



Chaotic marine predators algorithm for global optimization of real-world engineering problems



Sumit Kumar ^a, Betul Sultan Yildiz ^b, Pranav Mehta ^c, Natee Panagant ^d, Sadiq M. Sait ^e, Seyedali Mirjalili ^{f,g}, Ali Riza Yildiz ^{b,*}

^a Australian Maritime College, College of Sciences and Engineering, University of Tasmania, Launceston, 7248, Australia

^b Department of Mechanical Engineering, Bursa Uludag University, Gorukle, Bursa, Turkey

^c Department of Mechanical Engineering, Dharmsinh Desai University, Nadiad 387001, India

^d Sustainable Infrastructure Research and Development Centre, Department of Mechanical Engineering, Faculty of Engineering, Khon Kaen University, 40002, Thailand

^e Department of Computer Engineering, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia

^f Centre for Artificial Intelligence Research and Optimization, Torrens University, 90 Bowen Terrace, Fortitude Valley QLD 4006, Australia

^g Yonsei Frontier Lab, Yonsei University, Seoul, South Korea

ARTICLE INFO

Article history:

Received 24 September 2022

Received in revised form 21 November 2022

Accepted 10 December 2022

Available online 15 December 2022

Keywords:

Marine Predators Algorithm

Chaotic maps

Global optimization

Engineering design problems

Metaheuristic algorithms

ABSTRACT

A novel metaheuristic called Chaotic Marine Predators Algorithm (CMPA) is proposed and investigated for the optimization of engineering problems. CMPA integrates the exploration merits of the recently proposed Marine Predators Algorithm (MPA) with the chaotic maps exploitation capabilities. Several chaotic maps were applied in the proposed CMPA to govern MPA parameters that eventually led to controlled exploration and exploitation of search. This study makes an initial attempt to explore and employ CMPA in decoding complex and challenging design and manufacturing problems. For performance evaluation of the proposed algorithm, CEC 2020 numerical problems having different dimensions and five widely adopted constrained design problems were solved. For all problems, both qualitative and quantitative results are examined and discussed. Moreover, two case studies of multi-pass turning were examined by the proposed CMPA algorithm to optimize the cutting operation with a minimum cost of production per unit objective. Furthermore, the suggested CMPA algorithm has been investigated for solving a real-world structural topology optimization problem. Statistical analysis is performed, and the results of CMPA are compared with twelve distinguished algorithms. Outcomes of the proposed variant algorithm on the benchmarks demonstrate its significantly improved performance relative to other optimizers including a variant of MPA and two state-of-the-art IEEE CEC competitions winners algorithms. Findings from the manufacturing process exhibit CMPA proficiency in solving arduous real-world design problems.

© 2022 Published by Elsevier B.V.

1. Introduction

Solving challenging design problems to achieve the best value (minimum or maximum) for different system parameters with economic aspects refers to an optimization process [1]. Due to their ungoverned nature, different dimensionality, and diverse objective and constraints, real-world optimization problems are difficult to solve and thus requires intelligent techniques like machine learning (ML) and artificial intelligence (AI) [2,3]. Conventional and metaheuristics (MHs) are two types of approaches

that are typically adopted for solving optimization issues. The former approach is the traditional one that involves gradient-based techniques. However, it loses its ground due to inefficiency in solving non-linear and non-differentiable design issues. Moreover, while solving practical engineering problems with large search space, they often take large computational time and fail to obtain optimum solutions. Contrarily, MHs are gradient-free and stochastic optimization methods that prove their competency in solving many design problems in different fields. Thus, recently many researchers have developed state-of-the-art MH algorithms that provide great harmony between the exploration and exploitation phase, fetch global optima with less computational time and provide superior solutions [4]. These widely adopted MHs can be categorized into four sub-classes as Evolutionary algorithms (EA), Swarm intelligence algorithms (SI), Physics-based algorithms, and Human-inspired algorithms [5–7].

* Corresponding author.

E-mail addresses: sumit21sep1990@gmail.com (S. Kumar), betulyildiz@uludag.edu.tr (B.S. Yildiz), pranavmehta.mech@ddu.ac.in (P. Mehta), natepa@kku.ac.th (N. Panagant), sadiq@kfupm.edu.sa (S.M. Sait), ali.mirjalili@gmail.com (S. Mirjalili), aliriza@uludag.edu.tr (A.R. Yildiz).

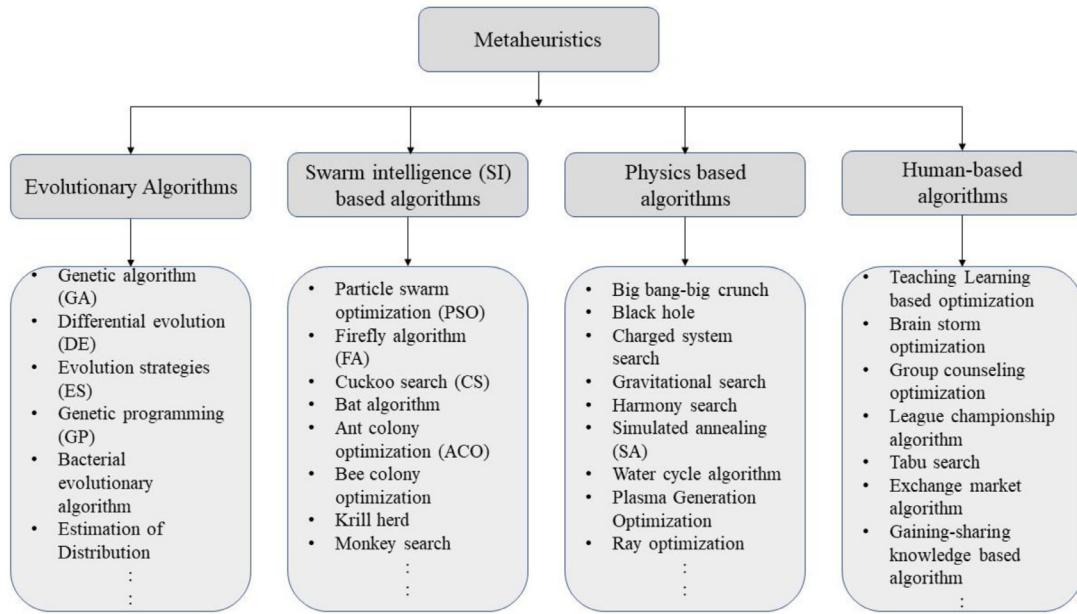


Fig. 1. A generalized metaheuristic classification based on their inspiration source.

The EAs were developed by considering the biological conduct of different systems, such as hybridization mutation [8] and the **Earthworm optimization algorithm** [9]. In SI algorithms, the social behaviour of insects, birds, and animals is mimicked, such as food foraging techniques and territorial fights to obtain the best solution from the search space [10] and **Elephant herding optimization** [11]. Physics laws and hypotheses help to built-up physics-based algorithms such as Plasma generation techniques and Henry's solubility law for gas [12]. Human-based algorithms are inspired by the social conduct of human beings, such as the gaining sharing knowledge-based algorithm [13]. The generalized MHs classification is described in Fig. 1.

Subsequently, efficient and quality searches within the search space should always be appreciable to the optimizer.

A balance between the exploration and exploitation phase creates evidence of the superiority of any MHs algorithm as these two parameters are the key phase of any optimizers for performance validation. The first phase (exploration) provides myriads of population-based parameters to explore feasible outputs from the search space. Whereas, in the second phase, it obtains the optimum solution from the available search space, which may be global or local [14,15]. However, harmony between these two characteristics is not encouraged by inefficient optimizers. Usually, population-based algorithms initiate the process with randomly generated solutions from the search space. This will be forwarded with the precise and optimized solution by the applied optimizer. Hence, search space, decision variables, and constraints may differ according to the nature of the problem. So, it is quite challenging to achieve the solution in a single step of the iteration process [16]. MHs such as GA and SA are also criticized for their slow convergence rate, which leads to high computation time for finding a promising solution. Apart from that, often, for large and complex problems, MHs like PSO and ACO are reported to be trapped in local optimum solutions [17]. To overcome such limitations and performance of MHs several improvement strategies have been adopted by many scholars so far, including parameter tuning [18,19], self-adaptive [20–23], opposition based learning [24–26], mutation and migration techniques [27], chaos strategy [28,29], elitism strategy [30,31], belief network [32,33], random solution selection process [34], modification approaches [35–37], hybridization [38].

The notion of chaos has been extensively applied in diverse applications with the growth of non-linear dynamic systems that are highly sensitive to the initial condition [39]. Owing to the ergodicity, randomness, and regularity features of chaos, chaotic maps were successfully applied with various MHs in several engineering optimization processes. Their initial population diversity enhancement was often found helpful in averting local optima entrapment and premature convergence [40]. Moreover, the implementation of the chaotic component in optimization is found to be a performance-improving factor in many algorithms [41,42]. For instance, scheduling and maintenance of the power generation unit are often found challenging. However, a chaotic differential evolution algorithm (CDEA) was reported to be helpful in resolving the scheduling difficulties of the short-term hydrothermal unit [2]. Similarly, proper power distribution in an electric system is an important factor when designing any power system. However, the economic dispatch system is the core challenge among the rest for optimized power distribution. For that, researchers developed a new MH called modified symbiotic organisms search (MSOS) with varieties of chaotic strategies applied to it to improve its performance [43]. Accordingly, implementing different logistic maps in the form of chaos helps the optimizer gain a global optimum solution, superior convergence rate, and more precise solution. Similarly, by applying three chaotic maps and five unique chaotic-based strategies in the Big bang-Big crunch (BBC) optimizer, the performance can be improved significantly [44]. Moreover, the local optimum solution is observed many times in the BBC optimizer, which can be overcome by utilizing chaotic maps. For instance, the approved best strategies from the available chaos BBC were applied to various standard unimodal and multimodal functions, and the superiority of chaotic-based optimizers can be verified from available results [44]. Many swarm intelligence algorithms utilize chaos maps in the form of Levy flights, which include myriads of strategy maps and exploration areas and flight paths. The cuckoo search algorithm (CSA) is improved by incorporating ten chaotic maps, and the performance of the chaotic CSA is tested by solving 27 benchmark problems [45]. The results reveal that chaotic CSA is superior compared to basic CSA in terms of convergence and quality solutions. Similarly, the particle-swarm krill herd optimizer is improved in terms of convergence and computational

time taken for solving complex problems by augmenting chaotic maps [46]. In a study, grey wolf optimizer (GWO) is made to operate with the ten specialized chaotic maps [29]. Furthermore, the developed chaotic GWO is tested for 13 benchmarks and five engineering design problems with various constraints; also, the results obtained from the chaotic GWO were compared with the GWO and other well-known MHS. Outcomes showed the supremacy of chaotic GWO among the traditional GWO, and the rest of the other tested MHS in terms of convergence, quality results of constrained problems, and global optimum solution [29]. Chaos theory is also applied to conventional GA to overcome its slow convergence, local optimum solutions, and lesser iteration speed. For instance, chaotic GA incorporates for obtaining optimal design of critical hydropower systems [47]. Here, chaos techniques improve the overall performance of the algorithm and solution outcome. The traditional krill herd algorithm is improved by chaotic theory, where it proves superior in terms of solution quality and convergence [48]. Several limitations have been detected while obtaining solutions from the dolphin swarm algorithm (DSA), such as failure of optimizer, incomplete solution, and trapping in local solution. However, the performance of chaotic DSA was significantly improved by augmenting eight chaotic logistic maps [49]. Moreover, performance was compared with established Whale optimization algorithm and conventional DSA; results show the superiority of chaotic DSA in terms of convergence rate. Similarly, the performance in terms of obtaining a global optimum solution is improved by hybridizing chaos and adaptive inertia weight factor (AIWF) with PSO; the results show that exploring quality has been proliferated with chaotic PSO [50]. Efficient tool path, motion, and handling are predicted with the help of the chaos ant colony algorithm (CACA); results demonstrate pocket milling could be optimized with effective tool trajectories with the help of CACA [51]. In another study, an artificial bee colony (ABC) optimizer is enhanced with various chaotic techniques for obtaining solutions to conventional benchmark test functions [52]. Accordingly, chaotic ABC was found superior to ABC in achieving global optimum solution and convergence rate. Moreover, many other states of the art optimizers are incorporated with chaos to improve, not limited but including shifting from local search solution to global optimum solution, convergence speed, computational time, and quality results [53–57].

The Marine Predators Algorithm (MPA) is a recently introduced MH inspired by the foraging strategies (i.e., Lévy and Brownian movements) of ocean predators and its encounters rate policy with prey [58]. MPA is a population-based nature-inspired MH that offers advantages like high flexibility, simplicity, precise computation, and least controlling parameters that lead to its wide application within a short span, interestingly in COVID-19 confirmed case forecasting [59]. Numerous numerical benchmark functions and engineering applications were used to assess the MPA's performance in the literature where it demonstrated its effectiveness when compared to other popular optimization algorithms like Harris hawk optimizer (HHO), GWO, Whale Optimization Algorithm (WOA), Slap Swarm Algorithm (SSA), GA and PSO [60–62]. However, due to its random strategy, it may cause blindness of predators and thus offer a limited exploitation rate in the search space. Hence, a proper exploitation guiding strategy can enhance its performance in non-linear, multi-constraints, and challenging design issues. A few improvised versions of MPA were also investigated for COVID-19 detection modelling [63] and as a COVID-19 image classification technique [64] that opens the door for further improvement through hybridization. Few efforts at performance improvement of MPA through chaos theory incorporation have also been seen in the literature [65,66]. However, these investigations are limited to only feature selection problems, one chaos map application, and outdated CEC 2014 test

suits. Thus, there is a requirement of to improve the performance of MPA to solve a wider range of real-world optimization problems and explore the opportunity that diverse chaos map offers in improving solution quality and efficiency. Moreover, as MPA is a state-of-the-art strategy and has a limited hybrid investigation so far, it is interesting to explore its potential with novel technique integration.

Motivated by the merits of chaotic maps and limitations of MPA, as discussed above, a novel hybrid algorithm called a Chaotic Marine Predators Algorithm (CMPA) is proposed in this study. CEC 2020 test suit, five benchmarks and two case studies of multi-pass turning (MpT) operations are investigated by using CMPA. The main objective of introducing a chaotic strategy is to overcome limitations of MPA such as local optimal trap and premature convergence and eventually enhance the exploration and exploitation potential of search. The outcomes of test examples were compared against several well-known recently introduced algorithms to evaluate the CMPA performance.

The present study contributes the following:

- A chaotic strategy is introduced in basic MPA to improve its exploration and exploitation skills along with local optimal trap and premature convergence prevention.
- A comprehensive performance analysis (both quantitative and qualitative) of CMPA against challenging CEC 2020 constraint benchmarks having diverse characteristics is performed.
- CMPA behaviour evaluation with diverse chaos theory and its comparison with twelve popular MHS on five engineering benchmarks problems from civil, mechanical, and industrial domains.
- Computation analysis CMPA with a range of maps for solving multi-constraint and non-linear practical multi-pass turning design and optimization problems with two different operation case studies.
- A real-world structure topology optimization by a proposed CMPA optimization algorithm and its statistical examination with different chaos theory.
- Extensive efficacy comparison of CMPA against many well-established MHS including an MPA variant and two state-of-the-art IEEE CEC competitions winners algorithms.

The remaining part of the study is structured as follows: Details of the considered marine predator's algorithm and chaotic maps are illustrated in Section 2, which is followed by Section 3, which governs the proposed hybrid approach in a detailed fashion. Section 4 demonstrates the details of the CEC 2020 numerical problems and five engineering test examples examined, while Section 5 presents the detailed mathematical model of the considered MpT optimization problem and provides the results and discussion of the two case studies investigated. Section 6 describes the real-world topology optimization of a vehicle structure, while Section 7 exhibits the overall concluding remark.

2. Overview of MPA and chaotic maps

2.1. Marine Predators Algorithm (MPA)

MPA is a recently introduced MH designed by mimicking ocean organism foraging behaviour (Lévy and Brownian motion) when predators search for their prey. It also accounts for the natural encounters rate policy in ocean settings among predator and prey. Initially presented by Faramarzi et al. [58] MPA is a population-based optimization method like GA, PSO, ACO, SSA, GOA, etc., albeit it is efficient, versatile, and simple.

MPA initializes its search by generating a certain number of random solutions (as per Eq. (1)) that are uniformly distributed

within specified lower (X_{min}) and upper boundaries (X_{max}) in the search arena.

$$X_0 = X_{min} + rand(X_{max} - X_{min}) \quad (1)$$

where $rand \in [0, 1]$ is a random vector

At first, the food location is discovered by predator/prey in the search space, and then they converge towards it to find its final location. The solutions are continuously updated to achieve the global solution with the mainframe of MPA. The fittest predator (best solutions) position is represented by an *Elite* matrix, generated based on the principle of natural selection. The matrix arrays shown below are used to supervise the quest and discovery of the prey by data collected on the prey's location.

$$\text{Elite} = \begin{bmatrix} X_{1,1}^I & X_{1,2}^I & \cdots & X_{1,d}^I \\ X_{2,1}^I & X_{2,2}^I & \cdots & X_{2,d}^I \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ X_{n,1}^I & X_{n,2}^I & \cdots & X_{n,d}^I \end{bmatrix}_{n \times d} \quad (2)$$

In the above matrix, n is the population size (number of predators) while d is the problem dimension. As the initial population is comprised of both predator and prey, the prey will also update its position while searching for its victim. Thus, a similar matrix is also defined for prey as follows:

$$\text{Prey} = \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,d} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,d} \\ X_{3,1} & X_{3,2} & \cdots & X_{3,d} \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ X_{n,1} & X_{n,2} & \cdots & X_{n,d} \end{bmatrix}_{n \times d} \quad (3)$$

It is worth noting that here the whole MPA process is based on these two matrices. While performing optimization is divided into three phases. All the three-phase governs the whole foraging actions of predator and prey based on *velocity ratio* (v). In each phase, a certain value of 'iteration count' is assigned. Note that these phases have been identified according to the movement of both predators and that prey in nature.

Phase 1: This phase governs the exploration in the search domain called the high-velocity ratio stage and occurs in primary 1/3rd iterations. Here prey moves with a higher velocity ($v > 10$) relative to the predator. It is interesting to note that the predator's best strategy in this step is defined as not moving at all. It is mathematically expressed as:

$$\text{While } Iter < \frac{1}{3} \text{Max_Iter}$$

$$\overrightarrow{\text{stepsize}}_i = \vec{R}_B \otimes (\overrightarrow{\text{Elite}}_i - \vec{R}_B \otimes \overrightarrow{\text{Prey}}_i) \quad i = 1, \dots, n \quad (4)$$

$$\overrightarrow{\text{Prey}}_i = \overrightarrow{\text{Prey}}_i + P \cdot \vec{R} \otimes \overrightarrow{\text{stepsize}}_i \quad (5)$$

In the above expressions, the vector containing random numbers is represented as R_B . It is in accordance with on 'Normal distribution', which reflects the Brownian motion. The symbol \otimes in the aforementioned equations indicates entry-wise multiplications. The simulation of the movement of prey is accomplished by multiplication of R_B with Prey. Constant $P = 0.5$ while random vector $R \in [0, 1]$. The large size of the step represents the high velocity.

Phase 2: This intermediate phase governs both the exploration and exploitation of search and is called the constant velocity-ratio stage. Here both prey and predators are moving with the same velocity($v \approx 1$) while searching for their respective food.

Therefore, 1/2 of the population is allocated for exploitation (prey moves in Lévy).

$$\begin{aligned} \text{while } \frac{1}{3} \text{Max_Iter} < Iter < \frac{2}{3} \text{Max_Iter} \\ \overrightarrow{\text{stepsize}}_i &= \vec{R}_L \otimes (\overrightarrow{\text{Elite}}_i - \vec{R}_L \otimes \overrightarrow{\text{Prey}}_i) \quad i = 1, \dots, n/2 \\ \overrightarrow{\text{Prey}}_i &= \overrightarrow{\text{Prey}}_i + P \cdot \vec{R} \otimes \overrightarrow{\text{stepsize}}_i \end{aligned} \quad (6)$$

where \vec{R}_L is a vector of random numbers based on Lévy distribution.

$$\begin{aligned} \text{and others } 1/2 \text{ for exploration (predator moves in Brownian)} \\ \overrightarrow{\text{stepsize}}_i &= \vec{R}_B \otimes (\vec{R}_B \otimes \overrightarrow{\text{Elite}}_i - \overrightarrow{\text{Prey}}_i) \quad i = n/2, \dots, n \\ \overrightarrow{\text{Prey}}_i &= \overrightarrow{\text{Elite}}_i + P \cdot CF \otimes \overrightarrow{\text{stepsize}}_i \end{aligned} \quad (7)$$

where $CF = (1 - \frac{Iter}{Max_Iter})^{(2 \times \frac{Iter}{Max_Iter})}$ is an adaptive step size controlling parameter.

