Development of Chaotic Metaheuristic Algorithm for Engineering Design Problems

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Computer Science and Engineering

Submitted By
Kamalinder Kaur Kaleka
(Roll No. 801732026)

Under the supervision of:
Dr. Vijay Kumar
Assistant Professor
TIET, Patiala

Dr. Prashant Singh Rana
Assistant Professor
TIET, Patiala

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY
PATIALA – 147004

June 2019

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled ***"Development of Chaotic Metaheuristic Algorithm for Engineering Design Problems ",*** in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Computer Science and Engineering* submitted in Computer Science and Engineering Department of Thapar Institute of Engineering and Technology, Patiala, is an authentic record of my own work carried out under the supervision of ***Dr. Vijay Kumar*** and ***Dr. Prashant Singh Rana*** and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

(Kamalinder Kaur Kaleka)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. Vijay Kumar)
Assistant Professor,
CSED
TIET, Patiala

(Dr. Prashant Singh Rana)
Assistant Professor,
CSED
TIET, Patiala

Acknowledgement

The research work would be incomplete without acknowledging the people who made it possible and whose constant guidance and encouragement secured the success. First of all I wish to acknowledge the benevolence of omnipotent God who gave me strength and courage to overcome all obstacles .With the profound sense of gratitude and heartiest regard, I express my deep gratitude to my supervisor Dr. Vijay Kumar, Assistant Professor, Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, Patiala, and Dr. Prashant Singh Rana, Assistant Professor, Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, Patiala, for their positive attitude, excellent guidance, constant encouragement, keen interest, invaluable co-operation and all their blessings have been a source of inspiration for me.

I am grateful to Dr. Maninder Singh, Head, Computer Science and Engineering Department and Dr. Ashutosh Mishra, P.G. Cordinator, Computer Science and Engineering Department for motivation and support for completion of the thesis. I would also like to express my gratitude to Dr. S.S.Bhatia. Senior Professor and Dean of Academic Affairs, for making provisions of infrastructure such as library facilities, computer labs equipped with net facilities, immensely useful for the learners to equip themselves with the latest in the field.

I am also thankful to the entire faculty and staff members of the Computer Science and Engineering Department for their help and cooperation. Last but not the least, I would like to thank my family for their motivation and encouragement, without their support none of this would have been possible.

Kamalinder Kaur Kaleka

(801732026)

Abstract

Optimization is a popular domain of research and is present in every field. The field of nature inspired computing and optimization techniques have evolved to solve the difficult optimization problems in diverse fields of engineering, science and technology. According to the "No free lunch theorem," there is no such efficient algorithm for all problems. As a result, many optimization algorithms have been developed and tried to use various improving techniques to enhance the capability of searching to see that if they can cope with these challenging optimization problems. Although many metaheuristic algorithms can accelerate the search speed, they still have major drawback of premature convergence and local optimum stagnation. Chaos theory is a novelty approach that has been widely used into various applications. Recently, numerous improvements, which rely on the chaos approach, have been proposed for metaheuristics algorithm. Spotted hyena optimizer (SHO) and Emperor penguin optimizer (EPO) are recently developed metaheuristic techniques and has been used for solving many optimization problems.

The aim of this research work is to propose the chaotic version of Spotted hyena optimizer and Emperor penguin optimizer. Ten different chaotic maps will be used in this approach. The proposed approach is tested on 29 well known benchmark test functions and a comparative analysis is made with other well-known metaheuristic algorithms. The proposed chaotic algorithms are also validated on constrained engineering design problems.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

CEPO	Chaotic Emperor Penguin Optimizer
CSHO	Chaotic Spotted Hyena Optimizer
EPO	Emperor Penguin Optimizer
GWO	Grey Wolf Optimizer
GSA	Gravitational Search Algorithm
MFO	Moth-Flame Optimization
PSO	Particle Swarm Optimization
SHO	Spotted Hyena Optimizer

CHAPTER 1
INTRODUCTION

1.1 Preamble

With the growing interest in optimization various optimization techniques have been developed in recent years. Optimization is a procedure for finding optimal solution for the given parameters of a system from all the possible values to maximize or minimize its output. Since money, time and other resources required for solving problem are limited, to find optimal solution using these valuable resources under various constraints. An optimization algorithm is a process executed iteratively by comparing various solutions till an optimum solution is found [1]. No single optimizer can solve all the optimization problems. Leading to number of methods being designed and developed for solving complex real world problems.

Optimization techniques are broadly classified into two main categories: exact and approximate. The exact algorithms guarantee an optimal solution for given problem but in exponential computational time. Whereas the approximate algorithms find reasonably good solutions in a reasonable time [2]. Metaheuristic algorithms are well known approximate algorithms that are used to solve various optimization problems [3]. The metaheuristic concept was introduced by Glover in 1986. A metaheuristic is formally defined as an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space, learning strategies are used to structure the information in order to find efficiently near optimal solutions [4]. In the past decade, a great interest has been devoted to metaheuristics. Metaheuristics are those heuristic techniques applicable for a range of computational problems.

Many of these are inspired by natural phenomena. Metaheuristics are designed and developed to find a solution that is “good enough” in a computing time that is “small enough”. Metaheuristic techniques are successfully used in many areas like engineering, finance and product management etc. to solve complex problems. Metaheuristic algorithms are known to be problem independent framework. The metaheuristic algorithms are guided by different strategies and concepts for exploration and exploitation the given search space optimally. Metaheuristic algorithms properties i.e. inexpensive, flexibility, and simplicity make them suitable for applying to various computational problems [5].

Exploration and exploitation are two important phases of metaheuristics, which help in finding the global optima [6]. The capability of exploration phase is used in avoiding local optima stagnation and keep on searching the promising regions in the given search space. Exploitation phase converges towards best optimal solution. Taking in concern balancing of these two components leads researchers to development of new metaheuristic algorithm [7, 8].

1.2 Classification

Metaheuristics are divided into following three categories namely swarm based metaheuristics methods, evolutionary based methods and physics based methods.

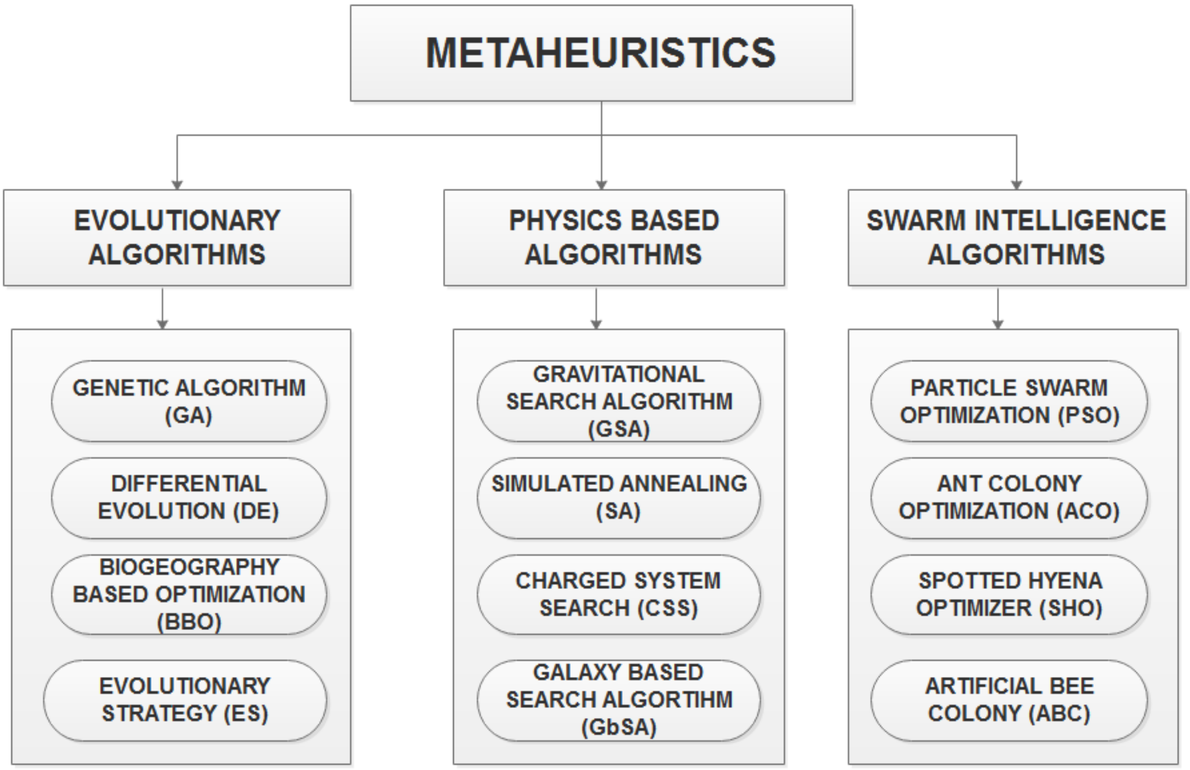


Fig. 1.1 Classification of metaheuristic algorithms

Swarm based algorithms are proposed being inspired and mimicking behaviour of social creatures. The algorithms are mathematically modelled by taking inspiration from how swarms interact and behaviour whether alone or in groups. Grey wolf optimization [5], Whale optimizer algorithm [9], Bat-inspired Algorithm (BA) [10], Seagull optimization algorithm [11] are few swarm based algorithms.

Next evolutionary category algorithms mimics biological evolution i.e. reproduction, mutation, recombination and selection. From the pool of population, a member survives on the fitness evaluation in a given conditions. Particle Swarm Optimization [12], Ant Colony Optimization [13], Genetic Algorithms [14], and Differential Evolution [15] are some of evolutionary algorithms.

Lastly physics based algorithms are inspired from physics phenomenon. Inspirational mechanism for these algorithms is such as theory of gravitational field, laws of motion, quantum theory. The names of few algorithms are Simulated Annealing [16], Gravitational Search Algorithm [17], and Black Hole algorithm [18].

1.3 Characteristics

Metaheuristic algorithms are designed efficiently so as to handle the complex optimization problems while other techniques are not able to perform efficiently. Metaheuristic algorithms characteristics make them easy to implement for real world complex problems. There are several advantages of using meta-heuristic algorithms for optimization. These are:

- Broad applicability. Metaheuristic algorithms can be applied to any problems that can be formulated as function optimization problems [5].
- Hybridization. They can be combined with more traditional optimization techniques for enhancing performance [5].
- Ease of implementation. These are typically easier to understand and implement [19].
- Efficiency and flexibility. They can solve problems larger problems faster. Moreover, they are simple to design and implement, and are very flexible [19].

In order to find the best optimal solution many metaheuristic algorithms have been designed and developed by improving the searching technique namely simulated annealing are some amongst them [20]. Nature inspired metaheuristics recently are an area of interest for researchers to design algorithm those provide promising results. Natural behaviour and hunting mechanisms of social creatures inspires for designing effective optimizers. Ant colony optimization [10] and particle swarm optimization [11] are few nature inspired algorithms.

As "No free lunch theorem" [21] that state that one single optimizer cannot perform efficiently for all sort of problems. One optimizer may provide effective results for a set of problem, but may fail to show optimal result for other problems .This has motivated researchers to design more efficient algorithms. Spotted hyena optimizer [3] and Emperor penguin optimizer [22] are the recently developed algorithms.

1.4 Spotted Hyena Optimizer

The classical spotted hyena optimizer is a recently designed metaheuristic optimizer that mimics the social relation between hyenas. In SHO, the social behaviour and hunting technique of spotted hyenas is mathematically modelled to implement the algorithm. Spotted hyenas stay and hunt for food in clusters. This hunting technique i.e. of forming clusters is helpful in finding the best optimal solution. The main steps of SHO are searching for prey, encircling the prey, hunting and attacking prey in clusters [3].

1.4.1 Encircling prey

The current best candidate solution is considered as the target prey. The other hyenas update their positions according to the position of best search solution. The mathematical model of this encircling behaviour is given below:

$$\vec{D}_h = \left|\vec{B}.\vec{X}_P(t) - \vec{X}(t)\right| \tag{1.1}$$

$$\vec{X}(t+1) = \vec{X}_P(t) - \vec{E}.\vec{D}_h \tag{1.2}$$

where $\vec{D}_h$ define the distance between prey and spotted hyena, $\cdot$ and $|\ |$represents multiplication with vectors and absolute value, respectively. $\vec{B}$ and $\vec{E}$ both are co-efficient vectors, current iteration is represented using , $\vec{X}_P$ represents prey position vector, and $\vec{X}$ represnts hyena position vector

$\vec{B}$ and $\vec{E}$ computed as:

$$\vec{B} = 2.\vec{rd}_1 \tag{1.3}$$

$$\vec{E} = 2\vec{h}.\vec{rd}_2 - \vec{h} \tag{1.4}$$

$\vec{h} = 5 - (\text{Iteration} * (5 / Max_{Iteration}))$

where, Iteration = 1, 2, 3. . . $Max_{Iteration}$ (1.5)

$\vec{h}$ linearly varies from 5 to 0. $r\vec{d}_1$ and $r\vec{d}_2$ are random vector with range [0, 1].

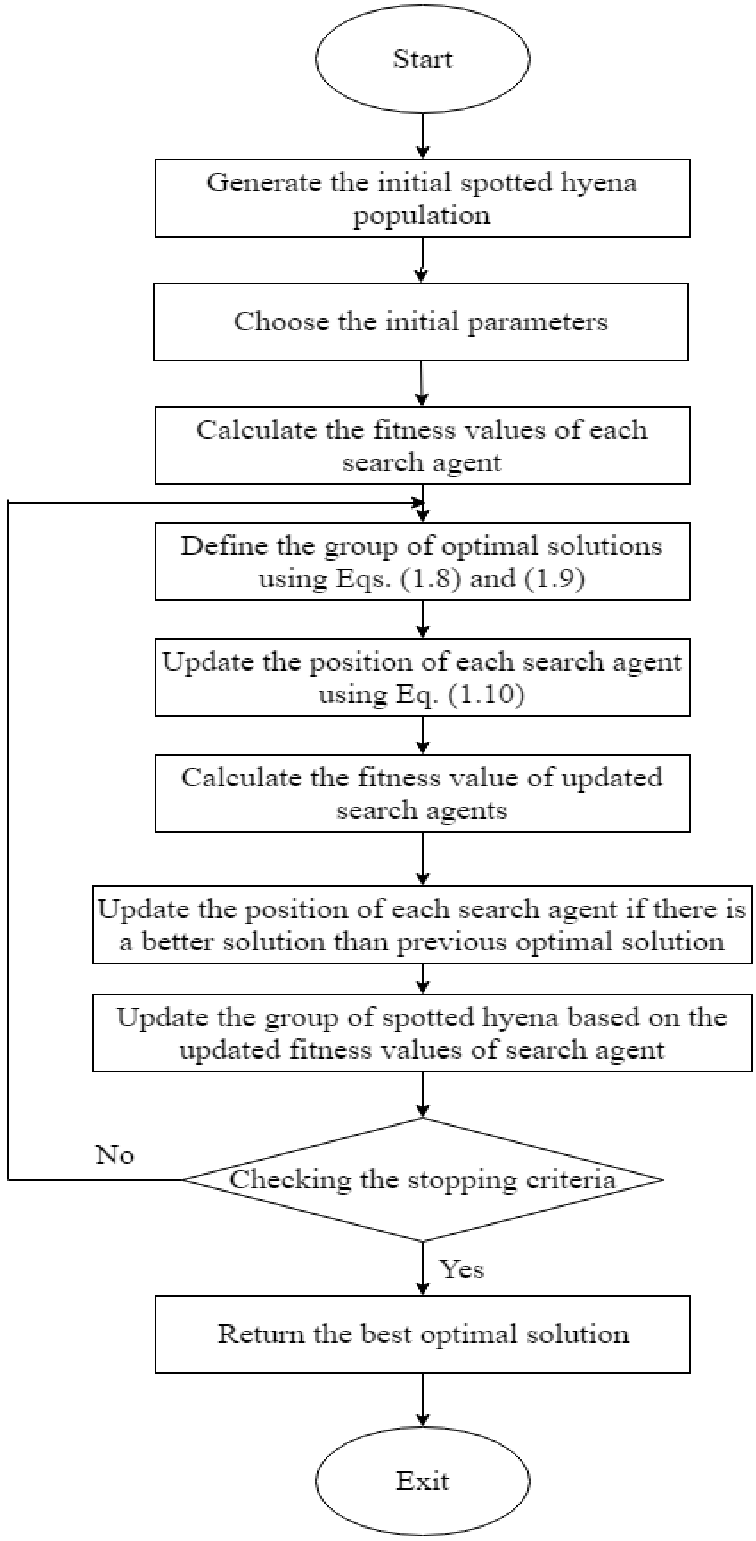


Fig. 1.2. Flowchart of Spotted Hyena Optimizer

1.4.2 Hunting prey

Hunting process is followed after recognizing the position of the prey. Assuming that the best search agent knows the position of prey. The other search agents form a group, updating their positions with respect to the best optimal solution. The following equations are proposed to implement the hunting behaviour of hyenas.

$$\vec{D}_h = \left|\vec{B}.\vec{X}_h - \vec{X}_k\right| \tag{1.6}$$

$$\vec{X}_k = \vec{X}_h - \vec{E}.\vec{D}_h \tag{1.7}$$

$$\vec{C}_h = \vec{X}_k + \vec{X}_{k+1} + \cdots + \vec{X}_{k+N} \tag{1.8}$$

where $\vec{X}_h$ is the position of best hyena, $\vec{X}_k$ represents t positions of other hyenas. *N* represents spotted hyenas number in a cluster, computed as

$$N = count_{nos}\left(\vec{X}_h + \vec{X}_{h+1} + \vec{X}_{h+2}, \dots \dots (\vec{X}_h + \vec{M})\right) \tag{1.9}$$

Here $\vec{C}_h$ is a cluster of *N* optimal solutions. Random vector $\vec{M}$ with range of [0.5, 1]. *nos* indicate the number of solutions which are almost similar to the best solution .

1.4.3 Attacking Prey

For attacking the prey, vector $\vec{h}$ is decreased over the iterations from 5 to 0. $\vec{E}$ vector variation is decreased that effects the $\vec{h}$ change.

$$\vec{X}(t+1) = \frac{\vec{C}_h}{N} \tag{1.10}$$

where $X(t + 1)$ save the best solution. According to the best optimal hyena i.e. the best search agent, positions of other search agents i.e. hyenas is updated.

1.4.4 Search for prey

According to the cluster formed by the search agents in vector $\vec{C}_h$, spotted hyenas search for the prey by moving away from each other. The search agents move away from the current targeted prey when $| E | > 1$ to find a better target. . $\vec{B}$ assigns the random weights to prey in exploration phase. On satisfying termination conditions the algorithm is stops.

1.5 Emperor Penguin Optimizer

The emperor penguin optimizer is a recently developed metaheuristic algorithm. The emperor penguins are known for searching and hunting for food altogether forming a group. Emperor penguins use huddling technique to survive in winters. Main mechanism for modelling of the algorithm is mathematically modelled into four phases [22]. These steps are explained in as follows:

1.5.1 Determine the huddle boundary

A polygon shaped huddle grid boundary is formed by emperor penguins. Two neighbour emperor penguins are randomly chosen from huddle for a emperor penguin. Let Φ defines the wind velocity, wind flow factor plays important role to determine boundary. Ψ acts as gradient of Φ. For the polygon plane F is defined as an analytical function

$$F = \nabla\Phi \quad (1.11)$$

Vector Ω is combined with Φ to generate the complex potential.

$$F = \Phi + i\Omega \quad (1.12)$$

where i is a imaginary constant

An emperor penguin located at the center of the formed huddle in the L-shaped polygon region with best fitness is chosen. According to the best fit emperor penguins other emperor penguins update their position .

1.5.2 Compute temperature profile

Basically to maximize the huddle temperature and save energy the emperor penguins form huddle. To mathematical model this, assume R defines radius of polygon and T is temperature profile. Exploration and exploitation depends upon the temperature. The temperature profile T' is computed as follows:

$$T' = \left(T - \frac{Max_{iteration}}{x - Max_{iteration}}\right) \quad (1.13)$$

$$T = \begin{Bmatrix} 0, if\ R > 1 \\ 1, if\ R < 1 \end{Bmatrix}$$

Where $Max_{iteration}$ defines the maximum number of iterations and x indicates the current iteration

1.5.3 Distance between emperor penguins

To update their positions other search agents make use of the current best optimal solution computed as follows:

$$\overrightarrow{D_{ep}} = Abs\left(S(\vec{A}).\overrightarrow{P(x)} - \vec{C}.\overrightarrow{P_{ep}(x)}\right) \tag{1.14}$$

where $\overrightarrow{D_{ep}}$ is distance between best fittest search agent and the emperor penguin , x is the current iteration.$\overrightarrow{A}$ and $\overrightarrow{C}$ responsible for collision avoidance amongst neighbours. $\overrightarrow{P}$ is best optimal solution .

$\overrightarrow{P_{ep}}$ represents the position of search agent. $S()$ is termed as social forces responsible for moving penguins towards best optimal agent. The vectors $\overrightarrow{A}$ and $\overrightarrow{C}$ are mathematical modelling:

$$\overrightarrow{A} = \left(M \times \left(T' + P_{grid}(Accuracy)\right) \times Rand()\right) - T' \tag{1.15}$$

$$P_{grid}(Accuracy) = Abs(\vec{P} - \overrightarrow{P_{ep}}) \tag{1.16}$$

$$\overrightarrow{C} = Rand(\,) \tag{1.17}$$

$P_{grid}(Accuracy)$ indicates the polygon grid and where M maintains appropriate gap between emperor penguins to avoid collision. M is defined as a movement parameter.Assume $2.T'$ as value of movement parameter. Rand() is a random function of [0, 1] range.

$$S(\vec{A}) = \left(\sqrt{f.e^{-x/l} - e^{-x}}\right)^2 \tag{1.18}$$

where f and l are used to explore and exploit. f range is [2, 3] and l range [1.5, 2] and e is the expression function.

1.5.4 Relocate the mover

With respect to the optimal solution, the positions of other search agents are updated. Mathematical model to update positions:

$$\overrightarrow{P_{ep}}(x+1) = \overrightarrow{P(x)} - \vec{A}.\overrightarrow{D_{ep}} \tag{1.19}$$

where $\overrightarrow{P_{ep}}(x+1)$ represents the updated position of search agent. For each iteration the huddling value is calculated again as the mover position is updated.

