Engineering with Computers (2023) 39:2463–2514
https://doi.org/10.1007/s00366-021-01591-5

ORIGINAL ARTICLE

A boosted chimp optimizer for numerical and engineering design optimization challenges

Ch. Leela Kumari[1] · Vikram Kumar Kamboj[1,2] · S. K. Bath[3] · Suman Lata Tripathi[1] · Megha Khatri[1] · Shivani Sehgal[1,4]

Received: 6 February 2021 / Accepted: 12 November 2021 / Published online: 24 March 2022

Abstract
Chimp optimization algorithm (ChoA) has a wholesome attitude roused by chimp's amazing thinking and hunting ability with a sensual movement for finding the optimal solution in the global search space. Classical Chimps optimizer algorithm has poor convergence and has problem to stuck into local minima for high-dimensional problems. This research focuses on the improved variants of the chimp optimizer algorithm and named as Boosted chimp optimizer algorithms. In one of the proposed variants, the existing chimp optimizer algorithm has been combined with SHO algorithm to improve the exploration phase of the existing chimp optimizer and named as IChoA-SHO and other variant is proposed to improve the exploitation search capability of the existing ChoA. The testing and validation of the proposed optimizer has been done for various standard benchmarks and Non-convex, Non-linear, and typical engineering design problems. The proposed variants have been evaluated for seven standard uni-modal benchmark functions, six standard multi-modal benchmark functions, ten standard fixed-dimension benchmark functions, and 11 types of multidisciplinary engineering design problems. The outcomes of this method have been compared with other existing optimization methods considering convergence speed as well as for searching local and global optimal solutions. The testing results show the better performance of the proposed methods excel than the other existing optimization methods.

Keywords CEC2005 · Hybrid search algorithms · Meta-heuristics search · Engineering optimization

1 Introduction

Nowadays, artificial intelligence as well as machine learning are rapidly increasing, because it is easy to implement to solve real-life issues which are continuous or discontinuous, constrained or unconstrained [1, 2]. For handling these characteristics using conventional approaches such as the quasi-Newton method, sequential quadratic programming, fast steepest and conjugate gradient, etc. faced difficulties to solve them [3, 4]. In the existing research, all these methods were tested experimentally and noticed that they are not exactly sufficient to obtain effectual solutions to non-continuous, non-differential problems and real-life multi-model problems [5]. Thus, the meta-heuristics algorithm came into the picture which is very simple to understand and easily be implemented to handle several issues. Generally, in optimization, techniques depend on inhabitants to find out the solution on optimal and sub-optimal which is closer to an exact optimal value, located at the nearest point. In this algorithm, the optimization process starts unless the population set of the individuals are generated and then relying on optimization method every individual act for candidate solution for the problem. Thus, by updating the present location with the best position, the population will be up-to-date by reaching maximum iterations. In modern research, the meta-heuristics algorithm which gives better efficiency, less expensive, and successful in implementation is given prior importance to utilize.

With such traits integrated, a new hybrid meta-heuristics optimization approach, ICHIMP-SHO algorithm is suggested in this research that depends on nature-lead and its

✉ Vikram Kumar Kamboj
vikram.23687@lpu.co.in

1 School of Electronics and Electrical Engineering, Lovely Professional University, Phagwara, India

2 Schulich School of Engineering, University of Calgary, Calgary, Alberta, Canada

3 Department of Electrical Engineering, GZSCCET-MRS Punjab Technical University, Bathinda, Punjab, India

4 DAV Institute of Engineering and Technology, Jalandhar, Punjab, India

mathematical formulation of search functions was designed to offer good competitiveness to current existing meta-heuristics optimizers. The intention to design this optimization technique is motivated by individual intelligence and sensual movement of social carnivores, named Chimps for their mass hunting mannerism in targeting the prey [6]. Hence, a stochastic and meta-heuristic mathematical model intended to handle various optimization problems and is verified by testing experimentally in this research work.

It is true that optimization technique is a large field of study, and researchers are rapidly applying new approaches to provide better answers to various issues that target specific obstacles and can succeed in their discoveries. In research, old techniques give way to new approaches, which use a hybrid unique strategy to eliminate inefficient ways from the present. In this suggested study, a collection of research articles is offered in the literature review to enumerate the flaws of modern algorithms.

Broadly speaking, meta-heuristics are of two types, named single solution-based meta-heuristics and population solution-based meta-heuristics. Improved Chimp (ICHIMP) variant belongs to swarm intelligence-based algorithm of the categories of population meta-heuristics, which is combined along with newly introduced swarm intelligence-based algorithm called Spotted Hyena Optimizer algorithm and named as Improved Chimp-Spotted Hyena Optimizer (ICHIMP-SHO) algorithm which is introduced in this paper. On the whole, this algorithm is simple to apply and involves very few operators than other population-based algorithms with minimum computational efforts.

The remaining parts of the present article contain literature review on related algorithms in Sect. 2, and concepts of improved chimp optimizer (ICHIMP) algorithm are discussed in Sect. 3. Sections 4 and 5 describe spotted hyena optimizer (SHO) algorithm and proposed ICHIMP-SHO algorithm, respectively. Standard benchmark functions are described in Sect. 6. Section 7 showcases the outcomes and comparison of results with other existing algorithms. Testing of 11 engineering-based optimization design problems are shown in Sect. 8, and finally, conclusion and future scope of the paper are presented in Sect 9.

2 Literature review

Meta-heuristics approaches have been frequently used in recent years due to their efficiency when compared to other approaches. These algorithms provide a more effective answer to real-world optimization problems. As a result, new meta-heuristics algorithms must be introduced to overcome these optimization challenges. Meta-heuristics optimization algorithms (MOAs) are important in the ever-increasing use of engineering applications. Because of the complexity of today's situations, the need for the most up-to-date MOAs is quickly growing.

It acquires distinct profits as: (i) Its natural algorithmic structure helps to implement it effortlessly; (ii) this suits real-life problems in engineering as it is a derivation-free mechanism; (iii) when compared to traditional optimization algorithms, this has better ability to minimize local optima; (iv) this is flexible in applying on different problems as its structure does not need any particular changes; (v) because of its simplicity and efficiency, this can be applied simultaneously in hardware applications as well as in computing applications. [like Field Programmable Gate Array (FPGA)] [6].

To limit the drawbacks of classical methods, meta-heuristics search algorithms were introduced. Few such algorithms are Biogeography-based optimization (BBO) [7], Artificial Bee Colony (ABC) [8], Differential Evolution (DE) [9], Genetic algorithm (GA) [10], Cuckoo Search algorithm (CSA) [11], Bacterial Foraging algorithm (BFA) [12], Flower pollination algorithm (FPA) [13], Chemical Reaction optimization (CRO) [14], Firefly algorithm (FA) [15], Immune algorithm (IA) [16], Teaching–Learning-based optimization algorithm [17], Particle Swarm optimization algorithm (PSO) [18], Grey wolf optimization (GWO) [19], Social spider for constrained optimization (SSO-C) [20], Gravitational Search algorithm (GSA) [21], and Bat algorithm (BA) [22]. The reasons how meta-heuristics algorithms are classified are explained in [23, 24], and with reference to [25, 26], meta-heuristics algorithms are considered by natural behavior and divided as single solution-based and population-based algorithms. Examples for single-based algorithms and population-based algorithms are: Variable Neighbourhood search (VNS) [27], Vortex search algorithm (VS) [28], whereas Simulated Annealing (SA) [29], Genetic algorithm (GA) [30], and Tabu search (TS) [31] have an emerging way to find a solution for combinatorial real-world problems in covering and scheduling, Cuckoo search algorithm (CSA) [32], Gravitational search algorithm (GSA) [33], Evolutionary programming (EP) [34] are a fast technique and classical evolutionary programmings were performed on real-world problems. Harmony search (HS) [35] is inspired using the music production cycle analogy. HS may not need the initial values of the variables for decision. Forest Optimization Algorithm (FOA) [36] is for finding maximum value and minimum value with a real application and found that the FOA can typically find solutions correctly. Grey Wolf Optimizer Algorithm (GWO) [19] work was inspired by a Swarm intelligence optimization through the grey wolves and the suggested model imitated the grey wolves' social hierarchical and hunting behavior. Moth Flame Optimizer (MFO) [37], the key influence of this optimizer is the moth navigation system called transverse orientation in nature. Moths migrate in darkness by keeping a preset moon angle, a very effective method for long-distance flying in a straight line. However, such fancy insects are stuck around artificial lights in a useless/deadly spiralling course. Stochastic Fractal Search Algorithm (SFS) [38] centered on random fractals to address global optimization problems with continuous variables, both constrained and unconstrained. In the entire optimization, if only one solution carries then it is known as a single solution-based algorithm and if there are many different solutions in the whole optimization phase, then it is a population-based algorithm, and as such, the solution may coincide with the optimum very nearly.

The two main components of meta-heuristics are exploitation and exploration [25]. Exploration extends searching widely to produce many different solutions, whereas exploitation focuses on searching in a specified area, assuming that area is the best for the present. It is very much important and necessary to balance these two components exploitation and exploration in MOA to keep away the fluctuations in the rate of convergence, as well preventing local and global optimum [39, 40]. Exploitation indicates single solution-based meta-heuristics and exploration indicates populated solution-based meta-heuristics.

