E-01	0.888	0.338	-0.648	0.892	1.217	-0.509	3.190
True Weights		0.840	0.310	-0.570	0.860	-1.500	-2.600	3.200

Table 4.28 represents the analysis for parameter vector estimated by EMA, CEMA1, CEMA2, CEMA3, CEMA4, CEMA5, CEMA6, CEMA7, CEMA8, CEMA9, CEMA10, COA, HHO and PSO at maximum iteration P=3000 and Pop =18 and noise level $E(\tau)$=1.91E-01. It can be seen that CEMA7 shows better results in terms of best fitness. While, CEMA4 estimated weights are very close to true weights of the IN-OE model.

Table 4.28: Analysis of INOE at 1.91E-01 noise level and population size =18

Methods	**Best FF**	**Estimated Weights**						
EMA	1.75E-02	0.836	0.309	-0.588	0.880	-1.265	-1.971	3.499
CEMA1	1.80E-02	0.832	0.307	-0.591	0.884	-1.442	-2.243	3.389
CEMA2	1.78E-02	0.837	0.310	-0.590	0.880	-1.031	-1.612	3.638
CEMA3	1.76E-02	0.835	0.309	-0.591	0.881	-1.159	-1.815	3.556
CEMA4	1.78E-02	0.837	0.310	-0.590	0.880	-1.027	-1.608	3.641
CEMA5	1.76E-02	0.835	0.309	-0.588	0.880	-1.313	-2.045	3.469
CEMA6	1.76E-02	0.836	0.309	-0.590	0.881	-1.172	-1.833	3.550
CEMA7	1.75E-02	0.835	0.309	-0.590	0.881	-1.230	-1.931	3.510
CEMA8	1.76E-02	0.835	0.308	-0.592	0.882	-1.157	-1.824	3.548
CEMA9	1.79E-02	0.834	0.308	-0.589	0.881	-1.430	-2.232	3.394
CEMA10	1.76E-02	0.835	0.309	-0.588	0.880	-1.316	-2.055	3.465
COA	6.46E+00	0.819	0.272	-1.227	1.693	-0.718	0.478	3.076
HHO	2.14E-01	0.788	0.274	-0.755	1.003	1.466	0.781	3.963
PSO	6.55E-01	0.879	0.357	-0.619	1.069	-1.454	-1.850	3.223
True Weights		0.840	0.310	-0.570	0.860	-1.500	-2.600	3.200

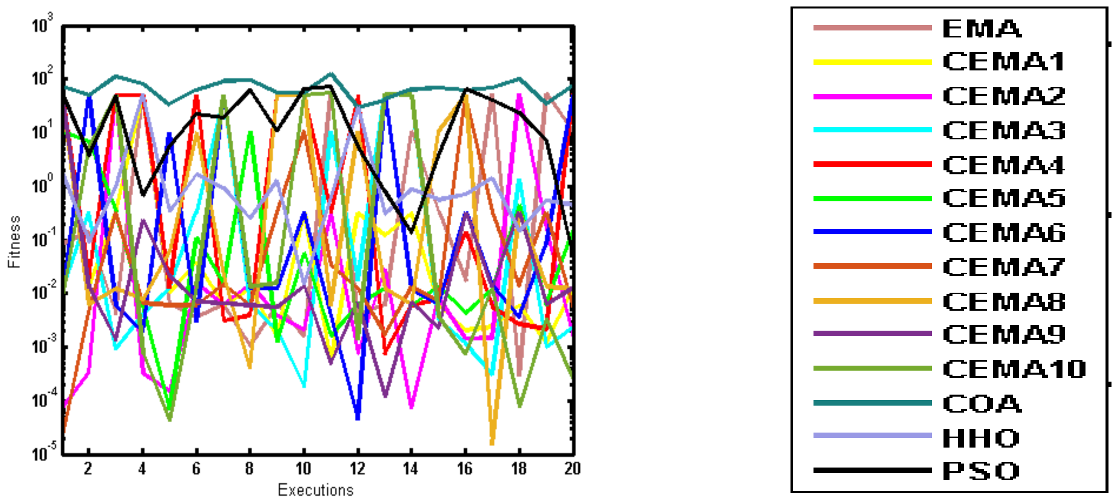
Table 4.29 represents the analysis for parameter vector estimated by EMA, CEMA1, CEMA2, CEMA3, CEMA4, CEMA5, CEMA6, CEMA7, CEMA8, CEMA9, CEMA10, COA, HHO and PSO at maximum iteration P=3000 and Pop =50 and noise level $E(\tau)$=1.91E-01. It can be seen that EMA and its chaotic variants show better results in terms of best fitness. While, CEMA5 estimated weights are very close to true weights of the IN-OE model.

Table 4.29: Analysis of INOE at 1.91E-01 noise level and population size =50

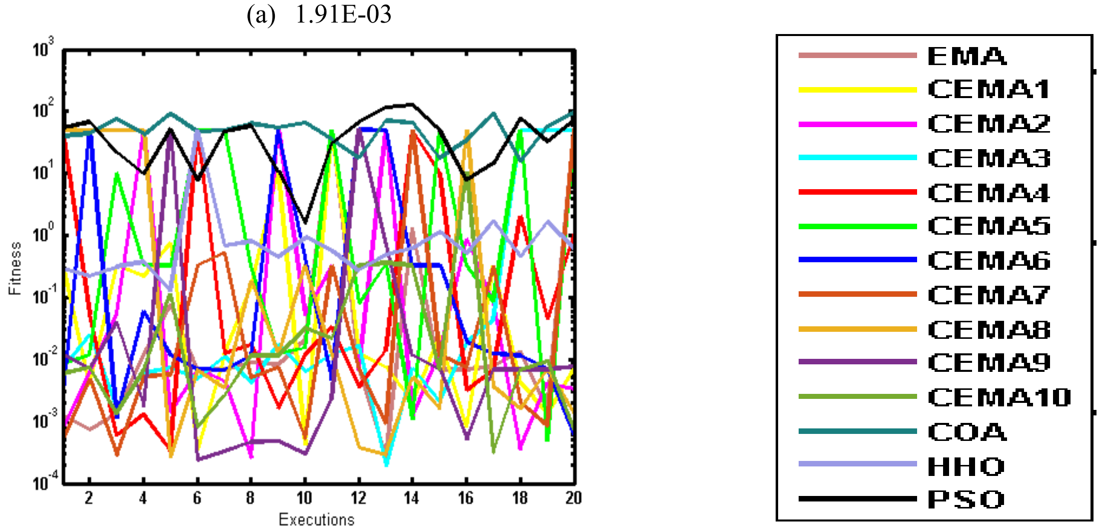
Methods	**Best FF**	**Estimated Weights**						
EMA	1.75E-02	0.836	0.309	-0.589	0.880	-1.220	-1.906	3.523
CEMA1	1.75E-02	0.836	0.309	-0.589	0.880	-1.219	-1.905	3.523
CEMA2	1.75E-02	0.836	0.309	-0.589	0.880	-1.217	-1.902	3.524
CEMA3	1.75E-02	0.836	0.309	-0.589	0.880	-1.227	-1.916	3.519
CEMA4	1.75E-02	0.836	0.309	-0.589	0.880	-1.232	-1.924	3.516
CEMA5	1.75E-02	0.836	0.309	-0.589	0.880	-1.239	-1.934	3.513
CEMA6	1.75E-02	0.836	0.309	-0.589	0.880	-1.191	-1.861	3.541
CEMA7	1.75E-02	0.836	0.309	-0.589	0.880	-1.225	-1.914	3.520
CEMA8	1.75E-02	0.836	0.309	-0.589	0.880	-1.238	-1.934	3.512
CEMA9	1.75E-02	0.836	0.309	-0.589	0.880	-1.226	-1.915	3.519
CEMA10	1.75E-02	0.836	0.309	-0.589	0.880	-1.205	-1.884	3.531
COA	1.17E+01	0.861	0.288	-1.306	1.915	0.798	-0.076	1.580
HHO	1.16E-01	0.808	0.288	-0.677	0.922	-0.996	-3.290	2.396
PSO	5.85E-01	0.876	0.321	-0.695	0.793	2.382	-1.687	2.143
True Weights		0.840	0.310	-0.570	0.860	-1.500	-2.600	3.200

After detailed analysis of EMA, its chaotic variants and other metaheuristic algorithms on IN-OE model with different noise levels and population size, it is seen from Tables 4.24-4.29 that chaotic variants of EMA achieves the lowest best fitness (B Fitness) and most accurate parameters for all variants than EMA, COA, HHO and PSO.

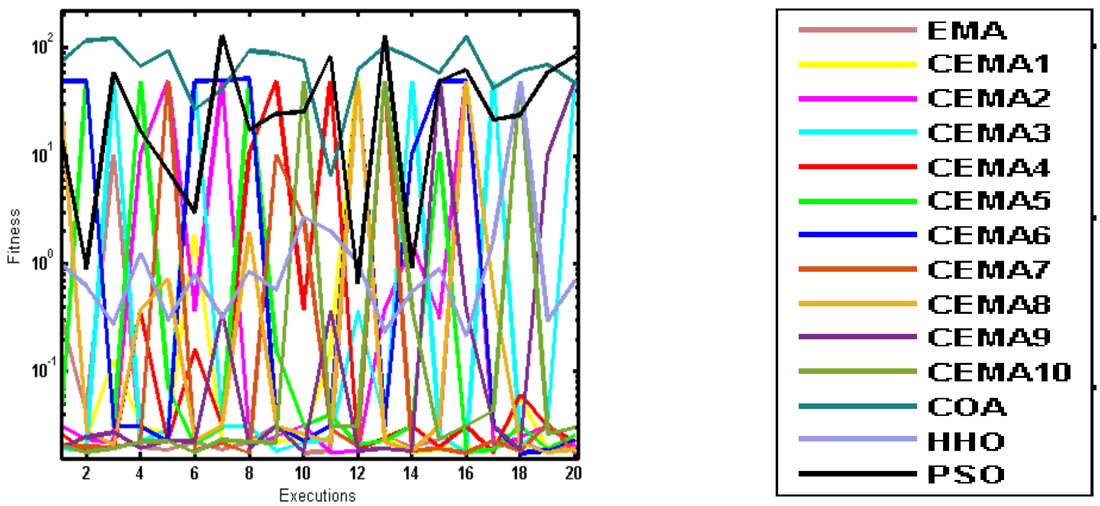
The statistical analysis of EMA, its chaotic variants and other metaheuristic algorithms are performed for the identification of IN-OE model parameters. Figure 4.7(a-c) and Figure 4.8(a-c) present the statistical parameter vector analysis of the IN-OE model for EMA, CEMA1, CEMA2, CEMA3, CEMA4, CEMA5, CEMA6, CEMA7, CEMA8, CEMA9, CEMA10, COA, HHO and PSO at maximum iteration P =3000, Pop =18, 50 and noise levels $E(\tau)$=1.91E-03, 1.91E-02 and 1.91E-01 respectively. It is seen from Figure 4.7(a-c) and Figure 4.8(a-c) that chaotic variants of EMA estimate parameter more accurately than EMA, COA, HHO and PSO for all twenty executions.