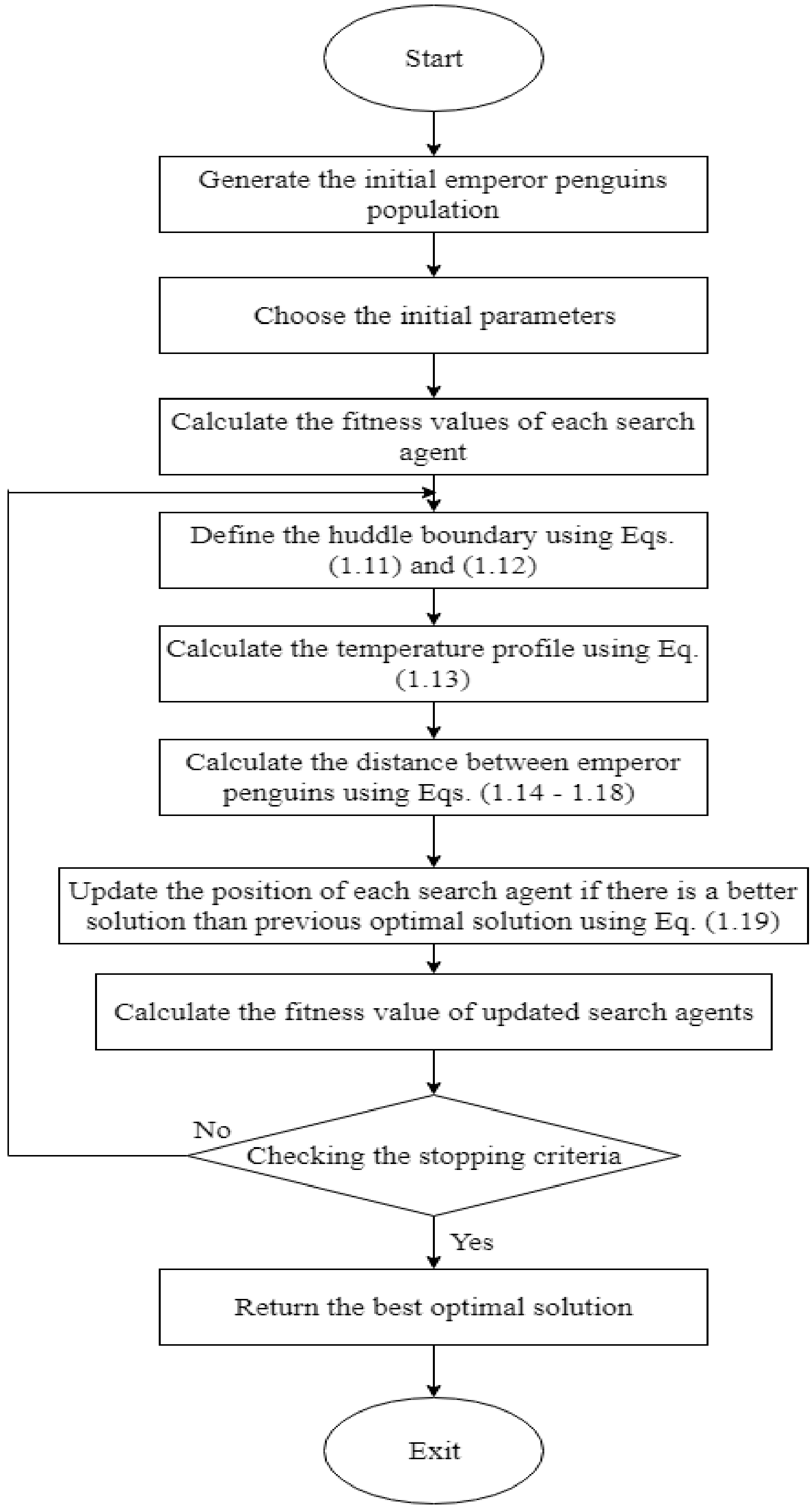


Fig. 1.3. Flowchart of Emperor Penguin Optimizer

1.6 Chaotic theory

Chaos theory is an approach integrated with optimization that has been widely used into various applications. Integration of chaos theory into optimization is a recent technique resulting in effective results [22].The efficiency of a metaheuristic algorithm depends on the utilization of randomization that is controlled by exploration and exploitation phases. Integration of various performance improving approaches with the recently many metaheuristic algorithms have effectively enhanced the exploration phase [22]. Chaos is defined as a bounded nonlinear system with ergodic and stochastic properties. It is very sensitive to the initial condition and the parameters. Many metaheuristic algorithm improved versions, have been designed by integration of chaos.

Integrating of chaos theory into metaheuristic algorithms is an effective approach for improving both exploration and exploitation [24, 25]. The randomness and ergodicity of chaos is a great mechanism in avoiding local search process. A chaotic map improves the convergence ability of algorithms [26]. The chaos theory is used to enhance the searching behaviour, which further improves the efficiency of algorithm. The randomness behaviour of chaos with dynamic property has better performance than the stochastic component of metaheuristic algorithms [27].

Chaos based metaheuristics are divided into two categories. In first category chaos technique is used for controlling parameters rather than random number generators to control parameter value for metaheuristics. In the second category, chaotic search uses chaotic map is incorporated into metaheuristics to enhance the searching behaviour of these algorithms and avoid local optima. Firstly chaotic map is mapped over search space and then chaotic dynamics are used for search purpose. In this process, a chaotic number is generated using selected chaotic map that is used to generate sequences of points [28].

1.6.1 Chaotic Map

A chaotic map is used instead of random number generators to generate random number sequence. In recent times, numbers of chaotic maps are available in literature and most of chaotic has been successfully integrated with metaheuristic algorithms.

Table 1.1 depicts the detail description of chaotic maps. In these chaotic maps, the number lies in the range of (0, 1) or the range of chaotic map can be chosen as the initial value [29].

Therefore, a random number is needed, which can be generated by iterating one step of the chosen chaotic map. A chaotic map is a guide that shows some sort of chaotic behaviour [27].

Table 1.1 Description of chaotic maps

S.no.	Map name	Map equation
1	Logistic map	$x_{t+1} = a.x_t(1-x_t)$
2	Piecewise map	$x_{t+1} = \begin{Bmatrix} \frac{x}{a} & 0 \le x_t \le a \\ \frac{x_t-a}{0.5-a} & a \le x_t \le 0.5 \\ \frac{1-a-x_t}{0.5-a} & 0.5 \le x_t \le 1-a \\ \frac{1-x_t}{a} & 1-a \le x_t \le 1 \end{Bmatrix}$
3	Sine map	$x_{t+1} = \sin(\pi x_t)$
4	Singer map	$x_{t+1} = a(7.86x_t - 23.31x_t^2 + 28.75x_t^{313}.302875x_t^4)$
5	Sinusoidal map	$x_{t+1} = a.x_t^2\sin(\pi x_t)$
6	Tent map	$x_{t+1} = \begin{Bmatrix} {}^{x_t}/_{0.7} & x_t < 0.7 \\ {}^{10}/_{3}(1-x_t) & x_t \ge 0.7 \end{Bmatrix}$
7	Chebyshev map	$x_{t+1} = cos(a.cos^{-1}x_t)$
8	Circle map	$x_{t+1} = x_t + b - \left(\frac{a}{2\pi}\right)\sin(2\pi x_t)\,mod(1)$
9	Gauss map	$x_{t+1} = \begin{Bmatrix} 0, & x_t = 0 \\ ({}^{1}/_{x_t})mod(1) & x_t \ne 0 \end{Bmatrix}$
10	Iterative map	$x_{t+1} = abs\left(sin\left(\frac{a}{x_t}\right)\right), a\epsilon(0,1)$

Recently many of the metaheuristic algorithms have been successfully integrated with chaotic mechanisms. Chaos maps have been introduced into various metaheuristic proposing chaotic grey wolf optimizers [30], chaotic whale optimization algorithm [31]. Chaotic moth flame optimization [33], chaotic gravitational search algorithm [34, 35] chaotic particle swarm optimization 36], chaotic bat algorithm [37], chaotic krill herd [38] and chaotic firefly algorithm [39].The chaos theory shows positive results when integrated with many metaheuristic algorithms [28].

Based on the results obtained chaotic maps have a considerable positive impact on the performance of meta-heuristics. These have shown the effectiveness and efficiency of chaos theory

1.7 Motivation

Motivation for this research work comes from "No free lunch theorem" [21] that state that one single optimizer cannot perform efficiently for all sort of problems. In result many new metaheuristic algorithms are designed another way is to integrate performance improving approaches with metaheuristic algorithm. Integration of chaos theory and metaheuristic algorithms has successfully resulted in improving performance.

Many metaheuristic algorithms face premature convergence and local optima stagnation issues [23]. Spotted hyena optimization and Emperor penguin algorithm are recently developed metaheuristic techniques for solving optimization problems. This research work aim to propose the chaotic version of spotted hyena optimization and emperor penguin algorithm to overcome these issues to improve their performance.

1.8 Thesis Organization

The organization of thesis is structured as follows with outline of each chapter given below:

Chapter 1 Introduction: This chapter gives the basic concept about metaheuristics, classification and characteristics of metaheuristics. The detailed description of SHO and EPO is also presented. Chaos theory is also discussed in this chapter.

Chapter 2 Literature survey: Summarizes metaheuristic algorithms and chaotic technique of few metaheuristic algorithms

Chapter 3 Problem Statement: This chapter describes the problem that is identified and has been chosen in this research. The research gaps and objectives are also mentioned.

Chapter 4 Chaotic Spotted hyena optimizer: Chaotic version of spotted hyena is presented in this chapter. It has been tested on twenty nine well known benchmark functions and constrained engineering design problems.

Chapter 5 Chaotic Emperor penguin optimizer: This chapter contains the proposed algorithm of chaotic emperor penguin optimizer. It has been tested on twenty nine well known benchmark functions and constrained engineering design problems.

Chapter 6 Conclusion and Future Scope: This chapter includes the conclusion and the future scope.

CHAPTER 2

LITERATURE SURVEY

This chapter presents a comprehensive review of metaheuristic and their chaotic version.

2.1 Chaotic Metaheuristic Algorithms

Chaos theory is an approach integrated with optimization that has been widely used into various applications [23]. Randomness property of chaos can be effectively integrated to improve searching capabilities of metaheuristic algorithms .The efficiency of a metaheuristic algorithm depends on the utilization of randomization that is controlled by exploration and exploitation phases. Chaos can enhance the exploration phase [22]. Chaos is defined as a bounded nonlinear system with ergodic and stochastic properties. Many metaheuristic algorithm improved versions, have been designed by integration of chaos. Following is the literature survey of various metaheuristic algorithms hybridized with chaos theory.

Grey Wolf Optimizer is basically inspired from the behaviours of grey wolves. Hunting technique of grey wolves is mathematically model in the algorithm.GWO was proposed by Seyedali Mirjalili in 2014 [5].Where as the chaotic grey wolf optimizer [30] is a chaotic version of grey wolf optimizer proposed by Mehak kohli and Sankalap Arora in 2017.

Whale optimization algorithm is a novel metaheuristic algorithm which is motivated from the bubble net hunting approach of the humpback whales [16].Chaotic whale optimization algorithm [31] was proposed by Gaganpreet Kaur and Sankalap Arora in 2017.

Moth Flame Optimization is a algorithm proposed by Mirjalili in 2015 [32], inspired from the moths navigation technique. Transverse orientation for navigation is the mechanism used by moths which the main motivation for this algorithm. Chaotic version of moth flame optimization [33] was designed by Hossam M. Zawbaa and Eid emery in 2016.

Gravitational Search Algorithm was proposed in 2009 by Rashedi et al. Gravitational search algorithm is physics based algorithm inspired from the law of gravitation and law of motion. Position, inertial mass, active gravitational mass, and passive gravitational mass are working parameters in GSA. Chaotic gravitational search algorithm [34] was proposed in 2017 by Seyedali Mirjalili, Amir H. Gandomi. Another chaotic version of GSA was proposed by

Mittal H., Pal R., Kulhari A. and Saraswat M. is Chaotic Kbest gravitational search algorithm in 2016 [35].

Particle Swarm Optimization was proposed in 1995 [11]. Kennedy and Eberhart proposed algorithm taking inspiration from the behaviour of the swarms in nature like birds that are searching for the food. Particle swarm optimization is a metaheuristic motivated from the social behaviour of birds hunting for food. Chaotic particle swarm optimization a chaos version of PSO was proposed by Bilal Alatas, Erhan Akin, A. Bedri Ozer in 2007[36].

Bat-inspired Algorithm (BA) [17] is inspired by the echolocation behaviour of bats. Bat algorithm was proposed by Yang in 2010. Chaotic bat algorithm [37] was designed by H. Gandomia, Xin-She Yangb in 2013.

The krill herd algorithm designed by Gandomi in 2012 is inspired from the krill herds. Chaotic krill herd[36] by proposed by Saremi, Shahrzad, Mirjalili, Seyed Mohammad, Mirjalili , Seyedali in 2014. In the chaotic KH [38], researchers used chaotic maps in tuning the random vector; it improves the ability of KH to avoid local optimum.

Firefly algorithm (FA) was designed by Yang in 2007 mimics the on the flash patterns and behaviour of fireflies. Chaotic firefly algorithm by Gandomi A.H., Yang X.S., S. Talatahari, Alavi in 2013[40].

CHAPTER 3
PROBLEM STATEMENT

This chapter includes the description of the problem that is identified and that has been chosen in this research. Objectives that are needed to attain for successful completion of work is discussed.

3.1 Problem Statement:

As "No free lunch theorem" [21] that state that one single optimizer cannot perform efficiently for all sort of problems. In result many new metaheuristic algorithms are designed another way is to integrate performance improving approaches with metaheuristic algorithm. Integration of chaos theory and metaheuristic algorithms has successfully resulted in improving performance.

Many metaheuristic algorithms face premature convergence and local optima stagnation issues [23]. Even though recently developed algorithms are designed algorithms have improved searching speed .But the premature convergence and getting stick into a local optimum are still issues in them [23].The efficiency of a metaheuristic algorithm depends on the utilization of randomization that is controlled by exploration and exploitation phases. Integration of chaos theory and metaheuristic algorithms has successfully resulted in improving performance. Chaos can enhance the exploration phase [22]. Many metaheuristic algorithm improved versions, have been designed by integration of chaos.

Spotted hyena optimization and Emperor penguin algorithm are recently developed metaheuristic techniques for solving optimization problems. This research work aim to propose the chaotic version of spotted hyena optimization and emperor penguin algorithm to overcome these issues to improve their performance.

3.2 Objectives

In this work, by integrating chaotic maps into SHO and EPO aim to study the effectiveness of chaos theory for improving the performance.

1. To study and analyze original Spotted hyena optimizer, Emperor penguin optimizer and chaos theory.

2. To develop chaotic version of Spotted hyena optimizer and Emperor penguin optimizer for better exploration and exploitation.

3 To test and validate the proposed algorithms on benchmark test functions

4. To apply the proposed algorithm on constrained engineering design problem.

CHAPTER 4
CHAOTIC SPOTTED HYENA OPTIMIZER

This chapter chaotic version of Spotted hyena optimizer is designed and evaluated on standard benchmark test functions as well as constrained engineering problems for performance evaluation. The main objective is to improve exploration phase and enhance its performance.

4.1 Motivation

Spotted hyena optimization is recently developed metaheuristic technique. Since one single optimizer cannot perform efficiently for all sort of problems. Also many metaheuristic algorithms face premature convergence and local optima stagnation issues [23]. The efficiency of a metaheuristic algorithm depends on the utilization of randomization that is controlled by exploration and exploitation phases. Integration of chaos theory and metaheuristic algorithms has successfully resulted in improving performance.

Chaos can enhance the exploration phase [22]. Many metaheuristic algorithm improved versions, have been designed by integration of chaos. In this research work chaotic version of spotted hyena optimization is proposed.

4.2 Proposed algorithm

Chaotic SHO algorithm is designed by integrating chaotic theory in original spotted hyena optimizer. In the proposed algorithm, parameter h is adjusted using chaotic maps

In the first step the population of spotted hyenas is initialized. Next a chaotic map is selected with first chaotic number and a variable. Initial parameters h, B, E, N are initialized which are same as in SHO. Also, the chaotic number of the chaotic map is initialized to adjust the parameter 'h'.Calculate fitness for all spotted hyenas i.e. search agents and obtain best search agent. Define the group of optimal solutions, i.e. cluster using Eqs. (1.8) and (1.9). Update the positions of search agents using Eq. (1.10) .Calculate fitness value for updated agents and replace f value is better than previous value and update cluster. Parameter h keeps on updating with course of iterations.

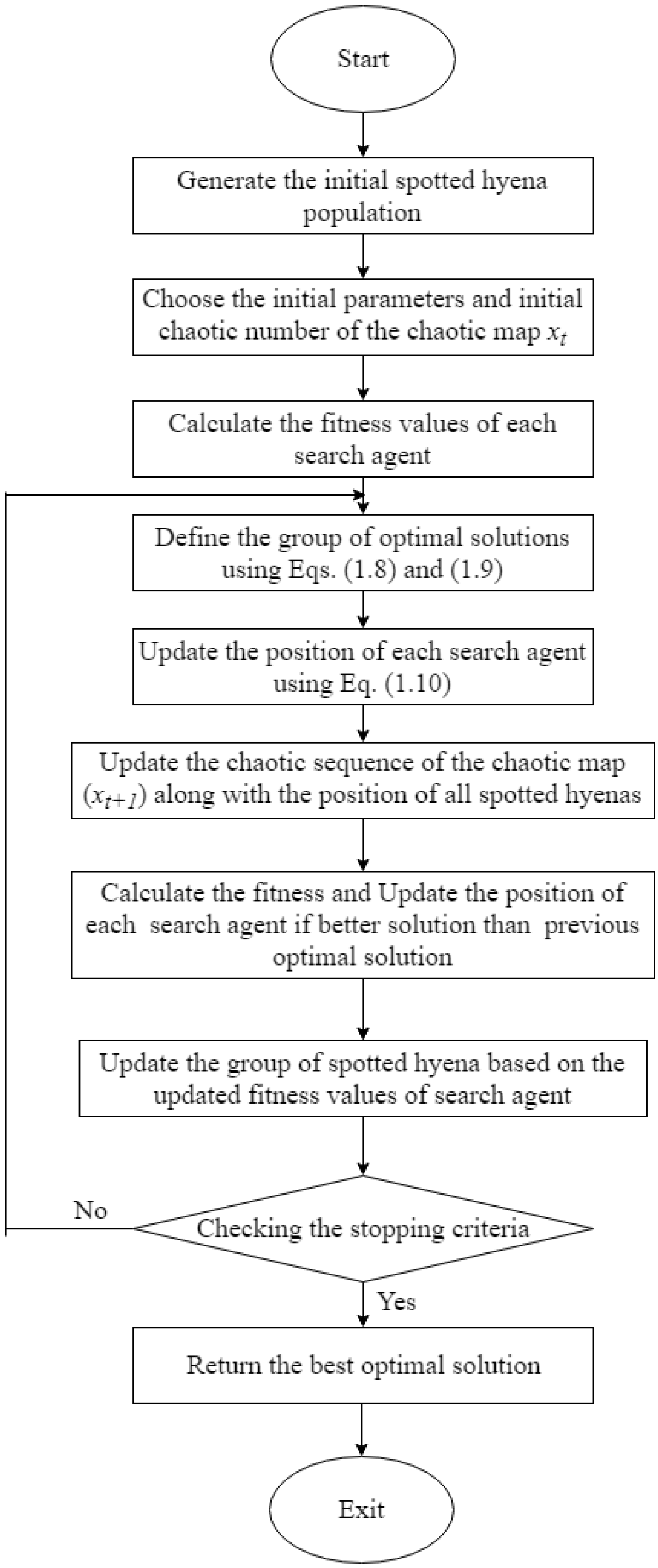


Fig. 4.1. Flowchart of proposed Chaotic SHO

4.3 Experimental Results on Benchmark Test Functions

Twenty nine standard benchmark test functions are used to test and evaluate proposed algorithm. These benchmark functions are detailed in Appendix A. The ten different chaotic maps are evaluated on benchmark functions and the most efficient map is chosen for further evaluation. The results are compared with original SHO as well as GWO, PSO, GSA and MFO.

4.3.1 Benchmark test functions

Benchmark test functions are categorized in following categories: first category is unimodal test functions [41], second is multimodal test functions [39], third category is fixed-dimension multimodal [39, 41] and last is composite functions [42, 43]. Benchmark test functions are described in Appendix A. Dimension and Range of the test functions are indicates in the appendix A. Table A.1 describes the unimodal $(F1 - F7)$. The multimodal categories $(F8 - F16)$ and fixed-dimension modal $(F14 - F23)$ are represented in Tables A.2 and A.3 respectively. Composite test functions $(F24 - F29)$. are detailed in A.4

4.3.2 Experimental setup

CSHO and other metaheuristic algorithms for comparison purpose parameters are described in Appendix C. The average and standard deviation are the metrics based on which comparison is made by obtaining optimal solution. Proposed algorithm is run for 30 independent runs with1000 iterations.

Table 4.1 Parameters setting of involved algorithms

Algorithms	**Parameters**	**Values**
Chaotic Spotted Hyena Optimizer(CSHO)	Search Agents	30
	Control Parameter($\vec{h}$)	[5,0]
	$\overrightarrow{M}$ Constant	[0.5,1]
	Number of Generations	1000
Spotted Hyena Optimizer (SHO)	Search Agents	30
	Control Parameter($\vec{h}$)	[5,0]
	$\vec{M}$ Constant	[0.5,1]
	Number of Generations	1000

Grey Wolf Optimizer (GWO)	Search Agents	30
	Control Parameter($\vec{a}$)	[2,0]
	Number of Generations	1000
Particle Swarm Optimization(PSO)	Number of Particles	30
	Inertia Coefficient	.75
	Cognitive and Social Coeff	1.8,2
	Number of Generations	1000
Gravitational Search Algorithm(GSA)	Search Agents	30
	Gravitational Constant	100
	Alpha Coefficient	20
	Number of Generations	1000
Moth-Flame Optimization(MFO)	Convergence Constant	[-1,-2]
	Logarithmic Spiral	.75
	Number of Generations	1000

4.3.3 Performance evaluation

Proposed algorithm uses ten different chaotic maps for evaluation on benchmark functions as shown in Tables 4.2-4.5. Many chaotic maps are able to enhance the performance of SHO algorithm successfully such as sine, singer, tent and logistic. However amongst all tent map shows much improved results than the others and offers competitive results with SHO.

Table 4.6 shows that CSHO evaluation on unimodal functions with other algorithms namely SHO, GWO, PSO, GSA, and MFO. CSHO is the most efficient algorithm for functions $F1, F2, F3, F4, F5$ and $F7$ with the best optimal solution

Multimodal benchmark functions results are shown in Tables 4.7, while Table 4.8 for fixed-dimension multimodal functions .According to these tables , CSHO shows effective results in $F9, F11, F15, F16, F17, F18, F20$ and $F22$. CSHO algorithm shows enhance results.

Composite benchmark functions results are shown in Table 4.9 .CSHO algorithm shows better results for functions $F26, F28$ and $F29$.