Optimization problems can find solutions by nature-inspired MOAs' physical or biological behavior implementation. They are classified into four main classes (Fig. 1) [24, 41]: Swarm Intelligence based algorithm, Evolutionary algorithms (EAs), Human-based, and Physics-based algorithms. The below is the survey made on the algorithms which fall under these four categories. Among them, first, the Evolutionary algorithms replicate features of biological generation like recombining, mutation, and selecting processes [23]. The famous Evolutionary algorithms are Differential Evolution (DE) which presented the minimization of potentially nonlinear and non-differentiable continuous space functions. It only requires some strong control variables, taken from a perfectly defined number interval, Evolutionary Strategy (ES) [42], Biogeography-based optimization (BBO) made analysis of biological species, that can be used to deduce algorithms suitable for optimization. Evolutionary Programming (EP) and Genetic algorithm (GA) are drawn from Darwinian Theory. Second, as per [41, 43], Physics-based algorithms are analogous to natural physical laws. The famous algorithms are Quantum Mechanics-Based (QMBA) and Gravitational Search (GSA) which were influenced by the Gravitational Law and the theory of mass interaction. GSA utilizes Newtonian mechanics theory, and its search agent is the set of masses. Few more physics-based algorithms are Central Force Optimization (CFO) [44], Charged System Search (CSS) [45], Electromagnetism Like Algorithms (ELA) [46], Lightning Attachment Procedure Optimization (LAPO) [41], Big-Bang Big-Crunch (BBBC) [47], and Adaptive gbest-guided gravitational search algorithm (AGBGSA) [48]. Third, MOAs

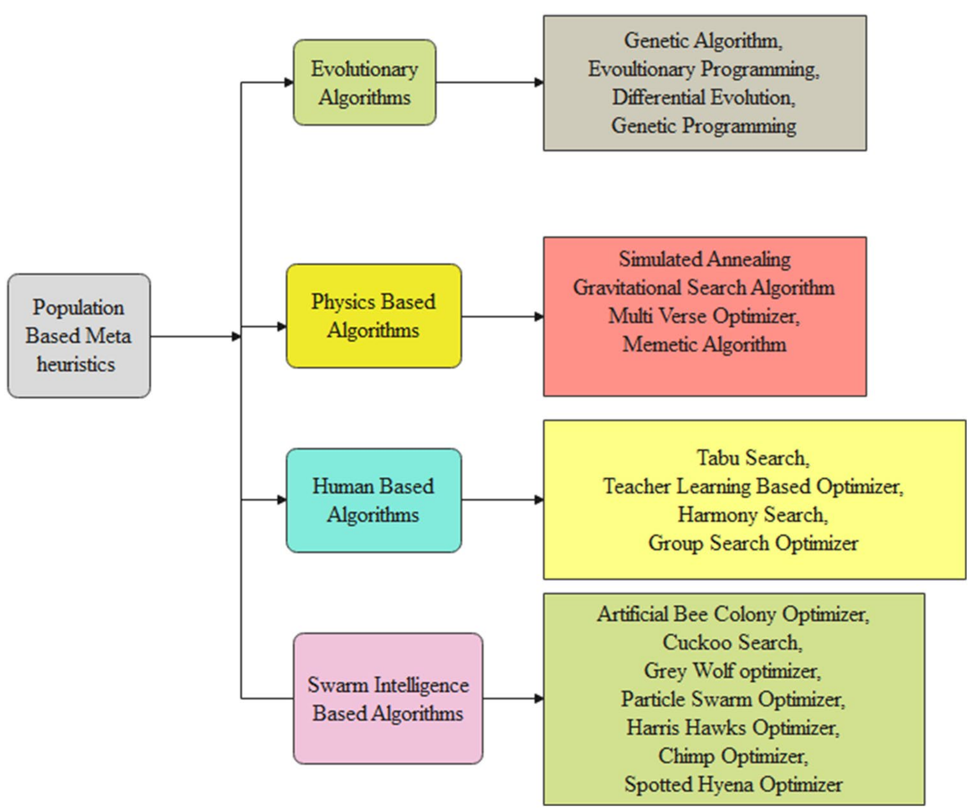


Fig. 1 Classifications of population-based meta-heuristics search algorithms

are inspired by natural human behavior. The best examples of them are Teaching–Learning-based optimization (TLBO) which comprises of two phases, teaching phase and learner phase, Imperialist Competitive Algorithm (ICA) [49], and Socio Evolution and Learning Optimization (SELO) [50]. Fourth, MOAs imitate the social behavior of organisms like swarms, shoals, flocks, or herds [51]. Few algorithms under this class are Particle Swarm optimization (PSO), Bat algorithm (BA), Ant colony optimization (ACO), Improved monarch butterfly optimization algorithm (MBO) [52], Cuckoo Search algorithm (CSA), Krill herd (KH) [53], Grey wolf optimizer (GWO), Multi-Objective Grasshopper optimization algorithm (MOGOA) [54], binary salp swarm algorithm (BSSA) [55], hybrid dragonfly optimization algorithm and MLP (DOA-MLP) [56], and Improved Whale Trainer [57].

A brief of recently developed algorithms to find solution for optimization problems: Harris Hawks optimizer (HHO) [25] is being introduced to tackle different tasks of optimization. The strategy is influenced by nature's cooperative activities and by the patterns of predatory birds, Harris' hawks. Henry Gas Solubility Optimization Algorithm (HGSO) [58] imitates the procedures of Henry’s rule. HGSO aimed at matching the production and conservation capabilities of check room and overcome local optimum. Photon Search Algorithm (PSA) [59] got inspired by the properties of photons in the field of physics. Chaotic Krill Herd Algorithm (CKH) [60] combined chaos theory with Krill Herd Optimization procedure to speed up global convergence. Bird Swarm Algorithm (BSA) [61] depends on social interactions of swarm intelligence with bird swarm. Lightning Search algorithm (LSA) [62] is a meta-heuristic technique used to resolve problems on constraint optimization by following lightning phenomenon applying the concept of fast-moving particles called projectiles. Multi-Verse Optimizer (MVO) [63], an environment lead heuristic algorithm, relies on three stages named: wormhole, black hole, and white hole. Virus Colony search (VCS) [64] is an environment-inspired method that affects the spreading and infection stages of the host cells followed by the virus for its survival in the cell environment. To find solutions for real-time problems, the Grasshopper Optimization algorithm (GOA) [65] follows grasshopper swarms behavior. Based on the thinking ability of the chicken swarm, the Chicken Swarm Optimization algorithm (CSO) [66] came into existence. Grey Wolf Optimizer-Sine Cosine Algorithm (GWO-SCA) [67] is a meta-heuristics optimizer correlating the nature of wolf with mathematical sine–cosine concepts. Crow Particle Swarm Optimization algorithm (CPO) [68] is a hybrid combination of crow search algorithm and particle swarm optimization. Whale Optimization technique (WOA) [69] is a hybridized combinatorial meta-heuristics technique of Whale and swarm human-based optimizers for finding perfect exploratory and convergence capabilities. Spotted Hyena Optimizer (SHO) [70] is a new meta-heuristic algorithm encouraged by the natural collaborative behavior of spotted hyenas in searching, encircling, and attacking the prey. Multi-Objective Spotted Hyena Optimizer (MOSHO) [71] is developed to reduce multiple objective functions. A modified adaptive butterfly optimization algorithm (BOA) [72] is developed based on butterfly observation that produces its fragrance when traveling in search of food from one place to another place. Binary Spotted Hyena Optimizer (SHO) [73] is a meta-heuristic algorithm introduced based on hunting behavior of spotted hyena which deals with discrete optimization problems. Hybrid Harris Hawks pattern search algorithm (HHO-PS) [74] is a meta-heuristic optimizer developed to figure out a newer version of Harris Hawks for finding a solution in local and global search. The Hybrid Harris Hawks-Sine–Cosine method (HHO-SCA) [75] is influenced by the virtuous behavior of Harris Hawks which added up with mathematical concepts of sine and cosine to increase its ability in exploration and exploitation phases. Bernstrain-Search Differential Evolution algorithm (EBSD) [76] belongs to a family of universal differential evolution algorithms, which is proposed based on mutation and crossover operators. Reliability-based design optimization algorithm (RBDO) [77] deals with the uncertainty factors like global convergence, complicated design variables. Table 1 presents a brief review on population based meta-heuristics.

2.1 Literature survey on CHIMP variants

A specific related study has been provided in this area to investigate information regarding current developments linked to CHIMP variations, and recently developed methods by various researchers are mentioned. As demonstrated by the stated literature studies, the researcher has built a wide range of meta-heuristic and hybrid versions of CHIMP to solve various sorts of stochastic challenges. Various academics evaluated real-time troubles such as data mining, climatic and environment concerns, medication and pharmaceuticals, engineering design issues, picture segmentation, power flow, solar PV modules, and so on using a heuristic technique. The capacity of any algorithm to find a suitable balance between intensification and diversity determines the accuracy of its answer. According to research, slow convergence is a common problem with most heuristic algorithms. As a result, the computational efficiency suffers. As a result, the use of hybrid algorithms to improve solution efficiency is becoming increasingly popular. Various CHIMP approaches have also been successfully employed by many researchers to maximize specific objective functions. The ultimate objective of these methods is to discover the optimal solution to a problem.