(a) 1.91E-03

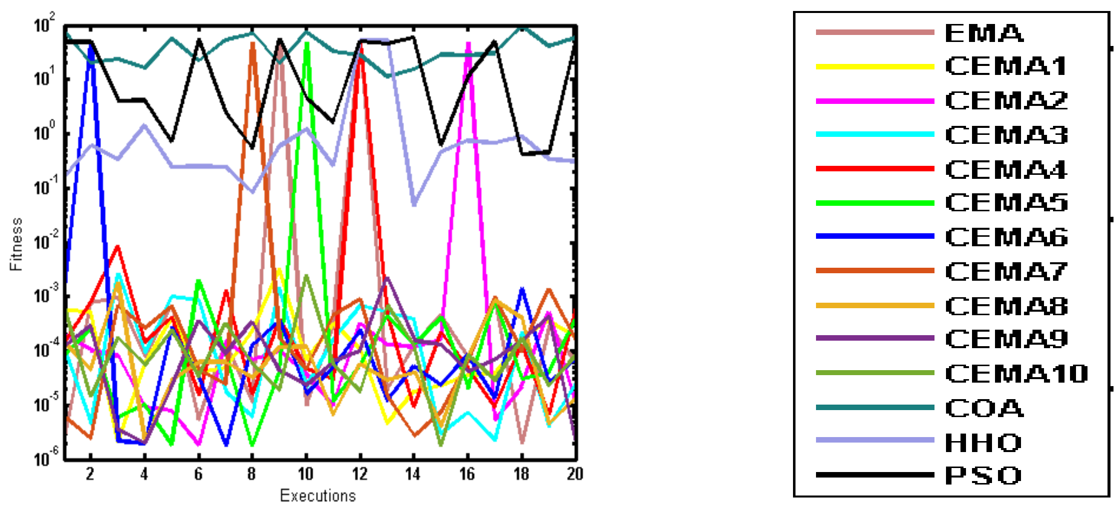


(b) 1.91E-02

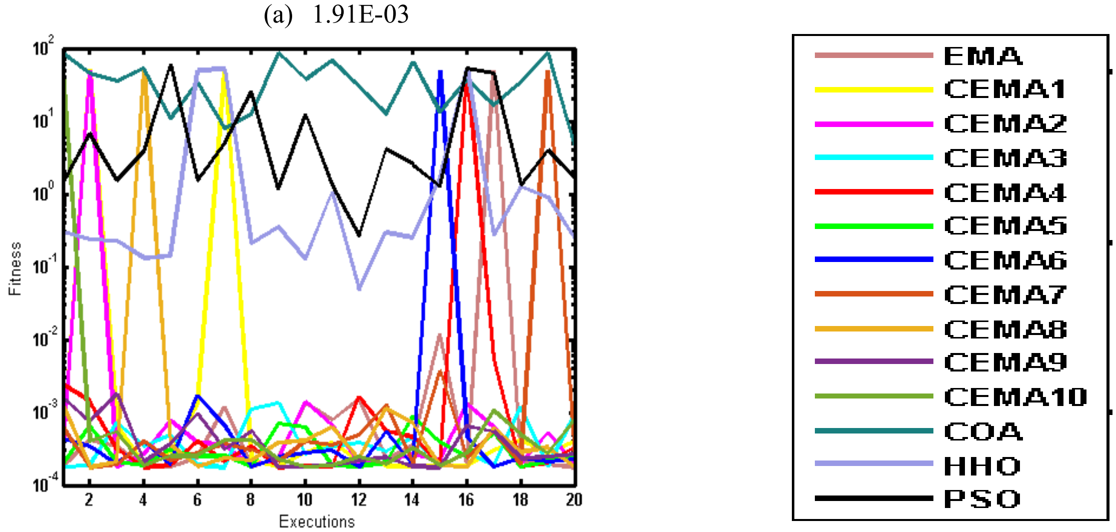


(c) 1.91E-01

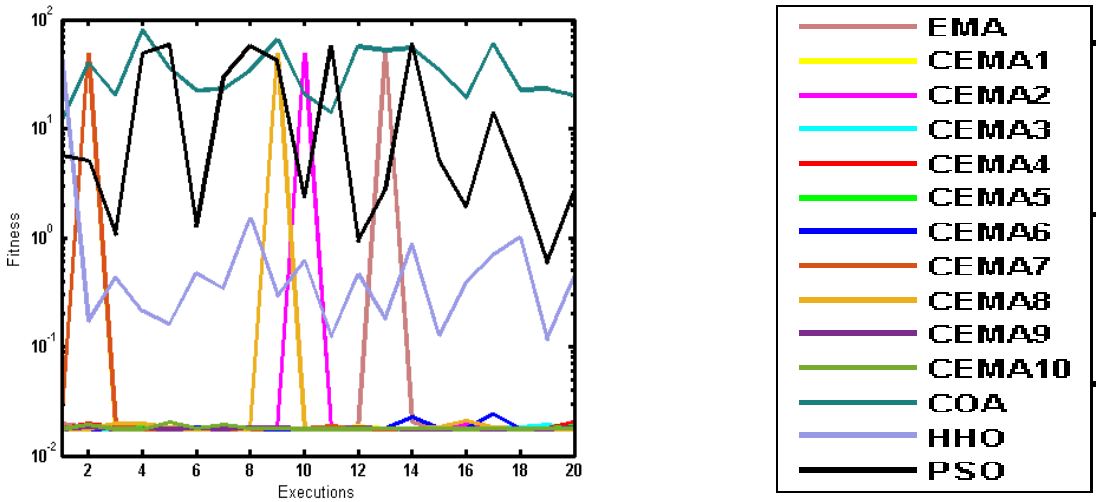
Figure 4.7: Statistical analysis of EMA and other metaheuristic algorithms at Pop=18



(a) 1.91E-03



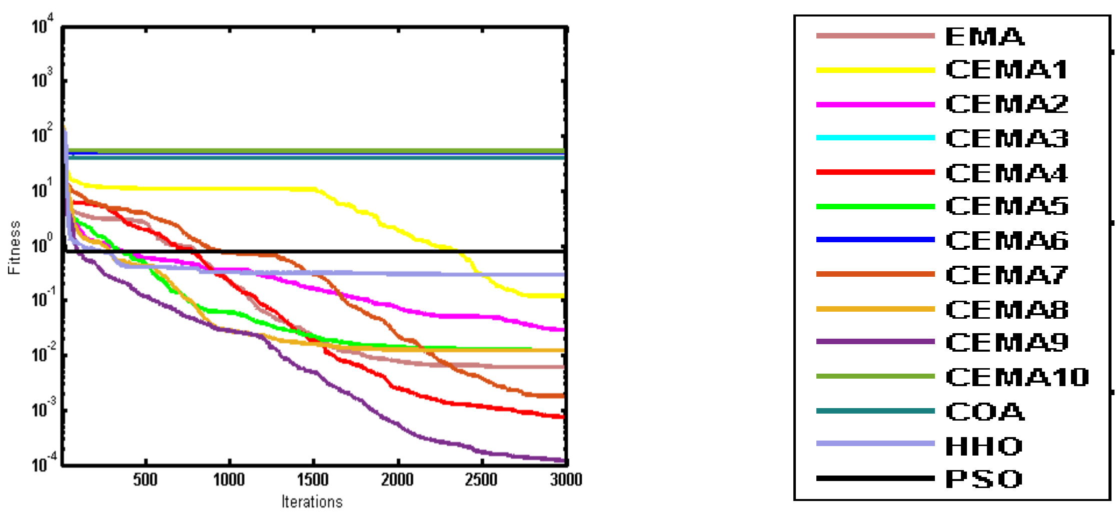
(b) 1.91E-02



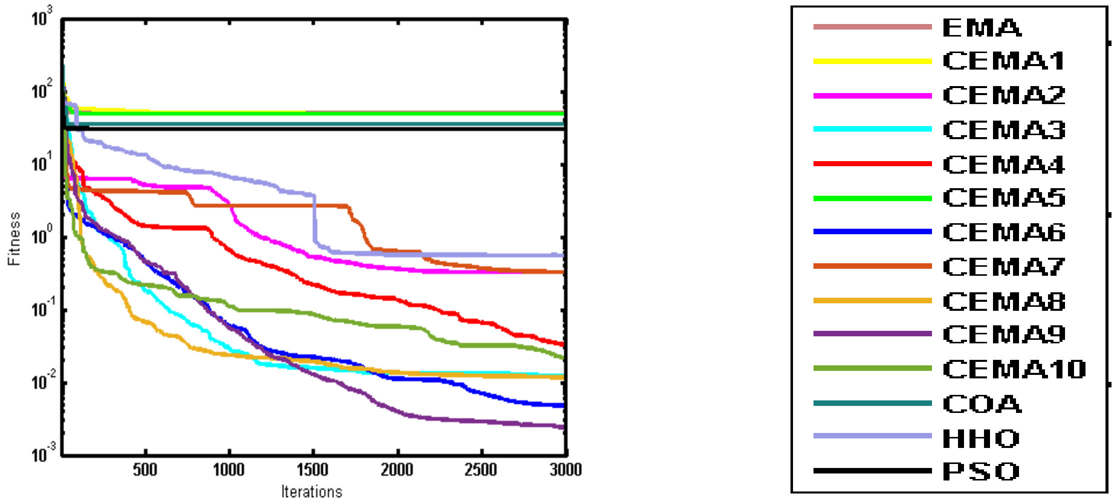
(c) 1.91E-01

Figure 4.8: Statistical analysis of EMA and other metaheuristic algorithms at Pop=50

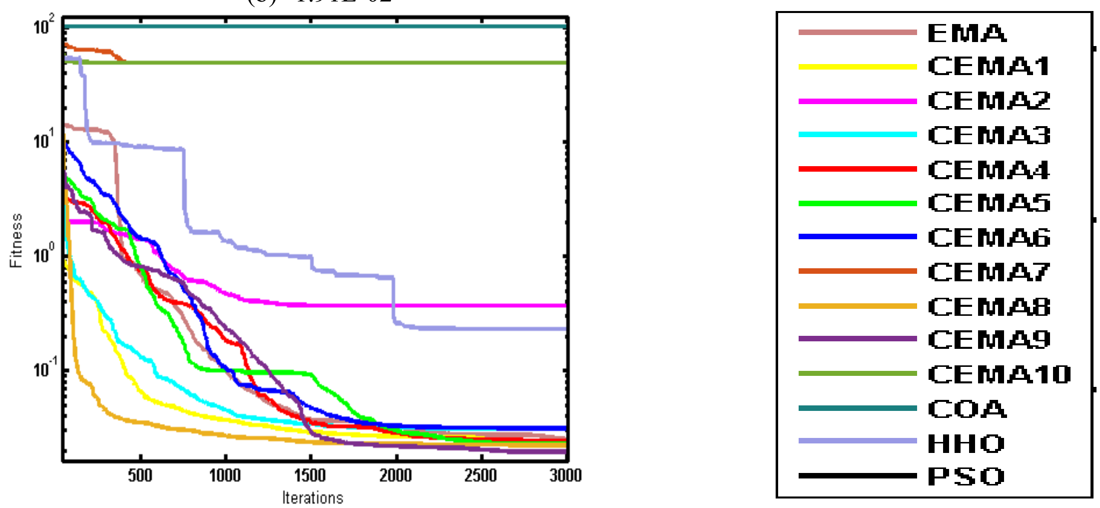
The convergence analysis of EMA, its chaotic variants and other metaheuristic algorithms are performed for the identification of IN-OE model parameters. Figure 4.9(a-c) and Figure 4.10(a-c) represent the convergence analysis of the IN-OE model for EMA, CEMA1, CEMA2, CEMA3, CEMA4, CEMA5, CEMA6, CEMA7, CEMA8, CEMA9, CEMA10, COA, HHO and PSO for three levels of $E(\tau)$ respectively. It is observed from Figure 4.9(a-c) and Figure 4.10(a-c) that higher level of $E(\tau)$ affects the fitness. Moreover it is observed from Figure 4.9(a-c) and Figure 4.10(a-c) that CEMA9 performs better than EMA, CEMA1, CEMA2, CEMA3, CEMA4, CEMA5, CEMA6, CEMA7, CEMA8, CEMA10, COA, HHO and PSO.