Table 4.2 Results obtained on chaotic maps for unimodal benchmark functions on CSHO

	F1	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**
SHO	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	6.61E-12 **(6.98E-12)**	8.73E+00 (5.95E-01)	2.66E-01 (1.89E-01)	3.50E-05 (2.45E-05)
CSHO-1 (Log)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	6.81E-11 (1.08E-11)	8.41E+00 (5.32E-01)	2.65E-01 (2.75E-01)	3.67E-05 (3.53E-05)
CSHO-2 (Pie)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	5.22E-11 (1.62E-11)	8.73E+00 (3.34E-01)	4.76E-01 (2.58E-01)	4.83E-05 (6.99E-05)
CSHO-3 (Sine)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	7.02E-12 (7.03E-12)	8.97E+00 (3.36E-01)	5.18E-01 (2.46E-01)	4.10E-05 (3.66E-05)
CSHO-4 (Singer)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	6.19E-11 (3.71E-11)	8.75E+00 (5.00E-01)	1.18E-01 (2.32E-01)	4.84E-05 (3.83E-05)
CSHO-5 (Sinu)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	6.36E-11 (9.95E-11)	8.84E+00 (4.98E-01)	2.19E-01 ()2.58E-01	4.05E-05 (4.29E-05)
CSHO-6 (Tent)	**0.00E+00** (0.00E+00)	**0.00E+00** (0.00E+00)	**0.00E+00** (0.00E+00)	**6.65E-12** (7.46E-12)	**8.41E+00** **(3.03E-01)**	2.14E-01 (**1.99E-01**)	**3.40E-05** (**1.39E-05**)
CSHO-7 (Che)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	8.34E-11 (9.72E-12)	8.79E+00 (5.40E-01)	9.62E-01 (2.59E-01)	4.32E-05 (4.60E-05)
CSHO-8 (Cir)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	7.49E-12 (8.88E-12)	8.83E+00 (4.22E-01)	**1.11E-01** (2.39E-01)	3.60E-05 (3.94E-05)
CSHO-9 (Gau)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	9.74E-11 (1.03E-11)	8.84E+00 (3.55E-01)	9.55E-01 (2.72E-01)	3.43E-05 (2.78E-05)
CSHO-10 (Icm)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	9.97E-12 (8.77E-12)	8.65E+00 (4.79E-01)	7.15E-01 (2.38E-01)	4.00E-05 (4.58E-05)

Table 4.3 Results obtained on chaotic maps for multimodal benchmark functions on CSHO

	F8	**F9**	**F10**	**F11**	**F12**	**F13**
SHO	**-1.11E+03** **(2.89E+02)**	0.00E+00 (0.00E+00)	2.33E+00 (1.49E+00)	0.00+00 (0.00+00)	3.85E-02 (2.42E-02)	9.99E-01 (9.63E-02)
CSHO-1 (Log)	-2.13E+03 (5.03E+02)	0.00E+00 (0.00E+00)	1.37E+00 (5.15E+00)	0.00+00 (0.00+00)	1.33E-02 (4.32E-02)	9.96E-01 (8.55E-02)
CSHO-2 (Pie)	-2.17E+03 (5.80E+02)	9.63E-01 (3.29E+00)	2.20E+00 (7.61E+00)	2.05E-03 (5.39E-03)	1.26E-02 (1.80E-02)	9.95E-01 (1.17E-01)
CSHO-3 (Sine)	-2.42E+03 (4.86E+02)	0.00E+00 (0.00E+00)	1.99E+00 (9.06E+00)	2.42E-02 (6.29E-02)	1.18E-02 (3.96E-02)	**8.92E-01** (**6.05E-02**)
CSHO-4 (Singer)	-2.14E+03 (6.56E+02)	0.00E+00 (0.00E+00)	5.35E+00 (**1.41E+00**)	2.31E-02 (6.21E-02)	1.07E-02 (3.39E-02)	9.97E-01 (5.38E-02)
CSHO-5 (Sinu)	-2.42E+03 (6.01E+02)	1.41E+00 (7.74E+00)	4.92E+00 (2.67E+00)	2.16E-03 (5.63E-03)	1.30E-02 (3.24E-02)	9.94E-01 (1.00E-01)
CSHO-6 (Tent)	-2.17E+03 (4.32E+02)	**0.00E+00** (**0.00E+00**)	**1.51E+00** (2.43E+00)	**0.00+00** (**0.00+00**)	1.19E-02 (2.86E-02)	9.94E-01 (9.14E-02)
CSHO-7 (Che)	-2.08E+03 (5.01E+02)	1.04E+00 (3.99E+00)	3.57E+00 (1.62E+00)	9.48E-02 (3.60E-02)	1.11E-02 (2.86E-02)	9.92E-01 (1.07E-01)
CSHO-8 (Cir)	-2.19E+03 (5.35E+02)	0.00E+00 (0.00E+00)	1.06E+00 (4.02E+00)	2.17E-01 (5.69E-02)	1.55E-02 (2.17E-02)	9.94E-01 (8.71E-02)
CSHO-9 (Gau)	-1.97E+03 (4.50E+02)	1.04E+00 (5.07E+00)	1.97E+00 (6.52E+00)	0.00+00 (0.00+00)	**1.05E-02** (**1.13E-02**)	9.94E-01 (9.63E-02)
CSHO-10 (Icm)	-2.33E+03 (6.17E+02)	0.00E+00 (0.00E+00)	9.78E+00 (5.11E+00)	1.45E-02 (4.45E-02)	1.25E-02 (6.51E-01)	9.97E-01 (6.84E-02)

Table 4.4 Results obtained on chaotic maps for fixed-dimension multimodal benchmark functions on CSHO

	F14	F15	F16	F17	F18	F19	F20	F21	F22	F23
SHO	9.70E+00	9.38E-04	-1.06E+01	3.97E-01	3.00E+00	-3.37E+00	-1.39E+00	**-3.80E+00**	1.64E+01	-1.73E+00
	(2.71E+00)	(1.17E-04)	(2.22E-11)	(2.45E-01)	(9.09E+00)	(4.33E-01)	(6.50E-01)	(3.81E-01)	(2.43E-04)	(2.70E-01)
CSHO-1	9.27E+00	9.14E-04	-1.06E+01	3.97E-01	3.00E+00	-3.37E+00	-1.72E+00	-4.64E+00	-1.90E+01	-1.84E+00
(Log)	(3.76E+00)	(1.02E-04)	(2.45E-11)	(3.25E-01)	(8.96E+00)	(3.97E-01)	(6.07E-01)	(3.56E-01)	(2.80E-04)	(2.46E-01)
CSHO-2	9.23E+00	9.18E-04	-1.03E+01	3.98E-01	3.00E+00	-3.39E+00	-1.41E+00	-5.48E+00	-1.84E+01	-1.93E+00
(Pie)	(4.03E+00)	(1.69E-04)	(3.74E-11)	(2.78E-01)	(9.66E+00)	(3.31E-01)	(6.42E-01)	(3.54E-01)	(3.46E-04)	(3.17E-01)
CSHO-3	9.41E+00	9.47E-04	-1.03E+01	3.98E-01	3.00E+00	-3.39E+00	-1.65E+00	-5.20E+00	-1.82E+01	-1.92E+00
(Sine)	(4.62E+00)	(1.38E-03)	(2.16E-11)	(3.13E-01)	(9.04E+00)	(3.35E-01)	(6.63E-01)	(3.55E-01)	(2.61E-04)	(2.06E-01)
CSHO-4	9.40E+00	9.68E-04	-1.03E+01	3.98E-01	3.00E+00	-3.39E+00	-1.57+00	-5.69E+00	-1.69E+01	-1.81E+00
(Singer)	(4.14E+00)	(1.09E-04)	(3.40E-11)	(3.01E-01)	(9.08E+00)	(3.32E-01)	(6.65E-01)	(3.96E-01)	(2.34E-04)	(2.51E-01)
CSHO-5	9.17E+00	9.39E-04	-1.03E+01	3.97E-01	3.00E+00	-3.38E+00	-1.57E+00	-5.20E+00	-1.73E+01	-1.87E+00
(Sinu)	(3.99E+00)	(1.52E-03)	(3.40E-11)	(4.53E-01)	(9.71E+00)	(3.89E-01)	(6.43E-01)	(3.74E-01)	(3.98E-04)	(5.43E-01)
CSHO-6	9.70E+00	**9.18E-04**	**-1.06E+01**	**3.97E-01**	**3.00E+00**	**-3.37E+00**	**-3.67E+00**	-3.89E+00	**-1.67E+01**	-1.66E+00
(Tent)	**(1.66E+00)**	**(1.04E-04)**	**(2.12E-11)**	**(2.31E-01)**	**(8.04E+00)**	**(3.13E-01)**	**(5.33E-01)**	**(3.34E-01)**	**(2.15E-04)**	(3.06E-01)
CSHO-7	9.25E+00	8.46E-03	-1.03E+01	3.98E-01	3.00E+00	-3.38E+00	-1.51E+00	-5.55E+00	-1.70E+01	**-1.66E+00**
(Che)	(4.33E+00)	(1.95E-03)	(3.48E-11)	(4.08E-01)	(9.95E+00)	(3.92E-01)	(6.34E-01)	(3.76E-01)	(5.37E-04)	**(1.25E-01)**
CSHO-8	9.83E+00	9.77E-04	-1.03E+01	3.99E-01	3.00E+00	-3.41E+00	-1.64E+00	-4.77E+00	-1.93E+01	-1.79E+00
(Cir)	(4.30E+00)	(1.39E-04)	(3.18E-11)	(4.49E-01)	(9.75E+00)	(3.67E-01)	(6.64E-01)	(3.46E-01)	(3.69E-04)	(2.25E-01)
CSHO-9	9.36E+00	7.48E-03	-1.03E+01	3.98E-01	3.00E+00	-3.37E+00	-1.49E+00	-5.17E+00	-1.98E+01	-1.87E+00
(Gau)	(4.31E+00)	(1.68E-03)	(3.21E-11)	(4.31E-01)	(9.07E+00)	(4.74E-01)	(6.79E-01)	(3.88E-01)	(4.14E-04)	(2.50E-01)
CSHO-	**9.03E+00**	9.47E-04	-1.03E+01	3.98E-01	3.00E+00	-3.41E+00	-1.28E+00	-6.06E+00	-1.97E-01	-1.82E-01
10 (Icm)	(4.10E+00)	(1.14E-04)	(3.79E-11)	(4.46E-01)	(9.97E+00)	(3.47E-01)	(6.79E-01)	(3.80E-01)	(2.93E-04)	(2.91E-01)

Table 4.5 Results obtained on chaotic maps for composite benchmark functions on CSHO

	F24	F25	F26	F27	F28	F29
SHO	2.33E+02 (1.39E+02)	**4.10E+02** (9.43E+01)	3.39E+02 (3.12E+01)	7.19E+02 (1.10E+02)	1.08E+02 (1.38E+01)	5.91E+01 (4.99E+00)
CSHO-1 (Log)	2.33E+02 (1.38E+02)	1.02E+03 (9.37E+01)	3.76E+02 (3.18E+01)	**7.11E+02** (**1.06E+02**)	1.28E+02 (1.48E+01)	5.88E+01 (4.79E+00)
CSHO-2 (Pie)	2.43E+02 (1.41E+02)	4.25E+02 (9.42E+01)	3.45E+02 (3.21E+01)	7.23E+02 (1.32E+02)	1.36E+02 (2.44E+01)	5.99E+01 (4.77E+00)
CSHO-3 (Sine)	2.50E+02 (1.36E+02)	4.21E+02 (9.41E+01)	3.48E+02 (3.16E+01)	7.26E+02 (1.23E+02)	1.25E+02 (1.67E+01)	5.96E+01 (4.67E+00)
CSHO-4 (Singer)	**2.33E+02** (**1.28E+02**)	1.34E+03 (9.39E+01)	3.51E+02 (3.19E+01)	8.02E+02 (1.24E+02)	1.18E+02 (2.89E+01)	5.97E+01 (4.98E+00)
CSHO-5 (Sinu)	2.47E+02 (1.29E+02)	4.11E+02 (9.54E+01)	3.64E+02 (3.24E+01)	7.54E+02 (1.76E+02)	1.15E+02 (1.54E+01)	5.92E+01 (5.01E+00)
CSHO-6 (Tent)	2.36E+02 (1.34E+02)	6.16E+02 (**1.34E+01**)	**3.39E+02** (**3.08E+01**)	7.31E+02 (1.13E+02)	**1.08E+02** **(1.32E+01)**	**5.87E+01** **(4.63E+00)**
CSHO-7 (Che)	2.39E+02 (1.44E+02)	4.69E+02 (9.47E+01)	3.40E+02 (3.26E+01)	7.67E+02 (1.56E+02)	1.09E+02 (1.76E+01)	5.95E+01 (4.99E+00)
CSHO-8 (Cir)	2.37E+02 (1.76E+02)	4.25E+02 (9.67E+01)	3.98E+02 (3.34E+01)	7.23E+02 (1.33E+02)	1.33E+02 (1.89E+01)	5.98E+01 (4.67E+00)
CSHO-9 (Gau)	2.33E+02 (1.38E+02)	1.47E+03 (9.32E+01)	3.45E+02 (3.43E+01)	7.26E+02 (1.56E+02)	1.19E+02 (1.36E+01)	5.97E+01 (4.92E+00)
CSHO-10 (Icm)	2.45E+02 (1.83E+02)	4.20E+02 (9.44E+01)	3.58E+02 (3.21E+01)	7.22E+02 (1.28E+02)	1.13E+02 (1.78E+01)	5.88E+01 (4.65E+00)

Table 4.6 Results obtained from CSHO on unimodal benchmark functions

	CSHO-6	SHO	GWO	PSO	GSA	MFO
F1	**0.00E+00** **(0.00E+00)**	0.00E+00 (0.00E+00)	1.79E-58 (1.98E-58)	3.38E-08 (4.61E-08)	1.23E-16 (9.72E-19)	2.77E-04 (2.99E-04)
F2	**0.00E+00** **(0.00E+00)**	0.00E+00 (0.00E+00)	8.46E-35 (7.18E-35)	3.94E-04 (8.47E-04)	5.62E-01 (8.50E-01)	3.57E+01 (2.52E+01)
F3	**0.00E+00** **(0.00E+00)**	0.00E+00 (0.00E+00)	8.88E-16 (1.54E-15)	1.72E+01 (8.82E+00)	4.57E+02 (1.46E+02)	1.49E+04 (1.18E+04)
F4	**6.65E-15** (6.16E-12)	6.61E-12 (6.98E-12)	1.38E-14 **(1.38E-14)**	6.53E-01 (1.60E-01)	1.29E+00 (1.07E+00)	6.67E+01 (9.3E+00)
F5	**8.41E+00** **(3.03E-01)**	8.73E+00 (5.95E-01)	2.68E+01 (6.37E-01)	7.25E+01 (5.18E+01)	4.52E+01 (5.01E+01)	2.67E+04 **(**3.45E+05)
F6	2.14E-01 (1.99E-01)	2.66E-01 (1.89E-01)	6.22E-01 (3.05E-01)	3.16E-08 (1.43E-07)	**1.04E-16** **(4.47E-17)**	1.32E-03 (3.44E-01)
F7	**3.40E-05** **(1.39E-05)**	3.50E-05 (2.45E-05)	7.99E-04 (3.70E-04)	7.36E-02 (2.99E-02)	5.07E-02 (1.71E-02)	3.68E-01 (1.05E-02)

Table 4.7 Results obtained from CSHO on multimodal benchmark functions

	CSHO-6	SHO	GWO	PSO	GSA	MFO
F8	-2.17E+03 (4.32E+02)	-1.11E+03 **(2.89E+02)**	-6.08E+03 (5.81E+02)	-6.60E+03 (1.04E+03)	-2.54E+03 (3.84E+02)	**-8.64E+03** (9.36E+02)
F9	**0.00E+00** **(0.00E+00)**	0.00E+00 (0.00E+00)	2.59E-01 (8.69E-01)	4.79E+01 (9.88E+00)	2.57E+01 (6.83E+00)	1.60E+02 (3.25E+01)
F10	1.51E+00 (2.43E+00)	2.33E+00 (1.49E+00)	**1.66E-14** **(3.32E-15)**	2.34E-04 (8.10E-04)	7.61E-09 (1.50E-09)	1.49E+01 (7.55E+00)
F11	**0.00+00** **(0.00+00**)	0.00+00 (0.00+00)	1.06E-03 (4.50E-03)	1.23E-02 (1.24E-02)	8.63E+00 (3.10E+00)	5.40E-02 (1.05E-01)
F12	1.19E-02 (2.86E-02)	3.85E-02 (2.42E-02)	4.46E-02 (1.94E-02)	**2.91E-10** **(2.63E-10)**	1.92E-01 (2.74E-01)	1.03E+00 (1.21E+00)
F13	9.94E-01 (9.14E-02)	9.99E-01 (9.63E-02)	4.94E-01 (2.13E-01)	4.36E-03 (6.09E-03)	**5.33E-32** **(6.58E-32)**	6.32E-01 (1.42E+00)

Table 4.8 Results obtained from CSHO on fixed dimension multimodal benchmark functions

	CSHO-6	SHO	GWO	PSO	GSA	MFO
F14	9.70E+00 **(1.66E+00)**	9.70E+00 (2.71E+00)	4.39E+00 (3.99E+00)	2.70E+00 (2.28E+00)	3.57E+00 (2.31E+00)	**2.21E+00** (1.86E+00)
F15	**9.184E-04** (**1.04E-04**)	9.38E-04 (1.17E-04)	3.04E-03 (6.91E-03)	9.92E-04 (2.11E-04)	6.05E-03 (6.76E-04)	1.95E-03 (3.67E-04)
F16	**-1.06E+01** (**2.12E-11**)	-1.06E+01 (2.22E-11)	-1.03E+00 (7.55E-09)	-1.03E+00 (6.71E-16)	-1.03E+00 (1.87E+00)	-1.03E+00 (1.77E+00)
F17	**3.97E-01** (**2.31E-01**)	3.97E-01 (2.45E-01)	3.97E-01 (7.38E-06)	3.97E-01 (8.91E-17)	3.97E-01 (1.54E-15)	3.97E-01 (1.78E-16)
F18	**3.00E+00** (8.04E+00)	3.00E+00 (9.09E+00)	3.00E+00 (1.47E-05)	3.00E+00 (6.85E-06)	3.00E+00 (2.70E-02)	3.00E+00 **(4.65E-15)**
F19	**-3.90E+00** (3.13E-01)	-3.37E+00 (4.33E-01)	-3.86E+00 (2.28E-03)	-3.89E+00 (**2.71E-15**)	-3.86E+00 (4.41E-01)	-3.86E+00 (3.43E-15)
F20	**-3.67E+00** (**.33E-01**)	-1.39E+00 (6.50E-01)	-3.25E+00 (8.65E-02)	-3.26E+00 (6.03E-02)	-1.32E+00 (5.45E-01)	-3.23E+00 (6.73E-02)
F21	-3.89E+00 **(1.34E-01)**	-3.80E+00 (3.81E-01)	**-9.30E+00** (1.92E+00)	-7.68E+00 (3.15E+00)	-4.99E+00 (1.43E+00)	-6.30E+00 (3.13E+00)
F22	**-1.67E+01** (**2.15E-04**)	-1.64E+01 (2.43E-04)	-1.04E+01 (2.46E-04)	-8.61E+00 (2.83E+00)	-6.02E+00 (2.42E+00)	-7.39E+00 (3.57E+00)
F23	-1.66E+00 (3.06E-01)	-1.73E+00 (2.70E-01)	**-1.00E+01** (**1.75E-04**)	-9.56E+00 (2.24E+00)	-9.05E+00 (2.42E+00)	-7.36E+00 (3.72E+00)

Table 4.9 Results obtained from CSHO on composite benchmark functions

	CSHO-6	SHO	GWO	PSO	GSA	MFO
F24	2.36E+02 (1.34E+02)	2.33E+02 (1.39E+02)	8.35E+01 (8.45E+01)	6.03E+01 (8.89E+01)	**4.32E-17** (**3.01E-17**)	1.17E+02 (7.39E+01)
F25	6.16E+02 (**1.34E+01**)	4.10E+02 (9.43E+01)	1.43E+02 (4.34E+01)	2.49E+02 (1.72E+02)	2.05E+02 (4.45E+02)	**9.18E+01** (1.46E+01)
F26	**3.39E+02** (**3.08E+01**)	3.39E+02 (3.12E+01)	3.76E+02 (5.42E+01)	3.39E+02 (8.56E+01)	3.78E+02 (8.89E+01)	4.11E+02 (1.19E+02)
F27	7.31E+02 (1.13E+02)	7.19E+02 (1.10E+02)	4.76E+02 (1.11E+02)	4.51E+02 (1.37E+02)	5.45E+02 (1.11E+02)	**3.39E+02** (**2.15E+01**)
F28	**1.08E+02** (**1.32E+01**)	1.08E+02 (1.38E+01)	1.38E+02 (2.11E+02)	2.43E+02 (4.23E+02)	1.42E+02 (1.33E+02)	1.15E+02 (9.32E+01)
F29	**5.87E+01** (**4.63E+00**)	5.91E+01 (4.99E+00)	8.20E+02 (1.86E+02)	8.26E+02 (1.76E+02)	8.21E+02 (4.56E+02)	8.82E+02 (2.43E+01)

4.3.3.1 Convergence analysis

The convergence curves analysis of CSHO, SHO, GWO, PSO, GSA and MFO are shown in Fig. 4.2.Convergence analysis is made on test functions $F3, F9, F15$ and $F26$. CSHO provides better results than other algorithms.

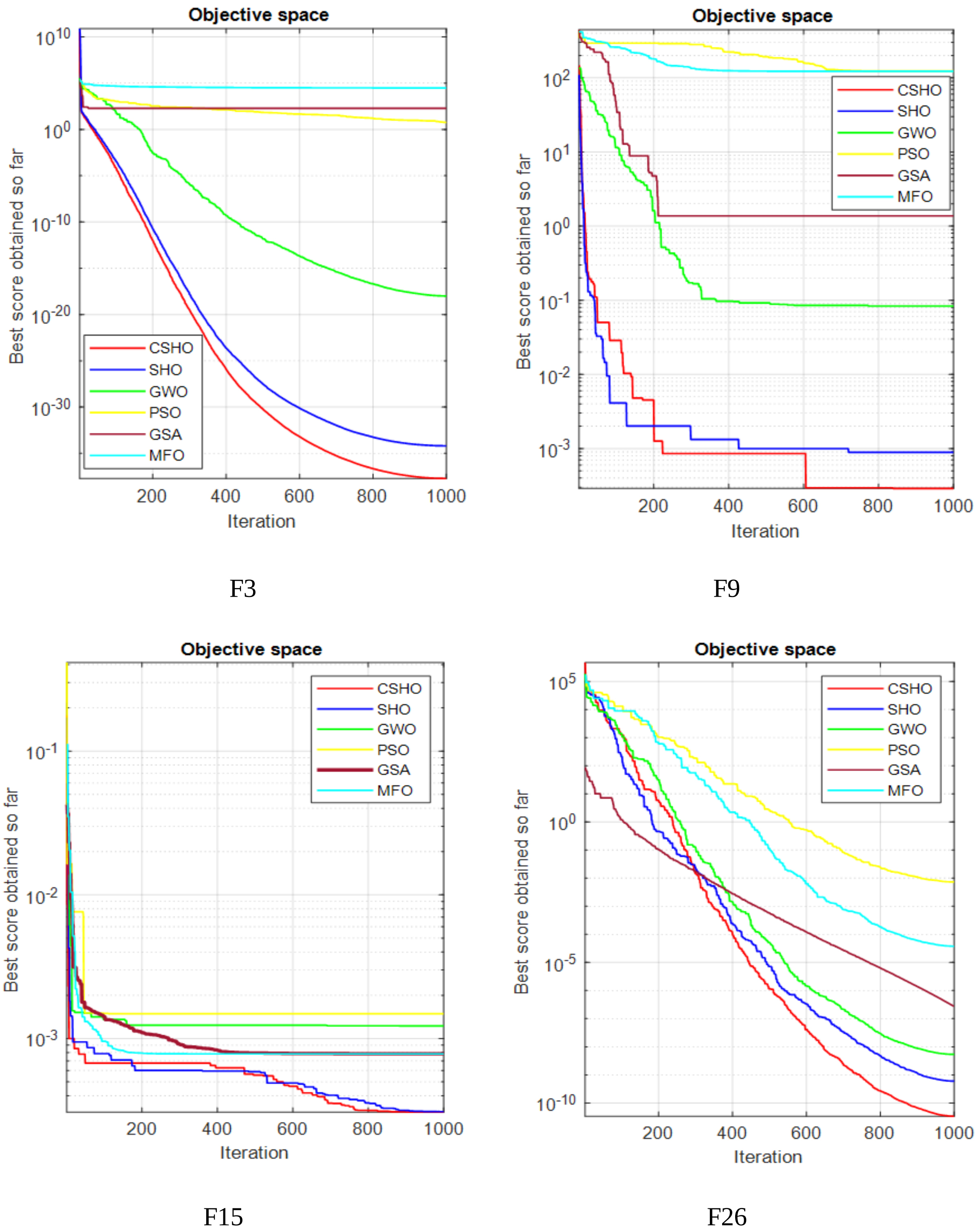
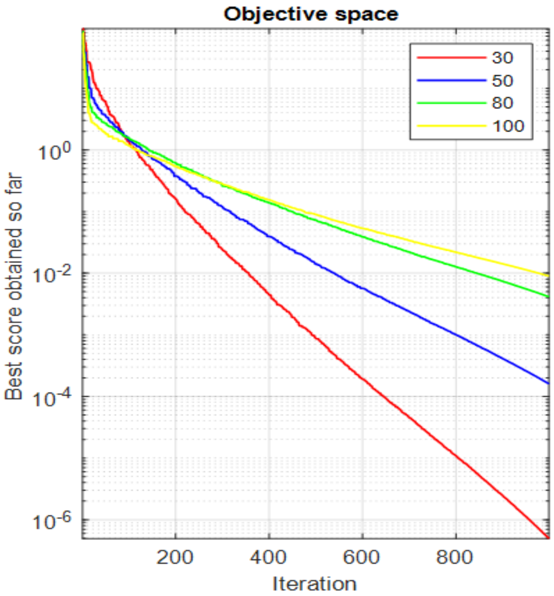


Fig. 4.2. Convergence analysis on CSHO

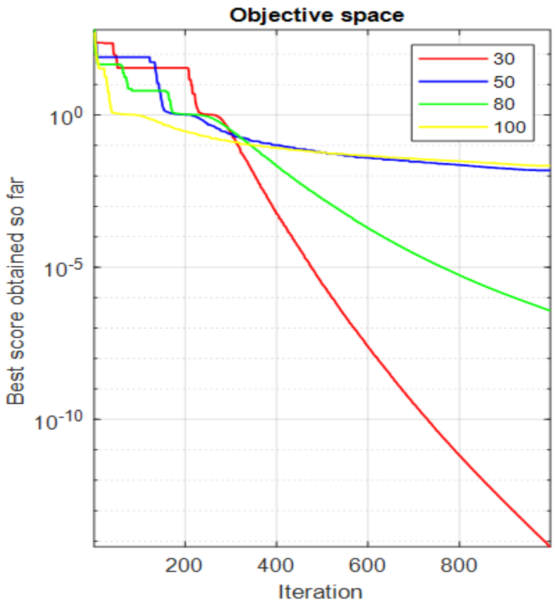
4.3.3.2 Sensitivity study

Sensitivity of proposed algorithm is discussed by varying number of search agents parameter and maximum number of iterations respectively ,while keeping other rest parameters fixed.