Phase 3: This is the low-velocity ratio phase and represents the exploitation potential of the search. Here the prey is moving with less velocity($v = 0.1$) relative to the predator that adopts Lévy motion.

$$\text{while } Iter > \frac{2}{3} \text{Max_iter}$$

$$\begin{aligned} \overrightarrow{\text{stepsize}}_i &= \vec{R}_L \otimes (\vec{R}_L \otimes \overrightarrow{\text{Elite}}_i - \overrightarrow{\text{Prey}}_i) \quad i = 1, \dots, n \\ \overrightarrow{\text{Prey}}_i &= \overrightarrow{\text{Elite}}_i + P \cdot CF \otimes \overrightarrow{\text{stepsize}}_i \end{aligned} \quad (8)$$

To avoid the local optima trap, MPA considers a long jump in search. This behaviour is modelled by observing the real scenario of motion change in marine predators due to surrounding effects (Eddy formation and FADs' effect)

$$\begin{aligned} \overrightarrow{\text{Prey}}_i &= \overrightarrow{\text{Prey}}_i + CF[\vec{X}_{min} + \vec{R} \otimes (\vec{X}_{max} - \vec{X}_{min})] \otimes \vec{U} \quad \text{if } r \leq FADs \\ &= \overrightarrow{\text{Prey}}_i + [FADs(1 - r) + r](\overrightarrow{\text{Prey}}_{r1} - \overrightarrow{\text{Prey}}_{r2}) \quad \text{if } r > FADs \end{aligned} \quad (9)$$

In Eq. (33), \vec{U} is a binary vector (0 or 1); $FADs = 0.2$ representing the probability of FADs effect; $r \in [0, 1]$; $r1$ and $r2$ are random prey indices.

Also, MPA considers predator memory features in the search process, where they remember the most successful location of foraging. This step act as a repository where the best iterative solutions are saved which is used for updating the *Elite* matrix. For more details, readers are referred to Faramarzi et al. [58].

2.2. Chaotic maps

The details of the well-recognized chaotic maps implemented in the proposed investigation are illustrated in Table 1.

3. Chaotic marine predator algorithm

Realistic engineering design problems are challenging and intricate, and thus conventional optimization methods often fail to solve these issues in reasonable time and computational cost. Thus, MHS were widely used to address these problems as they are simple and take the least computational cost. However, they may face premature convergence and local optima trap-like problems and may deviate from the optimal solution [76]. Also, they

Table 1

The popular chaotic maps accounted for in the present investigation.

No.	Map name	Mathematical expressions	Range
1	Chebyshev [67]	$x_{k+1} = \cos(k\cos^{-1}(x_k))$	(-1, 1)
2	Circle [68]	$x_{k+1} = x_k + b - (a - 2\pi)\sin(2\pi x_k) \text{mod}(1); a = 0.5 \text{ and } b = 0.2,$	(0, 1)
3	Gauss/Mouse [69]	$x_{k+1} = \begin{cases} \varepsilon + x_k + cC_k^n, & 0 < x_k \leq P \\ \frac{X_k - P}{1 + P}, & P < x_k < 1 \end{cases}; \\ 1/\frac{1}{x_k \text{mod}(1)} = \frac{1}{x_k} - \left[\frac{1}{x_k} \right]$	(0, 1)
4	Iterative [70]	$X_{k+1} = \sin\left(\frac{a\pi}{X_k}\right); \alpha = 0.7$	(-1, 1)
5	Logistic [71]	$X_{k+1} = \alpha X_k(1 - X_k); \alpha = 4$	(0, 1)
6	Piecewise [72]	$f(x) = \begin{cases} \frac{x_k}{P} & 0 \leq x_k < P \\ \frac{x_p - P}{0.5 - P} & P \leq x_k < \frac{1}{2} \\ \frac{1 - p - X_k}{0.5 - P} & \frac{1}{2} \leq x_k < 1 - P \\ \frac{1 - X_k}{P} & 1 - P \leq x_k < 1 \end{cases}; P = 0.4$	(0, 1)
7	Sine [73]	$x_{k+1} = \frac{\alpha}{4} \sin(\pi x_k); \alpha = 4$	(0, 1)
8	Singer [74]	$x_{k+1} = \mu(7.86x_k - 23.31x_k^2 + 28.75x_k^3 - 13.3028.75x_k^4); \mu \in (0.9, 1.08)$	(0, 1)
9	Sinusoidal [75]	$x_{k+1} = ax_k^2 \sin(\pi X_k); a = 2.3$ If $a = 2.3$ and $x_0 = 0.7$, $x_{k+1} = \sin(\pi x_k)$	(0, 1)
10	Tent [75]	$X_{k+1} = \begin{cases} \frac{x_k}{0.7}, & X_k < 0.7 \\ \frac{10}{3}(1 - X_k), & X_k \geq 0.7 \end{cases}$	(0, 1)

Table 2

PC specifications used for experimental computation.

CPU	Intel Core i5-9600K (3.7 GHz, 6 Cores)
RAM	16 GB DDR4
Operating system	Windows 10
Software	MATLAB 2021b

typically have low search consistency, parameter tuning prerequisites, and low population diversity issues. Moreover, a significant diversification of an algorithm opens the likelihood of finding the best solution but may reduce search efficiency [77]. Conversely, if an algorithm has high intensification, then it can take the search towards local optima. Consequently, a global optimization approach is always in search that can create a good balance between intensification (local search) and diversification (global search) [78]. To overcome the aforementioned challenges, in this investigation, a novel CMPA has been proposed that incorporates the chaotic maps merits in the latest discovered MPA optimization technique. The motivation behind this hybrid approach is to balance the exploration and exploitation behaviour of MPA and to generate a global optimization algorithm. Though MPA has a significant performance in complex design issues, it might get trapped in a local optimum [60,79]. This limitation is attempted to improve by the dynamic nature of chaotic maps in this investigation as they accelerate and assist the search process by jumping out of the local region.

For the proposed hybrid algorithm, the Chaos technique is integrated into the main search process of the original MPA leading to a hybrid MPA-Chaos. The main idea of the proposed MPA-Chaos is to integrate the Chaos feature of non-repeated positions into the main search process of MPA to improve global search performance. After finishing the main reproduction process of the MPA, the Logistic map chaos is used to update the newly

generated position from MPA search procedures based on the following equation:

$$x_i = ax_i(1 - x_i) \quad \text{if } \text{rand} < P_c \quad (10)$$

where a is a constant parameter in $[0, 4]$ and P_c is the probability of performing Chaos operation.

Pseudo-code of the proposed hybrid Marine Predators Algorithm and Chaos is shown in Algorithm 1

4. Empirical evaluation

The performance of CMPA is investigated through its implementation in CEC 2020 test suites and engineering problems, including a single objective pressure vessel, tension spring, and three-bar truss optimization problems. The outcomes of the suggested methodology are compared with the distinguished optimization methods available in the literature. Further to check CMPA efficacy in real-world design issues, two case studies of MpT operation have been modelled and solved. All the conducted experiments are performed on the same PC with the specification shown in Table 2. Moreover, all the experiments are reported on average, 30 independent runs having 12 000 evaluations.

4.1. Definition of CEC'2020 benchmark functions

The performance of the CMPA is investigated by employing it in the recently released CEC 2020 test suite that is comprised of the ten numerical constrained functions, as described in Table 3. The benchmarks in this intriguing suite are unimodal, multi-modal, new hybrid, and composite. These statistical functions are considered to find the optimal solution challenging because of the multi-modality, hybrid, and composite structure of these CEC 2020 problems.

Algorithm 1: the pseudo-code of CMPA is presented below

```

Initialize search agents (Prey) populations  $i = 1, \dots, n$ 
While termination criteria are not met
    Calculate the fitness, construct the Elite matrix, and accomplish memory saving
    If  $Iter < Max\_Iter/3$ 
        Update prey based on Eq. (12)
    Else if  $Max\_Iter/3 < Iter < 2 \times Max\_Iter/3$ 
        For the first half of the population ( $i = 1, \dots, n/2$ )
            Update prey based on Eq. (6)
        For the other half of the population ( $i = n/2, \dots, n$ )
            Update prey based on Eq. (7)
    Else if  $Iter > 2 \times Max\_Iter/3$ 
        Update prey based on Eq. (8)
    End (if)
    Update prey using chaotic maps based on Equations in Table 1
        Accomplish memory saving and Elite update
        Applying FADs effect and update based on Eq. (9)
Endwhile

```

Table 3
CEC 2020 benchmark test.

No.	Function description	F_i^*
Unimodal function		
F1	Shifted and Rotated Bent Cigar function	100
Multimodal shifted and rotated functions		
F2	Shifted and Rotated Schwefel's function	1100
F3	Shifted and Rotated Lunacek bi-Rastrigin function	700
F4	Expanded Rosenbrock's plus Griewangk's function	1900
Hybrid functions		
F5	Hybrid function 1 ($N = 3$)	1700
F6	Hybrid function 2 ($N = 4$)	1600
F7	Hybrid function 3 ($N = 5$)	2100
Composition functions		
F8	Composition function 1 ($N = 3$)	2200
F9	Composition function 2 ($N = 4$)	2400
F10	Composition function 3 ($N = 5$)	2500

4.1.1. Statistical results on CEC'2020 benchmark functions

As mentioned above, the CMPA method is used to solve CEC 2020 numerical problems, and the results are compared to well-known HHO, SCA, WOA, AOA, PSO, TLBO, GA, and basic MPA algorithms. For a fair comparison, the results of state-of-the-art SHADE [80] and LSHADE-EpSin [81] algorithms (statistically similar to MPA) and a variant NMPA [82] are also compared with CMPA. Table 4 presents the statistical results of MPA with different chaotic maps versus other metaheuristics on ten CEC 2020 benchmarks. Each algorithm is executed for 30 independent runs where the mean, maximum and minimum values obtained by each considered algorithm, along with their standard deviation (Std) of error, are presented to furnish a meaningful statistical comparison.

Table 4 shows that the CMPA having a piecewise map (MPA_Chaos6) obtained the best mean fitness values for one multimodal (F_4) and two hybrids (F_6 and F_7) functions out of ten. Similarly, LSHADE-EpSin finds the best mean solutions for one unimodal (F_1) and two multimodal (F_2 and F_3) functions.

MPA finds the best values for one hybrid (F_5) and one composite (F_{10}) function, while MPA with Chebyshev (MPA_Chaos1) and circle map (MPA_Chaos2) finds the best mean values for F_8 and F_9 , respectively. In terms of minimum function fitness values, LSHADE-EpSin finds three (F_1 , F_3 , and F_5), MPA_Chaos6 two (F_7 and F_9), and MPA two (F_8 and F_{10}) best values while MPA_Chaos7 and MPA_Chaos2 one each i.e., F_2 and F_6 respectively. Table 4 also demonstrates the Std values of the test problems that govern the stability behaviour of all considered methodologies. In this case, GA finds the best values for four functions (F_2 , F_3 , F_5 , and F_9), while LSHADE-EpSin and MPA_Chaos6 both realize superior values for two functions i.e., F_1 and F_4 , and F_7 and F_{10} respectively. These outcomes demonstrate that the introduction of chaos theory exhibits significant performance improvement of MPA relative to basic MPA and its variant (NMPA). It should be noted that in almost all benchmarks, MPA_Chaos6 finds either best or very close results to the best one. It is also interesting to find that other algorithms, including HHO, WOA, SCA, TLBO, NMPA, and AOA, failed to find the optimal solution for CEC 2020 numerical problems. Table 4 also illustrates the Friedman mean rank test results that are used to show the difference between the performance of the proposed algorithm and all other algorithms used in the comparison. It is evident from the tables that the incorporation of the piecewise map in MPA results in the best solution for the four CEC 2020 numerical problems out of ten, i.e., for one multimodal (F_4), two new hybrid (F_6 and F_7), and one composite (F_9) functions. It is worth mentioning that the introduction of chaos theory results in a better Friedman rank relative to other algorithms for all test problems. Table 4 also presents the overall mean Friedman ranks of all algorithms for CEC 2020 benchmarks. Based on that, one can find the proposed CMPA having a piecewise map has better performance than others and finds the first rank relative to LSHADE-EpSin and LSHADE, which are the winner of previous IEEE CEC competitions.

Moreover, the popular Wilcoxon rank-sum and Friedman tests are also performed, as shown in Table 5, to clearly visualize the statistical differences between the MPA_Chaos6 and the other optimization methods. In this test, three indicators (+/-/-) have

Table 4

The statistical results of fitness values for 30 runs obtained with the different algorithms on the CEC'2020 functions.

	HHO	SCA	WOA	AOA	GSA	PSO	TLBO	GA	MPA	MPA_Chao1	MPA_Chao2	MPA_Chao3	MPA_Chao4	MPA_Chao5	MPA_Chao6	MPA_Chao7	MPA_Chao8	MPA_Chao9	MPA_Chao10	L SHADE	L SHADE_EpSin	NMPA	Winner			
F01	mean	1.937E+07	1.095E+09	1.056E+08	9.339E+09	2.941E+10	2.931E+10	2.908E+10	2.928E+10	2.172E+02	8.038E+03	4.462E+03	2.818E+03	2.972E+03	9.855E+03	2.482E+02	2.714E+03	1.862E+04	1.275E+05	2.346E+03	1.013E+02	1.000E+02	1.732E+06	L SHADE_EpSin		
	max	3.355E+08	2.071E+08	4.791E+08	1.810E+10	2.946E+08	2.940E+08	2.950E+08	2.930E+08	1.637E+03	1.332E+05	3.699E+04	9.926E+03	9.261E+03	2.134E+05	1.009E+03	1.118E+04	1.546E+05	3.397E+05	7.431E+03	1.190E+02	1.003E+02	5.756E+02	L SHADE_EpSin		
	min	5.623E+05	2.843E+08	1.467E+07	3.951E+08	9.235E+08	10.292E+08	2.760E+10	2.928E+10	1.004E+02	1.158E+02	2.363E+02	1.203E+02	1.020E+02	1.129E+02	2.473E+02	6.084E+02	1.049E+02	1.000E+02	5.046E+05	L SHADE_EpSin					
	std	6.172E+07	4.508E+08	1.062E+08	1.939E+09	2.070E+07	3.316E+07	4.201E+08	3.644E+08	2.993E+03	2.382E+04	6.919E+03	2.373E+03	3.851E+04	2.284E+02	2.879E+03	3.402E+04	6.180E+05	2.319E+03	4.314E+00	8.075E+02	1.78E+06	L SHADE_EpSin			
F02	Frank	14.90	17.00	15.90	18.00	21.80	20.27	20.03	19.90	3.87	9.20	8.83	8.60	8.37	8.37	4.07	8.23	10.10	10.57	7.87	1.90	1.10	14.13	L SHADE_EpSin		
	mean	2.128E+03	2.532E+03	2.405E+03	2.315E+03	5.286E+03	5.261E+03	3.249E+03	5.238E+03	1.477E+03	1.565E+03	1.587E+03	1.515E+03	1.534E+03	1.655E+03	1.414E+03	1.497E+03	1.709E+03	1.582E+03	1.517E+03	1.350E+03	1.335E+03	2.191E+03	L SHADE_EpSin		
	max	2.723E+03	3.000E+03	3.104E+03	2.792E+03	5.296E+03	5.304E+03	4.213E+03	5.244E+03	1.722E+03	1.967E+03	2.067E+03	1.971E+03	1.908E+03	1.981E+03	1.703E+03	1.952E+03	2.215E+03	1.933E+03	1.901E+03	1.663E+03	1.525E+03	2.474E+03	L SHADE_EpSin		
	min	1.645E+03	1.886E+03	1.584E+03	1.855E+03	5.272E+03	5.237E+03	2.547E+03	5.237E+03	1.127E+03	1.145E+03	1.127E+03	1.127E+03	1.122E+03	1.234E+03	1.112E+03	1.104E+03	1.408E+03	1.119E+03	1.116E+03	1.167E+03	1.116E+03	1.915E+03	MPA_Chao7		
F03	std	3.005E+02	2.414E+02	4.336E+02	2.542E+02	5.294E+00	2.018E+01	4.282E+02	1.538E+00	1.529E+02	2.065E+02	2.075E+02	1.812E+02	1.986E+02	1.577E+02	1.963E+02	1.866E+02	1.953E+02	1.980E+02	1.417E+02	1.075E+02	1.528E+02	GA			
	Frank	14.63	17.10	16.10	16.17	21.83	21.10	18.83	20.07	6.13	7.90	8.43	7.33	7.67	9.73	6.70	10.67	8.27	6.73	3.93	3.33	15.23	L SHADE_EpSin			
	mean	7.921E+02	7.892E+02	7.919E+02	7.979E+02	8.645E+02	8.655E+02	8.380E+02	8.610E+02	7.276E+02	7.352E+02	7.398E+02	7.370E+02	7.338E+02	7.322E+02	7.261E+02	7.322E+02	7.392E+02	7.349E+02	7.292E+02	7.200E+02	7.172E+02	7.578E+02	L SHADE_EpSin		
	max	8.274E+02	8.092E+02	8.553E+02	8.172E+02	8.662E+02	8.805E+02	8.522E+02	8.615E+02	7.370E+02	7.508E+02	7.545E+02	7.594E+02	7.513E+02	7.489E+02	7.367E+02	7.463E+02	7.541E+02	7.454E+02	7.276E+02	7.214E+02	7.684E+02	L SHADE_EpSin			
F04	std	7.466E+02	7.690E+02	7.432E+02	7.719E+02	8.633E+02	8.608E+02	8.207E+02	8.608E+02	7.153E+02	7.179E+02	7.250E+02	7.202E+02	7.190E+02	7.183E+02	7.145E+02	7.209E+02	7.226E+02	7.171E+02	7.172E+02	7.144E+02	7.204E+02	7.398E+02	L SHADE_EpSin		
	Frank	16.50	16.27	16.20	16.77	21.63	21.20	18.90	20.17	6.07	8.67	10.27	9.23	8.13	7.27	5.17	7.67	9.93	8.70	6.13	2.60	14.03	L SHADE_EpSin			
	mean	1.909E+03	1.987E+03	1.913E+03	1.815E+03	1.984E+06	1.972E+06	1.774E+06	1.964E+06	1.901E+03	1.901E+03	1.901E+03	1.901E+03	1.904E+03	MPA_Chao6											
	max	1.915E+03	2.621E+03	2.007E+03	4.728E+03	1.987E+03	1.989E+06	1.970E+06	1.903E+03	1.902E+03	1.902E+03	1.902E+03	1.903E+03	1.903E+03	1.902E+03	1.902E+03	1.903E+03	1.903E+03	1.902E+03	1.902E+03	1.902E+03	1.902E+03	1.905E+03	MPA_Chao6		
F05	min	1.904E+03	1.914E+03	1.902E+03	0.63	1.841E+03	1.979E+03	1.964E+03	1.521E+03	1.964E+03	1.900E+03	1.901E+03	1.901E+03	1.901E+03	1.901E+03	1.902E+03	MPA									
	std	3.450E+00	1.375E+02	1.872E+01	1.434E+03	2.537E+03	8.013E+03	8.539E+04	1.034E+03	4.222E+01	3.651E+01	5.408E+01	5.109E+01	4.824E+01	2.598E+01	5.619E+01	1.056E+00	5.729E+00	9.530E+00	5.726E+00	9.868E+00	8.961E+00	6.940E+00	2.842E+00	7.071E+00	GA
	Frank	15.30	16.97	15.37	18.00	21.93	20.83	19.00	20.23	5.83	7.43	5.87	6.67	7.63	7.03	3.73	6.67	11.23	10.63	5.90	5.77	14.10	MPA_Chao6			
	mean	6.469E+04	1.007E+05	3.140E+05	4.732E+05	3.007E+07	2.934E+07	2.782E+05	2.934E+07	1.074E+03	1.788E+03	1.781E+03	1.806E+03	1.806E+03	1.755E+03	1.794E+03	2.932E+03	4.955E+03	1.857E+03	1.752E+03	1.763E+03	2.066E+03	MPA			
F06	max	2.720E+05	5.596E+05	1.697E+06	1.046E+06	3.059E+06	2.973E+07	2.934E+07	1.971E+06	2.934E+07	1.861E+03	2.027E+03	1.967E+03	1.951E+03	1.972E+03	1.923E+03	1.815E+03	1.916E+03	2.053E+03	1.852E+03	1.860E+03	2.259E+03	1.905E+03	MPA_Chao6		
	min	3.215E+03	1.222E+04	6.892E+03	3.907E+04	2.963E+07	2.934E+07	1.846E+03	2.934E+07	1.710E+03	1.753E+03	1.711E+03	1.714E+03	1.706E+03	1.706E+03	1.708E+03	1.708E+03	2.027E+03	2.156E+03	1.725E+03	1.709E+03	1.703E+03	1.920E+03	L SHADE_EpSin		
	std	6.419E+04	1.339E+05	4.562E+05	1.821E+05	3.008E+05	2.946E+01	4.916E+05	4.417E+01	3.394E+01	6.831E+01	7.016E+01	7.460E+01	5.490E+01	8.213E+01	4.589E+01	8.745E+02	3.966E+01	8.076E+01	4.131E+01	3.602E+01	8.949E+01	GA			
	Frank	16.00	16.53	17.03	18.50	22.00	20.93	16.10	20.07	3.70	9.27	5.10	5.40	5.87	4.47	5.70	14.13	8.33	4.27	4.90	11.90	5.90	MPA			
F07	mean	1.630E+03	1.603E+03	1.618E+03	1.618E+03	2.339E+03	2.336E+03	1.828E+03	2.336E+03	1.601E+03	1.601E+03	1.601E+03	1.601E+03	1.601E+03	MPA_Chao6											
	max	1.660E+03	1.616E+03	1.661E+03	1.644E+03	2.342E+03	2.336E+03	1.860E+03	2.336E+03	1.601E+03	1.601E+03	1.601E+03	1.601E+03	1.602E+03	L SHADE											
	min	1.602E+03	1.610E+03	1.601E+03	1.601E+03	2.336E+03	2.336E+03	1.601E+03	2.336E+03	1.600E+03	1.600E+03	1.600E+03	1.600E+03	1.601E+03	MPA_Chao2											
	std	1.603E+01	2.651E+00	1.209E+01	1.201E+02	1.290E+02	1.620E+02	6.887E+01	1.207E+01	3.144E+01	3.072E+01	2.706E+01	3.008E+01	2.304E+01	1.242E+01	3.424E+01	2.807E+01	3.452E+01	3.290E+01	1.498E+01	3.757E+01	2.692E+01	1.498E+01	PSO		
F08	Frank	17.43	15.37	15.73	16.87	22.00	20.43	18.00	20.57	6.33	6.43	9.47	7.13	5.90	7.47	10.53	6.93	4.67	5.87	9.36	10.23	6.07	5.53	13.47	MPA_Chao6	
	mean	2.566E+05	1.717E+04	1.240E+06	2.524E+09	2.520E+05	2.443E+05	2.520E+08	2.113E+03	2.180E+03	2.125E+03	2.130E+03	2.128E+03	2.118E+03	2.105E+03	2.128E+03	2.128E+03	2.128E+03	2.128E+03	2.128E+03	2.128E+03	2.128E+03	2.264E+03	MPA_Chao6		
	max	2.009E+06	3.650E+04	1.669E+07	6.889E+06	2.528E+09	2.520E+05	3.541E+05	6.250E+03	2.135E+03	2.307E+03	2.157E+03	2.225E+03	2.186E+03	2.176E+03	2.120E+03	2.167E+03	2.470E+03	3.795E+03	2.187E+03	2.127E+03	3.216E+03	2.403E+03	MPA_Chao6		
	min	2.701E+03	4.283E+03	4.440E+03	3.750E+03	2.029E+09	2.520E+09	2.520E+09	2.231E+03	2.520E+09	2.101E+03	2.104E+03	2.102E+03	2.101E+03	2.102E+03	2.102E+03	2.102E+03	2.102E+03	2.102E+03	2.102E+03	2.102E+03	2.102E+03	2.153E+03	MPA_Chao6		
F09	std	4.912E+05	8.874E+03	3.055E+06	2.115E+06	1.385E+06	2.784E+01	8.961E+05	6.482E+00	1.200E+01	6.685E+01	1.383E+01	4.281E+01	1.690E+01	1.692E+01	5.905E+00	1.627E+01	1.804E+02	3.621E+02	2.090E+01	9.136E+00	1.235E+01	6.108E+01	MPA_Chao6		
	Frank	17.10	16.43	18.07	17.53	22.00	20.90	15.17	20.10	6.03	6.67	7.67	7.17	5.43	2.40	7.73	13.00	14.07	4.00	4.33	12.10	12.10	MPA_Chao6			
	mean	2.363E+03	2.476E+03	2.514E+03	2.998E+03	5.160E+03	5.142E+03	3.271E+03	5.134E+03	2.290E+03	2.257E+03	2.293E+03	2.300E+03	2.300E+03	2.301E+03	2.279E+03	2.301E+03	2.295E+03	2.301E+03	2.297E+03	2.302E+03	2.300E+03	2.295E+03	MPA_Chao1		
	max	3.754E+03	4.100E+03	4.023E+03	3.969E+03	5.174E+03	5.028E+03	3.5150E+03	2.303E+03	2.304E+03	2.304E+03	2.306E+03	2.304E+03	2.303E+03	2.303E+03	2.303E+03	2.303E+03	2.303E+03	2.303E+03	2.303E+03	2.303E+03	2.303E+03	2.303E+03	L SHADE		
F10	min	2.307E+03	2.338E+03	2.309E+03	2.648E+03	5.146E+03	5.131E+03	3.200E+03	5.131E+03	2.200E+03	2.216E+03	2.217E+03	2.240E+03	2.235E+03	2.206E+03	2.206E+03	2.229E+03	2.229E+03	2.229E+03							