Table 1 A brief review on few of population meta-heuristics

Year	No. of benchmark functions	Technique and reference number	Name of authors	Complication
2021	29	Arithmetic optimization algorithm [78]	L L. Abualigah et al.	Engineering design problem
2021	30	Archimedes optimization algorithm [79]	F F.A. Hashim et al.	Engineering design optimization
2021	14	Modified butterfly optimization algorithm [72]	L. et al.	Engineering design problem
2021	23	hSMA-PS [80]	L. A. Bala Krishna et al.	Standard benchmark and engineering design problem
2021	23	Aquila optimizer [81]	L. L. Abualigah et al.	Standard benchmark and engineering design problem
2021	30	Spiral motion mode embedded grasshopper optimization algorithm [82]	L. Z. Xu et al.	Standard benchmark and engineering design problem
2021	NA	Hybrid variational mode decomposition (HVMD) [83]	Z. M. Neshat et al.	Wind turbine power output prediction
2021	NA	Modified krill herd [84]	A. Kaur et al.	Economic load dispatch problem
2021	23	A meliorated Harris Hawks optimizer [85]	A A. Nandi et al.	Combinatorial unit commitment
2021	23	Hunger game search algorithm [86]	A Y. Yang et al.	Standard benchmark and engineering design problem
2021	23	Soccer-inspired meta-heuristics [87]	Y E. Osaba et al.	Optimization problems
2021	32	Hybrid Harris Hawks pattern search algorithm (HHO-PS) [74]	Ardhala Balakrishna, Sohbit Saxena, Vikram Kumar Kamboj	Standard functions, multidisciplinary engineering problems
2021	29	Whale optimization algorithm (WOA) [69]	Vamshi Krishna Reddy, Venkata Lakshmi Narayana	Standard functions, multidisciplinary engineering problems
2020	89	Hybrid multi-population algorithm (HMPA) [88]	Y S. Barshandeh et al.	Standard Benchmark and Engineering Design Problem
2020	33	Slime mould algorithm [89]	S. S. Li et al.	Standard benchmark and engineering design problem
2020	29	Marine predators algorithm [90]	S. A. Faramarzi et al.	Engineering design optimization
2020	30	Chimp optimization algorithm (ChoA) [6]	M.Khishe, M. R. Mosavi	Standard benchmark functions
2020	NA	HSMA_WOA [91]	M. Abdel-Basset, V. Chang, and R. Mohamed	The image segmentation issue (ISP) connected to an infected person's X-ray owing to Covid-19 was investigated in this study
2020	8	K-Means clustering and chaotic slime mould algorithm [92]	Z. Chen and W. Liu	Standard benchmark functions
2020	NA	MOSMA: multi-objective slime mould algorithm [93]	M. Premkumar, P. Jangir, R. Sowmya, H. H. Alhelou, A. A. Heidari, and H. Chen	Multidisciplinary engineering problems
2020	NA	Chaotic Slime Mould Algorithm with Chebyshev Map [94]	J. Zhao and Z. M. Gao	Standard benchmark functions
2020	NA	Chaotic salp swarm algorithm [95]	S. K. Majhi, A. Mishra, and R. Pradhan	The authors conducted a thorough investigation of breast anomalies in thermal imaging using the CSSA algorithm, ensuring a healthy balance between the exploration and exploitation stages
2020	6	Modified Whale Optimization Algorithm [96]	Y. Li, M. Han, and Q. Guo	Standard benchmark functions
2020	31	Adaptive Chaotic Sine Cosine Algorithm [97]	Y. Ji et al	Standard benchmark functions

Table 1 (continued)

Year	No. of benchmark functions	Technique and reference number	Name of authors	Complication
2020	NA	Chaotic whale optimization algorithm [98]	C. Paul, P. K. Roy, and V. Mukherjee	In this study, a chaotic base whale optimization algorithm was used to investigate combined heat and power economic dispatch in order to reduce fuel costs and emissions. To investigate global issues, two separate nonlinear realistic power regions were used
2020	5	Reliability-based design optimization algorithm (RBDO) [77]	Zeng Meng et al	Engineering problems
2020	4	Bernstrain-search differential evolution algorithm (EBSD) [76]	Hoda zamani, Mohammad H.Nadimi-Shahraki, Shokooh Taghian, Mahdis Banaie-Dezfouli	Engineering design problems
2020	23	Hybrid Harris Hawks-Sine–Cosine algorithm (HHO-SCA) [75]	Vikram Kumar Kamboj, Ayani Nandi, Ashutosh Bhadoria, Shivani Sehgal	Standard functions, multidisciplinary engineering problems
2020	29	Binary spotted hyena optimizer (SHO) [73]	Vijay Kumar, Avneet Kaur	Standard benchmark functions
2020	14	Modified adaptive butterfly optimization algorithm (BOA) [72]	Kun Hu, Hao Jiang, Chen-Gaung Ji, Ze Pan	Standard benchmark functions
2020	20	Chicken Swarm Optimization algorithm (CSO) [66]	Sanchari Deb et al	Standard functions, multidisciplinary engineering problems
2020	23	Photon Search Algorithm (PSA) [59]	Y. Liu and R. Li	Standard benchmark functions
2019	13	Hybrid Particle Swarm and Spotted Hyena Optimizer algorithm(HPSSHO) [99]	Gaurav Dhiman, Amandeep Kaur	Standard benchmark functions and real-life engineering design problem
2019	47	Henry Gas Solubility Optimization Algorithm (HGSO) [58]	F.A Hashim et al	Standard benchmark functions
2019	29	Harris Hawks optimizer (HHO) [100]	A.A. Heidari et al	Standard benchmark functions, engineering problems
2019	28	Self-adaptive differential artificial bee colony algorithm [101]	X X. Chen et al	Optimization
2019	20	The Sailfish Optimizer [102]	S. Shadravan, H. R. Naji, V K. Bardsiri	Standard test function
2019	NA	Synthetic Minority Over-Sampling[103]	C. Verma, Z. Illes, and V. Stoffova	Data communication
2019	29	Harris Hawks optimizer [25]	A. Heidari, et al	Standard benchmark
2018	30	Multi-objective spotted hyena optimizer (MOSHO) [71]	Gaurav Dhiman, Vijay Kumar	Standard benchmark functions
2018	6	Crow Particle Swarm Optimization(CPO)algorithm [68]	Ko-Wei Huang et al	Standard benchmark functions
2017	29	Spotted Hyena Optimizer(SHO) [70]	Gaurav Dhiman, Vijay Kumar	Standard benchmark functions
2017	22	Grey Wolf Optimizer-Sine–Cosine Algorithm (GWO-SCA) [67]	N.Singh, S.B.Singh	Benchmark functions and real-life optimization
2017	19	Grosshopper Optimization algorithm (GOA) [65]	Shahrzad Saremi, Seyedali Mirjali, Andrew Lewis	Multidisciplinary engineering problems
2016	30	Virus colony search (VCS) [64]	Mu Dong Li et al.	Benchmark functions, engineering problems

Table 1 (continued)

Year	No. of benchmark functions	Technique and reference number	Name of authors	Complication
2016	24	Multi-verse optimizer (MVO) [63]	Seyedali Mirjali, Seyed Mohammad Mirjalili, Abdolreza Hatamlou	Standard benchmark functions, engineering problems
2016	18	Bird swarm algorithm (BSA) [61]	Xiang-Bing Meng et al.	Standard benchmark functions
2015	24	Lightning search algorithm (LSA) [62]	Hussain Shareef et al.	Standard benchmark functions
2015	23	Stochastic fractal search algorithm (SFS) [38]	H.Salimi	Standard benchmark functions
2015	36	Moth flame optimizer (MFO) [104]	S.Mirjalili	Standard benchmark functions, engineering problems
2014	22	Binary optimization using hybrid particle swarm optimization and gravitational search algorithm (PSOGSA) [105]	Seyedali Mirjalili et al.	Standard benchmark functions
2014	14	Chaotic Krill Herd Algorithm (CKH) [60]	Gai-Ge Wang et al.	Standard benchmark functions
2014	4	Forest Optimisation Algorithm (FOA) [36]	M. Ghaemi et al.	NA
2014	32	Grey Wolf Optimizer Algorithm (GWO) [19]	S.Mirjalili et al.	Standard benchmark functions, engineering problems
2012	13	Teaching learning based optimization algorithm (TLBO) [26]	R.V. Rao et al.	Standard benchmark functions
2009	23	Gravitational search (GSA) [106]	E. Rashedi et al.	Standard benchmark functions
2008	14	Biogeography-based Optimization (BBO) [107]	D. Simon	Standard benchmark functions
2001	NA	Harmony search (HS) [35]	Z.W. Geem et al.	Musical variables
1999	23	Evolutionary Programming (EP) [108]	Xin Yao, Yong Liu, Guangming lin	Standard benchmark functions
1997	30	Differential Evolution (DE) [9]	R. Storn and K. Price	Standard benchmark functions
1989	NA	Tabu Search (TS) [109]	Fred Glover	Real-world problems

Researchers have recently created novel CHIMP versions for a variety of applications one of which is the DCELM-ChOA algorithm; first, ELMs' parameters are tuned dimensionally, and then, ChOA is applied to acclimatize input layer weights and moreover bias ELM to eventually shoot up the system's stableness and reliability which was invented to obtain accurate X-ray for detection of COVID-19 positive [110]. RVFL-CHOA [111], the standard CHIMP, was enhanced with Random Vector Functional Link (RVFL); RVFL is used to foretell the instant power outcome of the network and the production of power of a solar dish/stirling power plant in a month. SSC [112] Sine–cosine and Spotted Hyena-based Chimp Optimization algorithm was introduced to fight against the limitations of slow convergence and stuck at local optima of ChoA technique and its efficacy was tested on six real-time engineering problems proving its effectiveness with other techniques. SChoA [113] deputes sine–cosine functions with chimp optimization algorithm to modify the equations of standard CHIMP in its hunting procedure in minimizing various limitations of ChoA technique.

The burning topic is the challenge of discovering solutions to difficulties for optimization. If the number of optimization parameters continues to grow, the complexity of optimization issues will increase. Furthermore, some of the proposed deterministic techniques are vulnerable to local optima entrapment. To solve such issues, meta-heuristic (MA) nature-inspired optimization approaches are used. The lack of starting assumptions and population dependency are two key features of these approaches. Even still, no optimization strategy has yet been discovered that can solve all optimization problems [114]. This inspired to create the Improved Chimp-Spotted Hyena Optimizer, a meta-heuristic hybrid variation optimizer (ICHIMP-SHO).

Chimp Optimization Algorithm (ChoA) [6] is designed based on the intelligence ability of Chimps in group hunts. This algorithm is developed to solve slow convergence speed, trapping in high-dimensional problems. Spotted Hyena optimizer (SHO) is a new upcoming optimizer influenced by the trapping behavior of spotted hyena. This technique benefits upon other meta-heuristics as follows:

(i) implementation of the algorithm is easy because of its simple structure;

(ii) it makes smooth continuous solutions in local optimum;

(iii) it has finer local and global search capability;

(iv) due to the continued diminution of search space, SHO convergence rate is faster. And this solves many types of engineering design problems [70].

Data mining feature selection and unit commitments are the major discrete optimization issues. To solve these problems, SHO is used. Feature selection targets unnecessary features and removes them from the data set and minimizes computation requirement, dimensionality, and results in better accuracy. In practice, real-time problems may have a huge number of features with relevant and irrelevant features. At that time, it is difficult for finding a solution. Then, the characteristic selection is treated as a combinatorial optimization problem. To solve this, selection feature problem binary meta-heuristics algorithms are used. Few examples are Binary Gravitational Search algorithm (BGSA) [115], Binary Grey Wolf optimizer (BGWO) [116], Binary Bat algorithm (BBA) [117, 118], and Binary Particle Swarm optimization (BPSO) [119].