(a) 1.91E-03

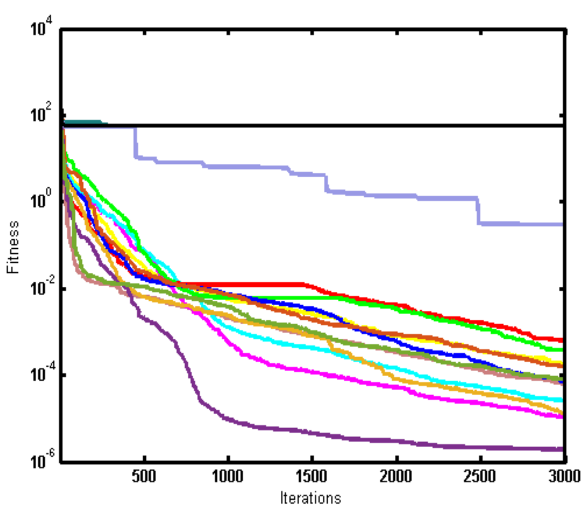


(b) 1.91E-02

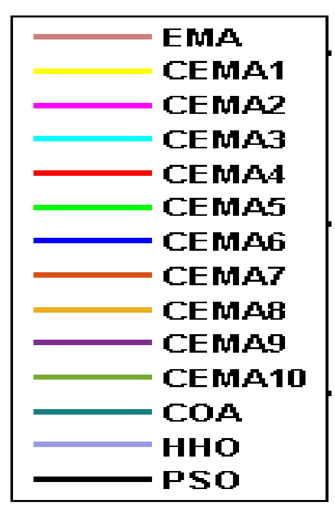


(c) 1.91E-01

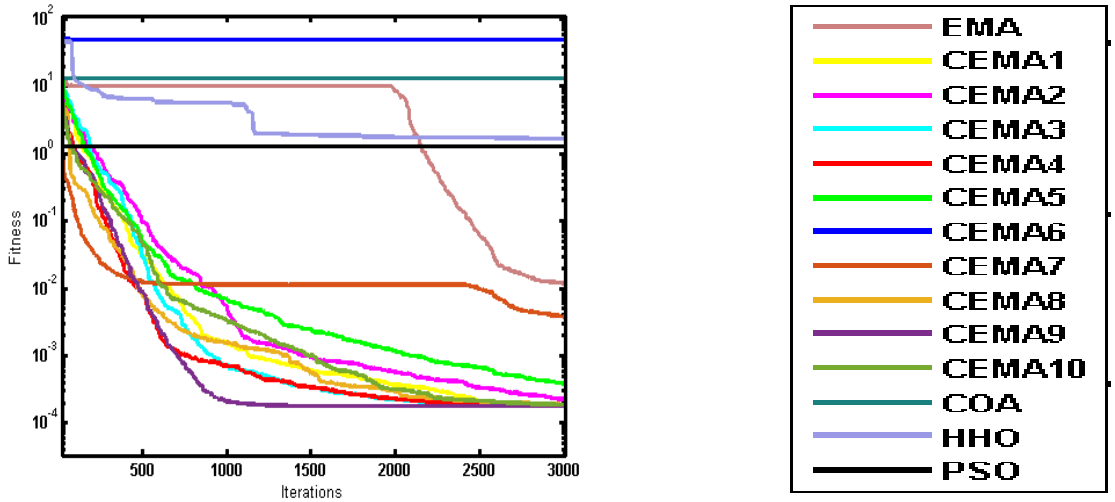
Figure 4.9: Convergence analysis of EMA and other metaheuristic algorithms at Pop=18



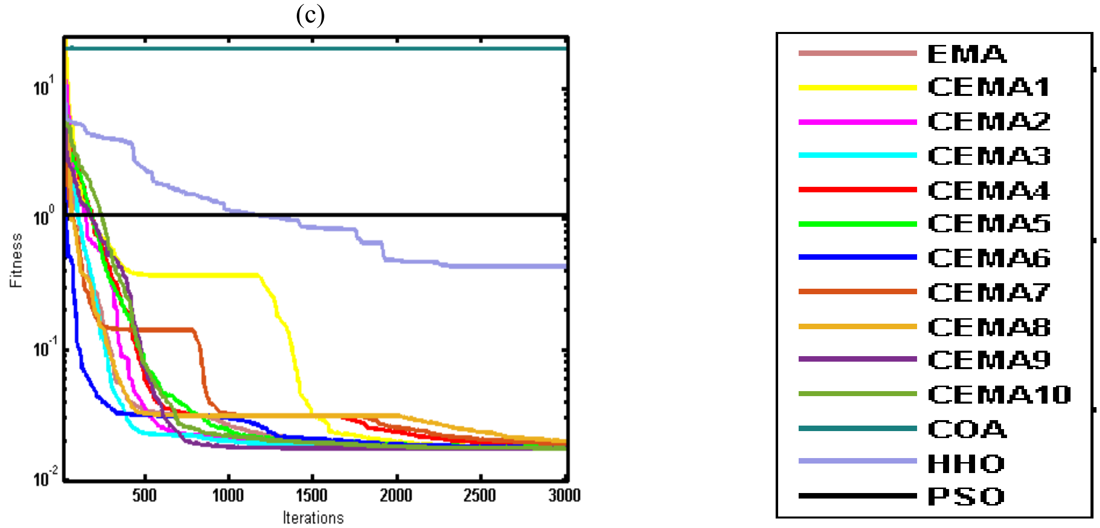
(a) 1.91E-03



(b) 1.91E-02



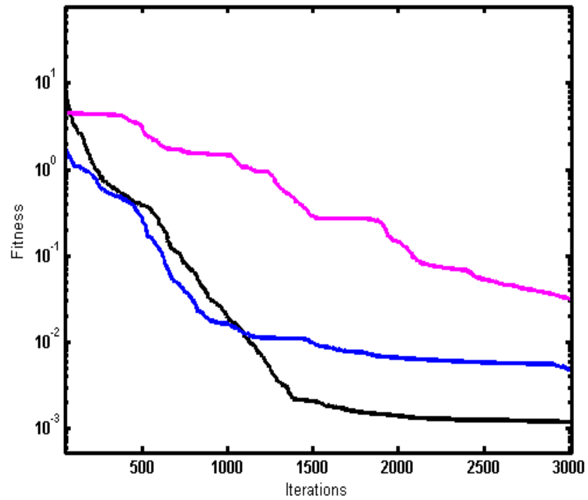
(c)



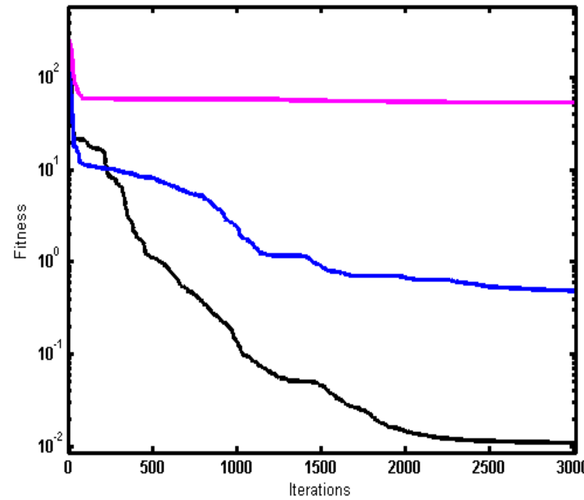
(d) 1.91E-01

Figure 4.10: Convergence analysis of EMA and other metaheuristic algorithms at Pop=50

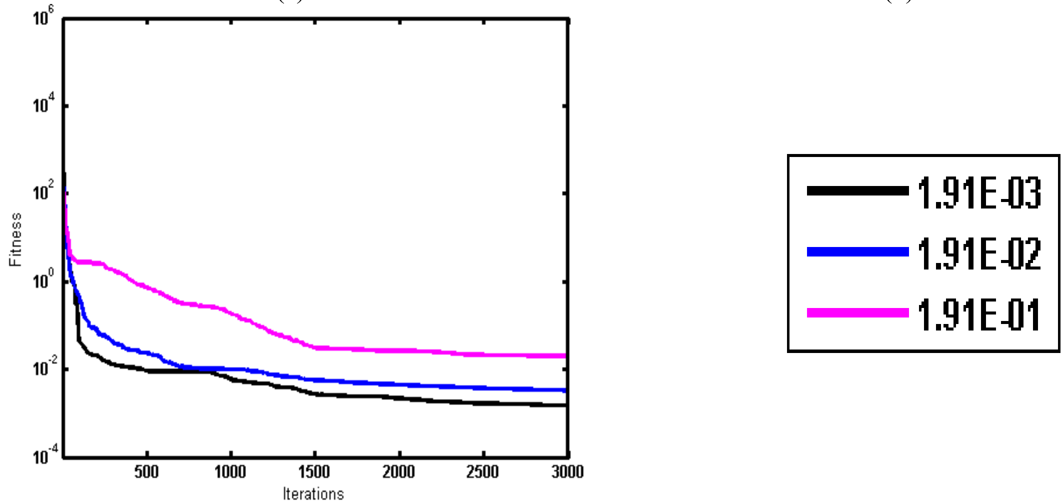
The performance of EMA, CEMA1, CEMA2, CEMA3, CEMA4, CEMA5, CEMA6, CEMA7, CEMA8, CEMA9, CEMA10, COA, HHO and PSO for the identification of IN-OE model at maximum iteration P =3000 and Pop =18, 50 respectively are presented in terms of different noise levels $E(\tau)$. Figures 4.11(a-c) to 4.14(a-c), 4.15(a-b), 4.16(a-c) to 4.19(a-c) and 4.20(a-b). It is perceived from the above mentioned figures that for all OM's the fitness increases with an increase in $E(\tau)$. However, chaotic variants of EMA achieve lowest fitness than EMA, COA, HHO and PSO for all variations.