Table 5

The Wilcoxon rank sum test and ranking of algorithms for the CEC'2020 functions.

Problems	HHO	SCA	WOA	AOA	GSA	PSO	TLBO	GA	MPA	MPA_Chaos1	MPA_Chaos2	MPA_Chaos3	MPA_Chaos4	MPA_Chaos5	MPA_Chaos6	MPA_Chaos7	MPA_Chaos8	MPA_Chaos9	MPA_Chaos10	L SHADE	L SHADE_EpSin	NMPA
1	15 (+)	17 (+)	16 (+)	18 (+)	22 (+)	21 (+)	19 (+)	20 (+)	3 (0)	10 (+)	9 (+)	7 (+)	8 (+)	11 (+)	4	6 (+)	12 (+)	13 (+)	5 (+)	2 (-)	1 (-)	14 (+)
2	14 (+)	18 (+)	17 (-)	16 (+)	22 (+)	21 (+)	19 (+)	20 (+)	4 (0)	9 (+)	11 (+)	6 (+)	8 (+)	12 (+)	3	5 (0)	13 (+)	10 (+)	7 (+)	2 (-)	1 (-)	15 (+)
3	17 (+)	15 (+)	16 (+)	18 (+)	21 (+)	22 (+)	19 (+)	20 (+)	4 (0)	10 (+)	13 (+)	11 (+)	8 (+)	6 (+)	3	7 (+)	12 (+)	9 (+)	5 (+)	2 (-)	1 (-)	14 (+)
4	15 (+)	17 (+)	16 (+)	18 (+)	22 (+)	21 (+)	19 (+)	20 (+)	3 (+)	8 (+)	4 (+)	7 (+)	11 (+)	6 (+)	1	10 (+)	13 (+)	12 (+)	9 (+)	5 (+)	2 (+)	14 (+)
5	15 (+)	16 (+)	18 (+)	19 (+)	22 (+)	21 (+)	17 (+)	20 (+)	1 (-)	11 (+)	5 (0)	6 (+)	8 (+)	10 (+)	3	7 (+)	13 (+)	14 (+)	9 (+)	2 (0)	4 (0)	12 (+)
6	18 (+)	15 (+)	17 (-)	16 (+)	22 (+)	20 (+)	19 (+)	21 (+)	6 (+)	7 (+)	3 (0)	10 (+)	13 (+)	8 (+)	1	4 (0)	11 (+)	12 (+)	9 (+)	2 (0)	5 (0)	14 (+)
7	17 (-)	15 (+)	19 (+)	18 (+)	22 (+)	21 (-)	16 (+)	20 (+)	4 (+)	11 (+)	7 (+)	10 (+)	9 (+)	5 (+)	1	8 (+)	13 (+)	14 (+)	6 (+)	2 (+)	3 (-)	12 (+)
8	15 (+)	16 (+)	17 (-)	18 (+)	22 (+)	21 (-)	19 (+)	20 (+)	3 (0)	1 (0)	4 (+)	9 (+)	8 (+)	12 (+)	2	6 (+)	13 (+)	7 (+)	14 (+)	11 (0)	10 (0)	5 (+)
9	17 (-)	16 (+)	15 (-)	18 (+)	22 (+)	21 (-)	19 (+)	20 (+)	5 (+)	7 (+)	1 (+)	2 (+)	6 (+)	12 (+)	3	4 (+)	13 (+)	10 (+)	9 (+)	14 (+)	11 (+)	8 (-)
10	15 (+)	17 (+)	16 (+)	18 (+)	22 (+)	21 (+)	19 (+)	20 (+)	1 (0)	10 (+)	8 (+)	7 (+)	4 (+)	9 (+)	6	2 (+)	13 (+)	11 (+)	3 (+)	12 (+)	14 (+)	5 (+)
Rank_av	15.8	16.2	16.7	17.7	21.9	21	18.5	20.1	3.4	8.4	6.5	7.5	8.3	9.1	2.7	5.9	12.6	11.2	7.6	5.4	5.2	11.3
+/-	10 0 0	10 0 0	10 0 0	10 0 0	10 0 0	10 0 0	10 0 0	10 0 0	4 1 5	9 0 1	8 0 2	10 0 0	10 0 0	NA	8 0 2	10 0 0	10 0 0	10 0 0	4 3 3	4 3 3	10 0 0	

Where + / - are best/not significantly differ from the best/significantly differ from the best.

Table 6
Runtime comparison of MPA and CMPA.

Runtime (s)	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
MPA	0.4552	0.2626	0.1577	0.1657	0.1629	0.1513	0.2068	0.2286	0.2154	0.2199
CMPA	0.6376	0.4083	0.3011	0.2914	0.3061	0.3014	0.4434	0.4195	0.3567	0.3426

been used to present the number of functions in which the MPA_Chaos6 performs better (+), the function in which it behaves similarly to other competitors (=), and the number of functions in which it has fallen short (-). The MPA_Chaos6 is the best variant in all functions in comparison with NMPA, HHO, SCA, WOA, GSA, PSO, and TLBO and shows a high or equal success in comparison with MPA and IEEE CEC winners LSHADE and LSHADE-EpSin algorithms. Table 5 results also illustrate the Friedman rankings for each function and the average rank according to which overall MPA_Chaos6 ranks first, which shows it is significantly better than other competitors. Furthermore, a computational analysis is performed between MPA and suggested CMPA techniques. The results from Table 6 show that the chaos theory may increase the computational time between 32%–67% and thus leave space for further improvement analysis.

4.1.2. Boxplot behaviour analysis

The boxplot analysis can show the properties of the data distribution. Because there are too many local minima affiliated with this class of functions, the boxplot of results for piecewise map-based CMPA algorithm and other chaos-based CMPA versus other methods for CEC 2020 functions is provided in Fig. 2 to help with understanding the distribution of outcomes. Boxplots are excellent displays to show quartile-based data distributions. The algorithm's lowest and highest data points, which are the edges of the whiskers, are the minimum and maximum. The edges of the rectangles define the lower and upper quartiles. A compact boxplot indicates strong data agreement. According to Fig. 2(a), CMPA has thin boxplots relative to other algorithms, which demonstrated the enhancement of the performance due incorporation of chaos theory in MPA. Fig. 2(b) shows that the boxplots of the piecewise map-based CMPA (i.e., MPA_chaos6) algorithm is relatively narrow and thus have the lowest value for the majority of functions except for F₈. It is also interesting to note that MPA_chaos6 realizes better performance than MPA, SHADE, LSHADE-EpSin, and NMPA.

4.1.3. Diversity and convergence behaviour analysis

The CMPA is suggested as a way to enhance the standard MPA exploratory and exploiting process for candidate solutions, and as a result, diversity and convergence curves are used in this study [83]. The average distance between all solutions during the course of iterations is shown by diversity curves, as illustrated in Fig. 3. For a mathematical description of this curve, readers are referred to Olorunda and Engelbrecht [84]. From these figures, it can be observed that the diversity in the population of the MPA_chaos6 is higher than the standard MPA. The chaos strategies maintain the diversity level in the population so that the probability of getting prone to the local optimal solutions is reduced during the search.

The trajectory of the candidate solutions is displayed in Fig. 4, in its first dimension, to help visualize the convergence behaviour of considered algorithms [83]. The trajectory demonstrates that the chaos tactics used in the CMPA produce significant changes in the movement of potential solutions throughout iterations, aiding in the provision of either the optimal solution or one that is extremely near to it. Moreover, CMPA with a piecewise map demonstrates superior convergence performance relatively.

4.2. Qualitative metrics analysis

Even if the previous result analyses have confirmed the suggested CMPA algorithm's excellent performance, doing additional tests and evaluations would allow us to reach firmer conclusions on the algorithm's effectiveness in solving actual problems. For example, one way to gain a better understanding of the optimization search process and algorithm convergence is to observe the behaviour of the particles or search agents. In this study, the detailed qualitative behaviour of MPA with a piecewise map has been examined by the performance index, average fitness history, exploration, and exploitation plots.

4.2.1. Performance index analysis

A performance index (PI) can be examined to assess how well the suggested MPA_Chaos6 and traditional MPA perform in terms of offering less error and requiring less computational time. For more details about the PI index, one can refer to Deep and Thakur [85] and Gupta et al. [83]. In this study, PI is computed for MPA_Chaos6 verse other algorithms on CEC 2020 benchmarks. All PI curves as illustrated in Fig. 5, show that MPA_Chaos6 can be favoured over traditional MPA when the user requires both error and calculation time minimum.

4.2.2. Average fitness history analysis

The average fitness history, or the fitness value averages as a function of iteration number, is shown in Fig. 6. This average shed light on the agents' typical behaviour and how they contribute to the optimization process. According to plots, the MPA_Chaos6 population becomes better with iteration as the history curve declines. Moreover, the MPA_Chaos6 ongoing development confirms a cooperative searching behaviour and backs the effectiveness of the evolving particle law.

4.2.3. Exploration and exploitation behaviour analysis

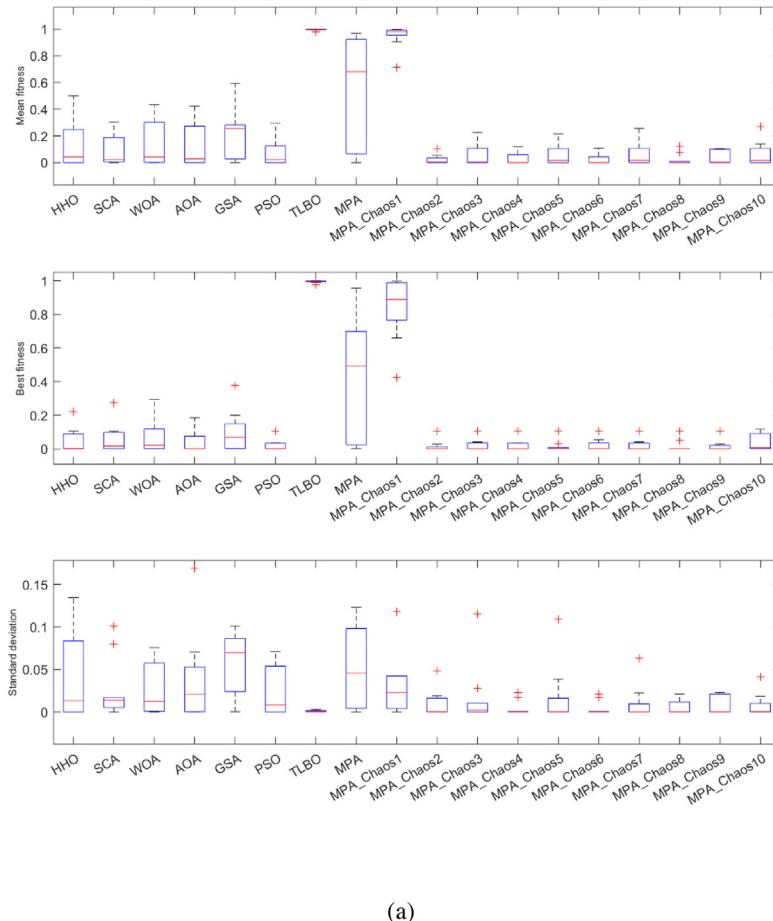
As mentioned in the introduction section, exploration and exploitation behaviour are the two crucial aspects of an effective algorithm, thus illustrated in Fig. 7(a) and (b), respectively, concerning function evaluation numbers. It is evident from the plots that the CMPA with a piecewise map has better exploration and exploitation performance relative to the compared algorithms in most of the numerical functions of CEC 2020 test suits.

4.3. Application of CMPA to engineering problems

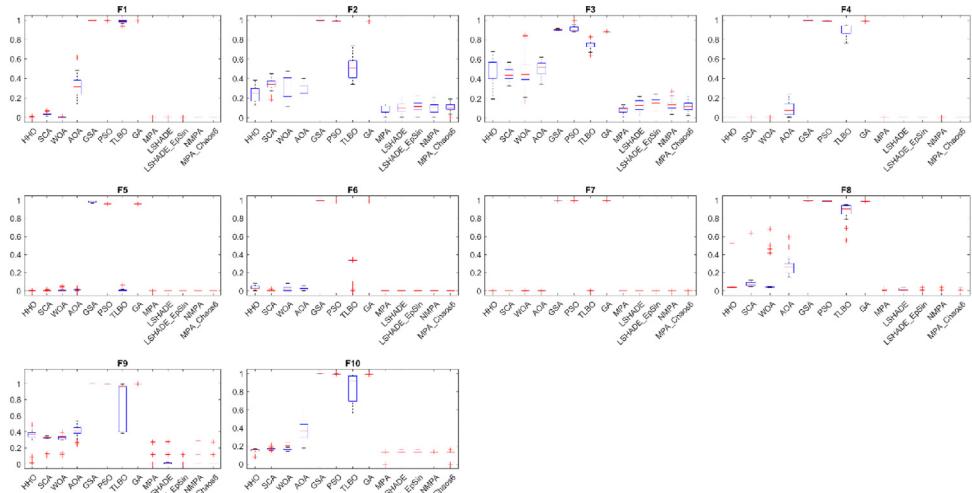
4.3.1. Tension spring optimization problem

The first test example considered in this investigation is the tension spring design problem [86,87], as illustrated in Fig. 8, in which weight minimization is the objective while the minimum shear stress, surge frequency, and deflection are the constraints of the problem. Three design variables viz. mean coil diameter (x_1), wire diameter (x_2), and many active coils (x_3) are present in this optimization problem that can be mathematically formulated as shown in Appendix A.1.

The experiment computation results for the considered test problem are illustrated in Table 7 in the form of best, mean, worst, standard deviation, and function evaluation values for all the algorithms executed. CMPA (circle map) achieved the best value of the minimum weight, i.e., 0.012665236, concerning other algorithms in the least function evaluation (FEs) of 15 000. The



(a)



(b)

Fig. 2. Boxplots of the results obtained by (a) all CMPA and (b) piecewise map-based CMPA against other algorithms for CEC'2020 functions.

standard deviation value computed by CMPA with a circle map is $2.61743E-06$ which is the minimum relative to different executed algorithms for this test example. Also, CMPA achieves lower

values for mean and worst. Consequently, we can say that the proposed hybridization improves the exploitation rate of MPA and presents an efficient and robust algorithm.