Some of Spotted Hyena optimizer algorithm variants are: HPSSHO algorithm targets in improving hunting tactic of spotted hyena by merging standard SHO with Particle Swarm Optimization and tested on standard benchmark functions to prove its effectiveness in regulating to validate the significance of the proposed HPSSHO performance in assessment with state-of-the art optimization techniques; the parametric tests have been conducted on the benchmark functions [120]. HMOSHSSA [121], hybrid technique, uses MOSHO exploration skill, and SSA updates global search for finding best solution than the standard SHO. MOSHEPO [122] combined Multi-objective Spotted Hyena optimizer and Emperor Penguin Optimizer to contemplate many physical and operational constraints. To reduce heating effect, providing ventilation and air conditioning in the systems, a modification is carried out by merging four different meta-heuristic techniques: salp swarm, spotted hyena, wind-driven, and whale optimization algorithm with multilayer perceptron neural network to conquer computation time [123].

2.2 Novelty of proposed research work

(i) The spotted hyena optimizer is used to improve the local search capacity of ICHIMP in the suggested study.

(ii) The specifications of ICHIMP are not changed to preserve the original features of ICHIMP.

(iii) The ICHIMP-SHO method has been successfully applied for seven standard uni-modal benchmark functions, six standard multi-modal benchmark functions, ten standard fixed-dimension benchmark functions, and 11 types of interdisciplinary engineering design challenges.

(iv) The efficacy of the suggested algorithm has been validated by Wilcoxon Rank test.

(v) According to the comparative analysis shown in the results section, the proposed technique performs very well in terms of fitness evaluation and solution precision.

2.3 Background of suggested work

Chimps (Chimpanzees) correspond to a family of African genus of huge chimpanzee. The living style of them is close to humans. Brain-to-body ratio (BBR) of Chimps and Dolphins are alike to humans. It is noticed that mammals along BBR are generally understood to be brilliant [124]. The DNA of human and Chimp are alike as they are from same solitary ancestors that existed a few million years back. Chimps hunt in group. All the chimps in a group are not same according to their ability and brilliance, but they perform their duties as a part of a chimp colony. The hunting procedure entails their natural capacity to communicate among group to drive, chase, and assault in lower canopy. If the prey manages to flee throughout this procedure, the chimps will regroup and launch another attack. In this process, each chimp may switch places. The exhausted victim eventually runs out of energy and is attacked by the chimps. In this procedure, each matching approach has a probability based on the locations of chimps in a group and the prey. Despite a good convergence rate, CHIMP struggles to identify the most optimal solution. As a result, an improved approach is introduced to reduce this effect while increasing its effectiveness.

The literature survey on some newly developed CHIMP variants is: The paper [125] presented ChOA for training artificial neural network and proved best than other existing algorithms. Abbas et al. [126] used a new chimp optimization algorithm to train radial basis function neural network which is the utilized as a detector and further improved to eradicate exploration and exploitation phases by upgrading ChOA and stood better with outstanding performance when compared with five well-noted algorithms. Heming Jia et al. applied enhanced chimp optimization algorithm (EChOA) in [127] and verified its effectiveness on standard benchmark functions in giving tough competition with other algorithms. Jianhao Wang et al. proposed Binary Chimp Optimization algorithm (BChOA) in [128] as the basic ChOA is not suitable in finding solutions for binary problems because of its continuous hunting nature. To evaluate its efficiency, it has been tested on 43 standard benchmark functions obtaining good results. ICHIMP in [129] is implemented to find solutions for dynamic economic load dispatch problems in single area. To overcome the drawbacks of ChOA to stuck in local optima, Di Wu et al. introduced Enhanced Chimp Optimization Algorithm (EChOA); here, highly disruptive polynomial mutation is involved to multiply the population in space to shoot up the diversity in the population, Spearman's rank correlation coefficient calculates the highest and lowest fitness among chimps, and later, Beetle Antenna Search Algorithm (BAS) is used to evade local optimum by chimps with lowest fitness. The combination of these three strategies enhances the exploration and exploitation phases and is tested on 17 benchmark datasets to prove its efficacy. Abdul Jabbar et al. [130] proposed a fresh hybrid algorithm by merging chimp optimization with conjugate gradient algorithm and tested on ten optimization functions, proving that the combination noted good results in gaining optimal solutions. Essam et al. [131] introduced opposition-based Levy Flight chimp optimizer (IChOA) in which opposition-based learning is involved in increasing pop in initializing stage of ChOA and Levy Flight is responsible for improving exploitation ability. This combination brought good results when compared with other algorithms in obtaining better thermography images to detect breast cancer. Bismin et al. [132] introduced Chimp-CoCoWa-AODV to enhance the MANET performance.

The recommended calculation aims to increase the local search capacity of CHIMP utilizing Improved Chimp Optimizer; in an effort to speed up ICHIMP, a combination of ICHIMP-SHO is introduced. Seven standard uni-modal benchmark functions, six standard multi-model benchmark functions, ten standard fixed-dimension benchmark functions, and 11 types of interdisciplinary engineering design challenges are all used to evaluate it. The findings are superior to those of other algorithms now in use.

3 Improved chimp optimizer

Chimps hunt very cleverly remembering the previous track of their attacks and are very closely related to swarm intelligence strategy, and based on this behavior, an innovative algorithm known as Chimp Optimization Algorithm (ChoA) is introduced. Chimps hunt in a group very intelligently based on two phases, namely, exploration and exploitation. Chimps are divided into four parties specifically named driver, barrier, chaser, and attacker. They streamline themselves by chasing, driving, blocking, and attacking in trapping the prey.

The mathematical equations [Eqs. (1) and (2)] represent driving and chasing of the prey

$$\vec{D} = \left|\vec{C}\vec{Y}_{\text{Prey}}(\text{iteration}) - \xi.\vec{Y}_{\text{Chimp}}(\text{iteration})\right| \tag{1}$$

$$\vec{Y}_{\text{Chimp}}(\text{iteration}+1) = \vec{Q}_k + \vec{Y}_{\text{Prey}}(\text{iteration}) - \vec{A}.\vec{D}. \tag{2}$$

Here, $\vec{A}$, ξ, and $\vec{C}$ = coefficient vectors, t = number of current iteration, Chimp location vector $= \vec{Y}_{\text{Chimp}}$, and $\vec{Y}_{\text{Prey}}$ = the vector of prey position.

Coefficient vectors $\vec{A}$, ξ, and $\vec{C}$ are found out using Eqs. (3), (4), and (5).

In the improved chimp optimizer, Eqs. (1) and (2) have been modified as follows:

$$\vec{Y}_{\text{Chimp}}(\text{iteration}+1)=\begin{cases}\vec{Y}_{\text{Prey}}(\text{iteration})-\vec{A}.\vec{D} & \text{if } \xi>0.5\\ Chaotic\_value & \text{if } \xi<0.5\end{cases}, \tag{2-i}$$

where ran(1) and ran(3) represent the random integer values and can be given by the following mathematical equation:

$$\text{ran(index)} = \text{randi([1, SAN],1,3)}, \tag{2-ii}$$

where SAN represents the search agent number;

$$\vec{A} = 2\vec{\eta}\nu_1 - \vec{\eta} \tag{3}$$

$$\vec{C} = 2\nu_2 \tag{4}$$

$$\xi = \text{chaotic vector} \tag{5}$$

$$x_{i+1} = 1.07x_i(7.86x_i - 23.31x_i^2 + 28.75x_i^3 - 13.302875x_i^4). \tag{6}$$

$\left|\vec{A}\right|$ Non-linearly decreases from 2.5 to 0 in both the phases iteratively. The vectors ν_1 and ν_2 are ranged [0, 1]. ξ the chaotic vector serves chimps in the process of trapping (Fig. 3a).

In this hunting process usually, an attacker chimp leads this operation followed by driver, barrier, and chaser. Mathematically, the actions of Chimps are imitated in the sequence initially starting from an attacker, driver, and then barrier; chaser will give better lead to notice the position of prey. Up till now, the location of Chimps is to be updated immediately and store the best positions of Chimps. This process is reflected mathematically in the Eqs. (7), (8), and (9)

$$\vec{D}_{\text{Attacker}} = abs\left|\vec{C}_1\vec{Y}_{Attacker} - \vec{Y}\right|. \tag{7a}$$

In the modify chimp algorithm, the $\vec{D}_{\text{Attacker}}$ has been selected with the help of the following equation:

$$\vec{D}_{\text{Attacker}}=\begin{cases}\left|\vec{C}\vec{Y}_{\text{Attacker}}(\text{iteration})-\xi.\vec{Y}(\text{iteration})\right|;\ |A|<1\\ \left|\vec{C}\vec{Y}_{\text{Attacker}}(ran(1), iteration)-\xi.\vec{Y}(ran(3), iteration)\right|;\ |A|>1\end{cases} \tag{7.a-i}$$

$$\vec{D}_{\text{Barrier}} = abs\left|\vec{C}_2\vec{Y}_{\text{Barrier}} - \vec{Y}\right|. \tag{7.b}$$

In the modify chimp algorithm, the $\vec{D}_{\text{Barrier}}$ has been selected with the help of the following equation:

$$\vec{D}_{\text{Barrier}}=\begin{cases}\left|\vec{C}\vec{Y}_{\text{Barrier}}(\text{iteration})-\xi.\vec{Y}(\text{iteration})\right|;\ |A|<1\\ \left|\vec{C}\vec{Y}_{\text{Barrier}}(ran(1), iteration)-\xi.\vec{Y}(ran(3), iteration)\right|;\ |A|>1\end{cases} \tag{7.b-i}$$

$$\vec{D}_{\text{Chaser}} = abs\left|\vec{C}_3\vec{Y}_{\text{Chaser}} - \vec{Y}\right|. \tag{7c}$$