(a) EMA

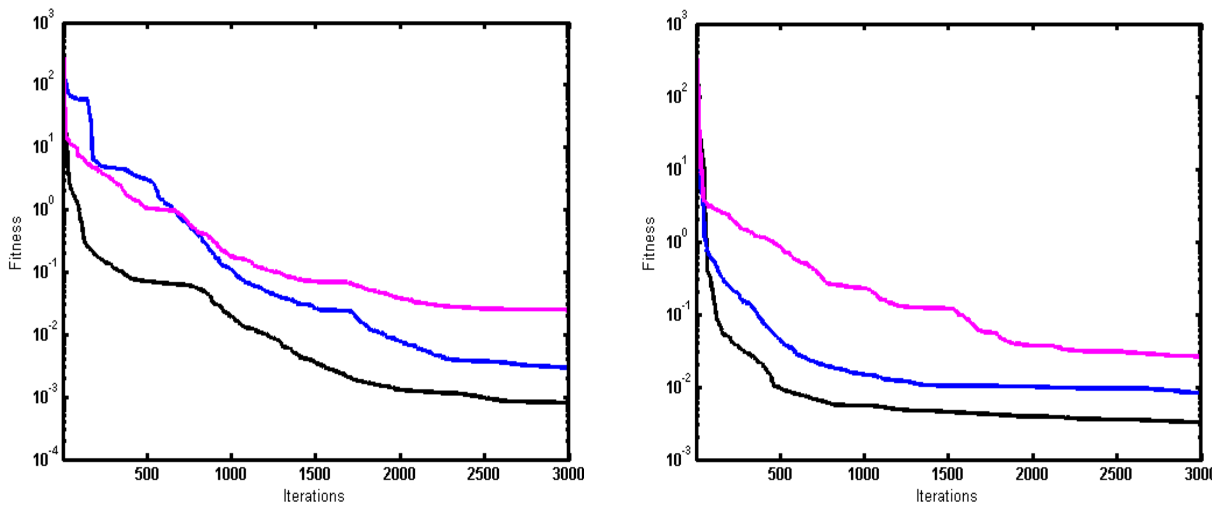


(b) CEMA1

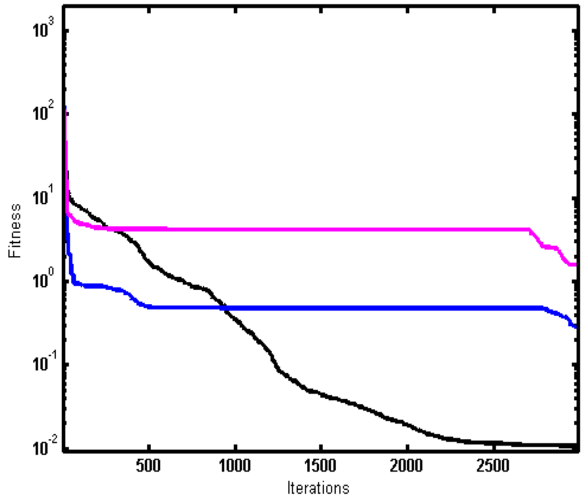


(c) CEMA2

Figure 4.11: Analysis of EMA, CEMA1 and CEMA2 for Pop =18



(a) CEMA3 (b) CEMA4



(c) CEMA5

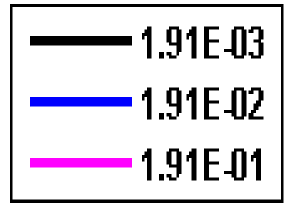
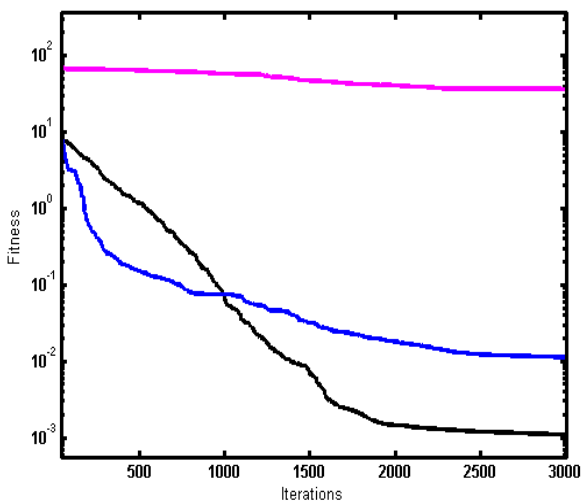
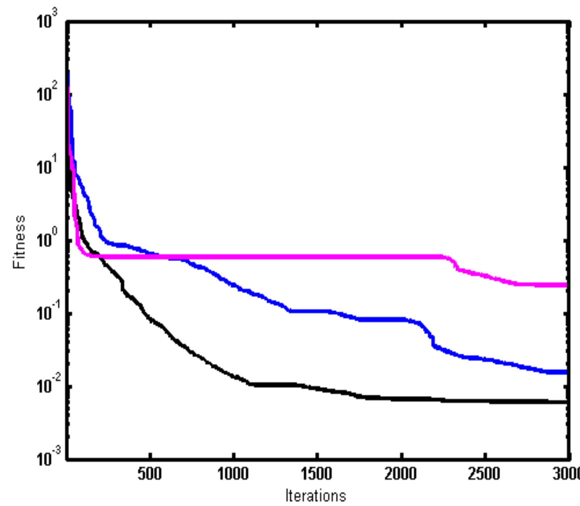


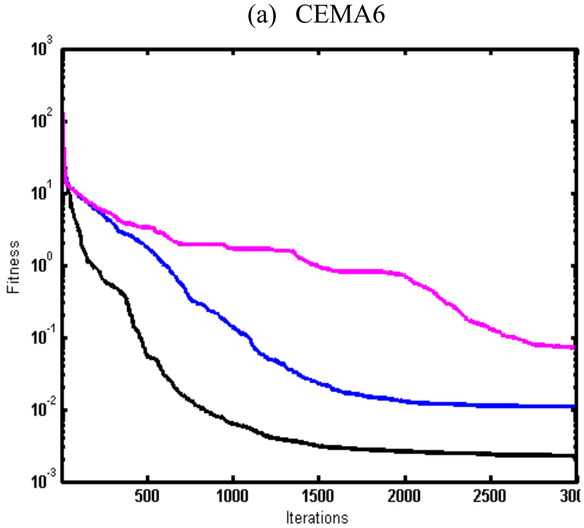
Figure 4.12: Analysis of CEMA3, CEMA4 and CEMA5 for Pop =18



(a) CEMA6



(b) CEMA7



(c) CEMA8

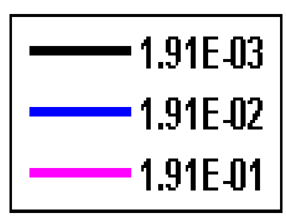
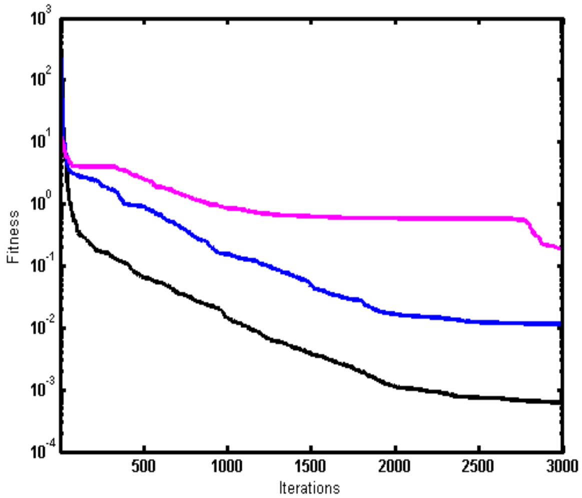
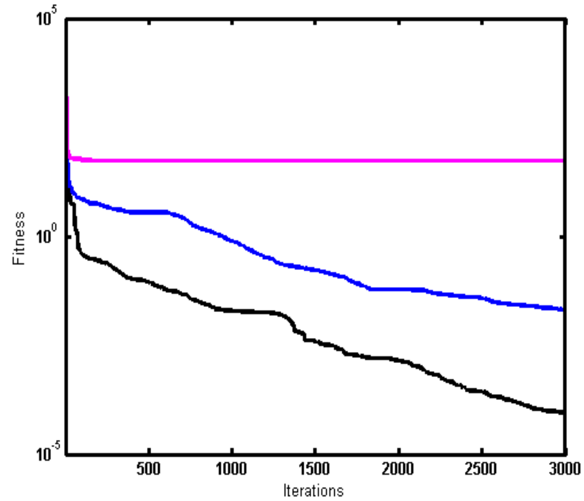


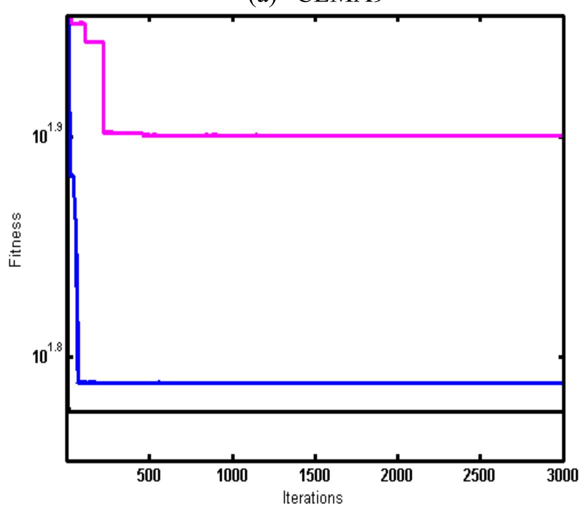
Figure 4.13: Analysis of CEMA6, CEMA7 and CEMA8 for Pop=18



(a) CEMA9



(b) CEMA10



(a) COA

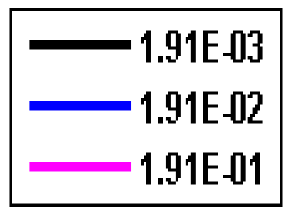
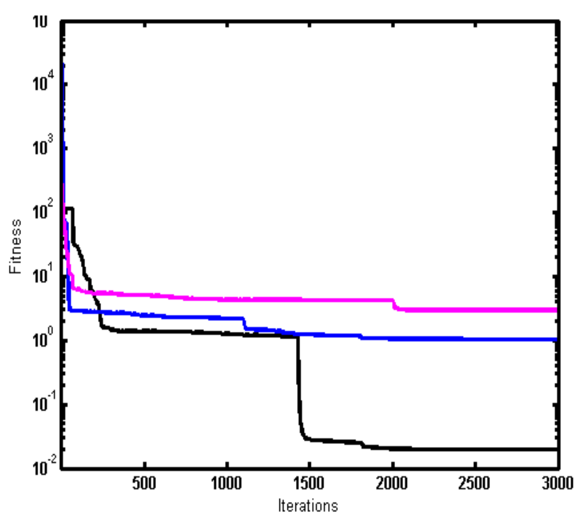
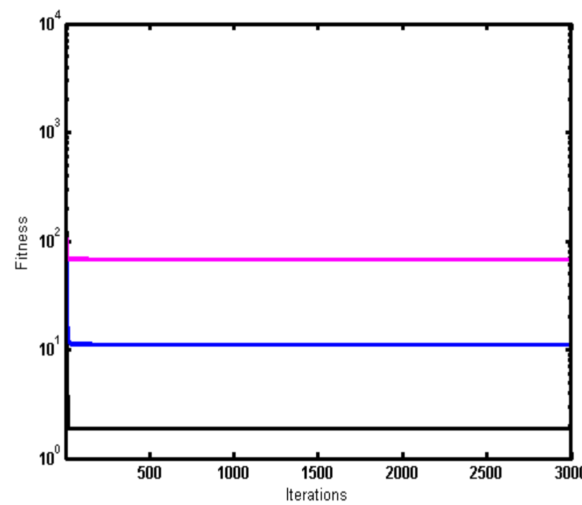