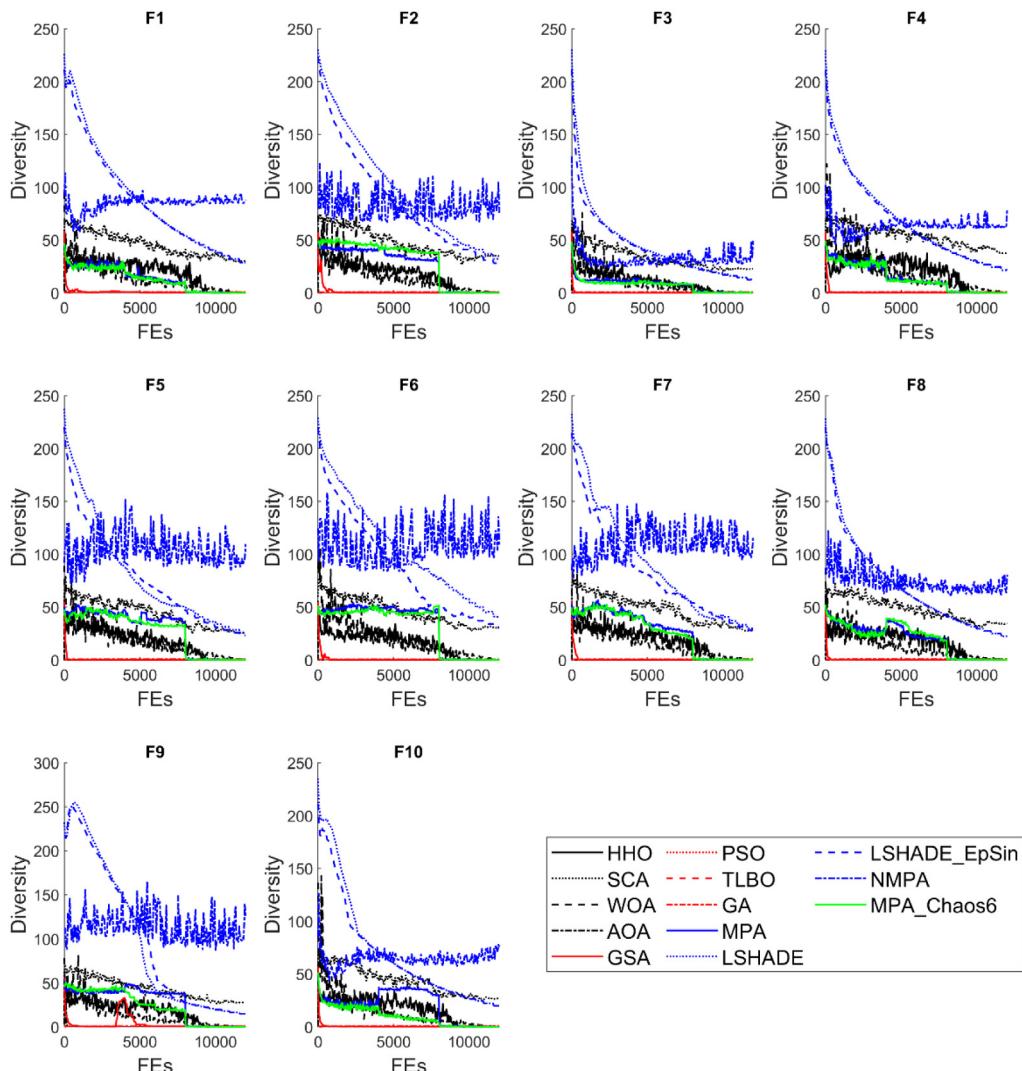


Fig. 3. Diversity curves.

Table 7
Comparison results between CMPA and the other algorithms for tension spring design.

Algorithms	Best	Mean	Worst	SD	FEs
CMPA with Chebyshev map	0.012665577	0.01266753	0.012675103	7.6853E-04	15 000
CMPA with circle map	0.012665236	0.012668055	0.012673936	2.61743E-06	15 000
CMPA with Gauss/Mouse map	0.012665245	0.01266821	0.012686979	5.58789E-06	15 000
CMPA with iterative map	0.012665237	0.012667107	0.012694557	6.61697E-06	15 000
CMPA with logistic map	0.012665244	0.012673548	0.012719054	1.63637E-05	15 000
CMPA with piecewise map	0.012665239	0.012667058	0.012697070	5.76256E-06	15 000
CMPA with sine map	0.012665242	0.012675698	0.012759987	2.25984E-05	15 000
CMPA with singer map	0.012665248	0.012668790	0.012697848	6.88418E-06	15 000
CMPA with sinusoidal map	0.012665243	0.012668731	0.012720545	1.03192E-05	15 000
CMPA with Chebyshev map	0.012665246	0.012668724	0.012692298	5.82987E-06	15 000
MPA	0.01266835	0.013025369	0.019963211	1.25369E-04	15 000
SSA [87]	0.012668411231	0.013119278	0.019625233	1.2691E-03	18 000
MVO [87]	0.012693497334	0.015961898	0.018075268	1.8089E-03	18 000
MFO [87]	0.012678015647	0.014306586	0.017773158	1.6344E-03	18 000
ASO [87]	0.011588881581	0.014579929	0.022603040	1.9396E-03	18 000
EO [87]	0.012667706272	0.013154360	0.017773158	1.3810E-03	18 000
ES [87]	0.012666529186	0.014725310	0.030454967	3.1009E-03	18 000

4.3.2. Pressure vessel optimization problem

In this study, a mixed-integer pressure vessel design problem, as shown in Fig. 9, is the second test example that is considered for the performance check of CMPA. This is one of the

widely used [88,89] test examples with the objective of total cost minimization that comprised of the cost of material, welding, and forming. Four design variables of this design problem are vessel cylindrical section length (L), cylinder head thickness (T_h),

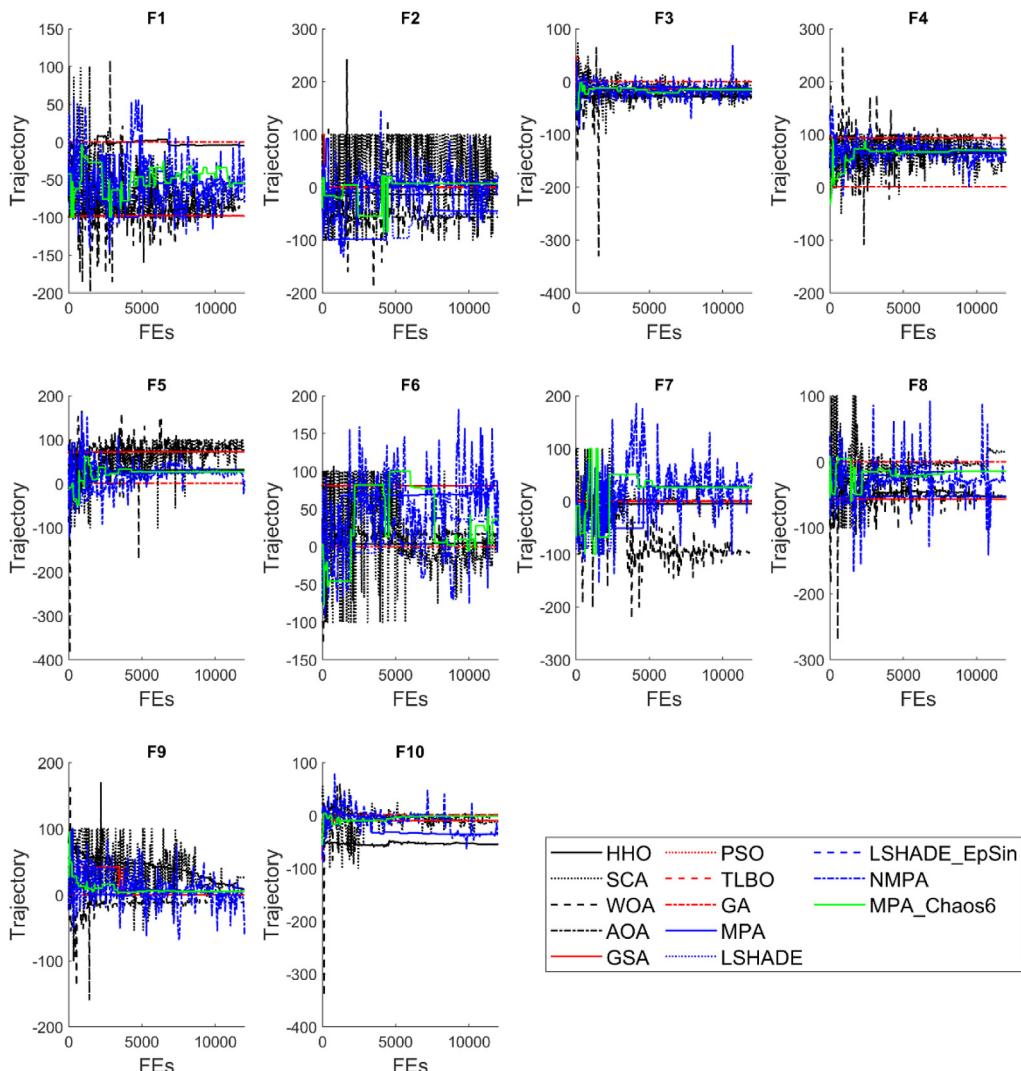


Fig. 4. Trajectory curves.

shell thickness (T_s), and cylinder inner radius (R). [Appendix A.2](#) provides a mathematical expression of the pressure vessel problem.

The results obtained by each method for the test example are presented in [Table 8](#), where the values of best, Std Dev, and function evaluation values are treated as the performance measure. The experiment results for the pressure vessel example are presented in [Table 8](#) which were obtained by all the executed algorithms. All the algorithms demonstrate better performance and may find the available optimal solution. The best value i.e., 6059.714456, is achieved by the CMPA (Gauss/Mouse map) algorithm. Overall, it is evident that the proposed hybridization improves the exploitation rate of MPA and presents an efficient and robust algorithm.

4.3.3. Three bar truss problem

The third test example considered in this study is a two-dimensional three-bar truss problem [90], as shown in [Fig. 10](#). The objective function of this design problem is considered as the minimization of weight subjected to three non-linear restraints, while the rod element areas are the two design variables owing to symmetry (i.e., $A_1 = A_3 = \mathbf{x}_1$ and $A_2 = \mathbf{x}_2$). This test problem

can be formulated as depicted in [Appendix A.4](#). The results of CMPA are contrasted with eleven distinguished algorithms from the literature to examine the performance. [Table 9](#) depicts the dominance of CMPA in comparison to the others as it achieves the best result in the least number of function evaluations of 5000. Hence, the proposed hybridization leads to a rise in the convergence rate of search and yields a robust global optimization algorithm.

4.3.4. Piston lever design

This is the fourth problem considered for CMPA performance evaluation. It is the widely adopted design problem initially introduced by Vanderplaats [87]. Here oil volume minimization is the objective, and the detailed mathematical formulation is presented in [Appendix A.4](#). As shown in [Fig. 11](#), piston elements (i.e. "H", "B", "D", and "X") are found when the lever is raised from 0° to 45° . By treating the variables X and D as constants, a 3D representation is also shown in [Fig. 5](#) to highlight the complexity of the problem.

The best functional and variable values obtained by all tested algorithms and the statistical results are presented in [Table 10](#). It is evident that CMPA performs better than other algorithms by

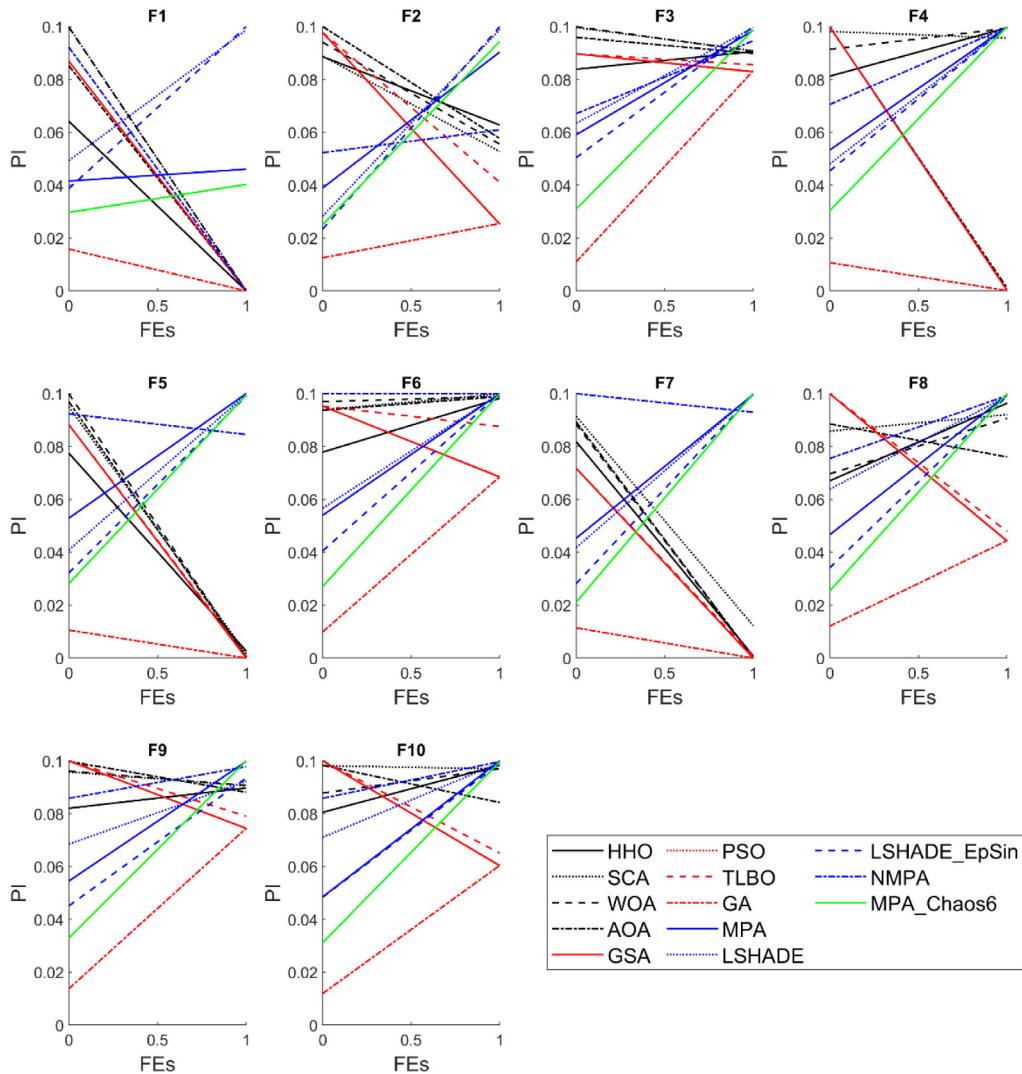
**Fig. 5.** Performance index for CEC 2020 problems.

Table 8
Comparison of results between CMPA and the other algorithms for pressure vessel design.

Algorithms	Best	Mean	Worst	SD	FEs
CMPA with Chebyshev map	6059.715239	6134.364905	6410.099	134.4304	5000
CMPA with circle map	6059.714901	6121.374571	6410.092	121.5436	5000
CMPA with Gauss/Mouse map	6059.714456	6095.832471	6370.807	55.4809	5000
CMPA with iterative map	6059.715356	6198.553283	7273.512	268.6529	5000
CMPA with logistic map	6059.714893	6154.469311	7332.842	249.0085	5000
CMPA with piecewise map	6059.714979	6124.565075	7198.013	217.2553	5000
CMPA with sine map	6059.715104	6134.617971	7332.852	243.2636	5000
CMPA with singer map	6059.715705	6108.386795	7050.681	180.5861	5000
CMPA with sinusoidal map	6059.715207	6223.311637	7332.842	354.8032	5000
CMPA with tent map	6059.715760	6158.180706	6771.602	198.5595	5000
MPA	6061.587963	6398.259875	6870.973	223.8965	5000
SSA [87]	6059.71443719	6406.3755192102	7433.3095187621	330.746	25 000
MVO [87]	6068.51149415	6465.6159464662	7335.9735321858	341.336	25 000
MFO [87]	6059.71433505	6434.5021580991	7544.4925179250	424.968	25 000
ASO [87]	6084.45342650	6713.6637975986	7438.6515953213	320.651	25 000
EBO [87]	6092.56710600	6501.0531798097	6830.7286320387	260.008	25 000
QSA [87]	6059.71433504	6078.3193234831	6370.7797127298	61.5629	25 000
EO [87]	6059.71433504	6581.6044020330	7544.4925179250	518.455	25 000
ES [87]	6059.71542118	6617.4323265386	7800.5318598906	471.316	25 000
HSOGA [87]	6059.71433504	6329.3848896881	7332.8415077595	367.658	25 000

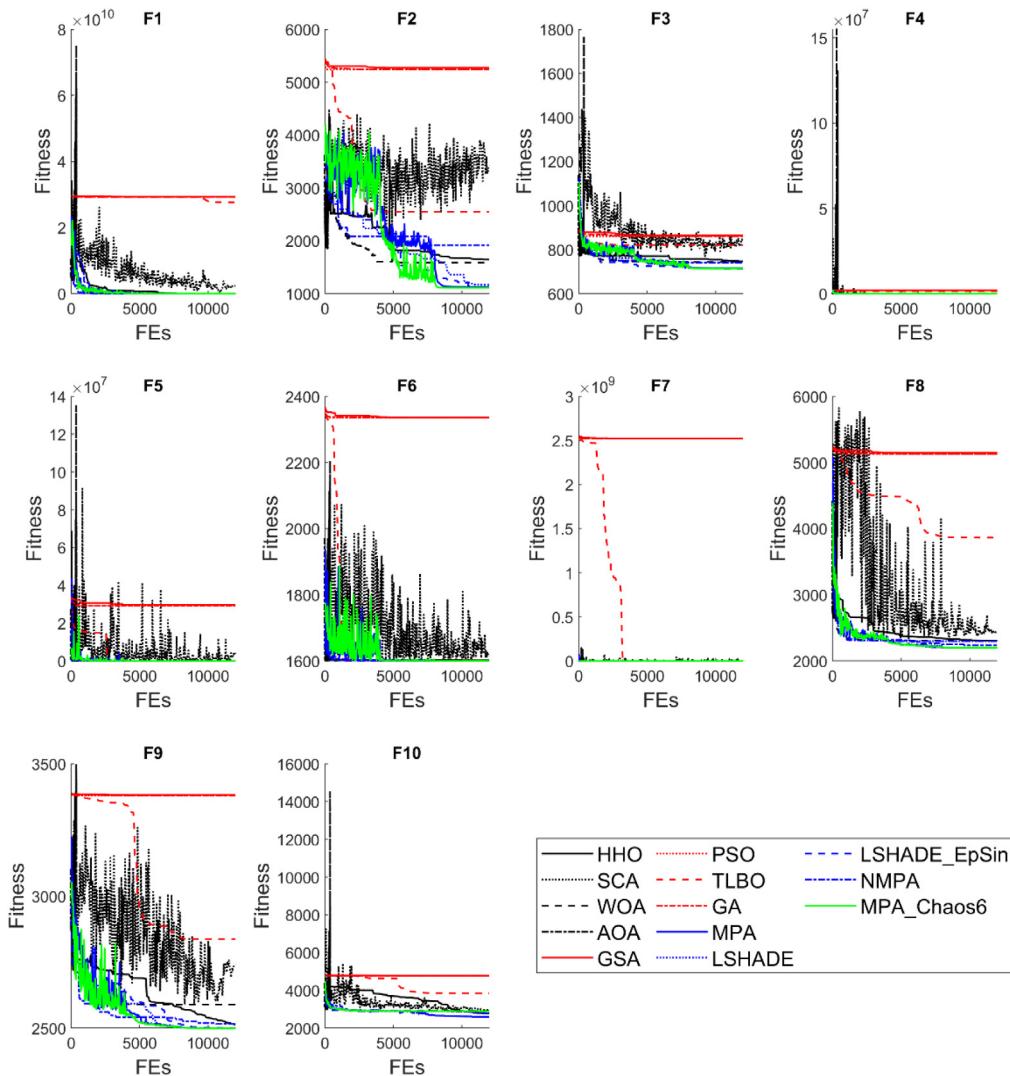
**Fig. 6.** Average fitness history.