In the modify chimp algorithm, the $\vec{D}_{\text{Chaser}}$ has been selected with the help of the following equation:

$$\vec{D}_{\text{Chaser}}=\begin{cases}\left|\vec{C}\vec{Y}_{\text{Chaser}}(\text{iteration})-\xi.\vec{Y}(\text{iteration})\right|;\ |A|<1\\ \left|\vec{C}\vec{Y}_{\text{Chaser}}(\text{ran}(1), \text{iteration})-\xi.\vec{Y}(ran(3), iteration)\right|;\ |A|>1\end{cases} \tag{7.c-i}$$

$$\vec{D}_{\text{Driver}} = abs\left|\vec{C}_4\vec{Y}_{\text{Driver}} - \vec{Y}\right|. \tag{7d}$$

In the modify chimp algorithm, the $\vec{D}_{\text{Driver}}$ has been selected with the help of the following equation:

$$\vec{D}_{\text{Driver}}=\begin{cases}\left|\vec{C}\vec{Y}_{\text{Driver}}(\text{iteration})-\xi.\vec{Y}(\text{iteration})\right|;\ |A|<1\\ \left|\vec{C}\vec{Y}_{\text{Driver}}(ran(1), iteration)-\xi.\vec{Y}(\text{ran}(3), \text{iteration})\right|;\ |A|>1.\end{cases} \tag{7.d-i}$$

Equation (2) mentioned above can be used to determine the spot of attacker, barrier, chaser, and driver as per Eqs. (8a)–(8d), respectively

$$\vec{Y}_1 = \vec{Y}_{\text{Attacker}} - \vec{A}_1.\vec{D}_{\text{Attacker}} \tag{8a}$$

$$\vec{Y}_2 = \vec{Y}_{\text{Barrier}} - \vec{A}_2.\vec{D}_{\text{Barrier}} \tag{8b}$$

$$\vec{Y}_3 = \vec{Y}_{\text{Chaser}} - \vec{A}_3.\vec{D}_{\text{Chaser}} \tag{8c}$$

$$\vec{Y}_4 = \vec{Y}_{\text{Driver}} - \vec{A}_4.\vec{D}_{\text{Driver}}. \tag{8d}$$

The overall final positions of all the chimps can be obtained by taking the mean of the attacker, barrier, chaser, and driver positions as per Eq. (9)

$$\vec{Y}(\text{iteration}+1) = \frac{(\vec{Y}_1+\vec{Y}_2+\vec{Y}_3+\vec{Y}_4)}{4}. \tag{9}$$

To generate the initial arbitrary position of search agents, the below mathematical equation can be adopted

$$\vec{Y}_{\mathrm{rand}} = LB_i + \xi \times (UB_i - LB_i)\,;\; i \in 1, 2, 3, \ldots, \mathrm{Dim}\,. \tag{10}$$

The PSEUDO code for calculations of Y1, Y2, Y3, and Y4 are given in Fig. 2a, b.

```
PSEUDO Code for calculation of Y1
r1=rand ();
r2=rand ();
A1=2*a*r1-a;
C1=2*r2;
if abs (A1)<1
D_Attacker=abs (C1*Y_Attacker (j)-Y (i, j));
   else
 D_Attacker=abs (C1*Y (rand_num (1), j)-Y
(rand_num (3), j)
If rand>CR
D_Attacker=Y_Attacker(j)
end
end
Y1=Y_Attacker(j)-A1*D_Attacker;
```

```
PSEUDO Code for calculation of Y2
r1=rand ();
r2=rand ();
 A2=2*a*r1-a;
C2=2*r2;
if abs (A2)<1
D_Attacker=abs (C2*Y_Barrier (j)-Y (i, j));
   else
 D_Barrier=abs (C1*Y (rand_num (1), j)-
Y(rand_num(3),j)
If rand>CR
D_Barrier=Y_Barrier(j)
end
end
Y2=Y_Barrier(j)-A2*D_Barrier;
```

(a) PSEUDO Code for Calculation of Y1 and Y2

```
PSEUDO Code for calculation of Y3
r1=rand ();
r2=rand ();
 A3=2*a*r1-a;
C3=2*r2;
if abs (A3) <1
D_Chaser=abs (C3*Y_Chaser (j)-Y (i, j));
   else
 D_Chaser=abs (C3*Y (rand_num (3), j)-Y
(rand_num (2),j)
If rand>CR
D_Chaser=Y_Chaser(j);
end
end
Y3=Y_Chaser(j)-A3*D_Chaser;
```

```
PSEUDO Code for calculation of Y4
r1=rand ();
r2=rand ();
 A4=2*a*r1-a;
C4=2*r2;
if abs (A4) <1
D_Driver=abs (C4*Y_Driver (j)-Y (i, j));
   else
 D_Driver=abs (C4*Y (rand_num (4), j)-Y
(rand_num (1),j)
If rand>CR
D_Driver=Y_Driver(j);
end
end
Y4=Y_Driver(j)-A4*D_Driver;
```

(b) PSEUDO Code for Calculation of Y3 and Y4

Fig. 2 **a** PSEUDO code for calculation of Y1 and Y2. **b** PSEUDO code for calculation of Y3 and Y4

PSEUDO code of Chimp Algorithm

Algorithm 1: Chimp

Initialize the Chimp population x_{i+1} *(i=1, 2… n)*

Initialize $\vec{\eta}$ **,** ξ **,** $\vec{A}$ ***and*** $\vec{C}$

Calculate the position of each chimp

Divide chimps randomly into independent groups

Until *stopping condition is satisfied*

Calculate the fitness of each chimp

$X_{Attacker}$*=the best search agent*

X_{Chaser}*=the second best search agent*

$X_{Barrier}$*=the third best search agent*

X_{Driver}*=the fourth best search agent*

while *(t <maximum number of iterations)*

for *each chimp:*

Extract the chimp's group

Use its group strategy to update $\vec{\eta}$ **,** ξ ***and*** $\vec{C}$

Use $\vec{\eta}$ *,* ξ *and* $\vec{C}$ *to calculate* $\vec{A}$ *and then* $\vec{D}$

Calculate Y1 and Y2 using Pseudo Code of Fig.2 (a)

Calculate Y3 and Y4 using Pseudo Code of Fig.2 (b)

Calculate Y=(Y1+Y2+Y3+Y4)/4

Update $\vec{\eta}$ **,** ξ **,** $\vec{A}$ ***and*** $\vec{C}$

Update $X_{Attacker}$ **,** X_{Driver} **,** $X_{Barrier}$**,** X_{Chaser}

I=I+1

end while

return $X_{Attacker}$

4 Spotted hyena optimizer

The spotted hyena lives in a group of no less than 100 individuals. They embark on hunting expeditions in groups. Spotted, striped, brown, and aardwolf are the four classifications. These are colossal hunters who know what they are doing. They create a sound that sounds like a human chuckle to communicate with one another. They have spots on their bodies. They devise coordinated arrays to encourage organizational understanding among hyenas.

SHO is mathematically illustrated by three stages, i.e., hunting, encircling, and finally attacking the prey. The present finest solution is prey which is nearer to optimum solution. Remaining hyenas renew their location once that finest solution is determined.

Mathematically, spotted hyenas encircling behavior is formulated using below equations

$$\vec{d}_h = \left|\vec{y}\vec{Q}_q(s) - \vec{Q}(s)\right| \tag{11}$$

$$\vec{Q}(s+1) = \vec{Q}_q(s) - \vec{z}\vec{d}_h, \tag{12}$$

where $\vec{d}_h$ = the gap among prey and hyena. $\vec{y}$ and $\vec{z}$ = coefficient vectors. s = the present iteration. $\vec{Q}_q$ = the vector spot of prey. $\vec{Q}$ = the vector spot of hyena. $\vec{y}$ and $\vec{z}$ are compared as follows:

$$\vec{y} = 2.\vec{r}_1 \tag{13}$$

$$\vec{z} = 2.\vec{H}\vec{r}_2 - \vec{H} \tag{14}$$

$$\vec{H} = 5 - (\text{Itr} \times (5/\text{Max}_{\text{itr}})), \tag{15}$$

where $\text{Itr} = 1, 2, 3, \ldots, \text{Max}_{\text{itr}}$.

Here, $\vec{H}$ from 5 to 0 linearly decreases during iteration process, and maintains steadiness between exploration and exploitation. The random vectors $\vec{r}_1, \vec{r}_2$ ranged [0, 1]. The $\vec{y}$ and $\vec{z}$ values are fine tuned, such that hyenas move to other area about the present position. Using Eqs. (11) and (12), hyenas renew their points randomly all over the prey.

To structure the hunting activities of spotted hyenas, we expect finest searching agent has awareness regarding prey position. Remaining search agents designs an array which is of devoted friends and renews the location for the finest search agent.

Mathematically hunting is formulated as

$$\vec{d}_h = \left|\vec{y}.\vec{Q}_h - \vec{Q}_k\right| \tag{16}$$

$$\vec{Q}_k = \vec{Q}_h - \vec{z}.\vec{d}_h \tag{17}$$

$$\vec{C}_h = \vec{Q}_k + \vec{Q}_{k+1} + \cdots + \vec{Q}_{k+N}, \tag{18}$$

where $\vec{Q}_h$ = first best position of spotted hyena. $\vec{Q}_k$ = the location of remaining spotted hyenas.

N = the count of spotted hyenas can be worked out as

$$N = \text{count}_{Ns}(\vec{Q}_h, \vec{Q}_{h+1}, \vec{Q}_{h+2}, \ldots(\vec{Q}_h + \vec{M})), \tag{19}$$

where vector $\vec{M}$ ranges [0.5, 1]. N_s = the number of candidate solutions, related to the superlative optimum solution in search space. $\vec{C}_h$ is group of N optimum solutions.

To explain the attacking stage, it is necessary to reduce the value of H. Thus, difference in $\vec{z}$ is also reduced due to change in H value which diminished from 5 to 0 during iteration runs.