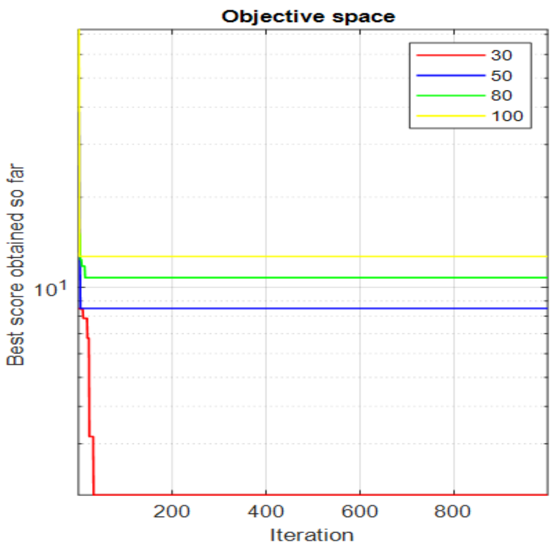
- Number of search agents: Number of search agents values taken as 30, 50, 80, and 100. In test functions F5,F11,F14,F26 as in Fig. 4.3 CSHO shows optimal solutions for value set to 30 for search agents



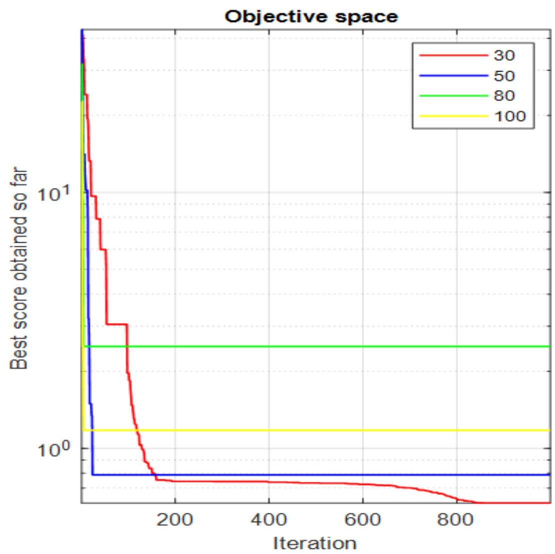
F5



F11



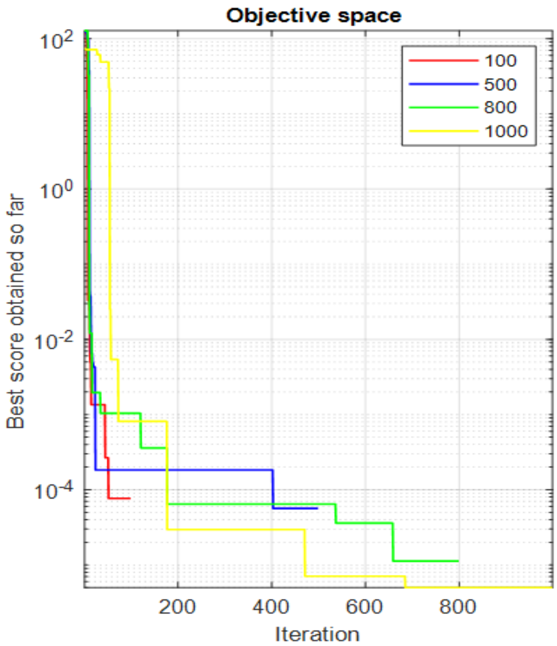
F14



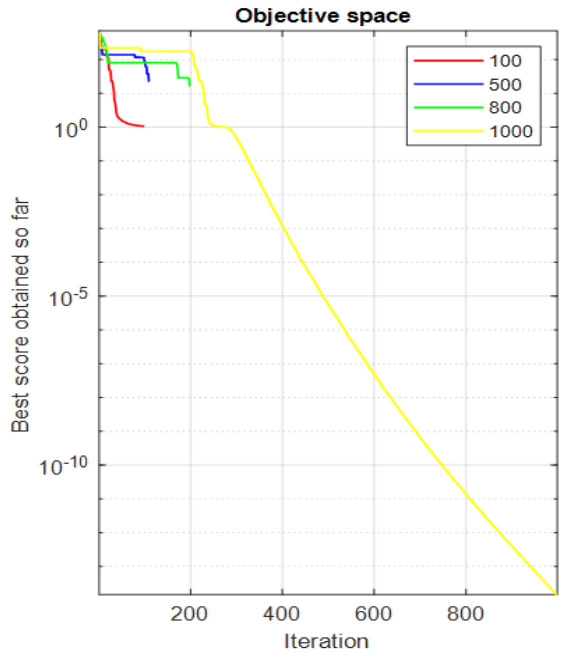
F26

Fig. 4.3. Sensitivity analysis of search agents on CSHO

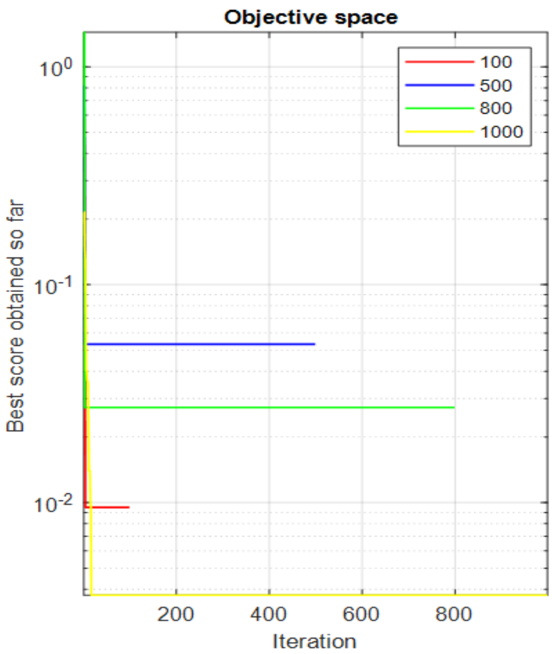
- Maximum number of iterations: 100, 500, 800, and 1000 are varied number of iterations. Fig. 4.4 CSHO shows best results for F7, F11, F15 and F26 test functions.



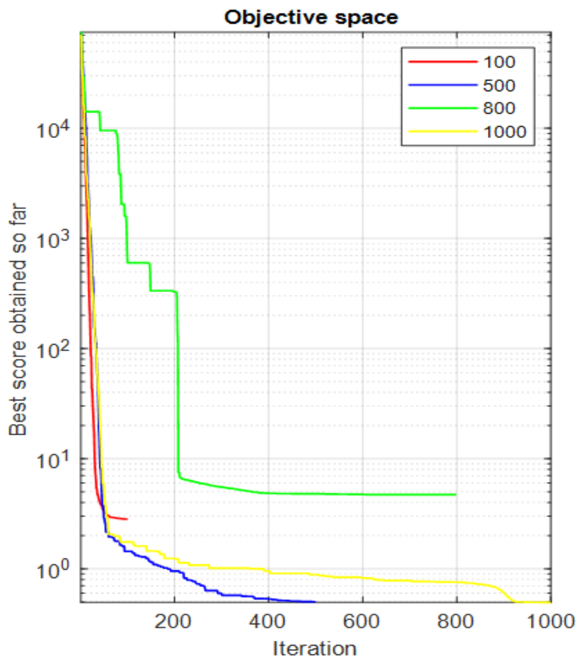
F7



F11



F15



F26

Fig. 4.4. Sensitivity analysis of number of iterations on CSHO

4.3.3.3 Scalability study

This section describes the effect of scalability on various benchmark test functions. The dimensionality of the test functions are changed from 30 to 50, 50 to 80, and 80 to100 are different dimension set on benchmark functions F3, F9, F17 andF26. Fig. 4.5 shows the performance of proposed CSHO on dimensionality change. Proposed algorithm shows different results with varied dimensions.

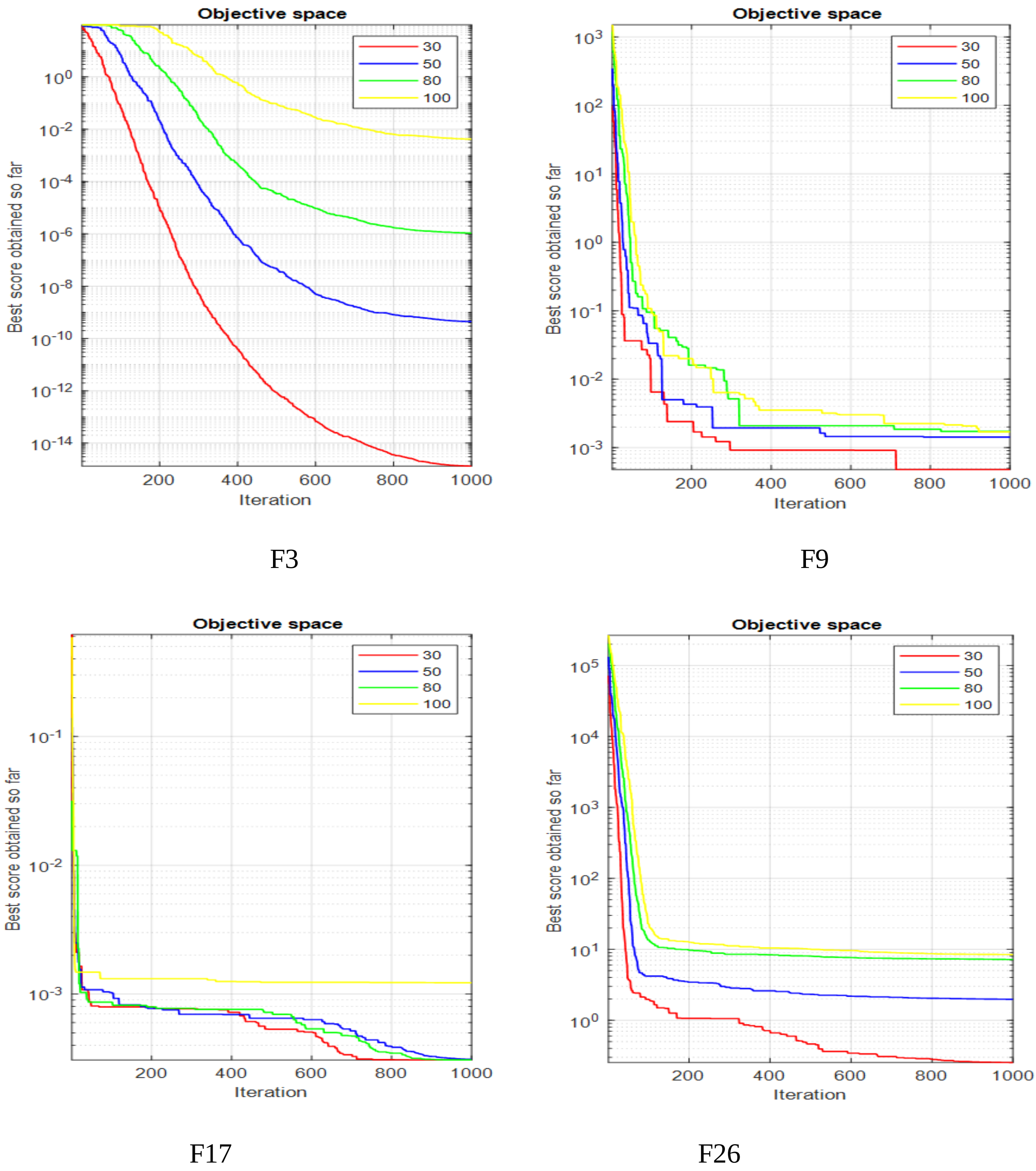


Fig. 4.5. Scalability analysis on CSHO

4.4 Engineering Design Problems

In this section, the performance of proposed algorithms is discussed by evaluating with constrained engineering problems. Constraints represent a feasible region which is nonempty and is filled with some restrictions or constraints to be followed by the solutions to solve a specific optimization problem [44]. CSHO, SHO, GWO, PSO, MFO, and GSA are metaheuristic algorithms used for comparison purpose.

4.4.1 Welded beam design

Welding beam design is a problem which aims to minimize the fabricating cost of the beam as shown in Fig. 4.6 [44]. Table 4.10 and Table 4.11. shows the results , CSHO algorithm obtains better solution than other metaheuristic algorithms.

The optimization constraints are:

- shear stress (τ)
- bending stress (θ) in the beam
- buckling load ($P c$) on the bar
- end deflection (δ) of the beam.

There are four optimization variables of this problem which are as follows:

- Thickness of weld (h)
- Length of the clamped bar (l)
- Height of the bar (t)
- Thickness of the bar (b)

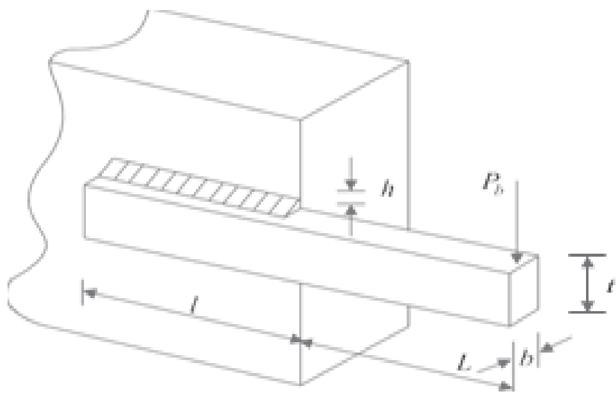


Fig. 4.6. Welded beam problem schematic view

Table 4.10 Comparison results for welded beam design problem on CSHO

Algorithm	Optimum Variables				Optimum Cost
	H	***T***	***l***	***b***	
CSHO	0.205312	3.474799	9.035791	0.205799	**1.725662**
SHO	0.205619	3.474856	9.035794	0.205823	1.725669
GWO	0.205673	3.475401	9.036978	0.206243	1.726984
PSO	0.197420	3.315069	10.00002	0.201376	1.820392
GSA	0.147100	5.490735	10.00000	0.217765	2.172843
MFO	0.203553	3.443022	9.230377	0.212353	1.732553

Table 4.11 Statistical results for welded beam design problem on CSHO

Algorithm	BEST	MEAN	WORST	STD DEV.	MEDIAN
CSHO	**1.725662**	**1.725823**	1.726044	**0.000278**	**1.725776**
SHO	1.725669	1.725839	1.726051	0.000279	1.725792
GWO	1.726984	1.727113	1.727557	0.001145	1.727076
PSO	1.820392	2.230321	**3.048232**	0.324534	2.244646
GSA	2.172843	2.544223	3.003643	0.255847	2.495104
MFO	1.732553	1.775229	1.802371	0.012392	1.812451

4.4.2. Tension/compression spring design problem

This problem aims to minimize the spring weight as shown in Fig. 4.7. The optimization constraints of this problem are:

- Shear stress,
- Surge frequency,
- Minimum deflection [45].

There are three design variables of this problem:

- wire diameter (d)
- mean coil diameter (D) and the number of active coils (N).

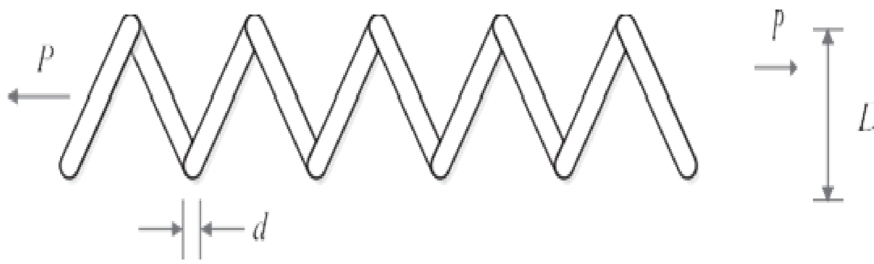


Fig. 4.7. Tension/compression spring problem schematic view

Table 4.12 and Table 4.13 shows results for all algorithms used for comparison purpose, where CSHO algorithm obtains better results than other metaheuristic algorithms.

Table 4.12 Comparison results for tension/compression spring design problem on CSHO

Algorithm	Optimum Variables			Optimum Cost
	d	D	N	
CSHO	0.051140	0.343751	12.0953	**0.012672000**
SHO	0.051146	0.343752	12.0958	0.012671145
GWO	0.050171	0.341539	12.07345	0.012678332
PSO	0.05000	0.310418	15.00000	0.013192578
GSA	0.05000	0.317316	14.22862	0.012873879
MFO	0.05000	0.313503	14.03281	0.012753904

Table 4.13 Statistical results for tension/compression spring design problem on CSHO

Algorithm	BEST	MEAN	WORST	STD DEV.	MEDIAN
CSHO	**0.012672000**	**0.012684108**	0.012715184	**0.000028**	**0.012687289**
SHO	0.012677000	0.012684109	0.012715198	0.000034	0.012687291
GWO	0.012678332	0.012697124	0.012720752	0.000043	0.012699679
PSO	0.013192578	0.014817169	**0.017862510**	0.002167	0.013192576
GSA	0.012873879	0.013438864	0.014211730	0.000245	0.013367889
MFO	0.012753904	0.014023643	0.017236589	0.001399	0.013896520

4.4.3. Pressure vessel design

This problem is to minimize the cost to form and weld the cylindrical vessel along with the material cost. Vessel is covered with hemispheres shaped heads at both sides at the end[47] as shown in Fig. 4.8 .

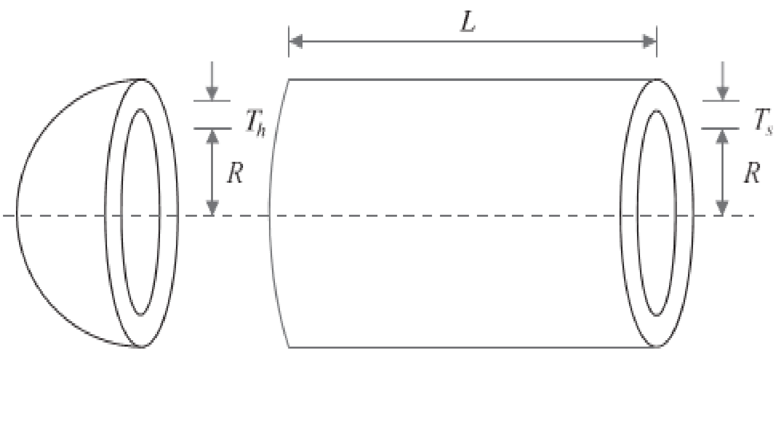


Fig. 4.8. Pressure vessel problem schematic view

There are four variables in this problem :

- T_S (thickness of the shell)
- T_h (thickness of the head)
- R (inner radius)
- L (length of the cylindrical section without considering the head).

Table 4.14. andTable 4.15 shows the comparison results CSHO performs better than all other metaheuristic algorithms.

Table 4.14 Comparison results for pressure vessel design problem on CSHO

Algorithm	Optimum Variables				Optimum cost
	T_S	T_h	R	L	
CSHO	0.778209	0.384867	40.315030	200.00000	**5885.5769**
SHO	0.778213	0.384879	40.315030	200.00069	5885.5570
GWO	0.779022	0.384654	40.327793	199.65031	5889.3682
PSO	0.778956	0.384681	40.320913	200.00000	5891.3872
GSA	1.085803	0.949619	49.345231	169.48739	11550.2977
MFO	0.835241	0.409834	43.578621	152.21515	6055.6358

Table 4.15 Statistical results obtained for pressure vessel design problem on CSHO

Algorithm	BEST	MEAN	WORST	STD DEV.	MEDIAN
CSHO	**5885.5769**	**5887.4439**	5892.3208	**002.889**	**5886.2279**
SHO	5885.5770	5887.4445	5892.3208	002.890	5886.2280
GWO	5889.3628	5891.5252	5894.6243	013.912	5890.6494
PSO	5891.3872	6531.5030	7394.5880	534.113	6416.1134
GSA	11550.2977	23342.2910	**333226.2555**	5790.631	24010.0425
MFO	6055.6358	6360.6859	7023.8539	365.589	6302.2305

4.4.4. Speed reducer design problem

As shown in Fig. 4.9 the motive of this problem is to minimize speed reducer weight with subject to constraints [35].

Constraints for this problem are:

- Bending stress of the gear teeth,
- Surface stress,
- Transverse deflections of the shafts,
- Stresses in the shafts.

There are seven optimization variables of this problem are:

- B represents the face width
- m represents module of teeth
- z represents number of teeth in the pinion
- $l1$ represents length of the first shaft between bearings
- $l2$ represents length of the second shaft between bearings
- $d1$ represents the diameter of first shafts, and
- $d2$ represents the diameter of second shafts

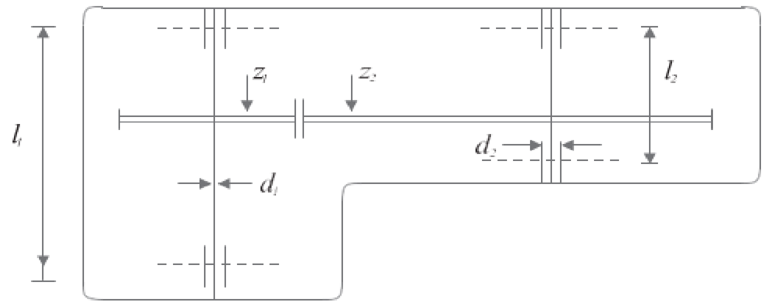


Fig. 4.9. Speed reducer problem schematic view

Table 4.16 and Table 4.17 shows results for speed reducer problem. CSHO shows optimal solution for minimum cost of this design.

Table 4.16 Comparison results for speed reducer design problem on CSHO

Algorithm	Optimum Variables							Optimum Cost
	B	**m**	**Z**	l_1	l_2	d_1	d_2	
CSHO	3.50161	0.7	17	7.3	7.8	3.35125	5.2889	**2998.5479**
SHO	3.50162	0.7	17	7.3	7.8	3.35127	5.28871	2998.5500
GWO	3.506689	0.7	17	7.380923	7.815722	3.357851	5.286756	3001.278
PSO	3.500014	0.7	17	8.3	7.8	3.352418	5.286718	3005.759
GSA	3.600002	0.7	17	8.3	7.8	3.369651	5.289219	3051.116
MFO	3.507526	0.7	17	7.302399	7.802364	3.323545	5.287529	3009.578

Table 4.17 Statistical results obtained for speed reducer design problem on CSHO

Algorithm	BEST	MEAN	WORST	STD DEV.	MEDIAN
CSHO	**2998.5479**	**2999.634**	3003.891	**1.93192**	**2999.191**
SHO	2998.5500	2999.639	3003.891	1.93192	2999.193
GWO	3001.278	3005.845	3008.759	5.83789	3004.523
PSO	3005.759	3105.258	3211.169	799.6389	3105.265

GSA	3051.116	3170.328	**3363.869**	92.5723	3156.752
MFO	3009.578	3021.251	3054.521	11.0245	3020.534

4.4.5. Rolling element bearing design problem

To maximize the dynamic load carrying capacity of a rolling element bearing as in Fig. 4.0 is the objective of this function. Decision variables of this problem which are:

- pitch diameter (D_m)
- ball diameter (D_b)
- number of balls (Z)
- inner (f_i)
- outer (f_o) raceway curvature
- coefficients K_{Dmin} , K_{Dmax}

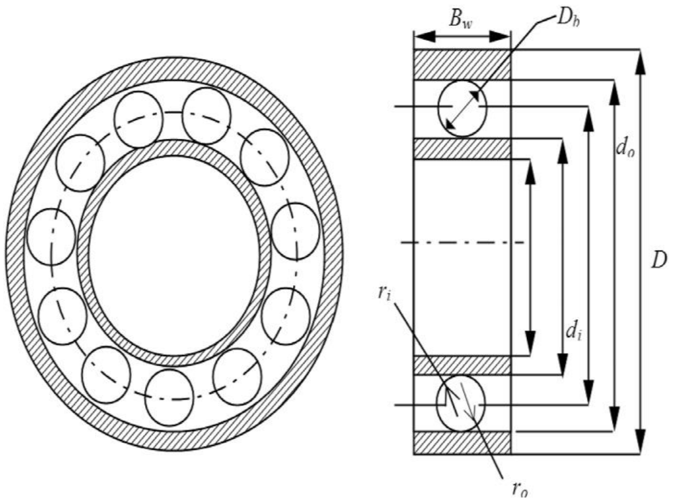


Fig. 4.10. Rolling element bearing problem schematic view

Table 4.18 and Table 4.19,shows results for proposed algorithm .CSHO algorithm is capable to achieve improved results.