Table 9
Comparison of results between CMPA and the other algorithms for three-bar truss design.

Algorithms	Best	Mean	Worst	SD	FEs
CMPA with Chebyshev map	263.8958430	263.9353	264.1014	0.034152	5000
CMPA with circle map	263.8959861	263.9495	264.2536	0.084216	5000
CMPA with Gauss/Mouse map	263.8958689	263.9650	264.6364	0.138221	5000
CMPA with iterative map	263.8962022	264.0166	264.4135	0.126599	5000
CMPA with logistic map	263.8958808	263.9397	264.1682	0.061137	5000
CMPA with piecewise map	263.8958468	263.9481	264.2064	0.060654	5000
CMPA with sine map	263.8958536	263.9667	264.4959	0.114493	5000
CMPA with singer map	263.8959730	263.9369	264.2358	0.048584	5000
CMPA with sinusoidal map	263.8958550	263.9573	264.1131	0.056365	5000
CMPA with Chebyshev map	263.8962483	263.9397	264.1963	0.080078	5000
MPA	263.9863023	263.9986	264.8976	0.135693	5000
ALO [91,92]	263.8958434	—	—	—	14 000
MVO [93]	263.8958499	—	—	—	15 000
GOA [94]	263.8958814	—	—	—	13 000
MFO [91,92]	263.8959796	—	—	—	15 000
MBA [95]	263.8958522	263.897996	263.915983	3.93E-03	13 280
CS [96]	263.97156	264.0669	—	0.00009	15 000

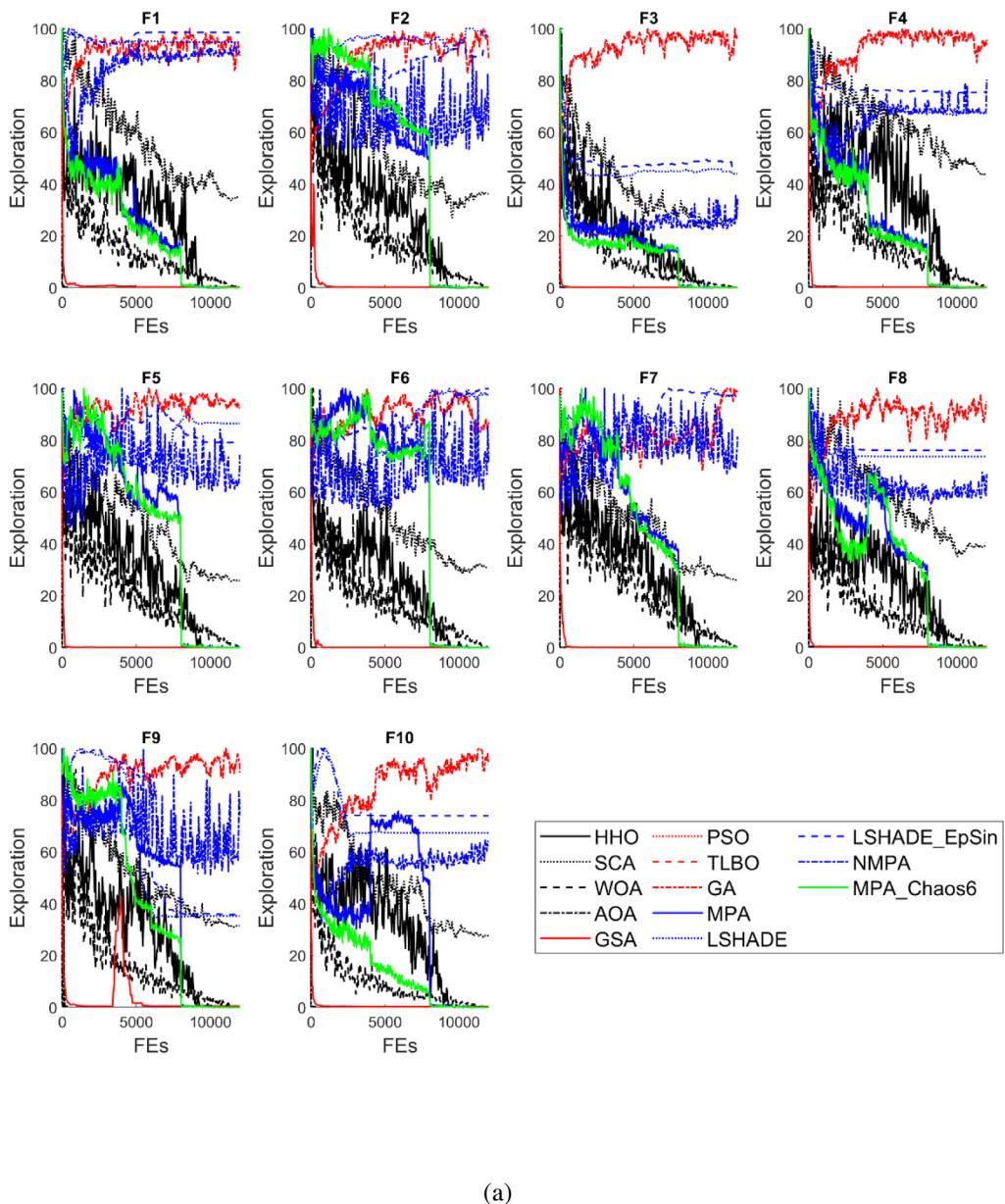


Fig. 7. (a) Exploration and (b) Exploitation plots of MPA_chaos6 against other algorithms.

realizing the best mean and SD value. Moreover, it took the least computational run, i.e., 20 000, to find the optimal solution.

4.3.5. Spur gear design problem

A widely investigated spur gear design issue [97] is the fifth problem considered in this study. Gear system weight minimization is the objective having eleven mixed design variables including module m , pinion teeth z , face width b , and pinion and gear shaft diameters (d_1, d_2). Its inequality constraints and mathematical expression are provided in Appendix A.5. Fig. 12 shows the schematic and 3D view (keeping d_1 and d_2 as constants) of the design problem.

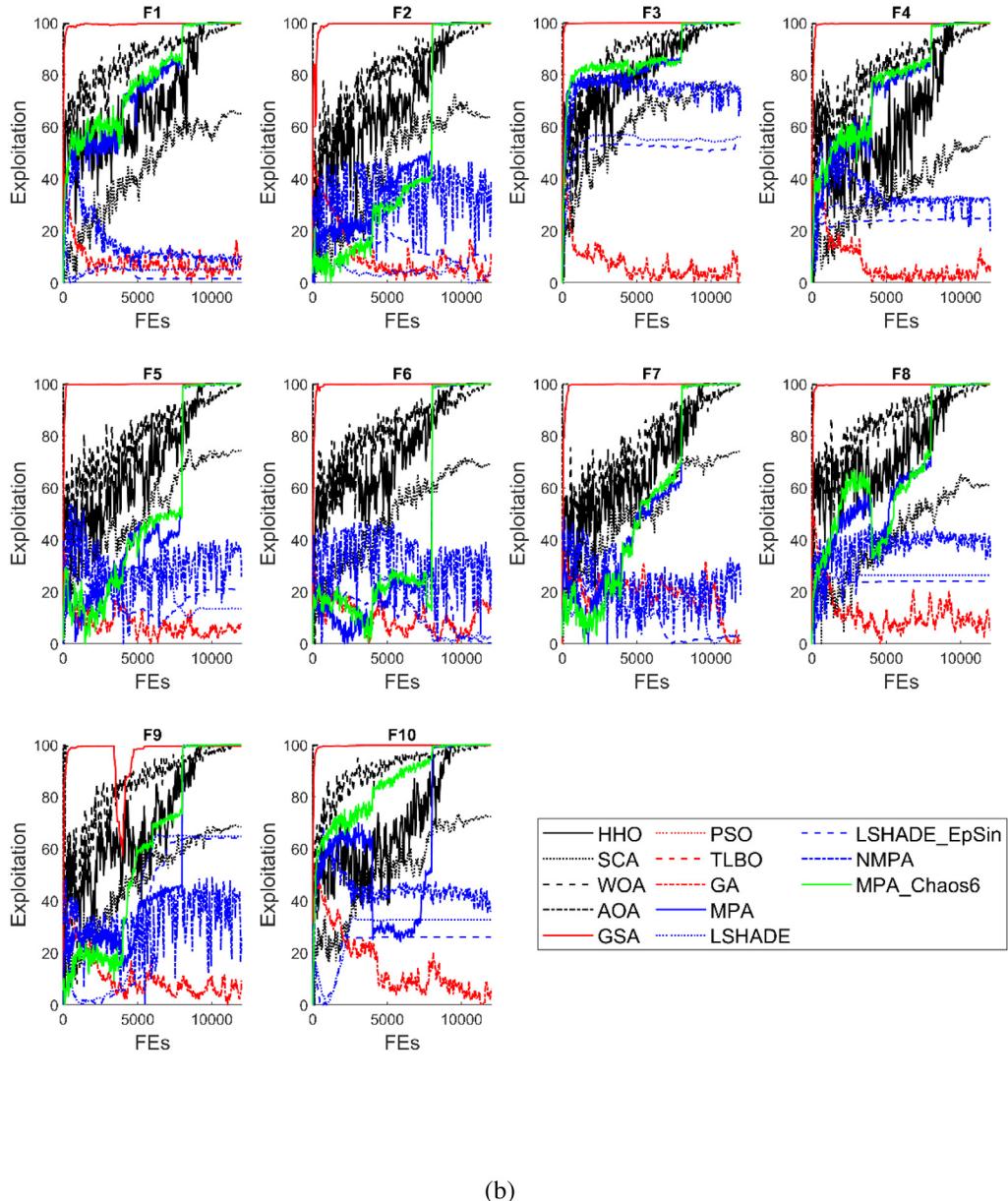
Table 11 shows the best variable and functional value obtained by all algorithms and the statistical results. From Table 11, it is evident that EO realizes the best mean and SD values relatively. However, the optimum value is realized by CMPA at least 20,000 FE number. This demonstrates its superior convergence rate in comparison to other applied algorithms.

5. Case study of computational machining optimization and formulation of the multi-pass turning operations model

Over the last couple of decades, turning-based metal cutting technology such as MpT operations has grown considerably. The optimization techniques have made a significant contribution to achieving different objectives of the machining process. High quality and the least cost with a minimum number of cuts are the primary objectives of MpT operations [98,99]. To achieve these objectives, process parameter optimization is essential that typically involves feed rate, depth of cut, cutting speed, and the number of passes optimal selection. A schematic view of a turning operation is shown in Fig. 13. In Fig. 14, the design variables of the turning problem solved in this paper are shown.

5.1. The objective function

In this study, a single objective MpT process model has an objective of unit production cost (C_{UP}) minimization while

**Fig. 7. (continued).**

machining a cylindrical job. Following the previous investigation conducted by Chen and Tsai [100] and Shin and Joo [101], a statistical model has been developed for the MpT operations where the production cost per unit is represented as follows:

$$F(X) = \min(C_{UP}) = C_M + C_I + C_R + C_T \quad (11)$$

$$\begin{aligned} C_{UP} = k_0 & \left[\frac{\pi DL}{1000V_r f_r} \left(\frac{d_t - d_s}{d_r} \right) + \frac{\pi DL}{1000V_s f_s} \right] \\ & + k_0 \left[t_c + (h_1 L + h_2) \left(\frac{d_t - d_s}{d_r} + 1 \right) \right] + \\ & k_0 \frac{t_e}{T_p} \left[\frac{\pi DL}{1000V_r f_r} \left(\frac{d_t - d_s}{d_r} \right) + \frac{\pi DL}{1000V_s f_s} \right] \\ & + \frac{k_t}{T_p} \left[\frac{\pi DL}{1000V_r f_r} \left(\frac{d_t - d_s}{d_r} \right) + \frac{\pi DL}{1000V_s f_s} \right] \end{aligned} \quad (12)$$

5.2. Multi-pass turning process constraints

In MpT operation, C_{UP} is subjected to numerous machining constraints during the roughing and finishing process.

5.2.1. Rough machining bounds

$$\text{Depth of cut: } d_{rL} \leq d_r \leq d_{rU} \quad (13)$$

$$\text{Feed: } f_{rL} \leq f_r \leq f_{rU} \quad (14)$$

$$\text{Cutting speed: } V_{rL} \leq V_r \leq V_{rU} \quad (15)$$

$$\text{Tool-life: } T_L \leq t_r \leq T_U \quad (16)$$

$$\text{Cutting force: } k_1 f_r^\mu d_r^\nu \leq F_u \quad (17)$$

Table 10

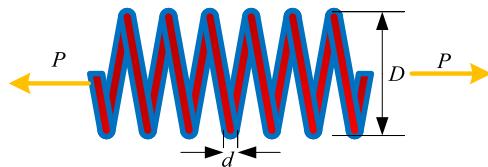
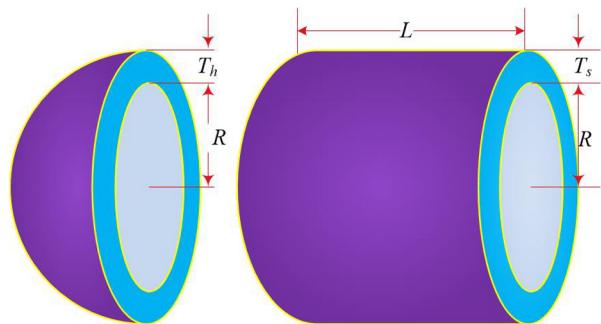
Comparison of various statistical values by the applied algorithms on the piston lever problem.

Algorithms	Best	Mean	Worst	SD	FEs
CMPA with Chebyshev map	8.412698323	167.4728025	167.4728025	40.35483216	20 000
CMPA with circle map	8.412700623	167.472844	167.472844	68.42493033	20 000
CMPA with Gauss/Mouse map	8.412698807	167.4727508	167.4727508	29.04026106	20 000
CMPA with iterative map	8.412698418	167.4727652	167.4727652	29.04027091	20 000
CMPA with logistic map	8.412698880	167.4728273	167.4728273	40.35483568	20 000
CMPA with piecewise map	8.412698679	167.4728129	167.4728129	2.44308E-05	20 000
CMPA with sine map	8.412698549	12.412831363	16.412578555	40.35482965	20 000
CMPA with singer map	8.412698737	167.4728236	167.4728236	29.04025878	20 000
CMPA with sinusoidal map	8.412698363	8.471023647	8.472749600	8.2985E-06	20 000
CMPA with Chebyshev map	8.41269832310	9.45863977	12.412737524	9.02878526	20 000
MPA	8.436985326	198.8963254	224.53698521	58.36599885	20 000
SSA [87]	8.42203784711	276.9405864277	653.4973462065	121.4145	25 000
MVO [87]	8.42899605472	138.4470035304	356.2368524801	138.5046	25 000
MFO [91,92]	8.41269832310	91.12391482201	167.4727300517	80.27315	25 000
ASO [87]	213.834591596	346.0264403230	778.6031806898	115.1357	25 000
EBO [87]	8.41269884770	110.9658310772	181.5427843584	75.01844	25 000
QSA [87]	8.41269832310	78.45855972953	167.4727300517	79.70457	25 000
EO [87]	8.41269832310	100.6675167283	167.4727300880	79.30246	25 000
ES [87]	8.41269832310	129.6027508165	215.5008965538	73.23548	25 000
HSOGA [87]	8.41271552698	128.0854411099	230.0218329665	84.22322	25 000

Table 11

Comparison of various statistical values by the applied algorithms on spur gear design problem.

Algorithms	Best	Mean	Worst	SD	FEs
CMPA with Chebyshev map	1538.94468183	1538.945011	1538.946258	0.000318	20 000
CMPA with circle map	1538.944823	1538.945785	1538.950481	0.00124	20 000
CMPA with Gauss/Mouse map	1538.944684	1543.349674	1608.819511	14.6675	20 000
CMPA with iterative map	1538.944759	1544.278687	1608.820276	17.09839	20 000
CMPA with logistic map	1538.944727	1543.096581	1570.080669	10.76417	20 000
CMPA with piecewise map	1538.944698	1539.982836	1570.077859	5.684042	20 000
CMPA with sine map	1538.944719	1539.984269	1570.078951	5.683979	20 000
CMPA with singer map	1538.944720	1539.983503	1570.078956	5.684124	20 000
CMPA with sinusoidal map	1538.944696	1627.692367	2870.152098	337.738	20 000
CMPA with Chebyshev map	1538.944782	1538.945695	1538.949024	0.001178	20 000
MPA	1538.9986325	1546.102369	1559.1519084	11.025365	20 000
SSA [87]	1538.94468499	1550.41308815	1654.55754051	25.845818	25 000
MVO [87]	1538.953765647	1539.027569357	1539.166647936	0.04791177	25 000
MFO [87]	1538.944681836	1545.965610384	1680.978949409	26.3350479	25 000
ASO [87]	1624.223602003	1713.750536474	1827.07275730	49.4135992	25 000
EBO [87]	1538.944681836	1544.548937677	1570.077785077	12.0822412	25 000
QSA [87]	1538.944681836	1539.567343901	1570.077785074	4.40288568	25 000
EO [87]	1538.944681836	1538.944681836	1538.944681836	1.0866E-12	25 000
ES [87]	1538.944681836	1586.281735297	2870.151908489	189.127219	25 000
HSOGA [87]	1538.944681836	1584.871507635	2870.151908490	188.079509	25 000

**Fig. 8.** Design variables for tension spring problem.**Fig. 9.** Design variables for pressure vessel.

$$\text{Power: } \frac{k_1 f_r^\mu d_r^\nu V_r}{6120\eta} \leq P_U \quad (18)$$

$$\text{Stable cutting region: } V_r^\lambda f_r d_r^\nu \geq S_C \quad (19)$$

$$\text{Chip-tool interface temperature: } Q_r = k_q V_r^\tau f_r^\phi d_r^\delta \leq Q_U \quad (20)$$

$$\text{depth of cut: } d_{sL} \leq d_s \leq d_{sU} \quad (21)$$

$$\text{feed: } f_{sL} \leq f_s \leq f_{sU} \quad (22)$$

$$\text{cutting speed: } V_{sL} \leq V_s \leq V_{sU} \quad (23)$$

$$\text{Tool-life: } T_L \leq t_s \leq T_U \quad (24)$$

$$\text{Cutting force: } k_1 f_s^\mu d_s^\nu \leq F_U \quad (25)$$

$$\text{Power: } \frac{k_1 f_s^\mu d_s^\nu V_s}{6120\eta} \leq P_U \quad (26)$$

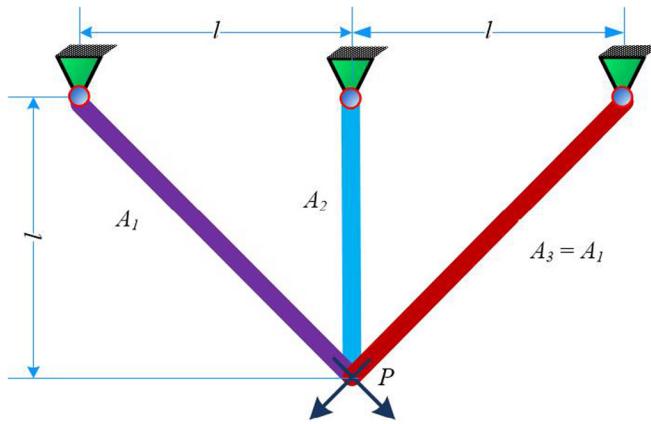


Fig. 10. Schematic view of three bar truss design.

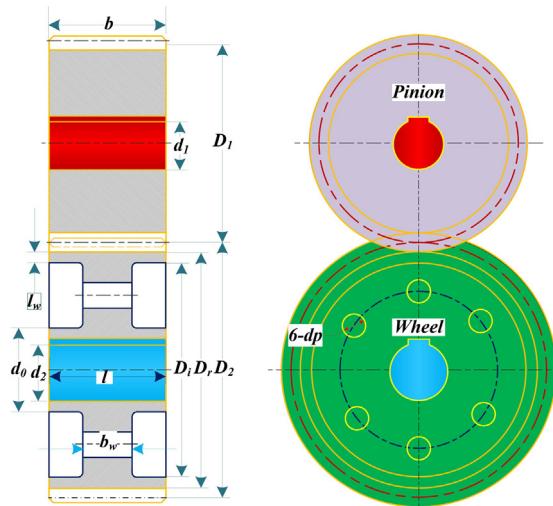


Fig. 12. Schematic view of spur gear geometry of the web-type structure.