The mathematically attacking the prey (exploitation) is prearranged by

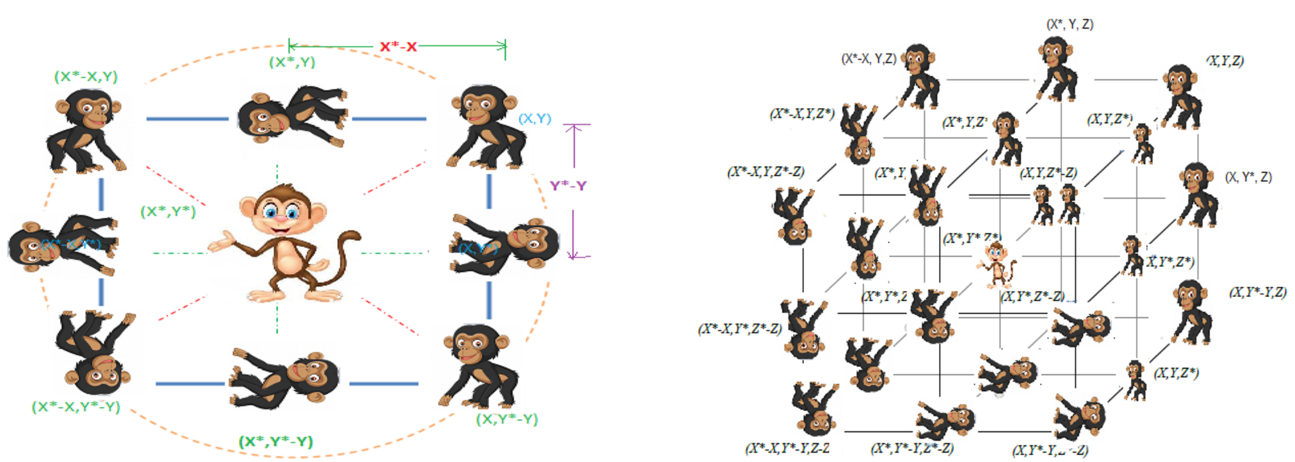
$$\vec{Q}(s+1) = \frac{\vec{C}_h}{N}, \tag{20}$$

where $\vec{Q}(s+1)$ accumulates finest solution and further search agents renew their locations by the positions of finest searching agent. SHO permits their hyenas to renew their locations and attack the prey.

The searching behavior explains the exploration ability of an algorithm. SHO algorithm guarantees the ability of using $\vec{z}$ with random values > 1 or < − 1.

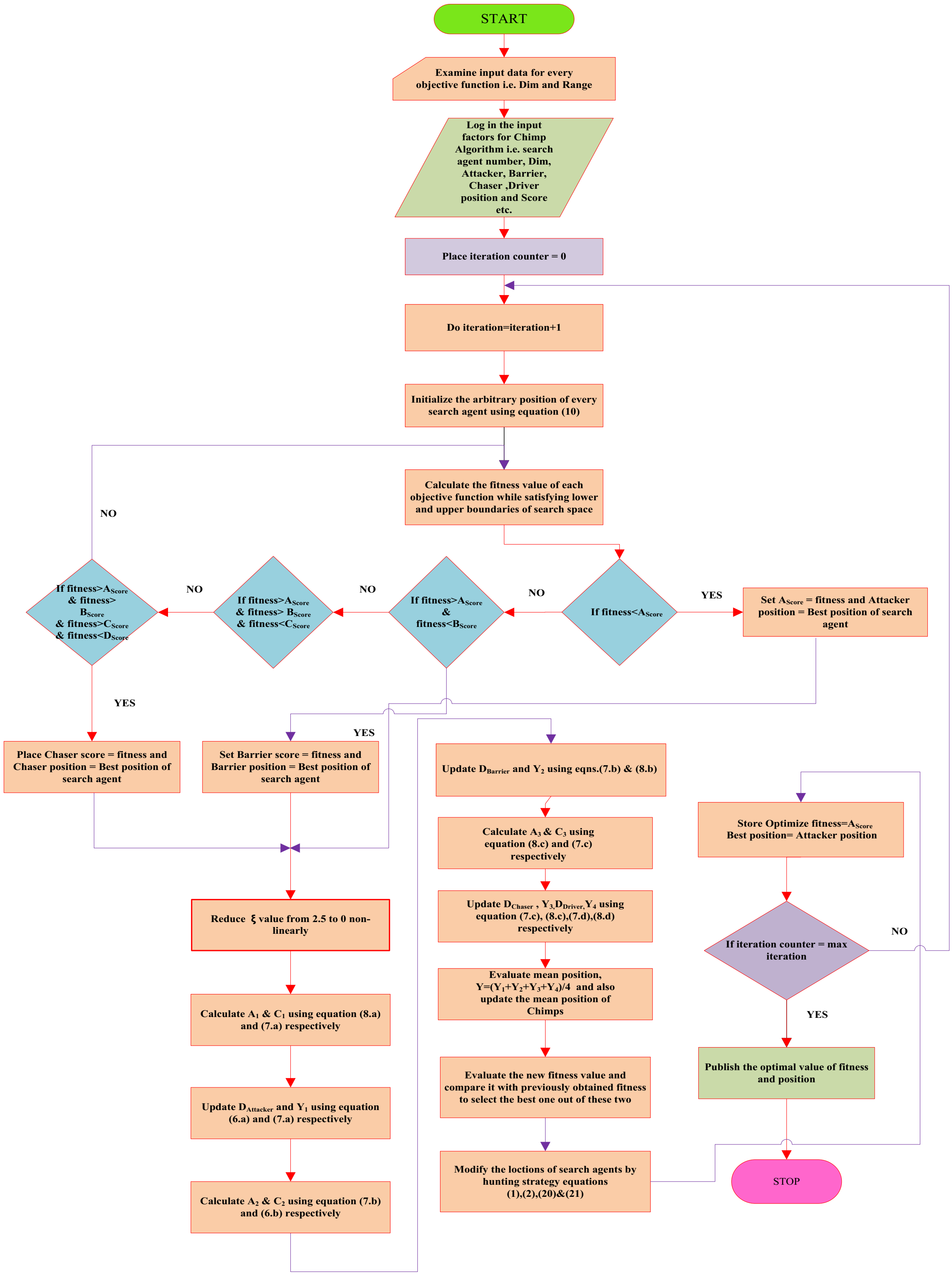
$\vec{y}$ takes the responsibility for more randomized behavior of SHO algorithm and avoids local optimal values.

Below Algorithm 2 depicts spotted hyena optimizer.



(a): 2D view for the Position of Prey and Chimp **(b):** 3D view for the Position of Prey and Chimp

Fig. 3 **a** 2D view for the position of prey and chimp, **b** 3D view for the position of prey and chimp, and **c** flowchart of proposed ICHIMP-SHO algorithm



(c) Flow Chart of proposed ICHIMP-SHO Algorithm

Fig. 3 (continued)

Algorithm 2: Spotted Hyena Optimizer

Initialize population of spotted hyenas Q_i(i=1, 2… n)
Initialize y, z ,H and N
Calculate the fitness value of each spotted hyena
Q_h is set to the best spotted hyena
***C_h** is set to group of optimal solution*
while(s<Max. no. of iterations)do
  for each spotted hyena do
    Update the location of present hyena by eqn. (11)
   end for
Update y ,z ,H and N
Adjust the boundary of hyenas if they go beyond the search space
Compute the fitness value of each spotted hyenas
Update Q_h if it is better than the previous solution
Update the cluster C_h w.r.t. Q_h
s=s+1
end while
return Q_h

5 Proposed improved chimp optimizer (ICHIMP-SHO)

This work extends an enhanced version of hunting behavior of Improved Chimp optimizer by means of spotted hyena, as depicted in Fig. 3c. To experience this consequence, the driving and chasing Eqs. (1) and (2) of IChimp along with hunting behavior of spotted hyena in Eq. (17) are considered to modify into Eq. (21). The pseudo code for the suggested ICHIMP-SHO algorithm is discussed in Algorithm 3

$$\vec{Y}_{\text{Chimp}}(\text{iteration} + 1) = \vec{Q}_k + \vec{Y}_{\text{Prey}}(\text{iteration}) - \vec{A}.\vec{D}. \tag{21}$$

PSEUDO code of Improved Chimp-SHO Algorithm

Algorithm 3: Imp-Chimp-SHO

Initialize the Chimp population x_{i+1} *(i=1, 2… n)*
Initialize $\vec{\eta}$ **,** ξ **,** $\vec{A}$ ***and*** $\vec{C}$
Calculate the position of each chimp
Divide chimps randomly into independent groups
Until *stopping condition is satisfied*
Calculate the fitness of each chimp
$\boldsymbol{X}_{Attacker}$*=the best search agent*
$\boldsymbol{X}_{Chaser}$*=the second best search agent*
$\boldsymbol{X}_{Barrier}$*=the third best search agent*
$\boldsymbol{X}_{Driver}$*=the fourth best search agent*
while *(t <maximum number of iterations)*
for *each chimp:*
Extract the chimp's group
Use its group strategy to update $\vec{\eta}$ **,** ξ ***and*** $\vec{C}$
Use $\vec{\eta}$ *,* ξ *and* $\vec{C}$ *to calculate* $\vec{A}$ *and then* $\vec{D}$
Calculate Y1 and Y2 using Pseudo Code of Fig. 2(a)
Calculate Y3 and Y4 using Pseudo Code of Fig. 2(b)
Calculate Y=(Y1+Y2+Y3+Y4)/4
Update Y further using SHO algorithm
Refer to pseudo code of SHO from algorithm 2
Update $\vec{\eta}$ **,** ξ **,** $\vec{A}$ ***and*** $\vec{C}$
Update $\boldsymbol{X}_{Attacker}$**,** $\boldsymbol{X}_{Driver}$**,** $\boldsymbol{X}_{Barrier}$**,** $\boldsymbol{X}_{Chaser}$
I=I+1
end while
return $\boldsymbol{X}_{Attacker}$

The two-dimensional and three-dimensional views for the position of chimp from the respective prey are depicted in Fig. 3a, b, respectively.

The suggested ICHIMP-SHO variant is beneficial above few population-based meta-heuristic techniques mainly in three aspects as follows.

The first aspect refers in combining two conventional techniques to frame a simple new efficient simulation method which executes faster with complex mathematical operations when compared with other existing methods. The features of standard ICHIMP are injected to SHO technique as initial parameters to strengthen its power which excels in processing and endeavors to optimize these values to boost up the ability of ICHIMP to consider the optimal value of optimization issue. This process is done without involving complex operations.