4.5 Summary

In this chapter, a chaotic version of SHO is designed and implemented. To analyze the performance the CSHO is validated on twenty-nine well known test functions. The results show that CSHO provides improved and competitive results than SHO and other metaheuristic algorithms used for comparison purpose. Proposed algorithm is very well able to handle the constraint problems. Therefore providing the optimal results than other algorithms.

Table 4.18 Comparison results for rolling element bearing design problem on CSHO

Algorithm	**Optimum Variables**										**Optimum Cost**
	D_m	D_b	**Z**	F_i	**f**	K_{dim}	K_{dmax}	**E**	E_i	**Z**	
CSHO	125	21.40727	10.93271	0.515	0.515	0.4	0.7	0.3	0.02	0.6	**85054.529**
SHO	125	21.40728	10.93271	0.515	0.515	0.4	0.7	0.3	0.02	0.6	85054.530
GWO	125.6219	21.35134	10.98789	0.515	0.515	0.5	0.68810	0.300151	0.03254	0.62701	84807.115
PSO	125	20.75389	11.17331	0.515	0.515	0.5	0.61506	0.3	0.05161	0.6	81691.209
GSA	125	20.85420	11.14945	0.515	0.517	0.5	0.61824	0.304068	0.02000	0.62463	82276.937
MFO	125	21.03279	10.96591	0.515	0.515	0.5	0.67579	0.300214	0.02397	0.61001	84002.530

Table 4.19 Statistical results obtained for rolling element design problem on CSHO

Algorithm	**BEST**	**MEAN**	**WORST**	**STD DEV.**	**MEDIAN**
CSHO	**85054.529**	**85024.861**	85853.872	**0186.64**	**85040.236**
SHO	85054.530	85024.865	85853.873	0186.64	85040.237
GWO	84807.115	84791.624	84517.932	0137.190	84960.142
PSO	81691.209	50435.020	**32761.559**	13962.161	42287.538
GSA	82276.937	78002.110	71043.110	3119.909	78398.865
MFO	84002.530	82357.489	83979.255	1401.519	84497.392

CHAPTER 5
CHAOTIC EMPEROR PENGUIN OPTIMIZER

This chapter presents the new proposed chaotic emperor penguin optimizer algorithm. The designed CEPO is validated on standard benchmark test functions and constrained engineering problems for performance evaluation. The main objective is to improve exploration phase and enhance its performance.

5.1 Motivation

Emperor penguin algorithm is recently developed metaheuristic techniques for solving optimization problems. Since one single optimizer cannot perform efficiently for all sort of problems. Also many metaheuristic algorithms face premature convergence and local optima stagnation issues [23]. The efficiency of a metaheuristic algorithm depends on the utilization of randomization that is controlled by exploration and exploitation phases. Integration of chaos theory and metaheuristic algorithms has successfully resulted in improving performance. Chaos can enhance the exploration phase [22]. Many metaheuristic algorithm improved versions, have been designed by integration of chaos. Although it has good convergence ability, still EPO may not always find the global optima for some optimization problems even. In this research work chaotic version of spotted hyena optimization is proposed.

5.2 Proposed algorithm

CEPO algorithm is algorithm is designed by integrating chaotic theory in original EPO. In the proposed algorithm, the adjustment of parameter M is controlled by the given chaos function Start by initialization of search agents i.e. emperor penguin. Next a chaotic map is selected with first chaotic number and variable. Initial parameters of the CEPO T',$\vec{A}$,$\vec{C}$ l and *S()*,*R* are initialized which are same as in EPO. Also, the chaotic number of the chaotic map is initialized to adjust the parameter 'M'.Calculate fitness for all search agents. Determine the huddle boundary of emperor penguins using Eqs. (11) and (12) the temperature profile T′ around the huddle using Eq. (13). Compute the distance between the emperor penguins using Eqs. (14)–(18). Update the positions of other search agents using Eq. (19).Parameter M keeps on updating with course of iterations. Calculate the updated search agent fitness value and update the position of previously obtained optimal solution. At the end of the last iteration, the emperor penguin with the highest fitness will be considered as the most optimal solution by the CEPO algorithm.

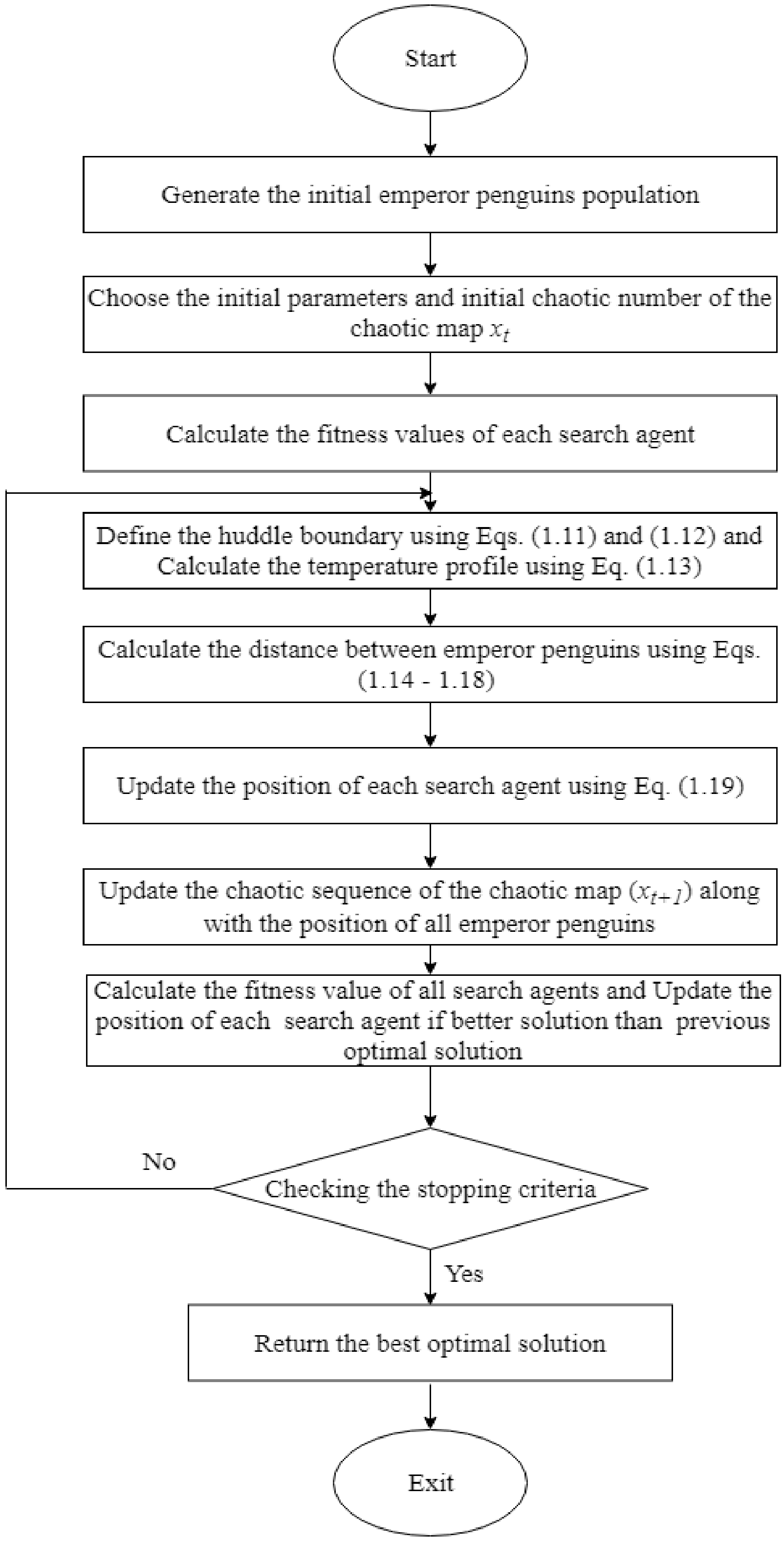


Fig. 5.1. Flowchart of proposed Chaotic EPO

5.3 Experimental results on benchmark test functions

Proposed algorithm is validated on twenty nine standard benchmark test functions. These benchmark functions are described in section below and Appendix A. The ten different chaotic maps are evaluated on benchmark functions and the most efficient map is chosen for further evaluation. For evaluation of results from proposed algorithm, the results are compared with original EPO as well as SHO, GWO, PSO, GSA and MFO.

5.3.1 Benchmark test functions

Benchmark test functions are categorized following categories: first category is unimodal test functions [41], second is multimodal test functions [39], third category is fixed-dimension multimodal [39, 41] and last is composite functions [42, 43]. Benchmark test function are described in Appendix A. Dimension and Range of the test functions are indicates in the appendix A. Tables A.1 and A.2 describes the unimodal $(F1 - F7)$ .The multimodal categories $(F8 - F16)$ and fixed-dimension modal $(F14 - F23)$ are represented in Tables A23 and A.3 respectively. Composite test functions $(F24 - F29)$. are detailed in A.4

5.3.2 Experimental setup

The parameter setting of the proposed algorithms CSHO and other metaheuristic algorithms for comparison purpose are described in Appendix C. The average and standard deviation are the metrics based on which comparison is made by obtaining optimal solution. Proposed algorithm is run for 30 independent runs with1000 iterations.

Table 5.1 Parameters setting of involved algorithms

Algorithms	Parameters	Values
Chaotic Emperor Penguin Optimizer(CEPO)	Search Agents	30
	Temperature Profile $(T^{'})$	[1,1000]
	$\overrightarrow{A}$ Constant	[-1.5,1.5]
	Function $S()$	[0,1.5]
	Parameter f	[2,3]
	Parameter M	2
	Parameter l	[1.5,2]
	Number of Generations	1000
Emperor Penguin Optimizer(EPO)	Search Agents	30
	Temperature Profile $(T^{'})$	[1,1000]

	$\overrightarrow{A}$ Constant	[-1.5,1.5]
	Function $S()$	[0,1.5]
	Parameter f	[2,3]
	Parameter M	2
	Parameter l	[1.5,2]
	Number of Generations	1000
Spotted Hyena Optimizer (SHO)	Search Agents	30
	Control Parameter($\vec{h}$)	[5,0]
	$\vec{M}$ Constant	[0.5,1]
	Number of Generations	1000
Grey Wolf Optimizer (GWO)	Search Agents	30
	Control Parameter($\vec{a}$)	[2,0]
	Number of Generations	1000
Particle Swarm Optimization(PSO)	Number of Particles	30
	Inertia Coefficient	.75
	Cognitive and Social Coeff	1.8,2
	Number of Generations	1000
Gravitational Search Algorithm(GSA)	Search Agents	30
	Gravitational Constant	100
	Alpha Coefficient	20
	Number of Generations	1000
Moth-Flame Optimization(MFO)	Convergence Constant	[-1,-2]
	Logarithmic Spiral	.75
	Number of Generations	1000

5.3.3 Performance evaluation

Proposed algorithm uses ten different chaotic maps for evaluation on benchmark functions as shown in Tables 4.2-4.5. Many chaotic maps are able to enhance the performance of EPO but the tent map shows best results than the others algorithms.

Table 5.6 shows that CEPO evaluation on unimodal functions with other algorithms namely EPO, SHO, GWO, PSO, GSA, and MFO. CEPO is the most efficient algorithm for functions $F4, F6$ and $F7$ with the best optimal solution

Tables 5.7 and 5.8 shows the results for multimodal and fixed-dimension multimodal functions. According to these tables results, CEPO was most efficient in $F8, F10, F13, F14, F15, F17, F18, F19$ and $F22$. These results show that the CEPO algorithm enhancing exploration phase. Table 5.9 shows CEPO algorithm performs better for functions $F25, F26, F27$ and F29 are very competitive in the rest other cases.

Table 5.2 Results obtained on chaotic maps for unimodal benchmark functions on CEPO

	F1	F2	F3	F4	F5	F6	F7
EPO	2.58E-30 5.98E-33	5.58E-58 2.98E-52	2.28E-20 8.98E-23	6.31E-19 **(4.98E-20)**	**6.73E+00** (2.15E-01)	6.66E-19 (4.99E-19)	3.50E-08 **(2.45E-07)**
CEPO-1 (Log)	1.00E-59 **6.10E-59**	**1 .00E-69** **6.10E-69**	1.00E-19 6.10E-19	6.81E-11 (1.08E-11)	8.41E+00 (5.32E-01)	2.65E-01 (2.75E-01)	3.67E-05 (3.53E-05)
CEPO-2 (Pie)	3.80E-20 3.41E-20	3 80E-33 3.41E-33	3.80E-30 3.41E-20	5.22E-10 (1.62E-13)	8.73E+00 (3.34E-01)	4.76E-01 (2.58E-01)	4.83E-05 (6.99E-05)
CEPO-3 (Sine)	4.00E-19 5.06E-32	4.00E-53 5.606E-32	**4.00E-39** 5.606E-29	7.02E-12 (7.03E-12)	8.97E+00 (3.36E-01)	5.18E-01 (2.46E-01)	4.10E-05 (3.66E-05)
CEPO-4 (Singer)	5.95E-30 4.45E-33	5.95E-24 4.45E-20	5.95E-10 4.45E-10	6.19E-11 (3.71E-11)	8.75E+00 (5.00E-01)	5.18E-01 (2.32E-01)	4.84E-05 (3.83E-05)
CEPO-5 (Sinu)	6.64E-08 2.23E-08	6.64E+18 2.23E+18	6.64E+02 2.23E+03	6.36E-11 (9.95E-11)	8.84E+00 (4.98E-01)	2.19E-01 ()2.58E-01	4.05E-05 (4.29E-05)
CEPO-6 (Tent)	**9.68E-60** 2.58E-50	9.68E-50 2.58E-52	9.68E-20 2.58E-20	**6.65E-22** (7.46E-12)	8.41E+00 **(3.03E-01)**	**2.14E-01** **(1.99E-01)**	**3.10E-05** (1.39E-05)
CEPO-7 (Che)	1.78E-30 7.71E-30	1.78E-30 7.71E-30	1.78E-20 7.71E-20	8.34E-11 (9.72E-14)	8.79E+00 (5.40E-01)	9.62E-01 (2.59E-01)	4.32E-05 (4.60E-05)
CEPO-8 (Cir)	2.58E-22 7.98E-32	2.58E-52 7.98E-52	2.58E-22 7.98E-42	7.49E-14 (8.88E-14)	8.83E+00 (4.22E-01)	3.31E-01 (2.39E-01)	3.60E-05 (3.94E-05)
CEPO-9 (Gau)	1.00E-49 6.10E-49	1.00E-39 6.10E-49	1.00E-29 6.10E-29	9.74E-15 (1.03E-12)	8.84E+00 (3.55E-01)	9.55E-01 (2.72E-01)	3.43E-05 (2.78E-05)
CEPO-10 (Icm)	3.80E-30 3.41E-32	3.80E-50 3.41E-20	3.80E-20 **3.41E-50**	9.97E-13 (8.77E-11)	8.65E+00 (4.79E-01)	7.15E-01 (2.38E-01)	4.00E-05 (4.58E-05)

Table 5.3 Results obtained on chaotic maps for multimodal benchmark functions on CEPO

	F8	F9	F10	F11	F12	F13
EPO	-8.11E+02 (6.89E+01)	7.06E-01 (5.80E-01)	8.73E-19 (1.09E-14)	**5.34E-05** (6.20E-10)	7.05E-04 (4.42E-03)	0.00E+00 (0.00E+00)
CEPO-1 (Log)	-2.13E+03 **(2.03E+01)**	**4.06E-01** (5.70E-01)	1.37E-19 (5.15E-10)	3.06E-01 **(4.80E-11)**	1.33E-02 (4.32E-02)	0.00E+00 (0.00E+00)
CEPO-2 (Pie)	-2.17E+03 (5.80E+02)	9.63E-01 (7.29E+00)	2.20E-14 (7.61E-11)	5.06E-01 (6.80E-10)	1.26E-02 (1.80E-02)	0.00E+00 (0.00E+00)
CEOO-3 (Sine)	-2.42E+03 (4.86E+02)	9.06E-01 **(3.80E-01)**	1.99E-10 (9.06E-10)	2.42E-02 (6.29E-02)	1.18E-02 (3.96E-02)	0.00E+00 (0.00E+00)
CEPO-4 (Singer)	-2.14E+03 (6.56E+02)	5.06E-01 (4.80E-01)	5.35E-13 (1.41E-10)	4.31E-02 (6.21E-04)	1.07E-02 (3.39E-02)	0.00E+00 (0.00E+00)
CEPO-5 (Sinu)	-2.42E+03 (6.01E+02)	6.06E-01 (4.80E-01)	4.92E-11 (2.67E-11)	5.16E-03 (5.63E-08)	1.30E-02 (3.24E-02)	0.00E+00 (0.00E+00)
CEPO-6 (Tent)	**-2.13E+03** (4.32E+02)	7.06E-01 (4.80E-01)	**1.51E-19** **(2.43E-19)**	2.06E-01 (8.80E-10)	1.19E-02 **(2.86E-02)**	**0.00E+00** **(0.00E+00)**
CEPO-7 (Che)	-2.08E+03 (5.01E+02)	11.04E+00 (3.99E+00)	3.57E-15 (1.62E-10)	9.48E-02 (3.60E-02)	1.11E-02 (2.86E-02)	0.00E+00 (0.00E+00)
CEPO-8 (Cir)	-2.19E+03 (5.35E+02)	6.06E-01 (4.80E-01)	1.06E-11 (4.02E-15)	2.17E-01 (5.69E-02)	1.55E-02 (2.17E-02)	0.00E+00 (0.00E+00)
CEPO-9 (Gau)	-1.97E+03 (4.50E+02)	9.04E+00 (5.07E+00)	1.97E-17 (6.52E-17)	7.06E-01 (2.80E-04)	1.05E-02 (1.13E-02)	0.00E+00 (0.00E+00)
CEPO-10 (Icm)	-2.33E+03 (6.17E+02)	7.06E-01 (4.80E-01)	9.78E-18 (5.11E-19)	1.45E-02 (4.45E-07)	**1.25E-02** (6.51E-01)	0.00E+00 (0.00E+00)

Table 5.4 Results obtained on chaotic maps for fixed-dimension multimodal benchmark functions on CEPO

	F14	F15	F16	F17	F18	F19	F20	F21	F22	F23
EPO	2.70E+00 (5.71E-03)	9.18E-04 (1.17E-04)	**-1.06E+01** **(7.22E-12)**	3.99E-01 **(2.22E-01)**	3.00E+00 (4.09E+00)	**-3.97E+00** (4.33E-03)	-2.39E+00 (4.50E-01)	-9.80E+00 (3.11E-01)	-9.64E+01 **(1.13E-04)**	-2.73E+00 (4.70E-03)
CEPO-1 (Log)	3.27E+00 (3.76E-02)	9.14E-04 (1.02E-04)	-1.06E+01 (2.45E-11)	3.97E-01 (3.25E-01)	3.00E+00 (8.96E+00)	-3.37E+00 (3.97E-01)	**-1.72E+00** (6.07E-01)	-4.64E+00 (3.56E-01)	-1.90E+01 (2.80E-04)	-1.84E+00 (2.46E-01)
CEPO-2 (Pie)	5.23E+00 (4.03E-01)	9.18E-04 (1.69E-04)	-1.03E+01 (3.74E-11)	3.98E-01 (2.78E-01)	3.00E+00 (2.66E+00)	-3.39E+00 (3.31E-01)	-1.41E+00 (6.42E-01)	-5.48E+00 (3.54E-01)	-1.84E+01 (3.46E-04)	-1.93E+00 (3.17E-01)
CEPO-3 (Sine)	6.41E+00 (4.62E-02)	9.47E-04 (1.38E-03)	-1.03E+01 (2.16E-11)	3.98E-01 (3.13E-01)	3.00E+00 (4.04E+00)	-3.39E+00 (3.35E-01)	-1.65E+00 (6.63E-01)	-5.20E+00 (3.55E-01)	-1.82E+01 (2.61E-04)	-1.92E+00 **(2.06E-04)**
CEPO-4 (Singer)	2.40E+00 **(4.14E-04)**	9.68E-04 (1.09E-04)	-1.03E+01 (3.40E-11)	3.98E-01 (3.01E-01)	3.00E+00 (5.08E+00)	-3.39E+00 (3.32E-01)	-1.57+00 (6.65E-01)	-5.69E+00 (3.96E-01)	-1.69E+01 (2.34E-04)	-1.81E+00 (2.51E-01)
CEPO-5 (Sinu)	2.17E+00 (3.99E-02)	9.39E-04 (1.52E-03)	-1.03E+01 (3.40E-11)	3.97E-01 (4.53E-01)	3.00E+00 (6.71E+00)	-3.38E+00 (3.89E-01)	-1.57E+00 (6.43E-01)	-5.20E+00 (3.74E-01)	-1.73E+01 (3.98E-04)	-1.87E+00 (5.43E-01)
CEPO-6 (Tent)	**2.01E+00** (1.66E-02)	**9.08E-04** **(1.04E-04)**	-1.06E+01 (2.12E-11)	**3.97E-01** (2.31E-01)	**3.00E+00** **(2.04E+00)**	-3.39E+00 **(2.13E-08)**	-3.67E+00 **(3.33E-01)**	-3.89E+00 **(3.04E-01)**	**-1.67E+01** (2.15E-04)	-1.66E+00 (3.06E-01)
CEPO-7 (Che)	5.25E+00 (4.33E-03)	8.46E-03 (1.95E-03)	-1.03E+01 (3.48E-11)	3.98E-01 (4.08E-01)	3.00E+00 (2.95E+00)	-3.38E+00 (3.92E-01)	-1.51E+00 (6.34E-01)	-5.55E+00 (3.76E-01)	-1.70E+01 (5.37E-04)	-1.66E+00 (1.25E-01)
CEPO-8 (Cir)	6.83E+00 (4.30E-03)	9.77E-04 (1.39E-04)	-1.03E+01 (3.18E-11)	3.99E-01 (4.49E-01)	3.00E+00 (3.75E+00)	-3.41E+00 (2.67E-01)	-1.64E+00 (6.64E-01)	-4.77E+00 (3.46E-01)	-1.93E+01 (3.69E-04)	-1.79E+00 (2.25E-01)
CEPO-9 (Gau)	9.36E+00 (4.31E-03)	7.48E-03 (1.68E-03)	-1.03E+01 (3.21E-11)	3.98E-01 (4.31E-01)	3.00E+00 (4.07E+00)	-3.37E+00 (6.74E-01)	-1.49E+00 (6.79E-01)	-5.17E+00 (3.88E-01)	-1.98E+01 (4.14E-04)	-1.87E+00 (2.50E-01)
CEPO-10 (Icm)	6.03E+00 (4.10E-03)	9.47E-04 (1.14E-04)	-1.03E+01 (3.79E-11)	3.98E-01 (4.46E-01)	3.00E+00 (5.97E+00)	-3.41E+00 (3.47E-01)	-1.28E+00 (6.79E-01)	**-6.06E+00** (3.80E-01)	-1.97E-01 (2.93E-04)	**-1.82E-01** (2.91E-01)

Table 5.5 Results obtained on chaotic maps for composite benchmark functions on CEPO