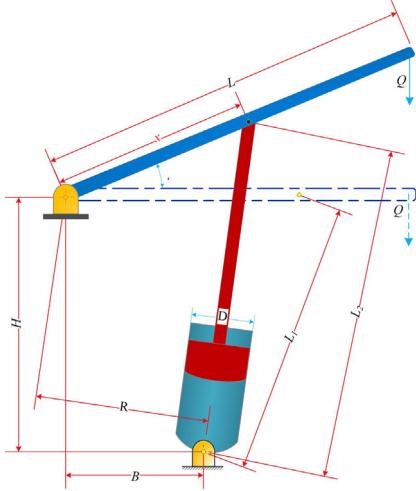


Fig. 11. The Schematic view of the piston problem design.

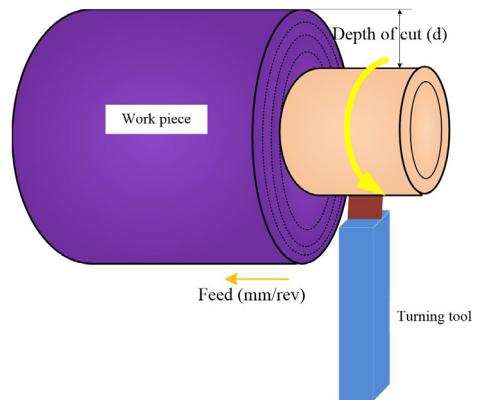


Fig. 13. A schematic view of turning operation.

$$\text{Stable cutting region: } V_s^\lambda f_s d_s^v \geq S_C \quad (27)$$

$$\text{Chip-tool interface temperature: } Q_S = k_2 V_s^\tau f_s^\phi d_s^\delta \leq Q_U \quad (28)$$

$$\text{Surface finish: } \frac{f_s^2}{8R} \leq SR_U \quad (29)$$

5.2.3. Parameter relations

$$V_s \geq k_3 V_r \quad (30)$$

$$f_r \geq k_4 f_s \quad (31)$$

$$d_r \geq k_5 d_s \quad (32)$$

where $k_3, k_4, k_5 \geq 1$

$$d_r = \frac{(d_t - d_s)}{n} \quad (33)$$

Apart from the above-mentioned technical and operational bounds, the total depth of cut (which is an additional but crucial constraint) is also accounted for in this case study. Finishing cut depth (d_s), and rough cutting depth (nd_r) combined govern the absolute depth of cut (d_t) of the turning operation in the present model. The optimal depth of the roughing operation is disregarded by the optimization algorithm as it can be determined

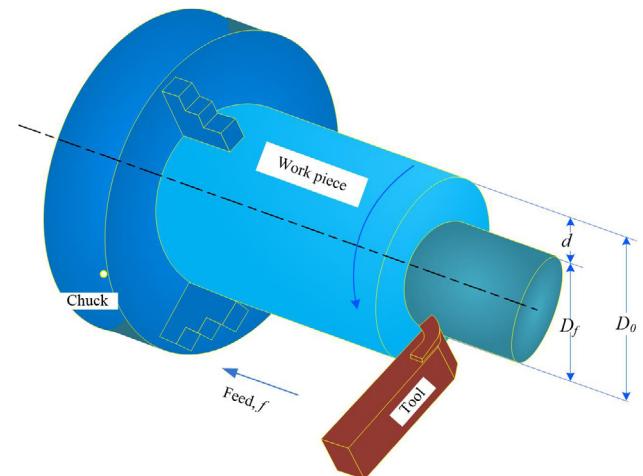
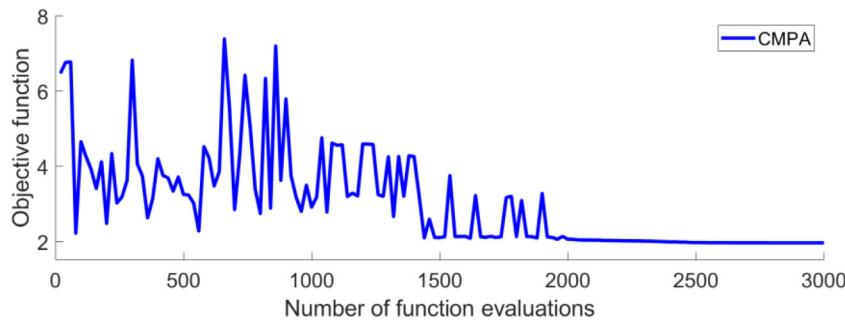


Fig. 14. Schematic illustration of depth of cut (d), and feed (f) is the turning.

statistically by Eq. (24).

$$d_s = d_t - nd_r \quad (34)$$

Thus, within the process optimization, the decision variable and equality constraint can be eradicated. The five process

Fig. 15. C_{UP} convergence behaviour for case 1.**Table 12**
Machining model data.

Parameter	Value	Parameter	Value	Parameter	Value
D	50 mm	L	300 mm	d_t	6 mm
V_{rU}	500 m/min	V_{rl}	50 m/min	f_{rU}	0.9 mm/rev
f_{rl}	0.1 mm/rev	d_{rU}	3 mm	d_{rl}	1 mm
V_{sU}	500 m/min	V_{sl}	50 m/min	f_{sU}	0.9 mm/rev
f_{sl}	0.1 mm/rev	d_{sU}	3 mm	d_{sl}	1 mm
ρ	5	q	1.75	r	0.75
u	0.75	ϑ	0.95	η	0.85
λ	2	v	-1	τ	0.4
ϕ	0.2	δ	0.105	R	1.2 mm
C_0	6×10^{11}	h_1	7×10^{-4}	h_2	0.3
T_L	25 min	t_c	0.75 min/piece	t_e	1.5 min/edge
P_U	5 kW	T_U	45 min	F_U	200 kgf
SC	140	SR_U	10 μm	Q_U	1000 °C
K_0	0.5 \$/min	k_1	108	k_2	132
k_3	1	k_4	2.5	k_5	1
k_f	2.5 \$/edge				

parameters that are investigated from considered turning model optimization are V_r , f_r , d_s , V_s , and f_s .

To examine the performance of the proposed CMPA in practical problems, an MpT optimization model was formulated and solved following data from the previous few studies [99,102]. Multiple constraints, variables, and non-linearity associated with this model make this design complex and challenging. As per Section 2, two case studies have been accounted for and solved considering the design data from Table 12.

This manufacturing problem is comprised of multiple rough cuts (roughing operation) and one finish cut (finishing operation), and for each pass feed rate, depth of cut and cutting speed are included as the operational parameters. Hence, a total of six design variables were present in this problem.

MATLAB tool is used to execute the algorithm, and the experiment is performed by Windows 10 ultimate operating system on a PC with Intel Core i5-9600K processor at 3.7 GHz and 16 GB DDR4 RAM. In particular, tool life Eqs. (10) and (13) were solved separately by the proposed CMPA (iterative map) in two cases. For this problem, CMPA is computed 30 times individually with a termination criterion of the 3000 number of function evaluations.

Tables 13 and 14 represent the best functional and variable values realized by all executed algorithms for cases 1 and 2, respectively. Moreover, for both case studies, the obtained C_{UP} convergence behaviour is plotted in Figs. 15 and 16. Further, the maximum FEs number (also employed as a termination criterion) specified in this investigation is significantly lower than the FEs number of other compared methodologies. CMPA (iterative map) provided better results for the turning problem relatively. Therefore, it is listed the best results of the CMPA between Tables 13–16.

As depicted in Tables 13 and 14, the best C_{UP} obtained by CMPA (iterative map) for case 1 and 2 is 1.9591 (\$/piece) and

Table 13
Detailed comparison of obtained results with CMPA (case 1: $T_p = T_r + T_s$).

	Variable	Range/Limit	CMPA (iterative map)
Cutting parameters	V_r	50–500	123.3430
	f_r	0.1–0.9	0.5655
	d_r	1.0–3.0	3.0000000000
	V_s	50–500	169.9785
	f_s	0.1–0.9	0.2262
	d_s	1.0–3.0	3
Tool life	T_r	25–45	25.0003
	T_s	25–45	25.0002
Roughing operation constraints	$g_1(X)$	≤ 0	-0.0076
	$g_2(X)$	≤ 0	-0.2580
	$g_3(X)$	≥ 0	2727.7
	$g_4(X)$	≤ 0	-93.0069
Finishing operation constraints	$g_5(X)$	≥ 0	-99.4092
	$g_6(X)$	≤ 0	-1.7131
	$g_7(X)$	≥ 0	2038.5
	$g_8(X)$	≤ 0	-141.5247
	$g_9(X)$	≤ 0	-9.9951
Constraint of variable relations	$g_{10}(X)$	≥ 0	46.6355
	$g_{11}(X)$	≥ 0	0
	$g_{12}(X)$	≥ 0	0
Unit production cost			1.9591 (\$/piece)

2.0351 (\$/piece) respectively. It is critical to note that all the solutions achieved through investigation are feasible for both case studies. Also, Figs. 9 and 10 demonstrated the good convergence characteristics of CMPA (iterative map) in finding the optimal solution. HPSO, FPA, and COA find the optimal solution after 62 500, 10 000, and 3500 function evaluations. The CMPA (iterative map) finds an optimal solution in 3000 function evaluation numbers with the least function evaluation number in the literature.

Tables 15 and 16 illustrate the comparative results obtained by CMPA with an iterative map and other compared algorithms for cases 1 and 2. The results show that the incorporation of iterative maps in MPA leads to better design variable values at the least cost.

6. Optimum structural design using CMPA

The topology optimization of a vehicle structure is carried out by a proposed CMPA optimization algorithm. The prime focus of the study is to minimize the mass of the element subjecting the critical constraints [113]. querymm2 has been changed to mm² in Eq. (36). Please check, and correct if necessary.

$$\text{Min } F(y) = f_1(y) \quad (35)$$

$$\text{Constraint: } g(y) \leq 300 \text{ N/mm}^2 \quad (36)$$

$$y_i^l \leq y_i \leq y_i^u, \quad i = 1, \text{NDV} \quad (37)$$

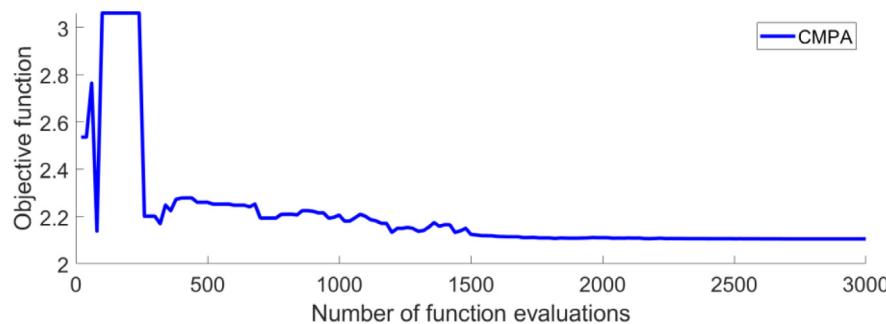
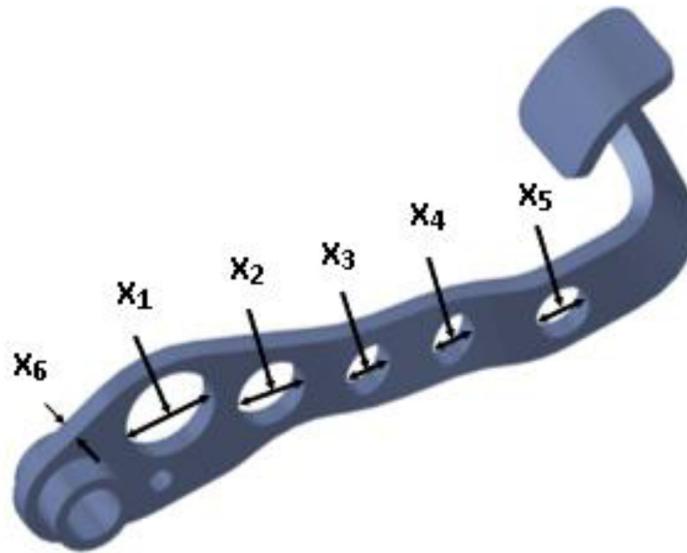
Fig. 16. C_{UP} convergence behaviour for case 2.

Fig. 17. Design variables.



Fig. 18. Optimum brake pedal design using the CMPA.

where \mathbf{y}^l and \mathbf{y}^u denote the least and upper values of the design variables.

The design variables of the component are shown in Fig. 17. The minimum and maximum limits of the design variables used in the optimization are considered as $6 < Y_1 < 13.9$, $35.1 < Y_2 < 45$, $25 < Y_3 < 35$, $18 < Y_4 < 26$, $18 < Y_5 < 26$, $6 < Y_6 < 13.9$ [114].

Table 14
Detailed comparison of obtained results with CMPA (iterative map) (case 2:
 $T_p = \theta T_r + (1 - \theta)T_s$).

	Variable	Range/Limit	CMPA (iterative map)
Parameter bounds	V_r	50–500	109.6630
	f_r	0.1–0.9	0.5655
	d_r	1.0–3.0	3
	V_s	50–500	169.9784
	f_s	0.1–0.9	0.2262
	d_s	1.0–3.0	3
Tool life	T_r	25–45	25.0604
	T_s	25–45	25.0005
Rough operation constraints	$g_1(X)$	≤ 0	-0.0076
	$g_2(X)$	≤ 0	-0.7840
	$g_3(X)$	≥ 0	2126.9
	$g_4(X)$	≤ 0	-134.6691
Finish operation constraints	$g_5(X)$	≤ 0	-99.4092
	$g_6(X)$	≤ 0	-1.7131
	$g_7(X)$	≥ 0	2038.5
	$g_8(X)$	≤ 0	-141.5249
	$g_9(X)$	≤ 0	-9.9951
Constraints of variable relations	$g_{10}(X)$	≥ 0	60.3154
	$g_{11}(X)$	≥ 0	0
	$g_{12}(X)$	≥ 0	0
Unit production cost			2.0351 (\$/piece)

The optimum design is accomplished using the proposed CMPA algorithm, as given in Fig. 18. The results of the CMPA,

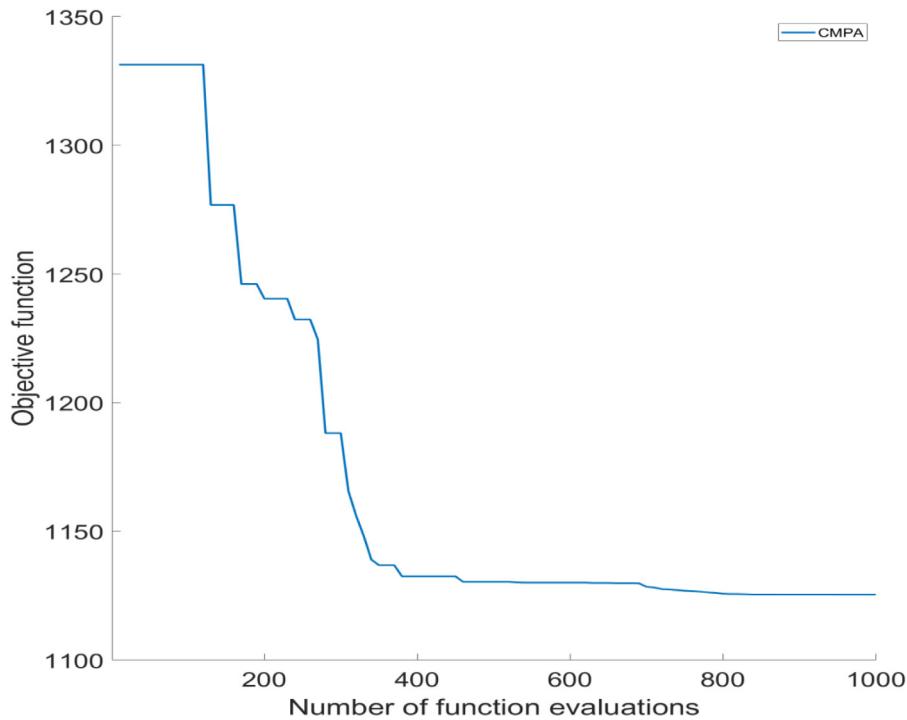


Fig. 19. Convergence history of the CMPA for the brake pedal problem.

Table 15

Comparison of different optimization methods (case 1: $T_p = T_r + T_s$).

Method	Cutting speed (m/min)		Feed rate (mm/rev)		Depth of cut (mm)		UC (\$/piece)	Violated constraints
	V_r	V_s	f_r	f_s	d_r	d_s		
CMPA (iterative map)	123.3429	169.9786	0.5655	0.2262	3	3	1.9591	0
FPA [103,104]	123.3431	169.9785	0.5655	0.2262	3	3	1.9591	0
COA [105]	123.1462	169.9876	0.5655	0.2262	3	3	1.959	0
GA [106]	114.22	164.369	0.7	0.2978	2.9745	2.9863	1.8450	27,28,29,35,42,43
PSO [107]	106.69	155.89	0.897	0.28	2	2	2.272	0
ACO [108]	103.05	162.02	0.9	0.24	–	–	1.626	44 not considered
HPSO [109]	123.3424	169.9783	0.5655	0.2262	3	3	1.959	0
SA-PS [109]	–	–	–	–	–	–	2.313	–
TLBO [110]	110	170	0.565	0.225	3	3	1.973	0
HRDE [98]	–	–	–	–	–	–	2.046	–
DERE [111]	–	–	–	–	–	–	2.046	–
ABC [111]	–	–	–	–	–	–	2.118	–
DE [111]	–	–	–	–	–	–	2.136	–
HABC [111]	–	–	–	–	–	–	2.046	–
HRTLBO [99]	–	–	–	–	–	–	2.046	–
GA-SQP [112]	94.464	162.289	0.866	0.258	3	3	1.814	27, 28
FA [113]	98.4102	162.2882	0.820	0.2582	3	3	1.824	28

MPA, grasshopper optimization algorithm (GOA), harris hawks optimization (HHO) algorithm, dragonfly optimization algorithm (DOA), and salp swarm optimization (SSO) algorithm are presented in Table 17. A minimum mass of 1121.6 g is obtained with the proposed CMPA (iterative map) algorithm.

The convergence history of the CMPA for the brake pedal problem is illustrated in Fig. 19. The CMPA converge to the global optimum with less function evaluation number of 1000 relative to GOA, HHO, DOA, SSO and MPA.

7. Conclusion and future works

A novel chaotic marine predator algorithm (CMPA) is proposed and investigated for engineering design optimization problems. Chaotic maps are integrated into the fundamental MPA algorithm that results in a hybrid algorithm. For performance evaluation,

five single objective test examples, two multi-pass turning operation case studies, and a vehicle brake pedal design problem taken from the industry are executed with the proposed algorithm. CMPA with a circle map, CMPA with a gauss/mouse map, CMPA with a Chebyshev map, CMPA with a sinusoidal map, CMPA with a Chebyshev map, CMPA with iterative map, and CMPA with an iterative map is ranked as the best for the spring design problem, pressure vessel, three bar truss design, piston lever problem, Spur gear design problem, turning problem, and brake pedal design problem, respectively. Additionally, chaos-based MPA is examined for problems of CEC 2020 test suits, and both quantitative and qualitative analysis is performed. The incorporation of chaos theory in MPA illustrates improved performance relative to basic MPA and the other twelve distinguished optimizers. Out of all maps, the piecewise map-based CMPA has been found most efficient on average for numerical functions and interestingly

Table 16Comparison of different optimization methods (case 2: $T_p = \theta T_r + (1 - \theta)T_s$).

Method	Cutting speed (m/min)		Feed rate (mm/rev)		Depth of cut (mm)		UC (\$/piece)	Constraint violations
	V_r	V_s	f_r	f_s	d_r	d_s		
CMPA (iterative map)	109.6630	169.9784	0.5655	0.2262	3	3	2.0351	0
FPA [103]	109.6631	169.9785	0.5655	0.2262	3	3	2.0351	0
COA [105]	117.9322	123.1993	0.5655	0.2262	3	3	2.239	0
HPSO [109]	109.663	169.97	0.5655	0.226	3	3	2.035	0

Table 17

Optimized results for the brake design obtained from the different algorithms.