The second aspect is the proposed new method succeeded in obtaining best results than the solution drawn by ICHIMP. The experimental result stands as proof in the result section displaying its performance in terms of numerically and experimentally. This makes difference between the suggested techniques with other techniques. Majorly, most of the techniques suffer to attain optimum solution with increasing number of iterations due to downside inability. The suggested method develops a vital and standard method to solve this issue which can be practiced by the other methods in optimization by considering the operating phases of this method.

The third aspect is the idea behind the ICHIMP-SHO method is to enhance the optimization strength of ICHIMP to attain the optimized values, but not the complexity of the algorithm. The suggested optimization technique is developed with incorporating SHO algorithm functionality to the ICHIMP. The above two mathematical models have independent structures for managing optimization. To combine them, the computational methods are utilized to transform the principles of one algorithm into the other algorithm. As such, in this research work, ICHIMP pattern is mapped into the SHO parameters and translating SHO attributes back to ICHIMP. Along with this procedure, new operators have been introduced to improve the sophistication of hybrid variants. To examine the proposed hybrid variant ICHIMP-SHO, 16 benchmark functions and 11 constrained engineering optimal issues are considered to verify with different types of parameter settings.

6 Standard benchmark functions

A cluster of unique benchmark functions [30, 105] is used to put the proposed ICHIMP-SHO optimization approach to the test. The standard benchmarks are categorized into uni-modal (UM), multi-modal (MM), and fixed dimensions (FD). Based on objective fitness, these standard benchmark functions are defined such as dimension, range limit, and optimum value ($f_{\min}$). The mathematical formulations for UM, MM, and FD are displayed in Tables 2, 3, and 4, and their results are described in outcomes and discussion section. Thirty trial runs are used to test the performance of standard benchmark functions. Table 5 illustrates the proposed algorithm's details of parameter setting.

The complete study is considered by 30 search agents, and maximum iterations of 500. The suggested ICHIMP-SHO was tested using the MATLAB R2016a software on an Intel corei3 processor laptop with a 7th generation CPU and 8GB RAM.

Table 2 Uni-modal (UM) standard benchmark functions

Functions	Dimensions	Range	$f_{\min}$
$F_1(U) = \sum_{m=1}^{z} U_m^2$	30	[− 100, 100]	0
$F_2(U) = \sum_{m=1}^{z} \lvert U_m \rvert + \prod_{m=1}^{z} \lvert U_m \rvert$	30	[− 10, 10]	0
$F_3(U) = \sum_{m=1}^{z} (\sum_{n-1}^{m} U_n)^2$	30	[− 100, 100]	0
$F_4(U) = \max_m \{\lvert U_m \rvert, 1 \le m \le z\}$	30	[− 100, 100]	0
$F_5(U) = \sum_{m=1}^{z-1} [100(U_{m+1} - U_m^2)^2 + (U_m - 1)^2]$	30	[− 38, 38]	0
$F_6(U) = \sum_{m=1}^{z} ([U_m + 0.5])^2$	30	[− 100, 100]	0
$F_7(U) = \sum_{m=1}^{z} mU_m^4 + random[0, 1]$	30	[− 1.28, 1.28]	0

Table 3 Multi-modal (MM) standard functions

Multi-modal (F8–F13) bench mark functions	Dim	Range	$f_{\min}$
$F_8(U) = \sum_{m=1}^{z} -U_m \sin(\sqrt{\lvert U_m \rvert})$	30	[− 500, 500]	− 418.98295
$F_9(U) = \sum_{m=1}^{z} [U_m^2 - 10\cos(2\pi U_m) + 10]$	30	[− 5.12, 5.12]	0
$F_{10}(U) = -20\exp(-0.2\sqrt{\left(\frac{1}{z}\sum_{m=1}^{z} U_m^2\right)}) - \exp(\frac{1}{z}\sum_{m=1}^{z}\cos(2\pi U_m) + 20 + d$	30	[− 32, 32]	0
$F_{11}(U) = 1 + \sum_{m=1}^{z} \frac{U_m^2}{4000} - \prod_{m-1}^{z} \cos\frac{U_m}{\sqrt{m}}$	30	[− 600, 600]	0
$F_{12}(U) = \frac{\pi}{z}\left\{10\sin(\pi\tau_1) + \sum_{m=1}^{z-1}(\tau_m - 1)^2[1 + 10\sin^2(\pi\tau_{m+1})] + (\tau_z - 1)^2\right\} + \sum_{m=1}^{z} g(U_m, 10, 100, 4)$ Where, $\tau_m = 1 + \frac{U_m+1}{4}$ $g(U_m, b, x, i) = \begin{cases} x(U_m - b)^i U_m > b \\ 0 \quad -b < U_m < b \\ x(-U_m - b)^i U_m < -b \end{cases}$	30	[− 50, 50]	0
$F_{13}(U) = 0.1\left\{\sin^2(3\pi U_m) + \sum_{m=1}^{z}(U_m - 1)^2[1 + \sin^2(3\pi U_m + 1)] + (x_z - 1)^2[1 + \sin^2]\right\}$	30	[− 50, 50]	0

Table 4 Fixed-Dimension (FD) standard functions

Fixed-modal (FD) (F14–F23) standard benchmark functions	Dimension	Range	$f_{\min}$
$F_{14}(U) = [\frac{1}{500} + \sum_{n=1}^{2} 5\frac{1}{n+\sum_{m=1}^{z}(U_m - b_{mn})6}]^{-1}$	2	[− 65.536, 65.536]	1
$F_{15}(U) = \sum_{m=1}^{11} [b_m - \frac{U_1(a_m^2 + a_m\eta_2)}{a_m^2 + a_m\eta_3 + \eta_4}]^2$	4	[− 5, 5]	0.00030
$F_{16}(U) = 4U_1^2 - 2.1U_1^4 + \frac{1}{3}U_1^6 + U_1U_2 - 4U_2^2 + 4U_2^4$	2	[− 5, 5]	− 1.0316
$F_{17}(U) = (U_2 - \frac{5.1}{4\pi 2}U_1^2 + \frac{5}{\pi}U_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos U_1 + 10$	2	[− 5, 5]	0.398
$F_{18}(U) = [1 + (U_1 + U_2 + 1)^2(19 - 14U_1 + 3U_1^2 - 14U_2 + 6U_1U_2 + 3U_2^2)]$ $x[30 + (2U_1 - 3U_2)^2 x(18 - 32U_1 + 12U_1^2 + 48U_2 - 36U_1U_2 + 27U_2^2)]$	2	[− 2, 2]	3
$F_{19}(U) = -\sum_{m=1}^{4} d_m \exp(-\sum_{n=1}^{3} U_{mn}(U_m - q_{mn})^2)$	3	[1, 3]	− 3.32
$F_{20}(U) = -\sum_{m=1}^{4} d_m \exp(-\sum_{n=1}^{6} U_{mn}(U_m - q_{mn})^2)$	6	[0, 1]	− 3.32
$F_{21}(U) = -\sum_{m=1}^{5} [(U - b_m)(U - b_m)^T + d_m]^{-1}$	4	[0, 10]	− 10.1532
$F_{22}(U) = -\sum_{m=1}^{7} [(U - b_m)(U - b_m)^T + d_m]^{-1}$	4	[0, 10]	− 10.4028
$F_{23}(U) = -\sum_{m=1}^{7} [(U - b_m)(U - b_m)^T + d_m]^{-1}$	4	[0, 10]	− 10.5363

Table 5 Parameter constraints for the proposed search method

Parameter setting	ICHIMP-SHO
Number of search agents	30
Number of iterations for benchmark problems (uni-modal, multi-modal, and fixed dimension)	500
Number of iterations for Engineering design problems	500
Number of trial runs for each function and engineering optimal designs	30

Table 6 Test observations of (F1–F7) functions using ICHIMP-SHO algorithm

Function	Mean	St. deviation	Best fitness value	Worst fitness value	Median	Wilcoxon rank sum test	*t* test	
						P value	*P* value	*h* value
F1	3.91443E−28	1.07214E−27	2.2954E−30	5.6993E−27	7.16499E−29	1.7344E−06	0.054971323	0
F2	4.70089E−17	3.86924E−17	7.04913E−18	1.59728E−16	4.01121E−17	1.7344E−06	2.69095E−07	1
F3	8.48976E−07	2.54158E−06	1.11728E−09	1.38683E−05	1.50394E−07	1.7344E−06	0.077611649	0
F4	3.64228E−08	3.08114E−08	2.77549E−09	1.26518E−07	3.2531E−08	1.7344E−06	4.36866E−07	1
F5	28.38318363	0.671703255	26.23392716	28.89070125	28.65306199	1.7344E−06	6.30357E−49	1
F6	1.481698857	0.405138911	0.742524222	2.257134021	1.732292069	1.7344E−06	1.57295E−18	1
F7	0.001127419	0.00056314	0.000275088	0.00238846	0.001036141	1.7344E−06	7.83249E−12	1

The aforementioned parametric settings are the ideal choice for testing the proposed optimizer for standard benchmarks and engineering design challenges.

7 Outcomes and discussion

In this research work, the introduced Improved Chimp-Spotted Hyena Optimizer algorithm is tested on three major classes of standard benchmark functions to verify the presentation of the developed ICHIMP-SHO technique. The exploitation and convergence rate of ICHIMP-SHO is tested by uni-modal benchmark functions which have a single minimum. As the name multi-modal replicates which have more than one minimum, hence, these functions are utilized to test for exploration and avoid local optimum. The design variables are obtained by the difference between multi-modal and fixed-dimension benchmark functions. The fixed-dimension benchmark functions will store these design variables, and maintain a chart of previous data of search space and compare with multi-modal benchmark functions.

Table 7 Execution Time for Uni-modal Benchmark Problems using ICHIMP-SHO algorithm

Function	Best time	Average time	Worst time
F1	1.4375	1.795833333	2.328125
F2	1.390625	1.759895833	1.9375
F3	1.859375	2.118229167	2.296875
F4	1.3125	1.472395833	1.671875
F5	1.34375	1.519270833	1.75
F6	1.34375	1.480729167	1.703125
F7	1.4375	1.60625	1.8125

For comprehensive comparison analysis, a record of results of the developed ICHIMP-SHO algorithm was framed which were tabulated in the criteria of mean value, standard deviation, median value, the best value, worst value, and parametric tests by performing with 500 iterations and maximum runs of 30.