	F24	F25	F26	F27	F28	F29
EPO	2.33E+02 (6.39E+02)	4.10E+02 (7.43E+01)	3.49E+02 **(3.12E+01)**	5.19E+02 **(3.10E+03)**	**1.08E+02** (3.38E+01)	6.91E+01 (2.99E+03)
CEPO-1 (Log)	2.33E+02 (5.38E+02)	1.02E+03 (9.37E+01)	3.76E+02 (3.18E+01)	7.11E+02 (4.06E+04)	1.28E+02 (1.48E+01)	5.88E+01 **(2.79E+00)**
CEPO-2 (Pie)	2.43E+02 (5.41E+02)	4.25E+02 (9.42E+01)	3.45E+02 (3.21E+01)	7.23E+02 (8.32E+03)	1.36E+02 (2.44E+01)	5.99E+01 (2.77E+00)
CEPO-3 (Sine)	**2.30E+02** (6.36E+02)	4.21E+02 (9.41E+01)	3.49E+02 (3.16E+01)	7.26E+02 (7.23E+04)	1.25E+02 **(1.27E+01)**	5.96E+01 (4.67E+00)
CEPO-4 (Singer)	2.33E+02 **(3.28E+02)**	1.34E+03 (9.39E+01)	3.51E+02 (3.19E+01)	6.02E+02 (7.24E+04)	1.18E+02 (2.89E+01)	5.97E+01 (3.98E+00)
CEPO-5 (Sinu)	2.47E+02 (7.29E+02)	4.11E+02 (9.54E+01)	3.64E+02 (3.24E+01)	5.54E+02 (3.76E+04)	1.15E+02 (1.54E+01)	5.92E+01 (5.01E+00)
CEPO-6 (Tent)	2.36E+02 (7.34E+02)	**4.06E+02** **(7.14E+01)**	**3.39E+02** (3.18E+01)	**4.91E+02** (2.13E+05)	1.08E+02 (1.32E+01)	**5.87E+01** (5.63E+00)
CEPO-7 (Che)	2.39E+02 (9.44E+02)	4.69E+02 (8.47E+01)	3.40E+02 (3.26E+01)	7.67E+02 (7.56E+04)	1.09E+02 (1.76E+01)	5.95E+01 (4.99E+00)
CEPO-8 (Cir)	2.37E+02 (8.76E+02)	4.25E+02 (8.67E+01)	3.98E+02 (3.34E+01)	9.23E+02 (8.33E+04)	1.33E+02 (1.89E+01)	5.98E+01 (3.67E+00)
CEPO-9 (Gau)	2.33E+02 (8.38E+02)	1.47E+03 (9.32E+01)	3.45E+02 (3.43E+01)	7.26E+02 (3.56E+04)	1.19E+02 (1.36E+01)	5.97E+01 (2.92E+00)
CEPO-10 (Icm)	2.45E+02 (6.83E+02)	4.20E+02 (9.44E+01)	3.58E+02 (3.21E+01)	9.22E+02 (3.28E+03)	1.13E+02 (1.78E+01)	5.88E+01 (2.95E+00)

Table 5.6 Results obtained from CEPO on unimodal benchmark functions

	CEPO-6	EPO	SHO	GWO	PSO	GSA	MFO
F1	9.68E-60 2.58E-50	2.58E-30 5.98E-33	**0.00E+00** **(0.00E+00)**	3.79E-38 (2.98E-42)	5.38E-10 (4.61E-08)	1.13E-14 (7.72E-15)	1.37E-05 (1.99E-06)
F2	9.68E-50 2.58E-52	5.58E-58 2.98E-52	**0.00E+00** **(0.00E+00)**	8.16E-35 (6.18E-35)	8.94E-04 (9.47E-04)	5.62E-01 (8.50E-01)	3.57E+01 (2.52E+01)
F3	9.68E-20 2.58E-20	2.28E-20 8.98E-23	**0.00E+00** **(0.00E+00)**	8.88E-16 (1.54E-15)	1.72E+01 (8.82E+00)	4.57E+02 (1.46E+02)	1.49E+04 (1.18E+04)
F4	**6.65E-22** (7.46E-12)	6.31E-19 **(4.98E-20)**	6.61E-12 (6.98E-12)	1.38E-14 (1.38E-14)	6.53E-01 (1.60E-01)	1.29E+00 (1.07E+00)	6.67E+01 (9.3E+00)
F5	8.41E+00 **(3.03E-01)**	**6.73E+00** (2.15E-01	8.73E+00 (5.95E-01)	2.68E+01 (6.37E-01)	7.25E+01 (5.18E+01)	4.52E+01 (5.01E+01)	2.67E+04 (3.45E+05)
F6	**2.14E-01** **(1.99E-01)**	6.66E-19 (4.99E-19)	2.66E-01 (1.89E-01)	6.22E-01 (3.05E-01)	3.16E-08 (1.43E-07)	1.04E-16 (4.47E-17)	1.32E-03 (3.44E-01)
F7	**3.10E-05** (1.39E-05)	3.50E-08 **(2.45E-07)**	3.50E-05 (2.45E-05)	7.99E-04 (3.70E-04)	7.36E-02 (2.99E-02)	5.07E-02 (1.71E-02)	3.68E-01 (1.05E-02)

Table 5.7 Results obtained from CEPO on multimodal benchmark functions

	CEPO-6	EPO	SHO	GWO	PSO	GSA	MFO
F8	**-2.13E+03** (4.32E+02)	--8.11E+02 (6.89E+01)	-1.11E+03 (2.89E+02)	-6.08E+03 **(1.81E+02)**	-6.60E+03 (3.04E+03)	-2.54E+03 (3.84E+02)	-8.64E+03 (9.36E+02)
F9	7.06E-01 (4.80E-01)	7.06E-01 (5.80E-01)	**0.00E+00** **(0.00E+00)**	2.59E-01 (8.69E-01)	4.79E+01 (9.88E+00)	2.57E+01 (6.83E+00)	1.60E+02 (3.25E+01)
F10	**1.51E-19** **(2.43E-19)**	8.73E-19 (1.09E-14)	2.33E+00 (1.49E+00)	1.66E-14 (3.32E-15)	2.34E-04 (8.10E-04)	7.61E-09 (1.50E-09)	1.49E+01 (7.55E+00)
F11	2.06E-01 (8.80E-10)	5.34E-05 (6.20E-10)	**0.00+00** **(0.00+00)**	1.06E-03 (4.50E-03)	1.23E-02 (1.24E-02)	8.63E+00 (3.10E+00)	5.40E-02 (1.05E-01)
F12	1.19E-02 (2.86E-02)	7.05E-04 (4.42E-03)	3.85E-02 (3.42E-02)	4.46E-02 (1.94E-02)	**2.11E-10** **(3.63E-10)**	1.92E-01 (7.74E-01)	1.03E+00 (9.21E+00)
F13	**0.00E+00** **(0.00E+00)**	0.00E+00 (0.00E+00)	9.99E-01 (9.63E-02)	4.94E-01 (2.13E-01)	4.36E-03 (6.09E-03)	5.33E-32 (6.58E-32)	6.32E-01 (1.42E+00)

Table 5.8 Results obtained from CEPO on fixed dimension multimodal benchmark functions

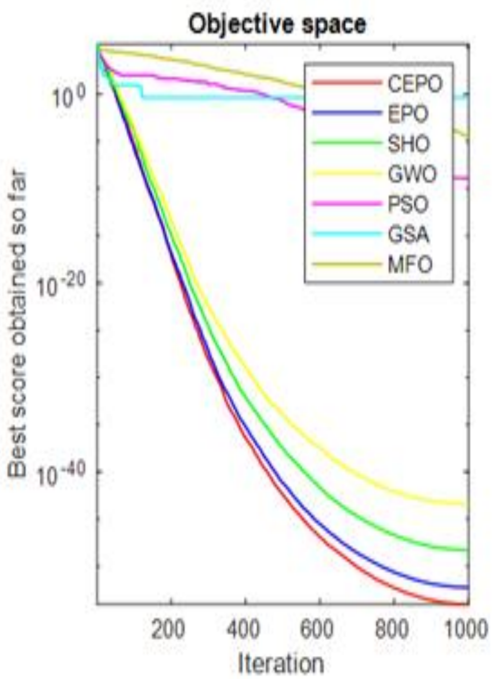
	CEPO-6	EPO	SHO	GWO	PSO	GSA	MFO
F14	**2.01E+00**	2.70E+00	7.70E+00	4.39E+00	7.70E+00	2.57E+00	2.21E+00
	(1.66E-02)	**(5.71E-03)**	(3.71E+00)	(5.99E+00)	(7.28E+00)	(2.31E+00)	(1.86E+00)
F15	**9.08E-04**	9.18E-04	9.38E-04	7.04E-03	9.92E-04	6.05E-03	1.95E-07
	(1.04E-04)	(1.17E-04)	(1.17E-04)	(6.91E-03)	(2.11E-04)	(6.76E-04)	(3.67E-04)
F16	-1.06E+01	-1.06E+01	**-1.01E+01**	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
	(2.12E-11)	**(7.22E-12)**	(1.82E-18)	(7.55E-19)	(6.71E-12)	(1.87E+00)	(2.27E+00)
F17	**3.97E-01**	3.99E-01	3.97E-01	3.97E-01	3.97E-01	3.97E-01	3.97E-01
	(2.31E-01)	**(2.22E-01)**	(4.45E-01)	(8.38E-01)	(8.91E-01)	(6.54E-01)	(5.78E-01)
F18	**3.00E+00**	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00
	(2.04E+00)	(4.09E+00)	(9.09E+00)	**(1.47E-05)**	(6.85E-06)	(2.70E-02)	(4.65E-15)
F19	-3.39E+00	**-3.37E+00**	-3.37E+00	-3.86E+00	-3.89E+00	-3.86E+00	-3.86E+00
	(2.13E-08)	(4.33E-03)	(4.33E-05)	(2.28E-03)	(2.71E-05)	(4.41E-01)	(3.43E-05)
F20	-3.67E+00	-2.39E+00	-1.39E+00	-3.25E+00	**-3.26E+00**	-1.32E+00	-3.23E+00
	(3.33E-01)	(4.50E-01)	(6.50E-01)	(8.65E-01)	(6.03E-01)	(5.45E-01)	(6.73E-01)
F21	-3.89E+00	-9.80E+00	-3.80E+00	**-9.30E+00**	-7.68E+00	-4.99E+00	-6.30E+00
	(3.04E-01)	(3.11E-01)	(3.81E-01)	(1.92E+00)	(3.15E+00)	(1.43E+00)	(3.13E+00)
F22	**-1.67E+01**	-9.64E+01	-1.64E+01	-1.04E+01	-8.61E+00	-6.02E+00	-7.39E+00
	(2.15E-04)	**(1.13E-04)**	(2.43E-04)	(2.46E-04)	(2.83E+00)	(2.42E+00)	(3.57E+00)
F23	-1.66E+00	-2.73E+00	-1.73E+00	**-1.00E+01**	-9.56E+00	-9.05E+00	-7.36E+00
	(3.06E-01)	(4.70E-03)	**(2.70E-01)**	(1.75E-04)	(2.24E+00)	(2.42E+00)	(3.72E+00)

Table 5.9 Results obtained from CEPO on composite benchmark functions

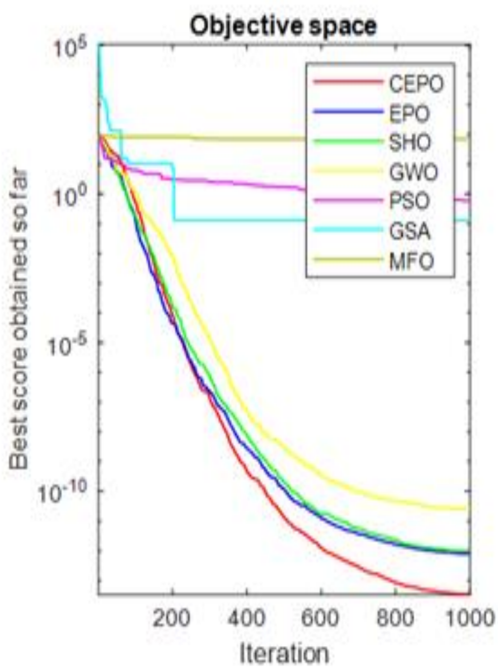
	CEPO-6	EPO	SHO	GWO	PSO	GSA	MFO
F24	2.36E+02 (7.34E+02)	2.33E+02 (6.39E+02)	9.13E+02 (4.39E+02)	8.35E+01 (8.45E+01)	6.03E+01 (8.89E+01)	**4.32E-11** **(3.01E-11)**	1.17E+02 (7.39E+01)
F25	**4.06E+02** **(7.14E+01)**	4.10E+02 (7.43E+01)	5.11E+02 (9.43E+01)	1.43E+02 (4.34E+01)	2.49E+02 (1.72E+02)	2.05E+02 (4.45E+02)	9.18E+01 (1.46E+01)
F26	**3.39E+02** (3.18E+01)	3.49E+02 **(3.12E+01)**	3.39E+02 (3.22E+01)	3.76E+02 (5.42E+01)	3.39E+02 (8.56E+01)	3.78E+02 (8.89E+01)	4.11E+02 (1.19E+02)
F27	**4.91E+02** (2.13E+05)	5.19E+02 **(3.10E+03)**	7.19E+02 (1.10E+03)	4.99E+02 (1.11E+03)	5.51E+02 (1.37E+03)	5.45E+02 (1.11E+03)	1.39E+03 (2.15E+05)
F28	1.08E+02 (1.32E+01)	1.08E+02 (3.38E+01)	**1.04E+02** **(1.28E+01)**	1.37E+02 (2.11E+02)	2.43E+02 (4.23E+02)	1.42E+02 (1.33E+02)	1.15E+02 (9.32E+01)
F29	**5.87E+01** (5.63E+00)	6.91E+01 (2.99E+03)	5.91E+01 **(4.99E+00)**	7.20E+02 (3.86E+03)	8.66E+02 (1.66E+02)	8.42E+02 (4.59E+02)	8.82E+02 (2.43E+01)

5.3.3.1 Convergence analysis

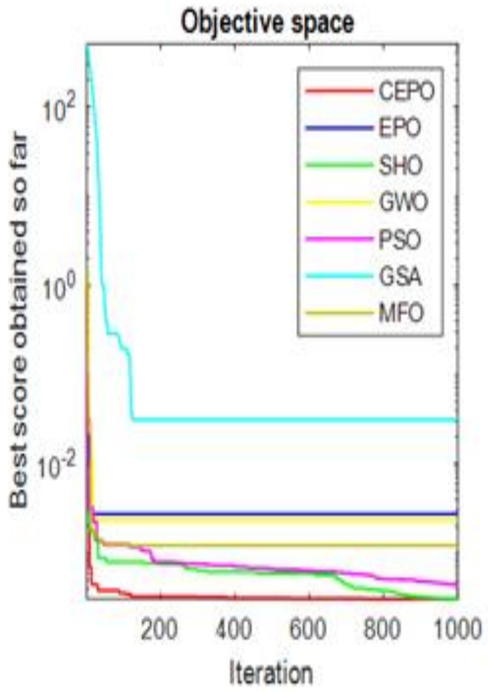
The convergence analysis of CEPO, EPO, SHO, GWO, PSO, GSA and MFO are shown in Fig. 5.2.Convergence analysis is made on test functions $F1, F7, F11$ and $F26$. CEPO provides better results than other algorithms.



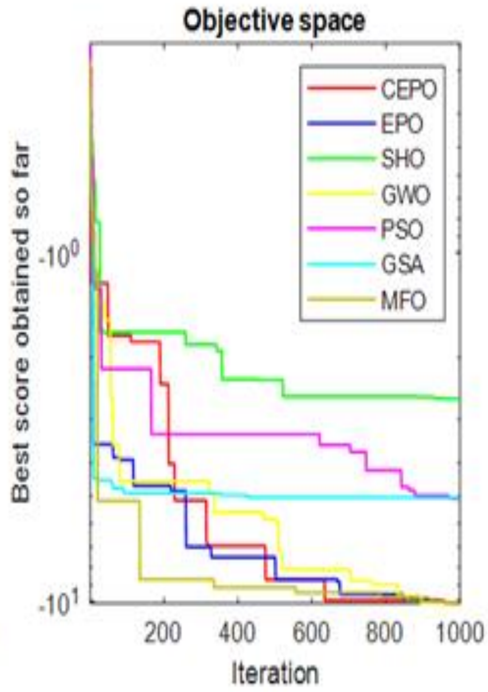
F1



F7



F11



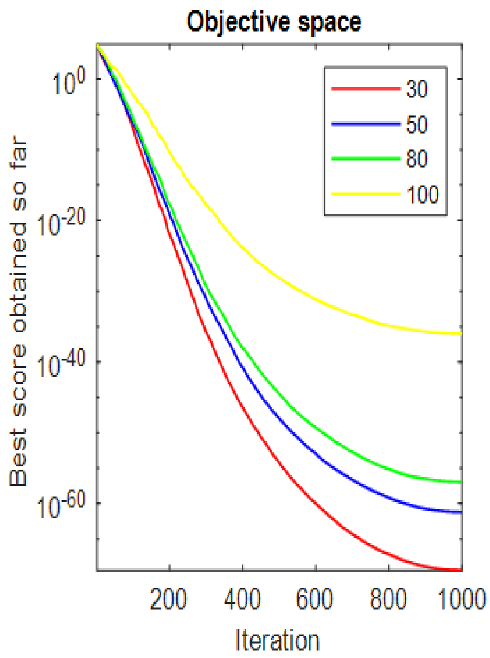
F26

Fig. 5.2. Convergence analysis on CEPO

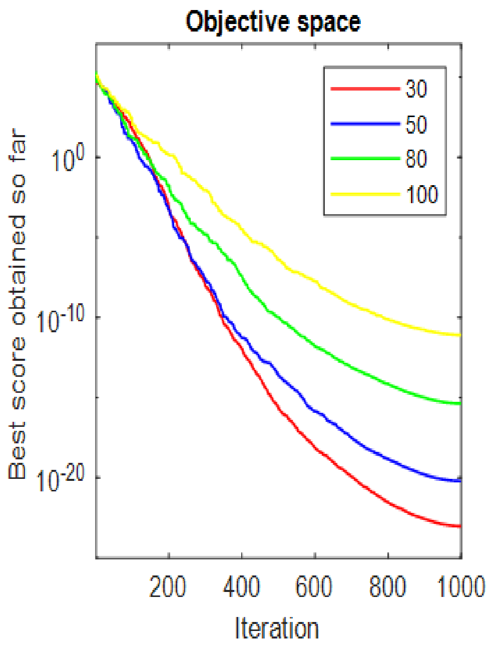
5.3.3.2 Sensitivity study

The sensitivity investigation of is discussed in this section below.

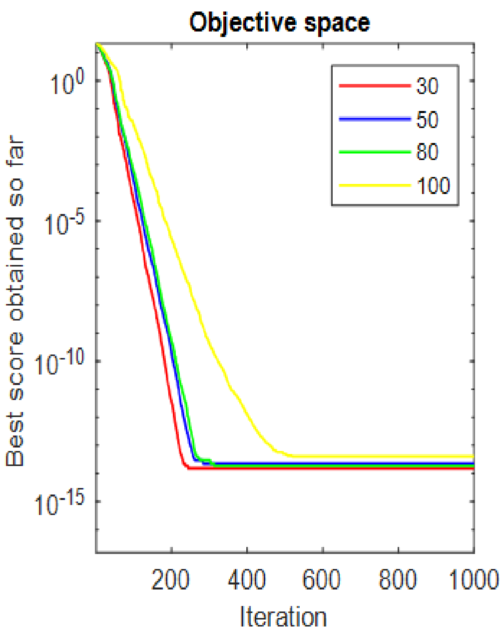
- Number of search agents: Values of search agents are set as 30, 50, 80, and 100. In test functions F2, F5, F13 and F27 as in Fig. 5.3 CEPO shows optimal solutions for value set to 30 for search agents.
- Maximum number of iterations: 100, 500, 800, and 1000 are varied number of iterations. Fig. 5.4 CSHO shows best results for F3,F7,F17,F27 test functions
- Variation in parameter f and parameter l: The values set for f are [1, 2], [2, 3], [3, 4], and [4,5]. l parameter are [.5, 1], [1, 1.5], [1.5, 2], and [2, 2.5]. Fig. 5.5 shows results on $F1$ for f parameter and Fig. 5.6 on function $F7$ for l parameter. CEPO obtained optimal results for f range [2, 3] and l range [1.5, 2].



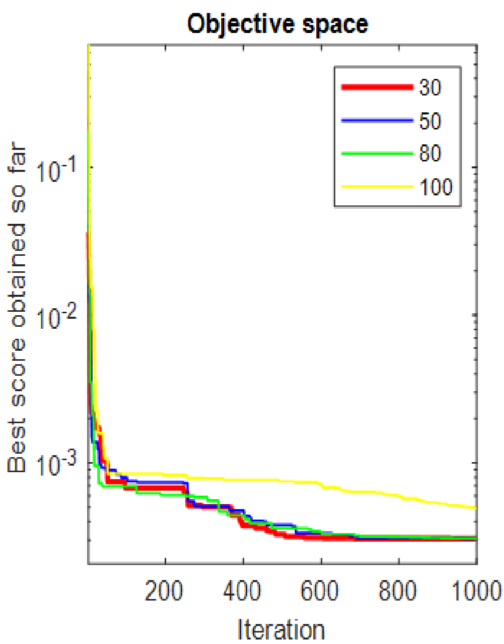
F2



F5

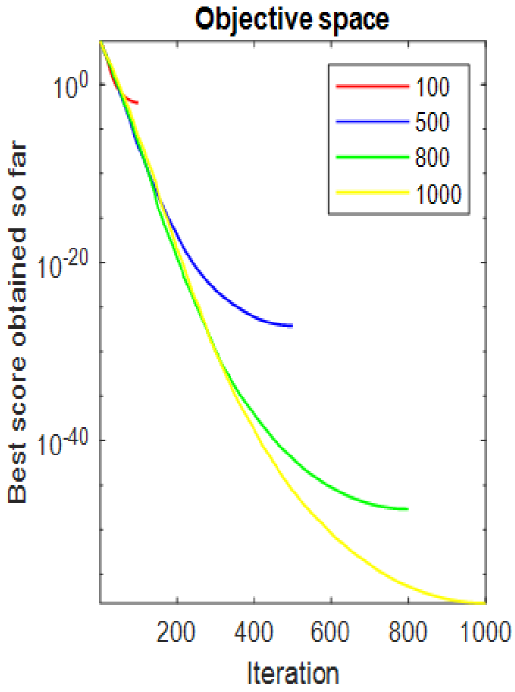


F13

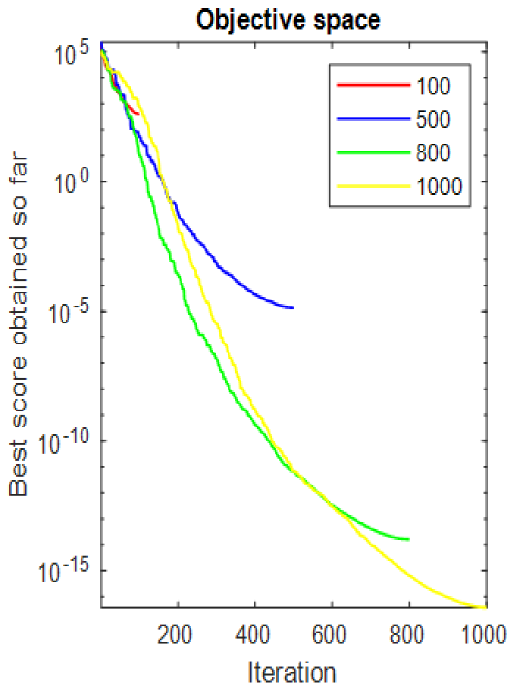


F27

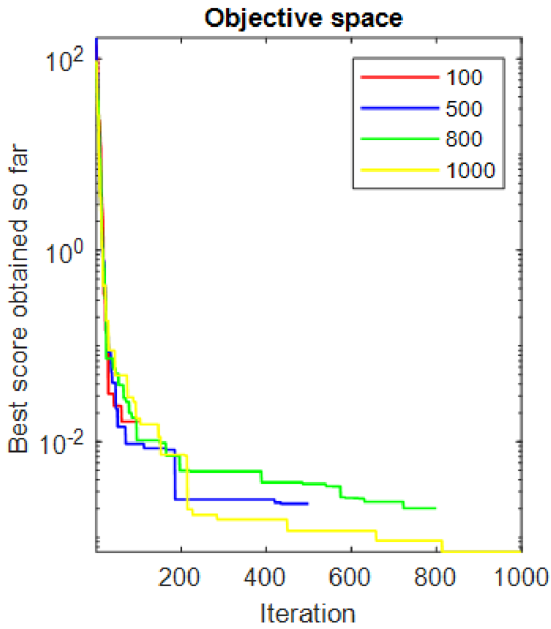
Fig. 5.3. Sensitivity analysis of search agents on CEPO.