Method	Best mass (g)	Mean	Worst	Std	Stress (MPa)	NFE
Initial design	1581	–	–	–	189	–
GOA [114]	1161	–	–	–	300	3000
HHO [114]	1140	–	–	–	300	3000
DOA [114]	1135	–	–	–	300	3000
SSO [114]	1125	–	–	–	300	3000
MPA	1126.658	1128.947	1142.081	6.742721	300	1000
CMPA with Chebyshev map	1124.05	1127.971	1138.897	3.337863	300	1000
CMPA with circle map	1123.693	1129.284	1150.37	5.183115	300	1000
CMPA with Gauss/Mouse map	1125.868	1133.748	1149.814	6.158808	300	1000
CMPA with iterative map	1121.645	1126.646	1128.626	2.18177	300	1000
CMPA with logistic map	1124.516	1127.444	1135.193	2.788228	300	1000
CMPA with piecewise map	1124.211	1128.659	1143.799	4.936401	300	1000
CMPA with sine map	1124.252	1129.197	1138.814	4.26157	300	1000
CMPA with singer map	1124.028	1128.993	1141.342	4.039193	300	1000
CMPA with sinusoidal map	1124.206	1127.485	1140.129	3.527734	300	1000
CMPA with Chebyshev map	1123.758	1128.231	1141.015	4.337494	300	1000

significantly better than a non-linear MPA variant and two state-of-the-art previous IEEE CEC competitions winners (LSHADE and LSHADE-EpSin) algorithms. CMPA realizes the best values for all engineering design problems as well while satisfying all design constraints. Moreover, in the manufacturing optimization problem, the proposed CMPA approach performs better than other algorithms and achieves the minimum processing cost. Comprehensively, the hybrid CMPA found an efficient and robust global optimization algorithm for solving real-world engineering design and manufacturing problems.

Every method has advantages and disadvantages, and the CMPA is no exception. With additional chaos mapping, the process may increase the computational time between 32%–67% for the multidimensional numerically constraint problems. Therefore in further investigation, we may combine all chaos mapping in the same algorithm and use some adaptive strategies to decide when each mapping will be activated. Also, the computational complexity of the algorithm will be improved in the next study. As a future work one can explore the proposed CMPA further for numerous two-dimensional and three-dimensional design optimization problems and other manufacturing operations like milling, grinding, drilling, etc. Its further comparisons with state-of-the-art variants such as CMA-ES, IPOP-CMAES, ELSHADE-SPACMA, IMODE and AGDE because these algorithms are widely tested and accepted in the IEEE competition. As CMPA is a state-of-the-art technique, its competency for non-linear, multi-objective design issues should be investigated. Moreover, interested scholars may further analyse their performance improvement with other feasible improvements.

CRediT authorship contribution statement

Sumit Kumar: Writing – original draft, Writing – review & editing, Visualization, Supervision. **Betul Sultan Yildiz:** Conceptualization, Methodology, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Supervision. **Pranav Mehta:** Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Visualization, Supervision. **Natee Panagant:** Writing – review & editing,

Visualization, Supervision. **Sadiq M. Sait:** Writing – review & editing. **Seyedali Mirjalili:** Resources, Validation, Writing – original draft, Writing – review & editing, Supervision. **Ali Riza Yildiz:** Conceptualization, Methodology, Formal analysis, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgement

The author of this project, Natee Panagant (Grant No. N42A650549) funded by the National Research Council Thailand (NRCT).

Appendix

The mathematical formulations of the five mechanical design optimization considered in this study are given as follows:

A.1. Compression/tension spring

Consider: $x = \{d, D, N\}$

Minimize: $f(x) = (N + 2)Dd^2$

Subject to:

$$g_1(x) = 1 - \frac{D^3 N}{71785 d^4} \leq 0$$

$$g_2(x) = \frac{4D^2 - dD}{12566 (Dd^3 - d^4)} + \frac{1}{5.108 d^2} \leq 0$$

$$g_3(x) = 1 - \frac{140.45 d}{D^2 N} \leq 0$$

$$g_4(x) = \frac{D + d}{1.5} - 1 \leq 0$$

where $0.05 \leq d \leq 2$, $0.25 \leq D \leq 1.3$ and $2 \leq N \leq 15$.

A.2. Pressure vessel

Consider: $x = \{T_s, T_h, R, L\}$

$$\begin{aligned} \text{Minimize: } f(x) &= 0.6224 T_s R L + 1.7781 T_h R^2 + 3.1661 T_s^2 L \\ &+ 19.84 T_s^2 R \end{aligned}$$

Subject to:

$$g_1(x) = -T_s + 0.0193 R \leq 0$$

$$g_2(x) = -T_h + 0.00954 R \leq 0$$

$$g_3(x) = -\pi R^2 L - \frac{(4\pi R^3)}{3} + 1296000 \leq 0$$

$$g_4(x) = L - 240 \leq 0$$

where $0.0625 \leq T_s \leq 6.1875$, $0.0625 \leq T_h \leq 6.1875$, $10 \leq R \leq 200$ and $10 \leq L \leq 200$.

A.3. Three bar truss problem

$$\text{Minimize: } f(x) = (2\sqrt{2A_1} + A_2) \times l$$

Subject to:

$$g_1(x) = \frac{\sqrt{2}A_1 + A_2}{\sqrt{2}A_1^2 + 2A_1A_2} p - \sigma \leq 0,$$

$$g_2(x) = \frac{A_2}{\sqrt{2}A_1^2 + 2A_1A_2} p - \sigma \leq 0$$

$$g_3(x) = \frac{1}{\sqrt{2}A_2 + A_1} p - \sigma \leq 0$$

where: $0 \leq A_1 \leq 1$, $0 \leq A_2 \leq 1$, $l = 100$ cm, $P = 2$ kN, $\sigma = 2$ kN/cm²

A.4. Piston lever

Consider: $y = \{H, B, D, x\}$

$$\text{Minimize: } f(y) = \frac{1}{4}\pi D^2 (L_2 - L_1) \quad (38)$$

Subject to:

$$g_1(y) = Q \times L(\cos \theta) - R \times F \leq 0 \text{ at } \theta = 45^\circ$$

$$g_2(y) = Q(L - x) - M_{max} \leq 0$$

$$g_3(y) = 1.2(L_2 - L_1) - L_1 \leq 0$$

$$g_4(y) = \left(\frac{D}{2}\right) - B \leq 0$$

where :

$$R = \frac{|-x(x \sin \theta + H) + H(B - x \cos \theta)|}{\sqrt{(x - B)^2 + H^2}}, F = \frac{\pi PD^2}{4},$$

$$L_1 = \sqrt{(x - B)^2 + H^2}$$

$$L_2 = \sqrt{(x \sin 45 + H)^2 + (B - x \cos 45)^2}, 0.05 \leq H, B, D \leq 500,$$

$$0.05 \leq x \leq 120$$

A.5. Spur gear design

Consider: $x = \{b, Z_1, d_1, d_2, m\}$

$$\begin{aligned} \text{Minimize: } f(x) &= \left(\frac{\pi \rho}{4000}\right) \times \{b \cdot m^2 \cdot Z_1^2 \cdot (1 + u^2) \\ &- (D_i^2 - d_0^2) \times (l - b_w) - n \cdot d_p^2 \cdot b_w - b \cdot (d_1^2 + d_2^2)\} \end{aligned}$$

Subject to:

$$g_1(x) = \left(\frac{S_n \cdot C_s \cdot K_r \cdot K_{ms} \cdot b \cdot J \cdot m}{K_v \cdot K_o \cdot K_m}\right) \geq \frac{(1000 \times P)}{v}$$

$$g_2(x) = \left(\frac{S_{fe}^2 \cdot C_l^2 \cdot C_r^2 \cdot b \cdot D_1 \cdot I}{C_p^2 \cdot K_v \cdot K_o \cdot K_m}\right) \geq \frac{(1000 \times P)}{v}$$

$$g_3(x) = \left(\frac{\sin(\phi)^2 \cdot D_1 \cdot (2D_2 + D_1)}{4m}\right) - D_2 - 1 \geq 0$$

$$g_4(x) = \left(\frac{b}{m}\right) \geq 8$$

$$g_5(x) = \left(\frac{b}{m}\right) \leq 16$$

$$g_6(x) = d_1^3 \geq \left(\frac{48.68 \times 10^6 \times P}{N_1 \cdot \tau}\right)$$

$$g_7(x) = d_2^3 \geq \left(\frac{48.68 \times 10^6 \times P}{N_2 \cdot \tau}\right)$$

$$g_8(x) = \frac{(1 + u) m \cdot Z_1}{2} \leq 250$$

where the pinion rotation speed $N_1 = 1500$ rpm, the transmission capacity $P = 7.5$ kW, the shaft shear strength $\tau = 19.62$ MPa, v is the linear velocity in the pitch circle (m/s), the bending reliability factor $K_r = 0.814$, the mean stress factor $K_{ms} = 1.4$, the mounting factor $K_m = 1.3$, the over load factor $K_o = 1$, the surface fatigue life factor $C_l = 1$, the elastic coefficient $C_p = 191$ (MPa), the surface reliability factor $C_r = 1$, and, the normal pressure angle $\phi = 25^\circ$. The dynamic factor $K_v = (78 + \sqrt{196.85v}/78)$, the standard moore endurance limit $S_n = 1.7236H$, the surface fatigue strength $S_{fe} = 2.8H - 69$, and, the geometry factor $I = (u * \sin\phi\cos\phi)/(2(u + 1))$. $b \in [10, 35]$ mm], $d_1 \in [10, 30]$ mm], $d_2 \in [10, 40]$ mm], $Z_1 \in [18, 25]$ mm], $m \in [1, 1.25, 1.5, 2, 2.75, 3, 3.5, 4]$, and $H \in [200, 400]$.

References

- [1] E. Cuevas, J. Gálvez, O. Avalos, Introduction to optimization and metaheuristic methods, in: E. Cuevas, J. Gálvez, O. Avalos (Eds.), Recent Metaheuristics Algorithms for Parameter Identification, Springer International Publishing, 2020, pp. 1–8, http://dx.doi.org/10.1007/978-3-030-28917-1_1.
- [2] M. Li, G.G. Wang, A review of green shop scheduling problem, Inform. Sci. 589 (2022) 478–496, <http://dx.doi.org/10.1016/j.ins.2021.12.122>.
- [3] W. Li, G.G. Wang, A.H. Gandomi, A survey of learning-based intelligent optimization algorithms, Arch. Comput. Methods Eng. 28 (5) (2021) 3781–3799, <http://dx.doi.org/10.1007/s11831-021-09562-1>.
- [4] Y. Feng, S. Deb, G.G. Wang, A.H. Alavi, Monarch butterfly optimization: A comprehensive review, Expert Syst. Appl. 168 (2021) 114418, <http://dx.doi.org/10.1016/j.eswa.2020.114418>.

- [5] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: Optimization by a colony of cooperating agents, *IEEE Trans. Syst. Man Cybern. B* 26 (1) (1996) 29–41, <http://dx.doi.org/10.1109/3477.484436>.
- [6] R. Eberhart, J. Kennedy, New optimizer using particle swarm theory, in: *Proceedings of the International Symposium on Micro Machine and Human Science*, 1995.
- [7] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm, *J. Global Optim.* 39 (3) (2007) 459–471, <http://dx.doi.org/10.1007/s10898-007-9149-x>.
- [8] J.H. Holland, Genetic algorithms, *Sci. Am.* 267 (1) (1992) 66–72, <http://dx.doi.org/10.1038/scientificamerican0792-66>.
- [9] G.G. Wang, S. Deb, L. Dos Santos Coelho, Earthworm optimisation algorithm: A bio-inspired metaheuristic algorithm for global optimisation problems, *Int. J. Bio-Inspir. Comput.* 12 (1) (2018) 1–22, <http://dx.doi.org/10.1504/ijbic.2018.093328>.
- [10] A. Chakraborty, A.K. Kar, Swarm intelligence: A review of algorithms, in: *Modeling and Optimization in Science and Technologies*, Vol. 10, 2017, pp. 475–494, http://dx.doi.org/10.1007/978-3-319-50920-4_19.
- [11] G.G. Wang, S. Deb, L.D.S. Coelho, Elephant herding optimization, in: *Proceedings - 2015 3rd International Symposium on Computational and Business Intelligence, ISCFBI 2015*, 2016.
- [12] F.A. Hashim, E.H. Houssein, M.S. Mabrouk, W. Al-Atabany, S. Mirjalili, Henry gas solubility optimization: A novel physics-based algorithm, *Future Gener. Comput. Syst.* 101 (2019) 646–667, <http://dx.doi.org/10.1016/j.future.2019.07.015>.
- [13] A.W. Mohamed, A.A. Hadi, A.K. Mohamed, Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm, *Int. J. Mach. Learn. Cybern.* 11 (7) (2020) 1501–1529, <http://dx.doi.org/10.1007/s13042-019-01053-x>.
- [14] L. Abualigah, A. Diabat, Z.W. Geem, A comprehensive survey of the harmony search algorithm in clustering applications, *Appl. Sci. (Switzerland)* 10 (11) (2020) 3827, <http://dx.doi.org/10.3390/app10113827>.
- [15] S. Salcedo-Sanz, Modern meta-heuristics based on nonlinear physics processes: A review of models and design procedures, *Phys. Rep.* 655 (2016) 1–70, <http://dx.doi.org/10.1016/j.physrep.2016.08.001>.
- [16] S. Kumar, G.G. Tejani, N. Pholdee, S. Bureerat, Performance enhancement of meta-heuristics through random mutation and simulated annealing-based selection for concurrent topology and sizing optimization of truss structures, *Soft Comput.* 26 (12) (2022) 5661–5683, <http://dx.doi.org/10.1007/s00500-022-06930-2>.
- [17] S. Kumar, P. Jangir, G.G. Tejani, M. Premkumar, MOTEQ: A novel physics-based multiobjective thermal exchange optimization algorithm to design truss structures, *Knowl.-Based Syst.* 242 (2022) 108422, <http://dx.doi.org/10.1016/j.knosys.2022.108422>.
- [18] M. Li, X. Feng, Research on PID parameter tuning based on improved artificial bee colony algorithm, *J. Phys. Conf. Ser.* (2020).
- [19] M. Mobin, S.M. Mousavi, M. Komaki, M. Tavana, A hybrid desirability function approach for tuning parameters in evolutionary optimization algorithms, *Measure.: J. Int. Measure. Confed.* 114 (2018) 417–427, <http://dx.doi.org/10.1016/j.measurement.2017.10.009>.
- [20] F. Goodarzian, A. Abraham, A.M. Fathollahi-Fard, A biobjective home health care logistics considering the working time and route balancing: A self-adaptive social engineering optimizer, *J. Comput. Des. Eng.* 8 (1) (2021) 452–474, <http://dx.doi.org/10.1093/jcde/qwaa089>.
- [21] S. Nama, A.K. Saha, S. Sharma, A novel improved symbiotic organisms search algorithm, *Comput. Intell.* 38 (3) (2022) 947–977, <http://dx.doi.org/10.1111/coin.12290>.
- [22] G.G. Tejani, V.J. Savsani, V.K. Patel, Adaptive symbiotic organisms search (SOS) algorithm for structural design optimization, *J. Comput. Des. Eng.* 3 (3) (2016) 226–249, <http://dx.doi.org/10.1016/j.jcde.2016.02.003>.
- [23] C.M. Wang, Y.F. Huang, Self-adaptive harmony search algorithm for optimization, *Expert Syst. Appl.* 37 (4) (2010) 2826–2837, <http://dx.doi.org/10.1016/j.eswa.2009.09.008>.
- [24] A.A. Ewees, M. Abd Elaziz, E.H. Houssein, Improved grasshopper optimization algorithm using opposition-based learning, *Expert Syst. Appl.* 112 (2018) 156–172, <http://dx.doi.org/10.1016/j.eswa.2018.06.023>.
- [25] S. Mahdavi, S. Rahnamayan, K. Deb, Opposition based learning: A literature review, *Swarm Evol. Comput.* 39 (2018) 1–23, <http://dx.doi.org/10.1016/j.swevo.2017.09.010>.
- [26] S.K. Sahoo, A.K. Saha, S. Nama, M. Masdari, An improved moth flame optimization algorithm based on modified dynamic opposite learning strategy, *Artif. Intell. Rev.* (2022) <http://dx.doi.org/10.1007/s10462-022-10218-0>.
- [27] G.G. Wang, S. Deb, Z. Cui, Monarch butterfly optimization, *Neural Comput. Appl.* 31 (7) (2019) 1995–2014, <http://dx.doi.org/10.1007/s00521-015-1923>.
- [28] G. Kaur, S. Arora, Chaotic whale optimization algorithm, *J. Comput. Des. Eng.* 5 (3) (2018) 275–284, <http://dx.doi.org/10.1016/j.jcde.2017.12.006>.
- [29] M. Kohli, S. Arora, Chaotic grey wolf optimization algorithm for constrained optimization problems, *J. Comput. Des. Eng.* 5 (4) (2018) 458–472, <http://dx.doi.org/10.1016/j.jcde.2017.02.005>.
- [30] V. Hajipour, M. Tavana, F.J. Santos-Arteaga, A. Alinezhad, D. Di Caprio, An efficient controlled elitism non-dominated sorting genetic algorithm for multi-objective supplier selection under fuzziness, *J. Comput. Des. Eng.* 7 (4) (2020) 469–488, <http://dx.doi.org/10.1093/jcde/qwaa039>.
- [31] R. Sindhu, R. Ngadiran, Y.M. Yacob, N.A.H. Zahri, M. Hariharan, Sine-cosine algorithm for feature selection with elitism strategy and new updating mechanism, *Neural Comput. Appl.* 28 (10) (2017) 2947–2958, <http://dx.doi.org/10.1007/s00521-017-2837-7>.
- [32] P. Wei, Y. Li, Z. Zhang, T. Hu, Z. Li, D. Liu, An optimization method for intrusion detection classification model based on deep belief network, *IEEE Access* 7 (2019) 8751965, <http://dx.doi.org/10.1109/ACCESS.2019.2925828>, 87593–87605.
- [33] Y. Wu, Z. Zhang, R. Xiao, P. Jiang, Z. Dong, J. Deng, Operation state identification method for converter transformers based on vibration detection technology and deep belief network optimization algorithm, *Actuators* 10 (3) (2021) 56, <http://dx.doi.org/10.3390/act10030056>.
- [34] S. Chakraborty, S. Sharma, A.K. Saha, A. Saha, A novel improved whale optimization algorithm to solve numerical optimization and real-world applications, *Artif. Intell. Rev.* 55 (6) (2022) 4605–4716, <http://dx.doi.org/10.1007/s10462-021-10114-z>.
- [35] S. Chakraborty, S. Nama, A.K. Saha, An improved symbiotic organisms search algorithm for higher dimensional optimization problems, *Knowl.-Based Syst.* 236 (2022) 107779, <http://dx.doi.org/10.1016/j.knosys.2021.107779>.
- [36] A.K. Saha, Multi-population-based adaptive sine cosine algorithm with modified mutualism strategy for global optimization, *Knowl.-Based Syst.* 251 (2022) 109326, <http://dx.doi.org/10.1016/j.knosys.2022.109326>.
- [37] S. Sharma, S. Chakraborty, A.K. Saha, S. Nama, S.K. Sahoo, mLBOA: A modified butterfly optimization algorithm with Lagrange interpolation for global optimization, *J. Bionic Eng.* 19 (4) (2022) 1161–1176, <http://dx.doi.org/10.1007/s42235-022-00175-3>.
- [38] S. Sharma, A.K. Saha, G. Lohar, Optimization of weight and cost of cantilever retaining wall by a hybrid metaheuristic algorithm, *Eng. Comput.* 38 (4) (2022) 2897–2923, <http://dx.doi.org/10.1007/s00366-021-01294-x>.
- [39] L.M. Pecora, T.L. Carroll, Synchronization in chaotic systems, *Phys. Rev. Lett.* 64 (8) (1990) 821, <http://dx.doi.org/10.1103/PhysRevLett.64.821>.
- [40] D. Yang, G. Li, G. Cheng, On the efficiency of chaos optimization algorithms for global optimization, *Chaos Solitons Fractals* 34 (4) (2007) 1366–1375, <http://dx.doi.org/10.1016/j.chaos.2006.04.057>.
- [41] H. Gezici, H. Livatliyi, Chaotic Harris hawks optimization algorithm, *J. Comput. Des. Eng.* 9 (1) (2022) 216–245, <http://dx.doi.org/10.1093/jcde/qwab082>.
- [42] H. Rezaie, M.H. Kazemi-Rahbar, B. Vahidi, H. Rastegar, Solution of combined economic and emission dispatch problem using a novel chaotic improved harmony search algorithm, *J. Comput. Des. Eng.* 6 (3) (2019) 447–467, <http://dx.doi.org/10.1016/j.jcde.2018.08.001>.
- [43] D.C. Secui, A modified Symbiotic Organisms Search algorithm for large scale economic dispatch problem with valve-point effects, *Energy* 113 (2016) 366–384, <http://dx.doi.org/10.1016/j.energy.2016.07.056>.
- [44] A. Rezaee Jordehi, A chaotic-based big bang-big crunch algorithm for solving global optimisation problems, *Neural Comput. Appl.* 25 (6) (2014) 1329–1335, <http://dx.doi.org/10.1007/s00521-014-1613-1>.
- [45] G.G. Wang, S. Deb, A.H. Gandomi, Z. Zhang, A.H. Alavi, Chaotic cuckoo search, *Soft Comput.* 20 (9) (2016) 3349–3362, <http://dx.doi.org/10.1007/s00500-015-1726-1>.
- [46] G.G. Wang, A.H. Gandomi, A.H. Alavi, A chaotic particle-swarm krill herd algorithm for global numerical optimization, *Kybernetes* 42 (6) (2013) 962–978, <http://dx.doi.org/10.1108/K-11-2012-0108>.
- [47] C.T. Cheng, W.C. Wang, D.M. Xu, K.W. Chau, Optimizing hydropower reservoir operation using hybrid genetic algorithm and chaos, *Water Resour. Manage.* 22 (7) (2008) 895–909, <http://dx.doi.org/10.1007/s11269-007-9200-1>.
- [48] L. Gharavian, M. Yaghoobi, P. Keshavarzian, Combination of krill herd algorithm with chaos theory in global optimization problems, in: *2013 3rd Joint Conference of AI and Robotics and 5th RoboCup Iran Open International Symposium: Learning, Glorious Future, RIOS 2013*, 2013.
- [49] W. Qiao, Z. Yang, Modified dolphin swarm algorithm based on chaotic maps for solving high-dimensional function optimization problems, *IEEE Access* 7 (2019) 2931910, <http://dx.doi.org/10.1109/ACCESS.2019.2931910>, 110472–110486.
- [50] B. Liu, L. Wang, Y.H. Jin, F. Tang, D.X. Huang, Improved particle swarm optimization combined with chaos, *Chaos Solitons Fractals* 25 (5) (2005) 1261–1271, <http://dx.doi.org/10.1016/j.chaos.2004.11.095>.
- [51] Y. Tian, P. Jiang, Optimization of tool motion trajectories for pocket milling using a chaos ant colony algorithm, in: *Proceedings of 2007 10th IEEE International Conference on Computer Aided Design and Computer Graphics, CAD/Graphics 2007*, 2007.