Figure 4.14: Analysis of CEMA9, CEMA10 and COA for Pop=18





(b) HHO (c) PSO

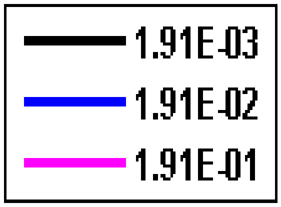
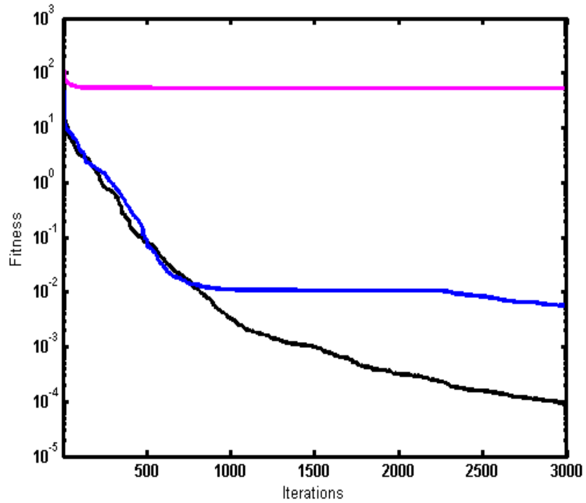
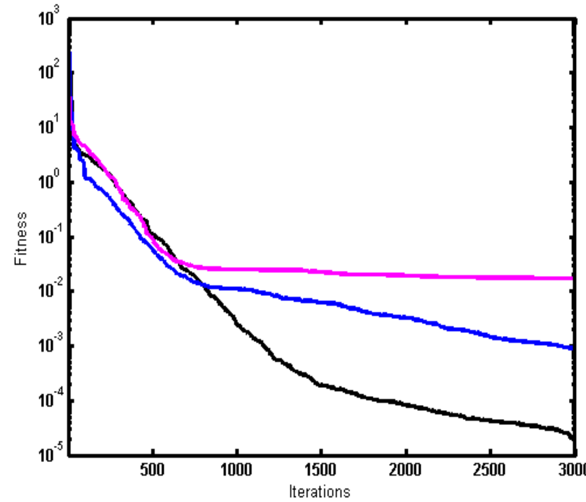


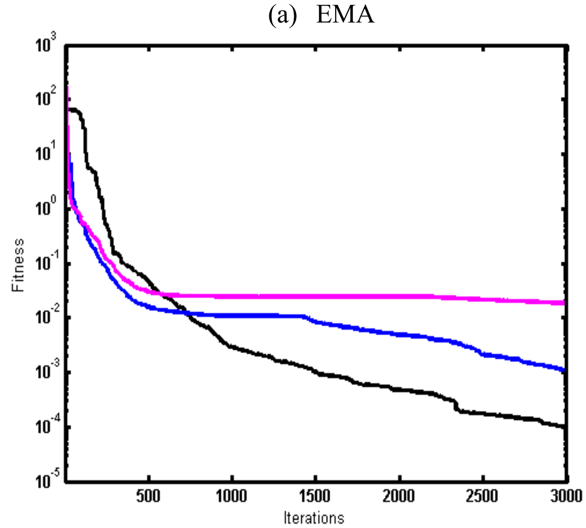
Figure 4.15: Analysis of HHO and PSO for Pop=18

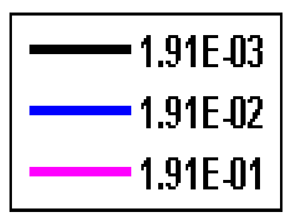


(a) EMA



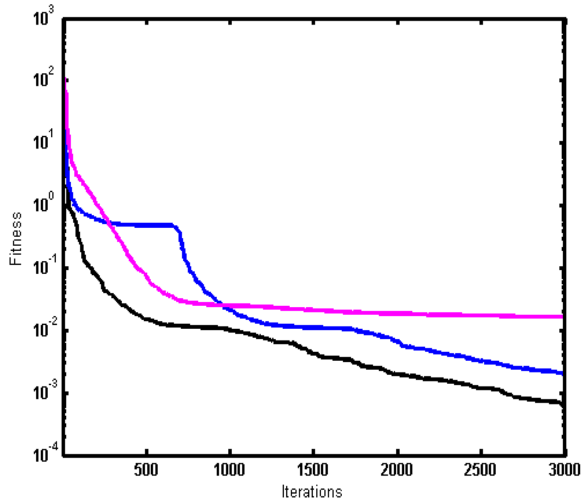
(b) CEMA1



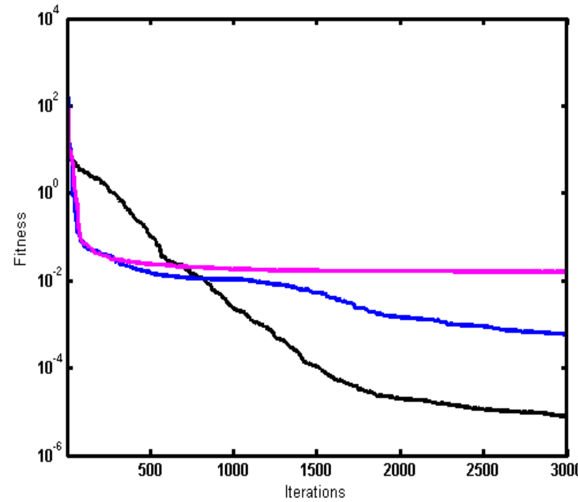


(c) CEMA2

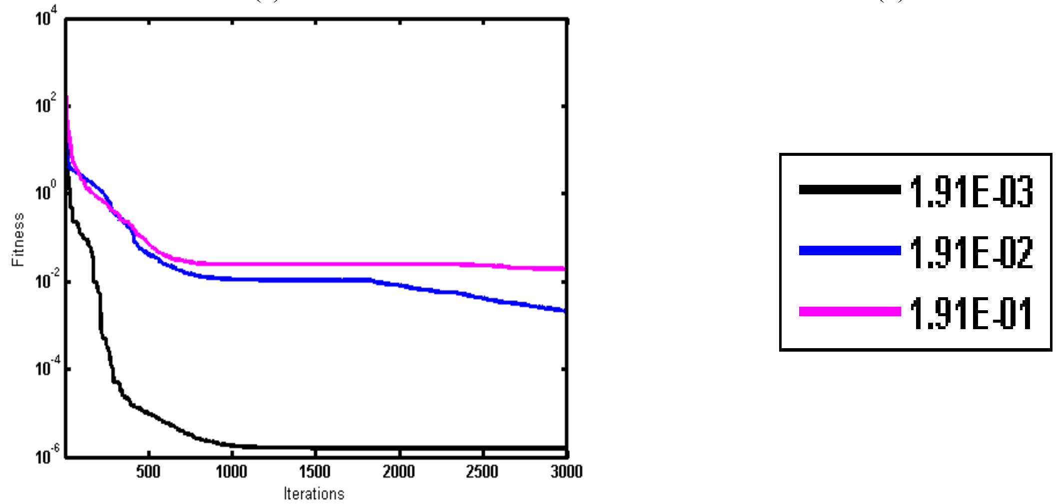
Figure 4.16: Analysis of EMA, CEMA1 and CEMA2 for Pop=50



(a) CEMA3

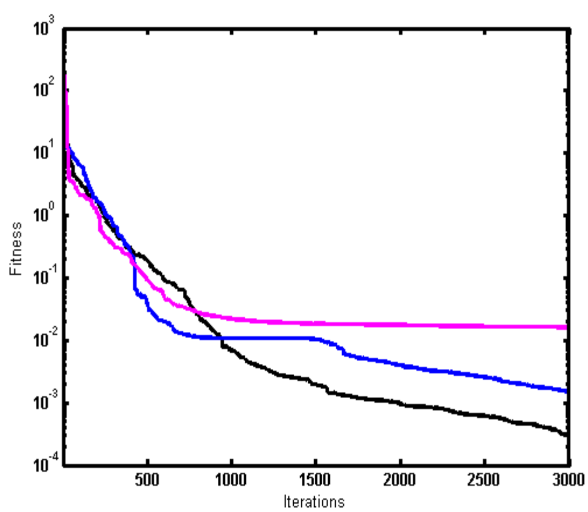


(b) CEMA4

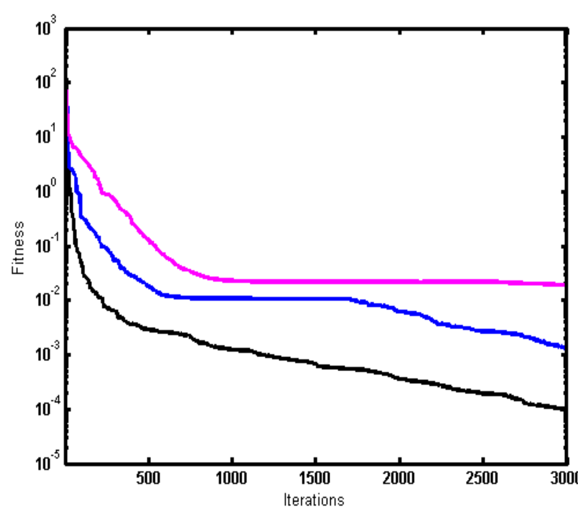


(c) CEMA5

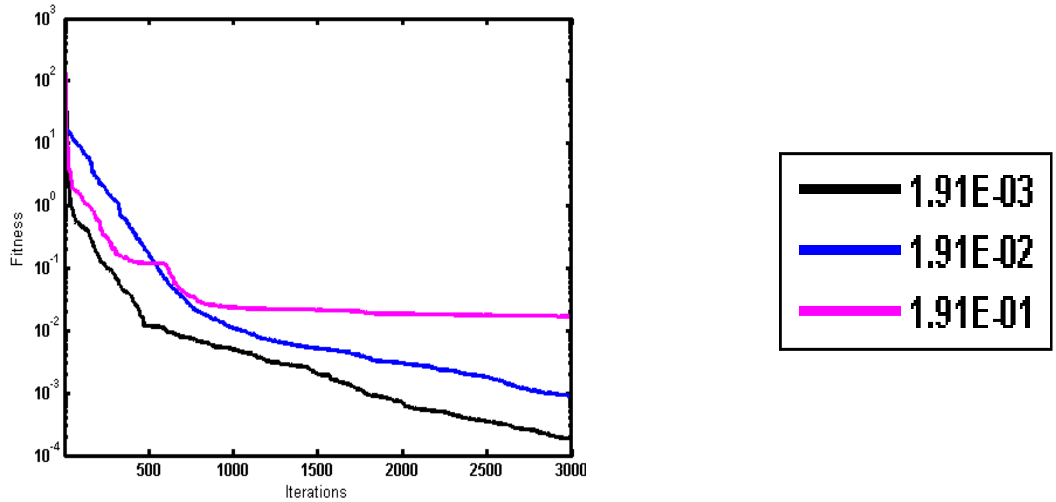
Figure 4.17: Analysis of CEMA3, CEMA4 and CEMA5 for Pop=50