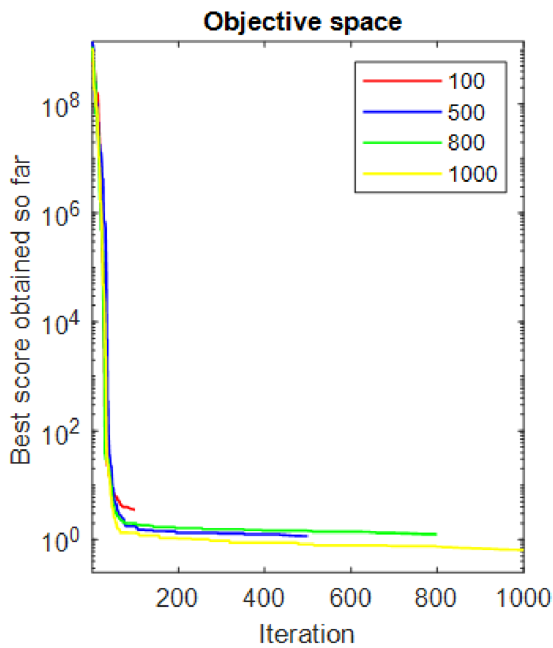
F3



F7

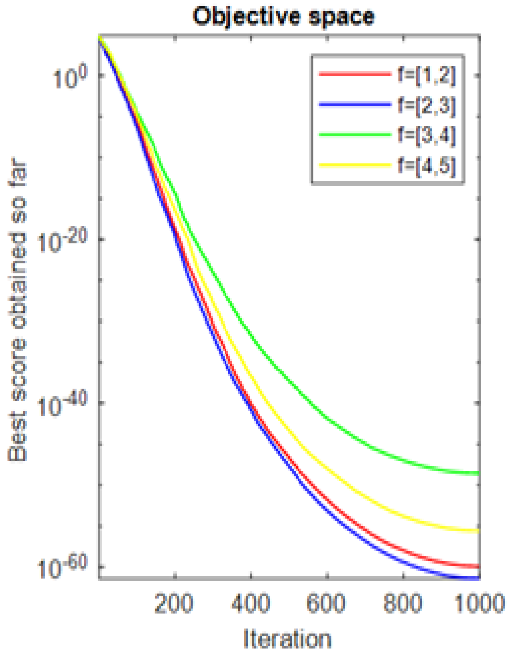


F17



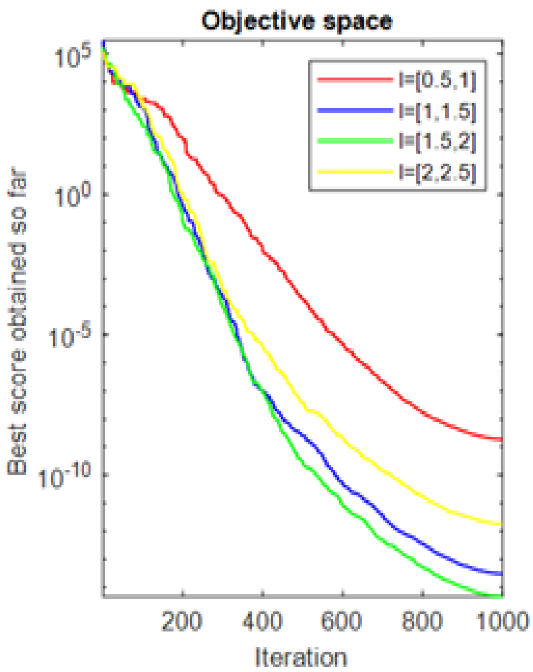
F27

Fig. 5.4. Sensitivity analysis of number of iterations on CEPO



F1

Fig. 5.5. Sensitivity analysis of *f* parameter on CEPO.

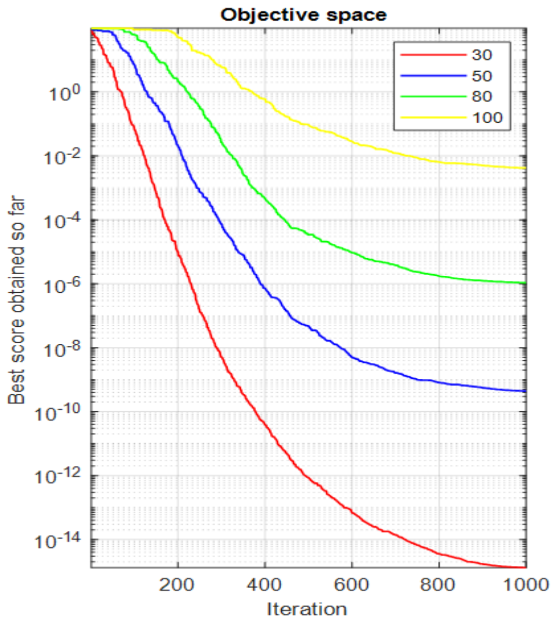


F7

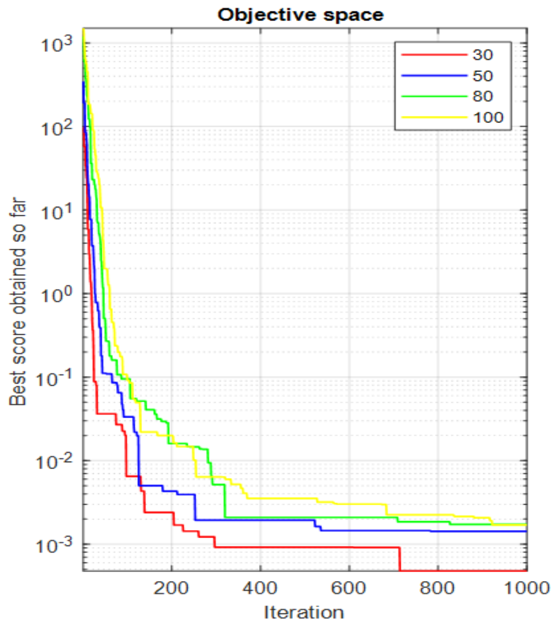
Fig. 5.6. Sensitivity analysis of ***l*** parameter on CEPO.

5.3.3.3 Scalability study

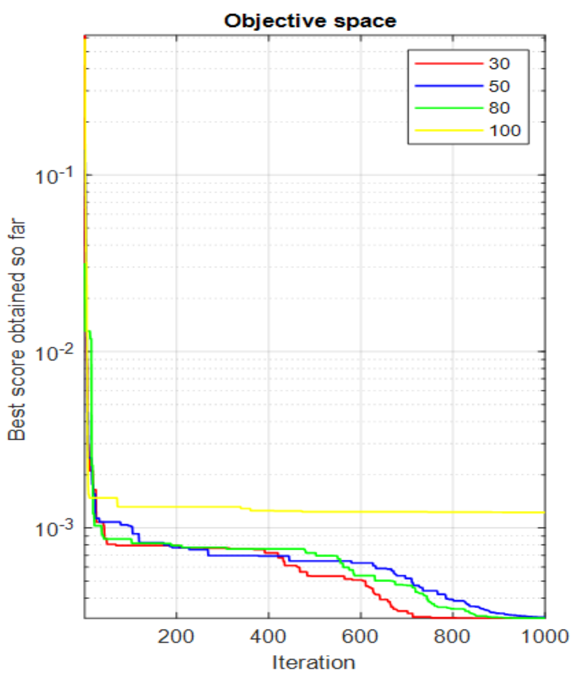
30 to 50, 50 to 80, and 80 to100 are different dimension set on benchmark functions F3,F7,F11, and F26. Fig. 5.7 shows the evaluation of CEPO algorithm on dimensionality change.Proposed algorithm shows different results with varied dimiensions.



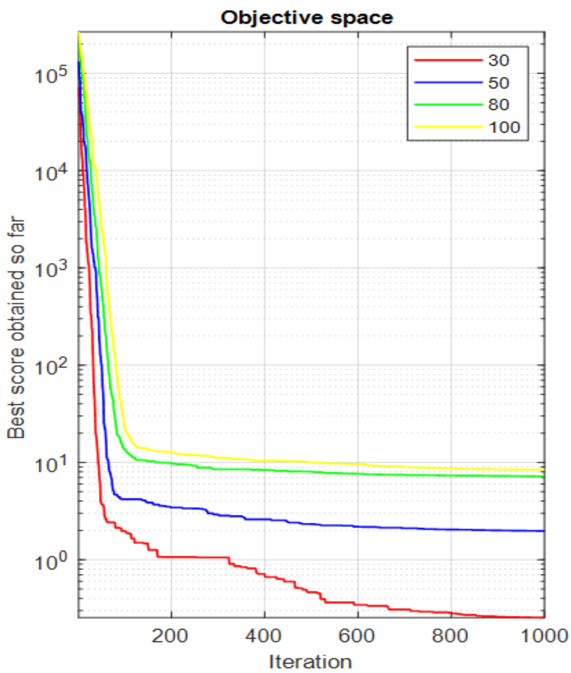
F3



F7



F11



F26

Fig. 5.7. Scalability analysis on CEPO

5.4 Engineering Design Problems

In this section proposed CEPO is evaluated on constrained engineering problems as discussed in section 4.5. CEPO, EPO SHO, GWO, PSO, MFO, and GSA are metaheuristic algorithms used for comparison purpose.

5.4.1. Welded beam design: Problem as earlier discussed in section 4.5.1. Table 5.10 and Table 5.11. shows the results, CEPO algorithm obtains better solution than other metaheuristic algorithms.

Table 5.10 Comparison results for welded beam design on CEPO

Algorithm	Optimum Variables				Optimum Cost
	H	***T***	***l***	***b***	
CEPO	0.205312	3.474790	9.035791	0.205799	**1.725660**
EPO	0.205322	3.474791	9.135792	0.205823	1.725661
SHO	0.205619	3.474856	9.035794	0.205823	1.725669
GWO	0.205673	3.475401	9.036978	0.206243	1.726984
PSO	0.197420	3.315069	10.00002	0.201376	1.820392
GSA	0.147100	5.490735	10.00000	0.217765	2.172843
MFO	0.203553	3.443022	9.230377	0.212353	1.732553

Table 5.11 Statistical results for welded beam design on CEPO

Algorithm	BEST	MEAN	WORST	STD DEV.	MEDIAN
CEPO	**1.725660**	**1.725820**	1.726044	**0.000277**	**1.725772**
EPO	1.725662	1.725823	1.726044	0.000279	1.725776
SHO	1.725669	1.725839	1.726051	0.000279	1.725792
GWO	1.726984	1.727113	1.727557	0.001145	1.727076
PSO	1.820392	2.230321	**3.048232**	0.324534	2.244646
GSA	2.172843	2.544223	3.003643	0.255847	2.495104
MFO	1.732553	1.775229	1.802371	0.012392	1.812451

5.4.2. Pressure vessel design: Problem as earlier discussed in section 4.5.2. Table 5.12. and Table 5.13 shows the comparison results CEPO performs better than all other algorithms.

Table 5.12 Comparison results for pressure vessel design problem on CEPO

Algorithm	Optimum Variables				Optimum cost
	T_S	T_h	**R**	**L**	
CEPO	0.778209	0.384867	40.315030	200.00000	**5885.5669**
EPO	0.778219	0.384873	40.315030	200.00069	5885.5570
SHO	0.778213	0.384879	40.315030	200.00069	5885.5570
GWO	0.779022	0.384654	40.327793	199.65031	5889.3682
PSO	0.778956	0.384681	40.320913	200.00000	5891.3872

GSA	1.085803	0.949619	49.345231	169.48739	11550.2977
MFO	0.835241	0.409834	43.578621	152.21515	6055.6358

Table 5.13 Statistical results for pressure vessel design problem on CEPO

Algorithm	BEST	MEAN	WORST	STD DEV.	MEDIAN
CEPO	**5885.5669**	**5887.6769**	5892.5270	**002.889**	**5884.2199**
EPO	5885.5769	5887.4439	5892.3308	002.889	5886.2279
SHO	5885.5770	5887.4445	5892.3435	002.890	5886.2280
GWO	5889.3628	5891.5252	5894.6243	013.912	5890.6494
PSO	5891.3872	6531.5030	7394.5880	534.113	6416.1134
GSA	11550.2977	23342.2910	**333226.2555**	5790.631	24010.0425
MFO	6055.6358	6360.6859	7023.8539	365.589	6302.2305

5.4.3. Speed reducer design problem: Problem as earlier discussed in section 4.5.4 Table 4.14 and Table 4.15 shows results for speed reducer problem. CEPO shows optimal solution.

Table 5.14 Comparison results for speed reducer design problem on CEPO

Algorithm	Optimum Variables							Optimum Cost
	B	m	Z	l_1	l_2	d_1	d_2	
CEPO	3.50161	0.7	17	7.3	7.8	3.35123	5.2889	**2998.5488**
EPO	3.50162	0.7	17	7.3	7.8	3.35127	5.28871	2998.5580
SHO	3.50162	0.7	17	7.3	7.8	3.35127	5.28871	2998.5500
GWO	3.506689	0.7	17	7.380923	7.815722	3.357851	5.286756	3001.278
PSO	3.500014	0.7	17	8.3	7.8	3.352418	5.286718	3005.759
GSA	3.600002	0.7	17	8.3	7.8	3.369651	5.289219	3051.116
MFO	3.507526	0.7	17	7.302399	7.802364	3.323545	5.287529	3009.578

Table 5.15 Statistical results for speed reducer design problem on CEPO

Algorithm	BEST	MEAN	WORST	STD DEV.	MEDIAN
CEPO	**2998.5488**	**2999.633**	3003.888	**1.93192**	**2999.192**
EPO	2998.5580	2999.637	3003.891	1.93192	2999.196
SHO	2998.5500	2999.639	3005.991	2.93566	2999.199
GWO	3001.278	3005.845	3008.759	5.83789	3004.523
PSO	3005.759	3105.258	3211.169	799.6389	3105.265
GSA	3051.116	3170.328	**3363.869**	92.5723	3156.752
MFO	3009.578	3021.251	3054.521	11.0245	3020.534

5.5 Summary

In this chapter, a chaotic version of EPO is designed and implemented. To analyze the performance the CEPO is validated on twenty-nine well known test functions. The results show that CEPO provides improved and competitive results than EPO and other metaheuristic algorithms used for comparison purpose. Proposed algorithm is very well able to handle constraint problems .Therefore provides the optimal results than other algorithms.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

To enhance the search process of SHO and EPO and overcome the premature convergence, chaos theory is introduced in SHO and EPO.

- Various chaotic maps have been utilized for proposed approach. The performance of proposed algorithms is evaluated on twenty nine benchmark test functions .The results obtained are better than the original SHO and EPO.
- Comparative analysis is performed with other metaheuristic algorithms confirmed that the proposed approach is better than other metaheuristic techniques.
- Integration of chaotic theory is able to significantly improve the performance of SHO and EPO.
- Among all the chaotic maps, the Tent map has considerably enhanced the performance for both SHO and EPO.
- The chaos theory integrated with the classical SHO and EPO to explore the search space is the responsible for enhancing the performance of both algorithms.
- Validation on constrained engineering problem has also shown significant results for both CSHO and CEPO.

6.2 Future Scope

- In future the proposed algorithms can be used for solving different optimization problems in field of research.
- Proposed algorithms can be extended to solve multi-objective optimization problems.
- In addition, more chaotic maps can be integrated for improving search strategy.

References

[1] Yang X. S., "Nature-inspired metaheuristic algorithms: success and new challenge", *Journal of Computer Engineering and Information Technology*, vol. 1, no.1, pp. 1–3, 2012.

[2] Kumar V., Chhabra J.K. and Kumar D. , "Parameter adaptive harmony search algorithm for unimodal and multimodal optimization problems", *Journal of Computational Science*, vol. 5, no. 2, pp144–155, 2014.

[3] Gaurav Dhiman and Vijay Kumar, "Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications", *Advances in engineering software*, vol. 114, pp.48-70, 2017.

[4] Osman, I. H., and Laporte, G., "Metaheuristic: A bibliography", *Annals of Operations Research*, vol. 63, no. 5, pp. 511–623, 1996.

[5] Mirjalili S, Mirjalili SM and Lewis A, "Grey wolf optimizer", *Advances in Engineering Software*, vol. 69, pp. 46-61, 2014.

[6] O. Olorunda and A.P. Engelbrecht, "Measuring exploration/exploitation in particle swarms using swarm diversity", *IEEE Congress on Evolutionary Computaion*, pp. 1128–1134, 2008.

[7]E. Alba and B. Dorronsoro, "The exploration/exploitation tradeoff in dynamic cellular genetic algorithms", *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 2, pp 126–142, 2005.

[8] M. Lozano and C. Garcia-Martinez, "Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: overview and progress report", *Computers and Operations research*, vol. 37, no. 3, pp. 481–497, 2010.

[9] Seyedali Mirjalili and Andrew Lewis, "The Whale Optimization Algorithm", *Advances in Engineering Software* ,vol. 95,pp.51–67, 2016.

[10] Yang, X.-S. and He, X., "Bat Algorithm: Literature review and applications", *International Journal of Bio-Inspired Computation*, vol. 5, no. 3, pp.141–149, 2013.

[11] Gaurav Dhimana and Vijay kumar, "Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems", *Knowledge based Systems*, vol. 165, pp. 169-196, 2019.

[12] Kennedy J. and Eberhart R.C., "Particle swarm optimization", *In: Proceedings of IEEE International Conference on Neural Networks*, pp. 1942–1948, 1995.

[13] Bijaya Kumar Nanda and Gyanesh Das, "Ant colony optimization - a computational intelligence technique", *International Journal of Computer & Communication Technology*, vol. 3, no. 2, 2013.

[14] I. Boussaïd, J. Lepagnot and P. Siarry, "A survey on optimization metaheuristics", *Information Sciences*, vol 237, pp. 82-117, 2013.

[15] Storn R. and Price K.,"Differential evolution -a simple and efficient heuristic for global optimization over continuous spaces", *Journal of Global Optimization*, vol.11, no. 4, pp. 341–359, 1997.

[16] Kirkpatrick S., Gelatt C.D. and Vecchi M.P., "Optimization by simulated annealing". *Science* ,vol.220, no. 4598, pp. 671–680, 1983.

[17]Rashedi E, Nezamabadi-pour H and Saryazdi S., "GSA: A gravitational search algorithm", *Information Sciences*, vol. 179, no. 13, pp.2232–2248, 2009.

[18] Hatamlou A, "Black hole: a new heuristic optimization approach for data clustering", *Information Sciences*, vol. 222, pp.175–184, 2013.

[19] Gholizadeh, S. and Barati, H., "Comparative study of three metaheuristics for optimum design of trusses", *International Journal of Optimization in Civil Engineering*, vol. 3, pp. 423-441, 2012.

[20] Talbi, E., "Metaheuristics: From Design to Implementation", *Wiley Publishing*, 2008.

[21] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization", *IEEE Transactions on Evolutionary Computation*, vol. 1,no. 1, pp. 67-82, 1997.

[22] Dhiman G. and Kumar V., " Emperor penguin optimizer: A bio-inspired algorithm for engineering problems", *Knowledge-Based Systems*, vol. 159, pp. 20-50, 2018.

[23] Rui Tang, Simon Fong and Nilanjan Dey, " Metaheuristics and Chaos Theory", *Chapter from the book Chaos Theory*, 2017.

[24] Kellert S.H., "In the wake of chaos: Unpredictable order in dynamical systems",*University of Chicago Press*, 1993.

[25] Li B. and Jiang W., "Optimization complex functions by chaos search", *Cybernetics and Systems*, vol. 29, no. 4, pp. 409–419, 1998.

[26] Yang J.J., Zhou J.Z., Wu W., Liu F., Zhu C.J. and Cao G.J., "A chaos algorithm based on progressive optimality and Tabu search algorithm", *International Conference on Machine Learning and Cybernetics*, vol. 5, pp. 2977–2981, 2005.

[27] S. Saremi, S. Mirjalili, and A. Lewis, "Biogeography-based optimisation with chaos", *Neural Computing and Applications*, vol. 25, no. 5, pp. 1077–1097, 2014.

[28] Razi Sheikholeslami and A. Kaveh, “A survey of chaos embedded metaheuristics”, *International journal of optimization in civil engineering*, vol. 3, no. 4, pp 617-633, 2013.

[29] Yang, D., Li, G. and Cheng, G., “On the efficiency of chaos optimization algorithms for global optimization”, *Chaos, Solitons and Fractals*, vol.34, no. 4, pp.1366–1375, 2007.

[30] Kohli M. and Arora S., “Chaotic grey wolf optimization algorithm for constrained optimization problems”. *Journal of Computational Design and Engineering*, vol. 3, no. 4, pp. 458-472, 2017.

[31] G. Kaur and S. Arora, “Chaotic Whale Optimization Algorithm”, *Journal of Computational Design and Engineering*, vol. 5, no. 3, pp. 275-284, 2018.

[32] S. Mirjalili, “Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm”, *Knowledge Based Systems*, vol. 89, pp. 228–249, 2015.

[33] Hossam M. Zawabaa and Eid Emay, “Impact of Chaos Functions on Modern Swarm Optimizers”, Plos one, vol. 11, no. 7, 2016.

[34] Seyedali Mirjalili and Amir H.Gandomi, “Chaotic gravitational constants for the gravitational search algorithm”, Applied soft computing, vol 53,pp. 407-419, 2017.

[35] Mittal H., Pal R., Kulhari A. and Saraswat M., “ Chaotic Kbest gravitational search algorithm”, *Ninth International Conference on Contemporary Computing*, pp. 1-6, 2016.

[36] Gandomi A. H., Yun G. J., Yang X.-S. and Talatahari S. , “Chaos-enhanced accelerated particle swarm optimization”, *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no.2, pp 327–340, 2013.

[37] Gandomi A. H. and Yang X.-S., “Chaotic bat algorithm”, *Journal of Computational Science*, vol.5 no. 2, pp. 224–232, 2014.

[38] S. Saremi, S. M. Mirjalili, and S. Mirjalili, “Chaotic krill herd optimization algorithm”, *Procedia Technology*, vol. 12, pp. 180–185, 2014.

[39] Yang X, “Firefly algorithm, Stochastic test functions and design optimisation”, *International Journal of Bio-Inspired Computation*, vol. 2, no.2, pp. 78–84, 2010.

[40] A. Gandomi, X.-S. Yang, S. Talatahari, and A. Alavi, "Firefly algorithm with chaos", *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no.1, pp. 89-98, 2013.

[41] Digalakis J and Margaritis K., “On benchmarking functions for genetic algorithms”, *International Journal of Computer Mathematics*, vol. 77, no. 4, pp. 481–506, 2001.

[42] Liang J.J., Suganthan P.N. and Deb K., “Novel composition test functions for numerical global optimization”, *Proceedings IEEE Swarm Intelligence Symposium*, pp. 68–75, 2005.

[43] Suganthan P.N. , Hansen N. , Liang J.J. , Deb K. , Chen Y.-P. , Auger A . andet al., "Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization Technical Report", *Singapore: Nanyang Technological University*, 2005.

[44] Karaboga D. and Basturk B., "Artificial bee colony (abc) optimization algorithm for solving constrained optimization problems", *Foundations of fuzzy logic and soft computing*, pp. 789–798, 2007.

[45] Cagnina L. C., Esquivel, S. C. and Coello C. A. C., "Solving engineering optimization problems with the simple constrained particle swarm optimizer", *Informatica*, vol 32, no. 3, pp. 319-326, 2008.

[46] Coello C. A. C., "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art", *Computer Methods in Applied Mechanics and Engineering*, vol.191, no.12, pp.1245-1287, 2002.

List of publications

[1] Avneet Kaur , Kamalinder Kaur Kaleka and Vijay Kumar, "Design and Development of Soft Computing Toolbox Using Metaheuristics" , in *International Conference on Sustainable Computing in Science, Technology & Management(SUSCOM-2019)* [Published].

[2] Kamalinder Kaur Kaleka, Avneet Kaur, Vijay Kumar ,"A Conceptual Comparison of Metaheuristic Algorithms and Applications to Engineering Design Problems".International Journal of Intelligent Information and Database Systems [Communicated].

[3]Vijay Kumar, Kamalinder Kaur Kaleka, "Chaotic Spotted Hyena Optimizer for Engineering Design Problems", *Engineering with Computers* [Communicated].

[4] Vijay Kumar; Kamalinder Kaur Kaleka ,"Chaotic Emperor Penguin Optimizer for Solving Engineering Design Problems" [To be communicated].