- [52] B. Wu, S.H. Fan, Improved Artificial Bee Colony Algorithm with Chaos, in: Communications in Computer and Information Science, 2011, https://www.scopus.com/inward/record.uri?eid=2-s2.0-79960432424&doi=10.1007%2f978-3-642-22694-6_8&partnerID=40&md5=3f83e57d25e6f323387cc65aaa46b6a1.
- [53] M.A. El-Shorbagy, A.M. El-Refaey, A hybrid genetic-firefly algorithm for engineering design problems, *J. Comput. Des. Eng.* 9 (2) (2022) 706–730, <http://dx.doi.org/10.1093/jcde/qwac013>.
- [54] S. Gao, C. Vairappan, Y. Wang, Q. Cao, Z. Tang, Gravitational search algorithm combined with chaos for unconstrained numerical optimization, *Appl. Math. Comput.* 231 (2014) 48–62, <http://dx.doi.org/10.1016/j.amc.2013.12.175>.
- [55] M. Misaghi, M. Yaghoobi, Improved invasive weed optimization algorithm (IWO) based on chaos theory for optimal design of PID controller, *J. Comput. Des. Eng.* 6 (3) (2019) 284–295, <http://dx.doi.org/10.1016/j.jcd.2019.01.001>.
- [56] G.G. Wang, A.H. Gandomi, A.H. Alavi, G.S. Hao, Hybrid krill herd algorithm with differential evolution for global numerical optimization, *Neural Comput. Appl.* 25 (2) (2014) 297–308, <http://dx.doi.org/10.1007/s00521-013-1485-9>.
- [57] X. Yuan, J. Zhao, Y. Yang, Y. Wang, Hybrid parallel chaos optimization algorithm with harmony search algorithm, *Appl. Soft Comput.* 17 (2014) 12–22, <http://dx.doi.org/10.1016/j.asoc.2013.12.016>.
- [58] A. Faramarzi, M. Heidarinejad, S. Mirjalili, A.H. Gandomi, Marine Predators Algorithm: A nature-inspired metaheuristic, *Expert Syst. Appl.* 152 (2020) 113377, <http://dx.doi.org/10.1016/j.eswa.2020.113377>.
- [59] M.A.A. Al-Qaness, A.A. Ewees, H. Fan, L. Abualigah, M.A. Elaziz, Marine predators algorithm for forecasting confirmed cases of COVID-19 in Italy, USA, Iran and Korea, *Int. J. Environ. Res. Public Health* 17 (10) (2020) 3520, <http://dx.doi.org/10.3390/ijerph17103520>.
- [60] M. Abd Elaziz, S.B. Thanikanti, I.A. Ibrahim, S. Lu, B. Nastasi, M.A. Alotaibi, M.A. Hossain, D. Yousri, Enhanced Marine Predators Algorithm for identifying static and dynamic Photovoltaic models parameters, *Energy Convers. Manage.* 236 (2021) 113971, <http://dx.doi.org/10.1016/j.enconman.2021.113971>.
- [61] D.S.A. Elminaam, A. Nabil, S.A. Ibraheem, E.H. Houssein, An efficient marine predators algorithm for feature selection, *IEEE Access* 9 (2021) 9404154, <http://dx.doi.org/10.1109/ACCESS.2021.3073261>, 60136-60153.
- [62] E.H. Houssein, D.S. Abdelminaam, I.E. Ibrahim, M. Hassaballah, Y.M. Wazery, A hybrid heartbeats classification approach based on marine predators algorithm and convolution neural networks, *IEEE Access* 9 (2021) 9453821, <http://dx.doi.org/10.1109/ACCESS.2021.3088783>, 86194-86206.
- [63] M. Abdel-Basset, R. Mohamed, M. Elhoseny, R.K. Chakraborty, M. Ryan, A hybrid COVID-19 detection model using an improved marine predators algorithm and a ranking-based diversity reduction strategy, *IEEE Access* 8 (2020) 9079859, <http://dx.doi.org/10.1109/ACCESS.2020.2990893>, 79521-79540.
- [64] A.T. Sahlol, D. Yousri, A.A. Ewees, M.A.A. Al-qaness, R. Damasevicius, M.A. Elaziz, COVID-19 image classification using deep features and fractional-order marine predators algorithm, *Sci. Rep.* 10 (1) (2020) 15364, <http://dx.doi.org/10.1038/s41598-020-71294-2>.
- [65] A.F. Alrasheed, K.A. Alnowibet, A. Saxena, K.M. Sallam, A.W. Mohamed, Chaos embed marine predator (CMPA) algorithm for feature selection, *Mathematics* 10 (9) (2022) 1411, <http://dx.doi.org/10.3390/math10091411>.
- [66] C.J. Sun, F. Gao, A tent marine predators algorithm with estimation distribution algorithm and Gaussian random walk for continuous optimization problems, *Comput. Intell. Neurosci.* 2021 (2021) 7695596, <http://dx.doi.org/10.1155/2021/7695596>.
- [67] M.S. Tavazoei, M. Haeri, Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms, *Appl. Math. Comput.* 187 (2) (2007) 1076–1085, <http://dx.doi.org/10.1016/j.amc.2006.09.087>.
- [68] R.C. Hilborn, Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers, Oxford University Press on Demand, 2000, http://sutlib2.sut.ac.th/sut_contents/64688.pdf.
- [69] D. He, C. He, L.G. Jiang, H.W. Zhu, G.R. Hu, Chaotic characteristics of a one-dimensional iterative map with infinite collapses, *IEEE Trans. Circuits Syst. I* 48 (7) (2001) 900–906, <http://dx.doi.org/10.1109/81.933333>.
- [70] A. Erramilli, R.P. Singh, P. Pruthi, An application of deterministic chaotic maps to model packet traffic, *Queueing Syst.* 20 (1–2) (1995) 171–206, <http://dx.doi.org/10.1007/BF01158436>.
- [71] R.M. May, Simple mathematical models with very complicated dynamics, *Nature* 2615560 (1976) 459–467, <http://dx.doi.org/10.1038/261459a0>.
- [72] Y. Li, S. Deng, D. Xiao, A novel Hash algorithm construction based on chaotic neural network, *Neural Comput. Appl.* 20 (1) (2011) 133–141, <http://dx.doi.org/10.1007/s00521-010-0432-2>.
- [73] A.G. Tomida, Matlab toolbox and GUI for analyzing one-dimensional chaotic maps, in: Proceedings - the International Conference on Computational Sciences and Its Applications, ICCSA 2008, 2008.
- [74] R.L. Devaney, An introduction to chaotic dynamical systems: Second edition, 2018, <http://dx.doi.org/10.4324/9780429502309>.
- [75] B.S. Yildiz, S. Kumar, N. Pholdee, S. Bureerat, S.M. Sait, A.R. Yildiz, A new chaotic Lévy flight distribution optimization algorithm for solving constrained engineering problems, *Expert Syst.* (2022) <http://dx.doi.org/10.1111/exsy.12992>.
- [76] G.G. Tejani, S. Kumar, A.H. Gandomi, Multi-objective heat transfer search algorithm for truss optimization, *Eng. Comput.* 37 (1) (2021) 641–662, <http://dx.doi.org/10.1007/s00366-019-00846-6>.
- [77] S. Winyangkul, K. Wansaseub, S. Sleesongsom, N. Panagant, S. Kumar, S. Bureerat, N. Pholdee, Ground structures-based topology optimization of a morphing wing using a metaheuristic algorithm, *Metals* 11 (8) (2021) 1311, <http://dx.doi.org/10.3390/met11081311>.
- [78] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, *IEEE Trans. Evol. Comput.* 1 (1) (1997) 67–82, <http://dx.doi.org/10.1109/4235.585893>.
- [79] E.H. Houssein, M.A. Mahdy, A. Fathy, H. Rezk, A modified Marine Predator Algorithm based on opposition based learning for tracking the global MPP of shaded PV system, *Expert Syst. Appl.* 183 (2021) 115253, <http://dx.doi.org/10.1016/j.eswa.2021.115253>.
- [80] R. Tanabe, A. Fukunaga, Success-history based parameter adaptation for differential evolution, in: 2013 IEEE Congress on Evolutionary Computation, CEC 2013, 2013.
- [81] N.H. Awad, M.Z. Ali, P.N. Suganthan, R.G. Reynolds, An ensemble sinusoidal parameter adaptation incorporated with L-SHADE for solving CEC2014 benchmark problems, in: 2016 IEEE Congress on Evolutionary Computation, CEC 2016, 2016.
- [82] A.S. Sadiq, A.A. Dehkordi, S. Mirjalili, Q.V. Pham, Nonlinear marine predator algorithm: A cost-effective optimizer for fair power allocation in NOMA-VLC-B5G networks, *Expert Syst. Appl.* 203 (2022) 117395, <http://dx.doi.org/10.1016/j.eswa.2022.117395>.
- [83] S. Gupta, K. Deep, S. Mirjalili, An efficient equilibrium optimizer with mutation strategy for numerical optimization, *Appl. Soft Comput.* 96 (2020) 106542, <http://dx.doi.org/10.1016/j.asoc.2020.106542>.
- [84] O. Olorunda, A.P. Engelbrecht, Measuring exploration/exploitation in particle swarms using swarm diversity, in: 2008 IEEE Congress on Evolutionary Computation, CEC 2008, 2008.
- [85] K. Deep, M. Thakur, A new crossover operator for real coded genetic algorithms, *Appl. Math. Comput.* 188 (1) (2007) 895–911, <http://dx.doi.org/10.1016/j.amc.2006.10.047>.
- [86] J. Arora, Introduction to optimum design, 2012, <http://dx.doi.org/10.1016/C2009-0-61700-1>.
- [87] S. Gupta, H. Abderazek, B.S. Yildiz, A.R. Yildiz, S. Mirjalili, S.M. Sait, Comparison of metaheuristic optimization algorithms for solving constrained mechanical design optimization problems, *Expert Syst. Appl.* 183 (2021) 115351, <http://dx.doi.org/10.1016/j.eswa.2021.115351>.
- [88] S.S. Rao, Engineering optimization: Theory and practice, 2019, <http://dx.doi.org/10.1002/9781119454816>.
- [89] A.R. Yildiz, A comparative study of population-based optimization algorithms for turning operations, *Inform. Sci.* 210 (2012) 81–88, <http://dx.doi.org/10.1016/j.ins.2012.03.005>.
- [90] B.S. Yildiz, N. Pholdee, S. Bureerat, A.R. Yildiz, S.M. Sait, Enhanced grasshopper optimization algorithm using elite opposition-based learning for solving real-world engineering problems, *Eng. Comput.* (2021) <http://dx.doi.org/10.1007/s00366-021-01368-w>.
- [91] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, *Knowl.-Based Syst.* 89 (2015) 228–249, <http://dx.doi.org/10.1016/j.knosys.2015.07.006>.
- [92] S. Mirjalili, The ant lion optimizer, *Adv. Eng. Softw.* 83 (2015) 80–98, <http://dx.doi.org/10.1016/j.advsengsoft.2015.01.010>.
- [93] S. Mirjalili, S.M. Mirjalili, A. Hatamlou, Multi-Verse Optimizer: a nature-inspired algorithm for global optimization, *Neural Comput. Appl.* 27 (2) (2016) 495–513, <http://dx.doi.org/10.1007/s00521-015-1870-7>.
- [94] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper Optimisation Algorithm: Theory and application, *Adv. Eng. Softw.* 105 (2017) 30–47, <http://dx.doi.org/10.1016/j.advsengsoft.2017.01.004>.
- [95] A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems, *Appl. Soft Comput.* 13 (5) (2013) 2592–2612, <http://dx.doi.org/10.1016/j.asoc.2012.11.026>.
- [96] A.H. Gandomi, X.S. Yang, A.H. Alavi, Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems, *Eng. Comput.* 29 (1) (2013) 17–35, <http://dx.doi.org/10.1007/s00366-011-0241-y>.
- [97] T. Yokota, T. Taguchi, M. Gen, A solution method for optimal weight design problem of the gear using genetic algorithms, *Comput. Ind. Eng.* 35 (3–4) (1998) 523–526, [http://dx.doi.org/10.1016/s0360-8352\(98\)00149-1](http://dx.doi.org/10.1016/s0360-8352(98)00149-1).
- [98] A.R. Yildiz, Hybrid Taguchi-differential evolution algorithm for optimization of multi-pass turning operations, *Appl. Soft Comput.* 13 (3) (2013) 1433–1439, <http://dx.doi.org/10.1016/j.asoc.2012.01.012>.

- [99] A.R. Yıldız, Optimization of multi-pass turning operations using hybrid teaching learning-based approach, *Int. J. Adv. Manuf. Technol.* 66 (9–12) (2013) 1319–1326, <http://dx.doi.org/10.1007/s00170-012-4410-y>.
- [100] M.C. Chen, D.M. Tsai, A simulated annealing approach for optimization of multi-pass turning operations, *Int. J. Prod. Res.* 34 (10) (1996) 2803–2825, <http://dx.doi.org/10.1080/00207549608905060>.
- [101] Y.C. Shin, Y.S. Joo, Optimization of machining conditions with practical constraints, *Int. J. Prod. Res.* 30 (12) (1992) 2907–2919, <http://dx.doi.org/10.1080/00207549208948198>.
- [102] G.R. Miodragović, V. Đorđević, R.R. Bulatović, A. Petrović, Optimization of multi-pass turning and multi-pass face milling using subpopulation firefly algorithm, *Proc. Inst. Mech. Eng. C* 233 (5) (2019) 1520–1540, <http://dx.doi.org/10.1177/0954406218774378>.
- [103] S. Xu, Y. Wang, F. Huang, Optimization of multi-pass turning parameters through an improved flower pollination algorithm, *Int. J. Adv. Manuf. Technol.* 89 (1–4) (2017) 503–514, <http://dx.doi.org/10.1007/s00170-016-9112-4>.
- [104] X.S. Yang, Flower Pollination Algorithm for Global Optimization, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2012, https://www.scopus.com/inward/record.uri?eid=2-s2.0-84866704537&doi=10.1007%2f978-3-642-32894-7_27&partnerID=40&md5=672b4c43d7acb93b4c10d26b3416acd.
- [105] M.A. Mellal, E.J. Williams, Cuckoo optimization algorithm for unit production cost in multi-pass turning operations, *Int. J. Adv. Manuf. Technol.* 76 (1–4) (2015) 647–656, <http://dx.doi.org/10.1007/s00170-014-6309-2>.
- [106] G.C. Onwubolu, T. Kumalo, Optimization of multipass turning operations with genetic algorithms, *Int. J. Prod. Res.* 39 (16) (2001) 3727–3745, <http://dx.doi.org/10.1080/00207540110056153>.
- [107] J. Srinivas, R. Giri, S.H. Yang, Optimization of multi-pass turning using particle swarm intelligence, *Int. J. Adv. Manuf. Technol.* 40 (1–2) (2009) 56–66, <http://dx.doi.org/10.1007/s00170-007-1320-5>.
- [108] K. Vijayakumar, G. Prabhaharan, P. Asokan, R. Saravanan, Optimization of multi-pass turning operations using ant colony system, *Int. J. Mach. Tools Manuf.* 43 (15) (2003) 1633–1639, [http://dx.doi.org/10.1016/S0890-6955\(03\)00081-6](http://dx.doi.org/10.1016/S0890-6955(03)00081-6).
- [109] A. Costa, G. Celano, S. Fichera, Optimization of multi-pass turning economies through a hybrid particle swarm optimization technique, *Int. J. Adv. Manuf. Technol.* 53 (5–8) (2011) 421–433, <http://dx.doi.org/10.1007/s00170-010-2861-6>.
- [110] R.V. Rao, V.D. Kalyankar, Multi-pass turning process parameter optimization using teaching-learning-based optimization algorithm, *Sci. Iranica* 20 (3) (2013) 967–974, <http://dx.doi.org/10.1016/j.scient.2013.01.002>.
- [111] A.R. Yıldız, Optimization of cutting parameters in multi-pass turning using artificial bee colony-based approach, *Inform. Sci.* 220 (2013) 399–407, <http://dx.doi.org/10.1016/j.ins.2012.07.012>.
- [112] A. Belloufi, M. Assas, I. Rezgui, Optimization of turning operations by using a hybrid genetic algorithm with sequential quadratic programming, *J. Appl. Res. Technol.* 11 (1) (2013) 88–94, [http://dx.doi.org/10.1016/S1665-6423\(13\)71517-7](http://dx.doi.org/10.1016/S1665-6423(13)71517-7).
- [113] A. Belloufi, M. Assas, I. Rezgui, Intelligent selection of machining parameters in multipass turnings using firefly algorithm, *Modelling Simul. Eng.* 2014 (2014) 592627, <http://dx.doi.org/10.1155/2014/592627>.
- [114] B.S. Yıldız, A.R. Yıldız, The Harris hawks optimization algorithm, salp swarm algorithm, grasshopper optimization algorithm and dragonfly algorithm for structural design optimization of vehicle components, *Mater. Test.* 61 (8) (2019) 744–748.