(a) CEMA6

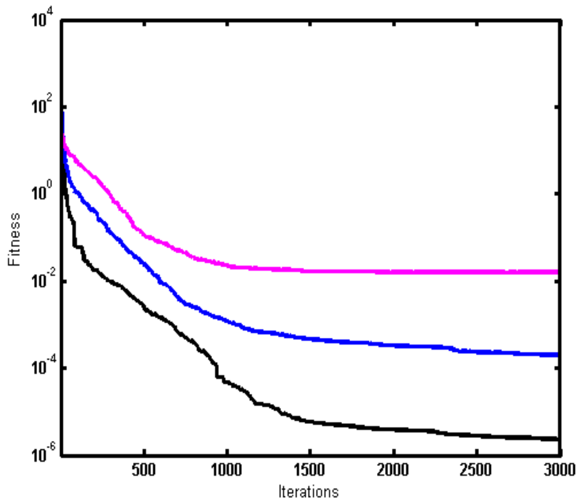


(b) CEMA7

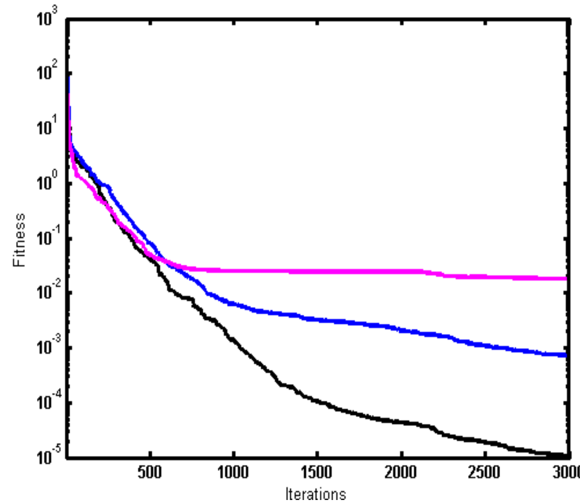


(c) CEMA8

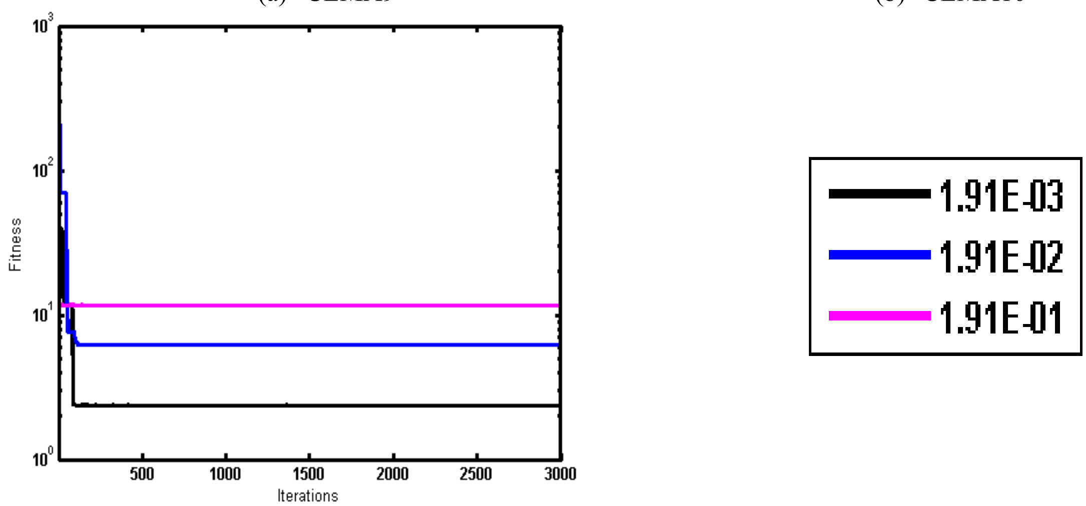
Figure 4.18: Analysis of CEMA6, CEMA7 and CEMA8 for Pop=50



(a) CEMA9

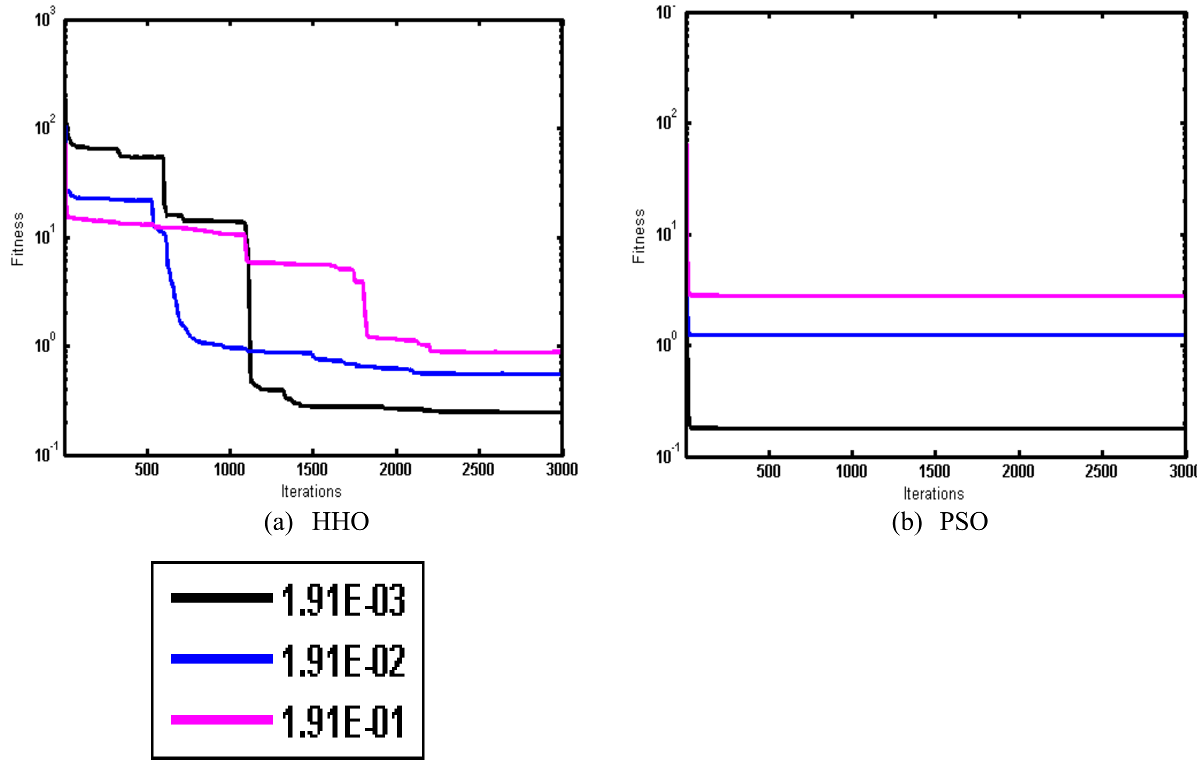


(b) CEMA10



(c) COA

Figure 4.19: Analysis of CEMA9, CEMA10 and COA for Pop=50



(a) HHO (b) PSO

Figure 4.20: Analysis of HHO and PSO for Pop=50

Table 4.30 shows the performance of EMA, CEMA1, CEMA2, CEMA3, CEMA4, CEMA5, CEMA6, CEMA7, CEMA8, CEMA9, CEMA10, COA, HHO and PSO in terms of average execution time and related standard deviation (STD) for $E(\tau)$=1.91E-03. It is observed from Table 4.30 that CEMA8 attain better results in terms of average execution time while maintaining lower fitness at pop = 18 and 50. It is also noted that CEMA1 shows better results in terms of STD at pop = 18.

Table 4.30: Statistical Analysis of EMA and other Metaheuristics

Methods	Population	Avg. Time	STD
EMA	18	6.47E+00	1.15E-01
	50	1.76E+01	3.16E-01
CEMA1	18	6.16E+00	9.77E-02
	50	1.66E+01	3.05E-01
CEMA2	18	6.70E+00	1.18E-01
	50	1.83E+01	3.46E-01

CEMA3	18	7.13E+00	1.18E-01
	50	1.94E+01	3.36E-01
CEMA4	18	6.35E+00	1.16E-01
	50	1.72E+01	2.69E-01
CEMA5	18	6.14E+00	1.15E-01
	50	1.66E+01	2.81E-01
CEMA6	18	6.46E+00	1.05E-01
	50	1.76E+01	2.91E-01
CEMA7	18	6.25E+00	1.01E-01
	50	1.70E+01	2.89E-01
CEMA8	18	5.81E+00	1.19E-01
	50	1.58E+01	2.45E-01
CEMA9	18	5.87E+00	1.05E-01
	50	1.59E+01	2.68E-01
CEMA10	18	6.45E+00	1.03E-01
	50	1.74E+01	2.87E-01
COA	18	1.74E+01	2.53E-01
	50	4.73E+01	5.25E-01
HHO	18	1.70E+01	3.44E-01
	50	4.67E+01	5.96E-01
PSO	18	7.05E+00	1.09E-01
	50	1.92E+00	3.69E-01

Table 4.31 presents the performance of EMA, CEMA1, CEMA2, CEMA3, CEMA4, CEMA5, CEMA6, CEMA7, CEMA8, CEMA9, CEMA10, COA, HHO and PSO for Friedman rank test analysis. It is observed from Table 4.31 that CEMA9 has highest rank than all other methods.

Table 4.31: Statistical Analysis using Friedman rank test

Methods	**Rank sum**	**Rank**
EMA	111.5	11
CEMA1	80.5	5
CEMA2	97.5	9
CEMA3	73.5	4
CEMA4	87.5	6
CEMA5	72.5	3
CEMA6	96.5	8
CEMA7	90	7
CEMA8	98.5	10
CEMA9	48.5	1
CEMA10	60	2
PSO	253	13
COA	275	14
HHO	133	12

The evaluation of EMA, CEMA1, CEMA2, CEMA3, CEMA4, CEMA5, CEMA6, CEMA7, CEMA8, CEMA9, CEMA10, COA, HHO and PSO for INOE model parameters is deliberated on the disturbance levels $E(\tau)$=[1.91E-03, 1.91E-02, 1.91E-01].

Detailed statistical, convergence, complexity and Freidman ranksum test show that chaotic variants of EMA achieves best performance against evolutionary mating algorithm (EMA) [52], coati optimization algorithm (COA) [55], Harris hawks optimization (HHO) [56], and particle swarm optimization (PSO) [41].

CHAPTER 5

Conclusion and Future Work

In this chapter, results of EMA and chaotic variants of EMA on IN-OE model will be concluded. Also a way forward for researchers will be proposed to optimize the parameters in system identification field.

5.1 Conclusion

The conclusion of this research after presenting considerable simulation results in previous chapter are given as follows:

- The evolutionary-based, EMA algorithm is proposed for identification of an IN-OE system, represented with key term separation technique.
- The chaotic EMA is established by assimilating the chaos theory with the conventional EMA exploration process.
- The simulations results show that EMA with a chaotic sinusoidal map (CEMA9) executes better results than CEMA1 to CEMA8, CEMA10, standard EMA, as well as recent metaheuristics based on PSO, COA and HHO for identification of IN-OE system.

5.2 Future Work

- This research can be considered as encouraging step to identify the IN-OE model parameters by using metaheuristic optimization algorithms.
- The proposed methodology can further be investigated in other engineering fields such as system identification of wiener Hammerstein models with colored noise based on hybrid signals [77].
- Hybrid model approach can also be designed by integrating chaotic variants of EMA with other metaheuristic algorithms to optimize the system parameters of Hammerstein and wiener models in system identification field.