Appendix A. Benchmark text functions

Table A.1 Unimodal benchmark test functions

Benchmark Test Function	Dim	Range	F_{min}
$F_1(Y)=\sum_{i=1}^{D} y_i^2$	5	$[-100,100]$	0
$F_2(Y)=\sum_{i=1}^{D}\lvert y_i\rvert+\prod_{i=1}^{D}\lvert y_i\rvert$	5	$[-10,10]$	0
$F_3(Y)=\sum_{i=1}^{D}\left(\sum_{j=1}^{i} y_j\right)^2$	5	$[-100,100]$	0
$F_4(Y)=\max_i\left\{\lvert y_i\rvert,\ 1\le i\le D\right\}$	5	$[-100,100]$	0
$F_5(Y)=\sum_{i=1}^{D-1}\left[100\left(y_{i+1}-y_i^2\right)^2+\left(y_i-1\right)^2\right]$	5	$[-30,30]$	0
$F_6(Y)=\sum_{i=1}^{D}\left(\lfloor y_i+0.5\rfloor\right)^2$	5	$[-100,100]$	0
$F_7(Y)=\sum_{i=1}^{D} iy_i^4+random[0,1)$	5	$[-1.28,1.28]$	0

Table A.2 Multimodal benchmark test functions

Objective Function	Dim	Range	F_{min}
$F_8(Y)=-\frac{1}{D}\sum_{i=1}^{D}\left(y_i\ sin\left(\sqrt{\lvert y_i\rvert}\right)\right)$	5	$[-500,500]$	-2094.914
$F_9(Y)=\sum_{i=1}^{D}\left[y_i^2-10\cos\left(2\pi y_i\right)+10\right]$	5	$[-5.12,5.12]$	0
$F_{10}(Y)=-20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}y_i^2}\right)-\exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos\left(2\pi y_i\right)\right)+20+e$	5	$[-32,32]$	0

$F_{11}(Y)=\frac{1}{4000}\sum_{i=1}^{D}y_i^2-\prod_{i=1}^{D}\cos\left(\frac{y_i}{\sqrt{i}}\right)+1$	5	$[-600,600]$	0
$F_{12}(Y)=\frac{\pi}{D}\left\{10\sin(\pi z_1)+\sum_{i=1}^{D-1}(z_i-1)^2\left[1+10\sin^2(\pi z_{i+1})\right]+(z_D-1)^2\right\}+\sum_{i=1}^{D}u(y_i,10,100,4)$ *where* $u(y_i,a,k,m)=\begin{cases}k(y_i-a)^m, & y_i>a,\\ 0, & -a\le y_i\le a,\\ k(-y_i-a)^m, & y_i<-a\end{cases}$ $z_i=1+\frac{1}{4}(y_i+1)$	5	$[-50,50]$	0
$F_{13}(Y)=0.1\left\{\sin^2(3\pi y_1)+\sum_{i=1}^{D-1}(y_i-1)^2\left[1+\sin^2(3\pi y_{i+1})\right]+(y_D-1)^2\left[1+\sin^2(2\pi y_D)\right]\right\}+\sum_{i=1}^{D}u(y_i,5,100,4)$	5	$[-30,30]$	0

Table A.3 Multimodal benchmark test functions with fixed dimension

Objective Function	Dim	Range	F_{min}
$F_{14}(Y)=\left(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right)^{-1}$	2	[–65, 65]	1
$F_{15}(Y)=\sum_{i=1}^{11}\left[a_i-\frac{y_1(b_i^2+b_iy_2)}{b_i^2+b_iy_3+y_4}\right]^2$	4	[–5, 5]	0.00030
$F_{16}(Y)=4y_1^2-2.1y_1^4+\frac{1}{3}y_1^6+y_1y_2-4y_2^2+4y_2^4$	2	[–5, 5]	-1.0316
$F_{17}(Y)=\left(y_2-\frac{5.1}{4\pi^2}y_1^2+\frac{5}{\pi}y_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos y_1+10$	2	[–5, 5]	0.398
$F_{18}(Y)=\left[1+(y_1+y_2+1)^2(19-14y_1+6y_1y_2+3y_2^2)\right]\times\left[30+(2y_1-3y_2)^2\times(18-32y_1+12y_1^2+48y_2-36y_1y_2+27y_2^2)\right]$	2	$[-2,2]$	3

$F_{19}(Y) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	3	$[1,3]$	-3.86
$F_{20}(Y) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	6	$[0,1]$	-3.32
$F_{21}(Y) = -\sum_{i=1}^{5}\left[\left(Y - a_i\right)\left(Y - a_i\right)^T + c_i\right]^{-1}$	4	$[0,10]$	-10.1532
$F_{22}(Y) = -\sum_{i=1}^{7}\left[\left(Y - a_i\right)\left(Y - a_i\right)^T + c_i\right]^{-1}$	4	$[0,10]$	-10.4028
$F_{23}(Y) = -\sum_{i=1}^{10}\left[\left(Y - a_i\right)\left(Y - a_i\right)^T + c_i\right]^{-1}$	4	$[0,10]$	-10.5363

Table A.4 Composite benchmark test functions

Objective Function	Dim	Range	F_{min}
$F_{24}(CF1)$: $f_1, f_2, f_3, ..., f_{10} = Sphere\ Function$ $[\sigma_1, \sigma_2, \sigma_3, ..., \sigma_{10}] = [1,1,1,.....,1]$ $[\lambda_1, \lambda_2, \lambda_3, ..., \lambda_{10}] = [5/100,\ 5/100,\ 5/100,\ ......, 5/100]$	10	$[-5,\ 5]$	0
$F_{25}(CF2)$: $f_1, f_2, f_3, ..., f_{10} = Griewank\text{'s}\ Function$ $[\sigma_1, \sigma_2, \sigma_3, ..., \sigma_{10}] = [1,1,1,.....,1]$ $[\lambda_1, \lambda_2, \lambda_3, ..., \lambda_{10}] = [5/100,\ 5/100,\ 5/100,\ ......, 5/100]$	10	$[-5,\ 5]$	0
$F_{26}(CF3)$: $f_1, f_2, f_3, ..., f_{10} = Griewank\text{'s}\ Function$ $[\sigma_1, \sigma_2, \sigma_3, ..., \sigma_{10}] = [1,1,1,.....,1]$ $[\lambda_1, \lambda_2, \lambda_3, ..., \lambda_{10}] = [1,1,1,.....,1]$	10	$[-5,\ 5]$	0

$F_{27}(CF4)$: $f_1, f_2 = Ackley's\ Function,\quad f_3,\ f_4 = Rastrigin's\ Function$ $f_5, f_6 = Weierstrass\ Function,\quad f_7,\ f_8 = Griewank's\ Function$ $f_9, f_{10} = Sphere\ Function$ $[\sigma_1, \sigma_2, \sigma_3, ..., \sigma_{10}] = [1,1,1,.....,1]$ $[\lambda_1, \lambda_2, \lambda_3, ..., \lambda_{10}] = [5/32, 5/32, 1, 1, 5/0.5, 5/0.5, 5/100, 5/100, 5/100, 5/100]$	10	[−5, 5]	0
$F_{28}(CF5)$: $f_1, f_2 = Rastrigin's\ Function,\quad f_3,\ f_4 = Weierstrass\ Function$ $f_5, f_6 = Griewank's\ Function,\quad f_7,\ f_8 = Ackley's\ Function$ $f_9, f_{10} = Sphere\ Function$ $[\sigma_1, \sigma_2, \sigma_3, ..., \sigma_{10}] = [1,1,1,.....,1]$ $[\lambda_1, \lambda_2, \lambda_3, ..., \lambda_{10}] = [1/5, 1/5, 5/0.5, 5/0.5, 5/100, 5/100, 5/32, 5/32, 5/100, 5/100]$	10	[−5, 5]	0
$F_{29}(CF6)$: $f_1, f_2 = Rastrigin's\ Function,\quad f_3,\ f_4 = Weierstrass\ Function$ $f_5, f_6 = Griewank's\ Function,\quad f_7,\ f_8 = Ackley's\ Function$ $f_9, f_{10} = Sphere\ Function$ $[\sigma_1, \sigma_2, \sigma_3, ..., \sigma_{10}] = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]$ $[\lambda_1, \lambda_2, \lambda_3, ..., \lambda_{10}] = \begin{bmatrix} 0.1\times 1/5, 0.2\times 1/5, 0.3\times 5/0.5, 0.4\times 5/0.5, 0.5\times 5/100, \\ 0.6\times 5/100, 0.7\times 5/32, 0.8\times 5/32, 0.9\times 5/100, 1\times 5/100 \end{bmatrix}$	10	[−5, 5]	0

Appendix B. Engineering Design Problems

B.1. Welded beam design problem

$$\textit{Minimize} \quad F(y) = 1.10471 y_1^2 y_2 + 0.0481 y_3 y_4 (14.0 + y_2)$$

$$\begin{aligned}
\textit{Subject to} \quad & g_1(y) = \tau(y) - \tau_{\max} \le 0, \\
& g_2(y) = \sigma(y) - \sigma_{\max} \le 0, \\
& g_3(y) = y_1 - y_4 \le 0, \\
& g_4(y) = \delta(y) - \delta_{\max} \le 0 \\
& g_5(y) = P - P_c(y) \le 0 \\
& g_6(y) = 0.125 - y_1 \le 0 \\
& g_7(y) = 1.10471 y_1^2 + 0.04811 y_3 y_4 (14.0 + y_2) - 5.0 \le 0
\end{aligned}$$

$$\textit{Range} \quad 0.1 \le y_1 \le 2, \quad 0.1 \le y_2 \le 10, \quad 0.1 \le y_3 \le 10, \quad 0.1 \le y_4 \le 2$$

where

$$\tau(y) = \sqrt{\tau_1^2 + 2\tau_1\tau_2\left(\frac{y_2}{2R}\right) + \tau_2^2} \quad ; \quad \tau_1 = \frac{P}{\sqrt{2} y_1 y_2} \quad ; \quad \tau_2 = \frac{MR}{J}$$

$$M = P\left(L + \frac{y_2}{2}\right) \; ; \quad J(y) = 2\left\{\sqrt{2} y_1 y_2 \left[\frac{y_2^2}{4} + \left(\frac{y_1 + y_3}{2}\right)^2\right]\right\} ; \quad R = \sqrt{\frac{y_2^2}{4} + \left(\frac{y_1 + y_3}{2}\right)^2}$$

$$P_c(y) = \frac{4.013E\sqrt{\frac{y_3^2 y_4^6}{36}}}{L^2}\left(1 - \frac{y_3}{2L}\sqrt{\frac{E}{4G}}\right); \quad \sigma(y) = \frac{6PL}{y_4 y_3^2} \; ; \quad \delta(y) = \frac{6PL^3}{E y_3^3 y_4}$$

$$P = 6000\,lb, \quad L = 15\,in, \quad G = 12 \times 10^6\ psi, \quad E = 30 \times 10^6\ psi$$

$$\delta_{max} = 0.25\,in, \quad \tau_{max} = 13600\,psi, \quad \sigma_{max} = 30000\,psi$$

B.2. Tension/compression spring design problem

$$\textit{Minimize} \quad F(y) = (y_3 + 2) y_2 y_1^2$$

$$\begin{aligned}
\textit{Subject to} \quad & g_1(y) = 1 - \frac{y_2^3 y_3}{7178 y_1^4} \le 0, \\
& g_2(y) = \frac{4y_2^2 - y_1 y_2}{12566\left(y_2 y_1^3 - y_1^4\right)} + \frac{1}{5108 y_1^2} - 1 \le 0, \\
& g_3(y) = 1 - \frac{140.45 y_1}{y_2^2 y_3} \le 0, \\
& g_4(y) = \frac{y_2 + y_1}{1.5} - 1 \le 0
\end{aligned}$$

B.3. Pressure vessel design problem

$$\textit{Minimize} \quad F(y) = 0.6224 y_1 y_2 y_3 y_4 + 1.7781 y_2 y_3^2 + 3.1661 y_1^2 y_4 + 19.84 y_1^3 y_3$$

$$\begin{aligned} \textit{Subject to} \quad & g_1(y) = -y_1 + 0.0193 y_3 \le 0, \\ & g_2(y) = -y_2 + 0.00954 y_3 \le 0, \\ & g_3(y) = -\pi y_3^2 y_4 - \frac{4}{3}\pi y_3^3 + 1296000 \le 0, \\ & g_4(y) = y_4 - 240 \le 0 \end{aligned}$$

B.4. Speed reducer design problem

$$\begin{aligned} \textit{Minimize} \quad F(Y) = {} & 0.785 y_1 y_2^2 \left(3.3333 y_3^2 + 14.9334 y_3 - 43.0934\right) - 1.508 y_1 \left(y_6^2 + y_7^2\right) + \ldots \\ & \ldots.\ 7.4777\left(y_6^3 + y_7^3\right) + 0.78054\left(y_4 y_6^2 + y_5 y_7^2\right) \end{aligned}$$

$$\begin{aligned} \textit{Subject to} \quad & g_1(y) = \frac{27}{y_1 y_2^2 y_3} - 1 \le 0, \qquad && g_2(y) = \frac{397.5}{y_1 y_2^2 y_3^2} - 1 \le 0, \\ & g_3(y) = \frac{1.93 y_4^3}{y_2 y_3 y_6^4} - 1 \le 0, && g_4(y) = \frac{1.93 y_5^3}{y_2 y_3 y_7^4} - 1 \le 0, \\ & g_5(y) = \frac{1.0}{110 y_6^3} \sqrt{\left(\frac{750.0 y_5}{y_2 y_3}\right)^2 + 16.9 \times 10^6} - 1 \le 0, \\ & g_6(y) = \frac{1.0}{85 y_7^3} \sqrt{\left(\frac{750.0 y_5}{y_2 y_3}\right)^2 + 157.5 \times 10^6} - 1 \le 0, \\ & g_7(y) = \frac{y_2 y_3}{40} - 1 \le 0, && g_8(y) = \frac{5 y_2}{y_1} - 1 \le 0, \\ & g_9(y) = \frac{y_1}{12 y_2} - 1 \le 0, && g_{10}(y) = \frac{1.5 y_6 + 1.9}{y_4} - 1 \le 0, \\ & g_{11}(y) = \frac{1.1 y_7 + 1.9}{y_5} - 1 \le 0 \end{aligned}$$

B.5. Rolling element bearing design problem

$$\begin{aligned} \textit{Maximize} \quad & C_d = f_c Z^{2/3} D_b^{1.8} && \textit{if } D \le 25.4\,mm \\ & C_d = 3.647 f_c Z^{2/3} D_b^{1.4} && \textit{if } D > 25.4\,mm \end{aligned}$$

Subject to $g_1(y)=\dfrac{\phi_0}{2sin^{-1}(D_b/D_m)}-Z+1\ge 0,$

$$g_2(y)=2D_b-K_{Dmin}(D-d)\ge 0,$$

$$g_3(y)=K_{Dmax}(D-d)-2D_b\ge 0,$$

$$g_4(y)=\zeta B_w-D_b\le 0,$$

$$g_5(y)=D_m-0.5(D+d)\ge 0,$$

$$g_6(y)=(0.5+e)(D+d)-D_m\ge 0,$$

$$g_7(y)=0.5(D-D_m-D_b)-\varepsilon D_b\ge 0,$$

$$g_8(y)=f_i\ge 0.515,$$

$$g_9(y)=f_0\ge 0.515,$$

where

$$f_c=37.91\left[1+\left\{1.04\left(\frac{1-\gamma}{1+\gamma}\right)^{1.72}\left(\frac{f_i(2f_0-1)}{f_0(2f_i-1)}\right)^{0.41}\right\}^{10/3}\right]^{-0.3}\times\left[\frac{\gamma^{0.3}(1-\gamma)^{1.39}}{(1+\gamma)^{1/3}}\right]\left[\frac{2f_i}{2f_i-1}\right]^{0.41}$$

$$\phi_0=2\pi-2\cos^{-1}\left(\frac{\left[\{(D-d)/2-3(T/4)\}^2+\{D/2-T/4-D_b\}^2-\{d/2+T/4\}^2\right]}{2\{(D-d)/2-3(T/4)\}\{D/2-T/4-D_b\}}\right)$$

$$\gamma=\frac{D_b}{D_m},\quad f_i=\frac{r_i}{D_b},\quad f_0=\frac{r_0}{D_b},\quad T=D-d-2D_b$$

$$D=160,\quad d=90,\quad B_w=30,\quad r_i=r_0=11.033$$

Range $0.4\le K_{Dmin}\le 0.5,\ 0.6\le K_{Dmax}\le 0.7,\ 0.3\le e\le 0.4,\ 0.02\le e\le 0.1,\ 0.6\le\zeta\le 0.85$

$0.5(D+d)\le D_m\le 0.6(D+d),\ 0.15(D-d)\le D_b\le 0.45(D-d),\ 4\le Z\le 50,\ 0.515\le f_i, f_0\le 0.6$

Kamalinder Thesis

ORIGINALITY REPORT

19%	11%	14%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Gaurav Dhiman, Vijay Kumar. "Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications", Advances in Engineering Software, 2017 Publication	2%
2	Gaurav Dhiman, Vijay Kumar. "Emperor penguin optimizer: A bio-inspired algorithm for engineering problems", Knowledge-Based Systems, 2018 Publication	1%
3	Mehak Kohli, Sankalap Arora. "Chaotic grey wolf optimization algorithm for constrained optimization problems", Journal of Computational Design and Engineering, 2017 Publication	<1%
4	mcf.gsfc.nasa.gov Internet Source	<1%
5	www.ornl.gov Internet Source	<1%

6	tesis.pucp.edu.pe Internet Source	<1%
7	Gaganpreet Kaur, Sankalap Arora. "Chaotic Whale Optimization Algorithm", Journal of Computational Design and Engineering, 2018 Publication	<1%
8	Gaurav Dhiman, Vijay Kumar. "Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems", Knowledge-Based Systems, 2018 Publication	<1%
9	Emary, E., and Hossam M. Zawbaa. "Impact of Chaos Functions on Modern Swarm Optimizers", PLoS ONE, 2016. Publication	<1%
10	Vijay Kumar, Avneet Kaur. "Binary spotted hyena optimizer and its application to feature selection", Journal of Ambient Intelligence and Humanized Computing, 2019 Publication	<1%
11	Advances in Metaheuristic Algorithms for Optimal Design of Structures, 2014. Publication	<1%
12	C. Wang, K. Z. Gao, J. Guo. "An improved gravitational search algorithm based on	<1%

neighbor search", 2013 Ninth International Conference on Natural Computation (ICNC), 2013
Publication

13	Gaurav Dhiman, Amandeep Kaur. "STOA: A bio-inspired based optimization algorithm for industrial engineering problems", Engineering Applications of Artificial Intelligence, 2019 Publication	<1%
14	B. K. Chatterjee, S. C. Roy. "Tables of Elastic Scattering Cross Sections of Photons in the Energy Range 50–1500 keV for All Elements in the Range 13 Z 104", Journal of Physical and Chemical Reference Data, 1998 Publication	<1%
15	code.google.com Internet Source	<1%
16	shengxinzhang.github.io Internet Source	<1%
17	wise-obs.tau.ac.il Internet Source	<1%
18	Draa, Amer, Samira Bouzoubia, and Imene Boukhalfa. "A sinusoidal differential evolution algorithm for numerical optimisation", Applied Soft Computing, 2015. Publication	<1%

19 tel.archives-ouvertes.fr
Internet Source
<1%

20 Chen, G.Q.. "Carbon emissions and resources use by Chinese economy 2007: A 135-sector inventory and input-output embodiment", Communications in Nonlinear Science and Numerical Simulation, 201011
Publication
<1%

21 Javier Miranda, Gregory Lapicki. "Experimental cross sections for L-shell x-ray production and ionization by protons", Atomic Data and Nuclear Data Tables, 2014
Publication
<1%

22 www.docstoc.com
Internet Source
<1%

23 Ke Ding, Ying Tan. "Comparison of random number generators in Particle Swarm Optimization algorithm", 2014 IEEE Congress on Evolutionary Computation (CEC), 2014
Publication
<1%

24 Vijay Kumar, Dinesh Kumar. "An astrophysics-inspired Grey wolf algorithm for numerical optimization and its application to engineering design problems", Advances in Engineering Software, 2017
Publication
<1%

25	"Six Sigma Performance Measurement System", Springer Nature, 2008 Publication	<1%
26	debian.mcc.ac.uk Internet Source	<1%
27	Gaurav Dhiman, Amandeep Kaur. "Spotted Hyena Optimizer for Solving Engineering Design Problems", 2017 International Conference on Machine Learning and Data Science (MLDS), 2017 Publication	<1%
28	"Soft Computing for Problem Solving", Springer Science and Business Media LLC, 2019 Publication	<1%
29	toyama.repo.nii.ac.jp Internet Source	<1%
30	Gaurav Dhiman, Vijay Kumar. "Multi-objective spotted hyena optimizer: A Multi-objective optimization algorithm for engineering problems", Knowledge-Based Systems, 2018 Publication	<1%
31	Qamar Abbas, Jamil Ahmad, Hajira Jabeen. "A Novel Tournament Selection Based Differential Evolution Variant for Continuous Optimization Problems", Mathematical Problems in Engineering, 2015	<1%

Publication

32	jjcit.org Internet Source	<1%
33	www-cfadc.phy.ornl.gov Internet Source	<1%
34	Kybernetes, Volume 42, Issue 6 (2013-09-21) Publication	<1%
35	Gaganpreet Kaur, Sankalap Arora. "Chaotic whale optimization algorithm", Journal of Computational Design and Engineering, 2018 Publication	<1%
36	Vainshtein, L.A.. "Dielectronic satellite spectra for highly charged H-like ions (2l'3l"-1s2l, 2l'3l"-1s3l) and He-like ions (1s2l'3l"-1s^22l, 1s2l'3l"-1s^23l) with Z = 6-33", Atomic Data and Nuclear Data Tables, 198007 Publication	<1%
37	Zhiming Li, Yongquan Zhou, Sen Zhang, Junmin Song. "Lévy-Flight Moth-Flame Algorithm for Function Optimization and Engineering Design Problems", Mathematical Problems in Engineering, 2016 Publication	<1%
38	Gaurav Dhiman, Amandeep Kaur. "Optimizing the Design of Airfoil and Optical Buffer Problems Using Spotted Hyena Optimizer",	<1%

Designs, 2018
Publication

39	"Advances in Swarm Intelligence", Springer Science and Business Media LLC, 2017 Publication	<1%
40	Roser Martí-Cid, David Huertas, Martí Nadal, Victoria Linares, Marta Schuhmacher, Joan O. Grimalt, José L. Domingo. "Dietary Exposure to Organochlorine Compounds in Tarragona Province (Catalonia, Spain): Health Risks", Human and Ecological Risk Assessment: An International Journal, 2010 Publication	<1%
41	Bo Xing, Wen-Jing Gao. "Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms", Springer Nature, 2014 Publication	<1%
42	M. A. Ordal. "Optical properties of the metals Al, Co, Cu, Au, Fe, Pb, Ni, Pd, Pt, Ag, Ti, and W in the infrared and far infrared", Applied Optics, 04/01/1983 Publication	<1%
43	Mirjalili, Seyedali, and Andrew Lewis. "The Whale Optimization Algorithm", Advances in Engineering Software, 2016. Publication	<1%

www.sto.fr

44 Internet Source <1%

45 www.mkg.se Internet Source <1%

46 Beigman, I.L.. "EXCITATION AND IONIZATION CROSS SECTIONS FOR HE I FROM NORMALIZED BORN AND K-MATRIX CALCULATIONS: @DS = 0 TRANSITIONS FROM n = 2, 3 EXCITED STATES", Atomic Data and Nuclear Data Tables, 200001 Publication <1%

47 D. R. White, R. V. Griffith, I. J. Wilson. "Report 46", Journal of the International Commission on Radiation Units and Measurements, 1992 Publication <1%

48 www.astronomy.ohio-state.edu Internet Source <1%

49 link.springer.com Internet Source <1%

50 Lijin Wang, Yiwen Zhong. "Cuckoo Search Algorithm with Chaotic Maps", Mathematical Problems in Engineering, 2015 Publication <1%

51 Pradeep Gupta, K. P. S. Rana, Vineet Kumar, Puneet Mishra, Jitendra Kumar, Sreejith S. Nair. "Development of a Grey Wolf Optimizer <1%

Toolkit in LabVIEW™", 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), 2015
Publication

52	www.medoc-ias.u-psud.fr Internet Source	<1%
53	goes.ngdc.noaa.gov Internet Source	<1%
54	epdf.pub Internet Source	<1%
55	Management of Environmental Quality: An International Journal, Volume 25, Issue 6 (2014-09-16) Publication	<1%

Exclude quotes	On	Exclude matches	< 8 words
Exclude bibliography	On		