References

[1] T. A. Khan et al., “A gazelle optimization expedition for key term separated fractional nonlinear systems with application to electrically stimulated muscle modeling,” Chaos Solitons Fractals, vol. 185, p. 115111, Aug. 2024, doi: 10.1016/j.chaos.2024.115111.

[2] K. Mehmood, N. I. Chaudhary, Z. A. Khan, K. M. Cheema, and M. A. Z. Raja, “Parameter estimation of nonlinear systems: dwarf mongoose optimization algorithm with key term separation principle,” J. Ambient Intell. Humaniz. Comput., vol. 14, no. 12, pp. 16921–16931, Dec. 2023, doi: 10.1007/s12652-023-04707-5.

[3] A. Mehmood and M. A. Z. Raja, “Novel design of weighted differential evolution for parameter estimation of Hammerstein-Wiener systems,” J. Adv. Res., vol. 43, pp. 123–136, Jan. 2023, doi: 10.1016/j.jare.2022.02.010.

[4] C. M. Nguyen, A.-T. Nguyen, and S. Delprat, “Neural-Network-Based Fuzzy Observer With Data-Driven Uncertainty Identification for Vehicle Dynamics Estimation Under Extreme Driving Conditions: Theory and Experimental Results,” IEEE Trans. Veh. Technol., vol. 72, no. 7, pp. 8686–8696, Jul. 2023, doi: 10.1109/TVT.2023.3249832.

[5] Y. Meng, X. Zhang, and X. Zhang, “Identification modeling of ship nonlinear motion based on nonlinear innovation,” Ocean Eng., vol. 268, p. 113471, Jan. 2023, doi: 10.1016/j.oceaneng.2022.113471.

[6] M. A. Balasubramani, R. Venkatakrishnaiah, and K. V. B. Raju, “Evolutionary artificial intelligence in the Industry 4.0 to simulate static loads in railway tracks for geosynthetic-reinforced soil structures based geogrid,” Int. J. Appl. Nonlinear Sci., vol. 4, no. 2, pp. 106–121, 2024, doi: 10.1504/IJANS.2024.137160.

[7] M. U. Hayat, A. Ali, B. Khan, K. Mehmood, K. Ullah, and M. Amir, “An improved spatial–temporal regularization method for visual object tracking,” Signal Image Video Process., vol. 18, no. 3, pp. 2065–2077, Apr. 2024, doi: 10.1007/s11760-023-02842-2.

[8] T. Ibuki, T. Hirano, R. Funada, and M. Sampei, “Optimization-based distributed safety control with applications to collision avoidance for mobile robotic networks,” Adv. Robot., vol. 37, no. 1–2, pp. 87–98, Jan. 2023, doi: 10.1080/01691864.2022.2119886.

[9] J. Liang, Y. Wang, Y. Chen, B. Yang, and D. Liu, “A Triangulation-Based Visual Localization for Field Robots,” IEEECAA J. Autom. Sin., vol. 9, no. 6, pp. 1083–1086, Jun. 2022, doi: 10.1109/JAS.2022.105632.

[10] S. Jeyanthi, R. Venkatakrishnaiah, and K. V. B. Raju, “Numerical study on geocell reinforced poor sand-based ground improvement using jellyfish optimisation algorithm,” Int. J. Appl. Nonlinear Sci., vol. 4, no. 2, pp. 157–173, 2024, doi: 10.1504/IJANS.2024.137179.

[11] N. A. Malik et al., “Parameter estimation of harmonics arising in electrical instruments of smart grids using cuckoo search heuristics,” Front. Energy Res., vol. 10, p. 1059132, Nov. 2022, doi: 10.3389/fenrg.2022.1059132.

[12] T. A. Khan, N. I. Chaudhary, Z. A. Khan, K. Mehmood, C.-C. Hsu, and M. A. Z. Raja, “Design of Runge-Kutta optimization for fractional input nonlinear autoregressive exogenous system identification with key-term separation,” Chaos Solitons Fractals, vol. 182, p. 114723, May 2024, doi: 10.1016/j.chaos.2024.114723.

[13] X. Wang, F. Ding, A. Alsaedi, and T. Hayat, “Auxiliary model-based iterative parameter estimation for a nonlinear output-error system with saturation and dead-zone nonlinearity,” Int. J. Robust Nonlinear Control, vol. 31, no. 9, pp. 4262–4286, Jun. 2021, doi: 10.1002/rnc.5468.

[14] Y. Tomita, A. A. Damen, and P. M. V. D. HOF, "Equation error versus output error methods," Ergonomics, vol. 35, no. 5-6, pp. 551-564, 1992.

[15] F. Ding, L. Xu, X. Zhang, and H. Ma, “Hierarchical gradient- and least-squares-based iterative estimation algorithms for input-nonlinear output-error systems from measurement information by using the over-parameterization,” Int. J. Robust Nonlinear Control, vol. 34, no. 2, pp. 1120–1147, Jan. 2024, doi: 10.1002/rnc.7014.

[16] C. Hu, Y. Ji, and C. Ma, “Joint two-stage multi-innovation recursive least squares parameter and fractional-order estimation algorithm for the fractional-order input nonlinear output-error autoregressive model,” Int. J. Adapt. Control Signal Process., vol. 37, no. 7, pp. 1650–1670, Jul. 2023, doi: 10.1002/acs.3593.

[17] K. Mehmood et al., “Design of Nonlinear Marine Predator Heuristics for Hammerstein Autoregressive Exogenous System Identification with Key-Term Separation,” Mathematics, vol. 11, no. 11, p. 2512, May 2023, doi: 10.3390/math11112512.

[18] F. Ding, L. Xu, X. Zhang, Y. Zhou, and X. Luan, “Recursive identification methods for general stochastic systems with colored noises by using the hierarchical identification principle and the filtering identification idea,” Annu. Rev. Control, vol. 57, p. 100942, 2024, doi: 10.1016/j.arcontrol.2024.100942.

[19] Y. Fan and X. Liu, “Auxiliary model-based multi-innovation recursive identification algorithms for an input nonlinear controlled autoregressive moving average system with variable-gain nonlinearity,” Int. J. Adapt. Control Signal Process., vol. 36, no. 3, pp. 521–540, Mar. 2022, doi: 10.1002/acs.3354.

[20] R. Salgotra, P. Sharma, S. Raju, and A. H. Gandomi, “A Contemporary Systematic Review on Meta-heuristic Optimization Algorithms with Their MATLAB and Python Code Reference,” Arch. Comput. Methods Eng., vol. 31, no. 3, pp. 1749–1822, Apr. 2024, doi: 10.1007/s11831-023-10030-1.

[21] L. Abualigah et al., “Meta-heuristic optimization algorithms for solving real-world mechanical engineering design problems: a comprehensive survey, applications, comparative analysis, and results,” Neural Comput. Appl., vol. 34, no. 6, pp. 4081–4110, Mar. 2022, doi: 10.1007/s00521-021-06747-4.

[22] C. A. U. Hassan, J. Iqbal, N. Ayub, S. Hussain, R. Alroobaea, and S. S. Ullah, “Smart Grid Energy Optimization and Scheduling Appliances Priority for Residential Buildings through Meta-Heuristic Hybrid Approaches,” Energies, vol. 15, no. 5, p. 1752, Feb. 2022, doi: 10.3390/en15051752.

[23] J. Lian et al., “Parrot optimizer: Algorithm and applications to medical problems,” Comput. Biol. Med., vol. 172, p. 108064, Apr. 2024, doi: 10.1016/j.compbiomed.2024.108064.

[24] S. Anosri et al., “A Comparative Study of State-of-the-art Metaheuristics for Solving Many-objective Optimization Problems of Fixed Wing Unmanned Aerial Vehicle Conceptual Design,” Arch. Comput. Methods Eng., vol. 30, no. 6, pp. 3657–3671, Jul. 2023, doi: 10.1007/s11831-023-09914-z.

[25] T. Luo, J. Xie, B. Zhang, Y. Zhang, C. Li, and J. Zhou, “An improved levy chaotic particle swarm optimization algorithm for energy-efficient cluster routing scheme in industrial wireless sensor networks,” Expert Syst. Appl., vol. 241, p. 122780, May 2024, doi: 10.1016/j.eswa.2023.122780.

[26] K. M. Cheema et al., “Knacks of marine predator heuristics for distributed energy source-based power systems harmonics estimation,” Heliyon, vol. 10, no. 15, p. e35776, Aug. 2024, doi: 10.1016/j.heliyon.2024.e35776.

[27] A. Slowik and H. Kwasnicka, “Evolutionary algorithms and their applications to engineering problems,” Neural Comput. Appl., vol. 32, no. 16, pp. 12363–12379, Aug. 2020, doi: 10.1007/s00521-020-04832-8.

[28] A. Singh and A. Kumar, “Applications of nature-inspired meta-heuristic algorithms: a survey,” Int. J. Adv. Intell. Paradig., vol. 20, no. 3/4, p. 388, 2021, doi: 10.1504/IJAIP.2021.119026.

[29] A. Alorf, “A survey of recently developed metaheuristics and their comparative analysis,” Eng. Appl. Artif. Intell., vol. 117, p. 105622, Jan. 2023, doi: 10.1016/j.engappai.2022.105622.

[30] A. E. Ezugwu et al., “Metaheuristics: a comprehensive overview and classification along with bibliometric analysis,” Artif. Intell. Rev., vol. 54, no. 6, pp. 4237–4316, Aug. 2021, doi: 10.1007/s10462-020-09952-0.

[31] H. Zamani, M. H. Nadimi-Shahraki, and A. H. Gandomi, “QANA: Quantum-based avian navigation optimizer algorithm,” Eng. Appl. Artif. Intell., vol. 104, p. 104314, Sep. 2021, doi: 10.1016/j.engappai.2021.104314.

[32] Y. Song and X. Fang, “An Improved Strength Pareto Evolutionary Algorithm 2 with Adaptive Crossover Operator for Bi-Objective Distributed Unmanned Aerial Vehicle Delivery,” Mathematics, vol. 11, no. 15, p. 3327, Jul. 2023, doi: 10.3390/math11153327.

[33] S. Mirjalili, “Genetic Algorithm,” in Evolutionary Algorithms and Neural Networks, vol. 780, in Studies in Computational Intelligence, vol. 780. , Cham: Springer International Publishing, 2019, pp. 43–55. doi: 10.1007/978-3-319-93025-1_4.

[34] S. Gao, K. Wang, S. Tao, T. Jin, H. Dai, and J. Cheng, “A state-of-the-art differential evolution algorithm for parameter estimation of solar photovoltaic models,” Energy Convers. Manag., vol. 230, p. 113784, Feb. 2021, doi: 10.1016/j.enconman.2020.113784.

[35] S. A. SALIH and T. A. RASHID, "A LEARNER PERFORMANCE-BASED BEHAVIOR ALGORITHM FOR SOLVING TRAVELLING SALESMAN PROBLEM," Journal of Duhok University, vol. 26, no. 1, pp. 291-304, 2023.

[36] A. Kaveh and V. R. Mahdavi, “Colliding bodies optimization: A novel meta-heuristic method,” Comput. Struct., vol. 139, pp. 18–27, Jul. 2014, doi: 10.1016/j.compstruc.2014.04.005.

[37] B. Doğan and T. Ölmez, “A new metaheuristic for numerical function optimization: Vortex Search algorithm,” Inf. Sci., vol. 293, pp. 125–145, Feb. 2015, doi: 10.1016/j.ins.2014.08.053.

[38] E. Cuevas, A. Echavarría, and M. A. Ramírez-Ortegón, “An optimization algorithm inspired by the States of Matter that improves the balance between exploration and exploitation,” Appl. Intell., vol. 40, no. 2, pp. 256–272, Mar. 2014, doi: 10.1007/s10489-013-0458-0.

[39] H. Abedinpourshotorban, S. Mariyam Shamsuddin, Z. Beheshti, and D. N. A. Jawawi, “Electromagnetic field optimization: A physics-inspired metaheuristic optimization algorithm,” Swarm Evol. Comput., vol. 26, pp. 8–22, Feb. 2016, doi: 10.1016/j.swevo.2015.07.002.

[40] A. Kaveh and M. Khayatazad, “A new meta-heuristic method: Ray Optimization,” Comput. Struct., vol. 112–113, pp. 283–294, Dec. 2012, doi: 10.1016/j.compstruc.2012.09.003.

[41] Y.-P. Chen et al., “A novel bacterial foraging optimization algorithm for feature selection,” Expert Syst. Appl., vol. 83, pp. 1–17, Oct. 2017, doi: 10.1016/j.eswa.2017.04.019.

[42] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, “Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems,” Adv. Eng. Softw., vol. 114, pp. 163–191, Dec. 2017, doi: 10.1016/j.advengsoft.2017.07.002.

[43] L. Abualigah et al., “Particle swarm optimization algorithm: review and applications,” in Metaheuristic Optimization Algorithms, Elsevier, 2024, pp. 1–14. doi: 10.1016/B978-0-443-13925-3.00019-4.

[44] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization,” IEEE Comput. Intell. Mag., vol. 1, no. 4, pp. 28–39, Nov. 2006, doi: 10.1109/MCI.2006.329691.

[45] S. Alzoubi, L. Abualigah, M. Sharaf, M. Sh. Daoud, N. Khodadadi, and H. Jia, “Synergistic Swarm Optimization Algorithm,” Comput. Model. Eng. Sci., vol. 139, no. 3, pp. 2557–2604, 2024, doi: 10.32604/cmes.2023.045170.

[46] A. Kaveh and S. Talatahari, "Optimum design of skeletal structures using imperialist competitive algorithm," Computers & structures, vol. 88, no. 21-22, pp. 1220-1229, 2010.

[47] H. Emami, “Stock exchange trading optimization algorithm: a human-inspired method for global optimization,” J. Supercomput., vol. 78, no. 2, pp. 2125–2174, Feb. 2022, doi: 10.1007/s11227-021-03943-w.

[48] T. T. Huan, A. J. Kulkarni, J. Kanesan, C. J. Huang, and A. Abraham, “Ideology algorithm: a socio-inspired optimization methodology,” Neural Comput. Appl., vol. 28, no. S1, pp. 845–876, Dec. 2017, doi: 10.1007/s00521-016-2379-4.

[49] A. Husseinzadeh Kashan, “League Championship Algorithm (LCA): An algorithm for global optimization inspired by sport championships,” Appl. Soft Comput., vol. 16, pp. 171–200, Mar. 2014, doi: 10.1016/j.asoc.2013.12.005.

[50] A. Kaveh, “Tug of War Optimization,” in Advances in Metaheuristic Algorithms for Optimal Design of Structures, Cham: Springer International Publishing, 2021, pp. 467–503. doi: 10.1007/978-3-030-59392-6_15.

[51] A. H. Gandomi, “Interior search algorithm (ISA): A novel approach for global optimization,” ISA Trans., vol. 53, no. 4, pp. 1168–1183, Jul. 2014, doi: 10.1016/j.isatra.2014.03.018.

[52] M. H. Sulaiman, Z. Mustaffa, M. M. Saari, H. Daniyal, and S. Mirjalili, “Evolutionary mating algorithm,” Neural Comput. Appl., vol. 35, no. 1, pp. 487–516, Jan. 2023, doi: 10.1007/s00521-022-07761-w.

[53] A. Irawan, M. H. Sulaiman, M. S. Ramli, and M. I. P. Azahar, “Pneumatic servo position control optimization using adaptive-domain prescribed performance control with evolutionary mating algorithm,” Results Control Optim., vol. 15, p. 100434, Jun. 2024, doi: 10.1016/j.rico.2024.100434.

[54] M. H. Sulaiman, Z. Mustaffa, N. F. Zakaria, and M. M. Saari, “Using the evolutionary mating algorithm for optimizing deep learning parameters for battery state of charge estimation of electric vehicle,” Energy, vol. 279, p. 128094, Sep. 2023, doi: 10.1016/j.energy.2023.128094.

[55] M. H. Sulaiman and Z. Mustaffa, “Using the evolutionary mating algorithm for optimizing the user comfort and energy consumption in smart building,” J. Build. Eng., vol. 76, p. 107139, Oct. 2023, doi: 10.1016/j.jobe.2023.107139.

[56] M. H. Sulaiman and Z. Mustaffa, “Forecasting solar power generation using evolutionary mating algorithm-deep neural networks,” Energy AI, vol. 16, p. 100371, May 2024, doi: 10.1016/j.egyai.2024.100371.

[57] B. Alvin, F. Husnayain, B. Sudiarto, and R. Setiabudy, “Energy Management System using Evolutionary Mating Algorithms for Optimizing Energy Usage and User Comfort in Office Bulding,” Int. J. Electr. Comput. Biomed. Eng., vol. 2, no. 2, Jun. 2024, doi: 10.62146/ijecbe.v2i2.50.

[58] M. Dehghani, Z. Montazeri, E. Trojovská, and P. Trojovský, “Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems,” Knowl.-Based Syst., vol. 259, p. 110011, Jan. 2023, doi: 10.1016/j.knosys.2022.110011.

[59] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, “Harris hawks optimization: Algorithm and applications,” Future Gener. Comput. Syst., vol. 97, pp. 849–872, Aug. 2019, doi: 10.1016/j.future.2019.02.028.

[60] M. S. Baptista, “Chaos for communication,” Nonlinear Dyn., vol. 105, no. 2, pp. 1821–1841, Jul. 2021, doi: 10.1007/s11071-021-06644-4.

[61] M. T. Elkandoz and W. Alexan, “Image encryption based on a combination of multiple chaotic maps,” Multimed. Tools Appl., vol. 81, no. 18, pp. 25497–25518, Jul. 2022, doi: 10.1007/s11042-022-12595-8.

[62] D. A. Trujillo-Toledo et al., “Real-time RGB image encryption for IoT applications using enhanced sequences from chaotic maps,” Chaos Solitons Fractals, vol. 153, p. 111506, Dec. 2021, doi: 10.1016/j.chaos.2021.111506.

[63] M. Alawida, A. Samsudin, and J. S. Teh, “Enhanced digital chaotic maps based on bit reversal with applications in random bit generators,” Inf. Sci., vol. 512, pp. 1155–1169, Feb. 2020, doi: 10.1016/j.ins.2019.10.055.

[64] K. Mehmood, N. I. Chaudhary, Z. A. Khan, K. M. Cheema, M. A. Z. Raja, and C.-M. Shu, “Novel knacks of chaotic maps with Archimedes optimization paradigm for nonlinear ARX model identification with key term separation,” Chaos Solitons Fractals, vol. 175, p. 114028, Oct. 2023, doi: 10.1016/j.chaos.2023.114028.

[65] B. Alatas, E. Akin, and A. B. Ozer, “Chaos embedded particle swarm optimization algorithms,” Chaos Solitons Fractals, vol. 40, no. 4, pp. 1715–1734, May 2009, doi: 10.1016/j.chaos.2007.09.063.

[66] E. Varol Altay and B. Alatas, “Bird swarm algorithms with chaotic mapping,” Artif. Intell. Rev., vol. 53, no. 2, pp. 1373–1414, Feb. 2020, doi: 10.1007/s10462-019-09704-9.

[67] K. Mehmood, Z. A. Khan, N. I. Chaudhary, K. M. Cheema, B. Siddiqui, and M. A. Z. Raja, “Design of chaotic Young’s double slit experiment optimization heuristics for identification of nonlinear muscle model with key term separation,” Chaos Solitons Fractals, vol. 189, p. 115636, Dec. 2024, doi: 10.1016/j.chaos.2024.115636.

[68] S. Kumar et al., “Chaotic marine predators algorithm for global optimization of real-world engineering problems,” Knowl.-Based Syst., vol. 261, p. 110192, Feb. 2023, doi: 10.1016/j.knosys.2022.110192.

[69] L. Li, Y. Yang, H. Peng, and X. Wang, “AN OPTIMIZATION METHOD INSPIRED BY ‘CHAOTIC’ ANT BEHAVIOR,” Int. J. Bifurc. Chaos, vol. 16, no. 08, pp. 2351–2364, Aug. 2006, doi: 10.1142/S0218127406016100.

[70] S. Saremi, S. Mirjalili, and A. Lewis, “Biogeography-based optimisation with chaos,” Neural Comput. Appl., vol. 25, no. 5, pp. 1077–1097, Oct. 2014, doi: 10.1007/s00521-014-1597-x.

[71] O. E. Turgut, M. S. Turgut, and E. Kırtepe, “Chaotic Aquila Optimization Algorithm for Solving Phase Equilibrium Problems and Parameter Estimation of Semi-empirical Models,” J. Bionic Eng., vol. 21, no. 1, pp. 486–526, Jan. 2024, doi: 10.1007/s42235-023-00438-7.

[72] X.-D. Li, J.-S. Wang, W.-K. Hao, M. Zhang, and M. Wang, “Chaotic arithmetic optimization algorithm,” Appl. Intell., vol. 52, no. 14, pp. 16718–16757, Nov. 2022, doi: 10.1007/s10489-021-03037-3.

[73] K. Mehmood, N. I. Chaudhary, Z. A. Khan, K. M. Cheema, and M. A. Z. Raja, “Variants of Chaotic Grey Wolf Heuristic for Robust Identification of Control Autoregressive Model,” Biomimetics, vol. 8, no. 2, p. 141, Mar. 2023, doi: 10.3390/biomimetics8020141.

[74] B. Alatas, “Chaotic harmony search algorithms,” Appl. Math. Comput., vol. 216, no. 9, pp. 2687–2699, Jul. 2010, doi: 10.1016/j.amc.2010.03.114.

[75] Z. Garip, E. Ekinci, K. Serbest, and S. Eken, “Chaotic marine predator optimization algorithm for feature selection in schizophrenia classification using EEG signals,” Clust. Comput., vol. 27, no. 8, pp. 11277–11297, Nov. 2024, doi: 10.1007/s10586-024-04511-6.

[76] N. El Ghouate et al., “Improving the Kepler optimization algorithm with chaotic maps: comprehensive performance evaluation and engineering applications,” Artif. Intell. Rev., vol. 57, no. 11, p. 313, Oct. 2024, doi: 10.1007/s10462-024-10857-5.

[77] F. Li and J. Han, "Parameters estimation for the Hammerstein-Wiener models with colored noise based on hybrid signals," International Journal of Adaptive Control and Signal Processing, vol. 38, no. 3, pp. 921-937, 2024.