7.1 Evaluation of (F1–F7) functions (exploitation)

The test results for uni-modal (F1–F7) benchmark functions of suggested technique are illustrated in Tables 6, 7. The mean value and standard deviation were considered for evaluation of the test results with few newly developed meta-heuristic algorithms named LSA [62], BRO [133], OEGWO [134], PSA [59], HHO-PS [74], SHO [70], HHO [100], ECSA [135], and TSO [136], and are presented in Table 8. Its characteristic curves, trial runs, and convergence comparative curves with other algorithms are depicted in Figs. 4, 5, 6.

Table 8 Evaluation for (F1–F7) problems

Algorithm	Parameters	(F1–F7) Uni-modal benchmark functions						
		F1	F2	F3	F4	F5	F6	F7
Lightning search algorithm (LSA) [62]	Mean	4.81067E−08	3.340000000	0.024079674	0.036806544	43.24080402	1.493275733	64.28160301
	St.Deviation	3.40126E−07	2.086007800	0.005726198	0.156233023	29.92194448	1.302827039	43.75576111
Battle Royale Optimization algorithm (BRO)[133]	Avg	3.0353E−09	0.000046	54.865255	0.518757	99.936848	2.8731E−08	0.000368
	St.Deviation	4.1348E−09	0.000024	16.117329	0.403657	82.862958	1.8423E−08	0.000094
Opposition based enhanced grey wolf optimization algorithm (OEGWO) [134]	Avg	2.49×10^{-34}	4.90×10^{-25}	1.01×10^{-1}	1.90×10^{-5}	2.72×10^{1}	1.40×10^{00}	3.63×10^{-4}
	St.Deviation	7.90×10^{-34}	6.63×10^{-25}	3.21×10^{-1}	2.43×10^{-5}	7.85×10^{1}	4.91×10^{-1}	2.68×10^{-4}
Photon Search Algorithm (PSA) [59]	Mean	15.3222	2.2314	3978.0837	1.1947	332.6410	19.8667	0.0237
	St.Deviation	27.3389	1.5088	3718.9156	1.0316	705.1589	33.4589	0.0170
Hybrid Harris Hawks Optimizer-Pattern Search algorithm (hHHO-PS) [74]	Avg	9.2×10^{-017}	8.31E	5.03×10^{-20}	6.20×10^{-54}	2.18×10^{-9}	3.95×10^{-14}	0.002289
	St.Deviation	5E−106	4.46×10^{-53}	1.12×10^{-19}	1.75×10^{-53}	6.38×10^{-10}	3.61×10^{-14}	0.001193
Spotted Hyena Optimizer (SHO) [70]	Avg	0	0	0	7.78×10^{-12}	8.59E+00	2.46×10^{-1}	3.29×10^{-5}
	St.Deviation	0	0	0	8.96×10^{-12}	5.53E−01	1.78×10^{-1}	2.43×10^{-5}
Harris Hawks Optimizer (HHO) [100]	Mean	1.06×10^{-90}	6.92×10^{-51}	1.25×10^{-80}	4.46×10^{-48}	0.015002	0.000115	0.000158
	St.Deviation	5.82×10^{-90}	2.47×10^{-50}	6.63×10^{-80}	1.70×10^{-47}	0.023473	0.000154	0.000225
Enhanced Crow search algorithm (ECSA) [135]	Mean	7.4323E−119	5.22838E−59	3.194E−102	3.04708E−52	7.996457081	0.400119079	1.30621E−05
	St.Deviation	4.2695E−118	2.86361E−58	1.7494E−101	1.66895E−51	0.661378213	0.193939866	8.39859E−06
Transient Search Optimization (TSO) [136]	Avg	1.18×10^{-99}	8.44×10^{-59}	3.45×10^{41}	1.28E−53	8.10×10^{-2}	3.35×10^{-3}	3.03×10^{-4}
	St.Deviation	6.44×10^{-99}	3.93×10^{-58}	1.26×10^{-41}	6.58×10^{-53}	11	6.82×10^{-3}	3.00×10^{-4}
ICHIMP-SHO (Proposed algorithm)	Mean	3.91443E−28	4.70089E−17	8.48976E−07	3.64228E−08	28.38318363	1.481698857	0.001127419
	St.Deviation	1.07214E−27	3.86924E−17	2.54158E−06	3.08114E−08	0.671703255	0.405138911	0.00056314

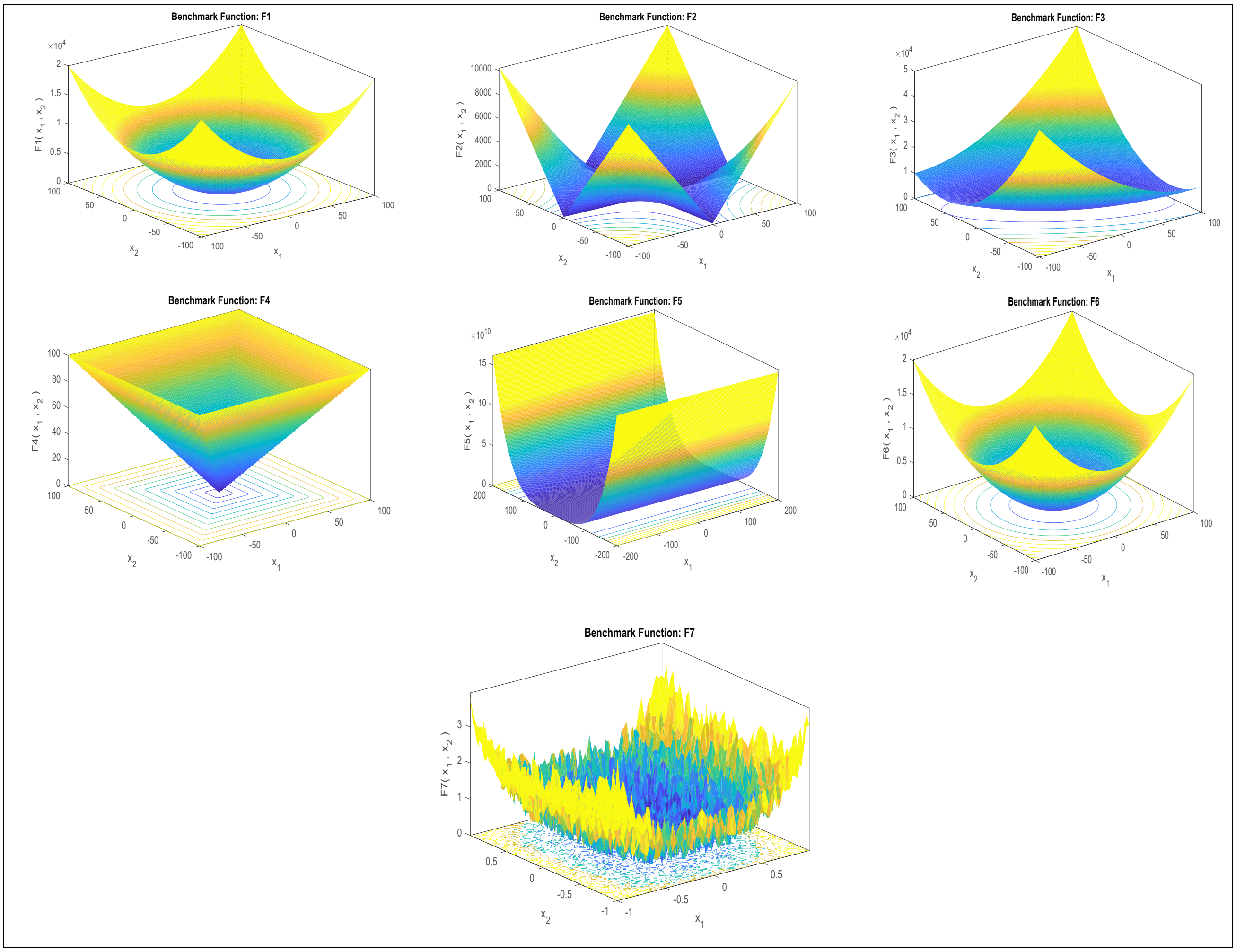


Fig. 4 3D view of uni-modal (UM) standard benchmark problems

7.2 Evaluation of (F8–F13) functions (exploration)

The multi-modal benchmark functions (F8–F13) show the design variables in the desired number in the exploration phase. The test results are tabulated in Tables 9, 10. As well, the comparison of results was done considering mean value and standard deviation with other algorithms, such as LSA [55], BRO [106], OEGWO [107], PSA [40], hHHO-PS [67], SHO [63], HHO [51], ECSA [108], and TSO [109], and is recorded in Table 11. Also, its characteristics curves, trial runs, and convergence comparative curves with other algorithms are depicted in Figs. 7, 8, 9.

7.3 Evaluation of (F14–F23) functions

The fixed-dimensional benchmark (F14–F23) functions do not manipulate the design variables, but prepare the previous search space record of multi-modal benchmark functions. Tables 12, 13 show the test results of proposed algorithm and Table 14 showcases the comparative analysis of mean value and standard deviation with LSA [55], ECSA [108], TSO [109], PSA [40], hHHO-PS [67], SHO [63], and HHO [51]. Figures 10, 11, 12 show characteristics curves, trial runs, and convergence comparative curves with other algorit hms.

Hence, the test results for UM, MM, and FD benchmarks problems are tabulated in Tables 6, 7, 8, 9, 10, 11, 12, 13, 14, and the assessment of the proposed optimizer with other meta-heuristics search algorithms for UM, MM, and FD benchmark problems is given in Figs. 5, 8 and 11 and trial run solutions for UM, MM, and FD benchmarks problems are shown in Figs. 6, 9, and 12. The above result clearly shows that the proposed optimizer presents much better than other algorithms. In subsequent sections, the proposed optimizers have been applied to 11 engineering optimization problems.

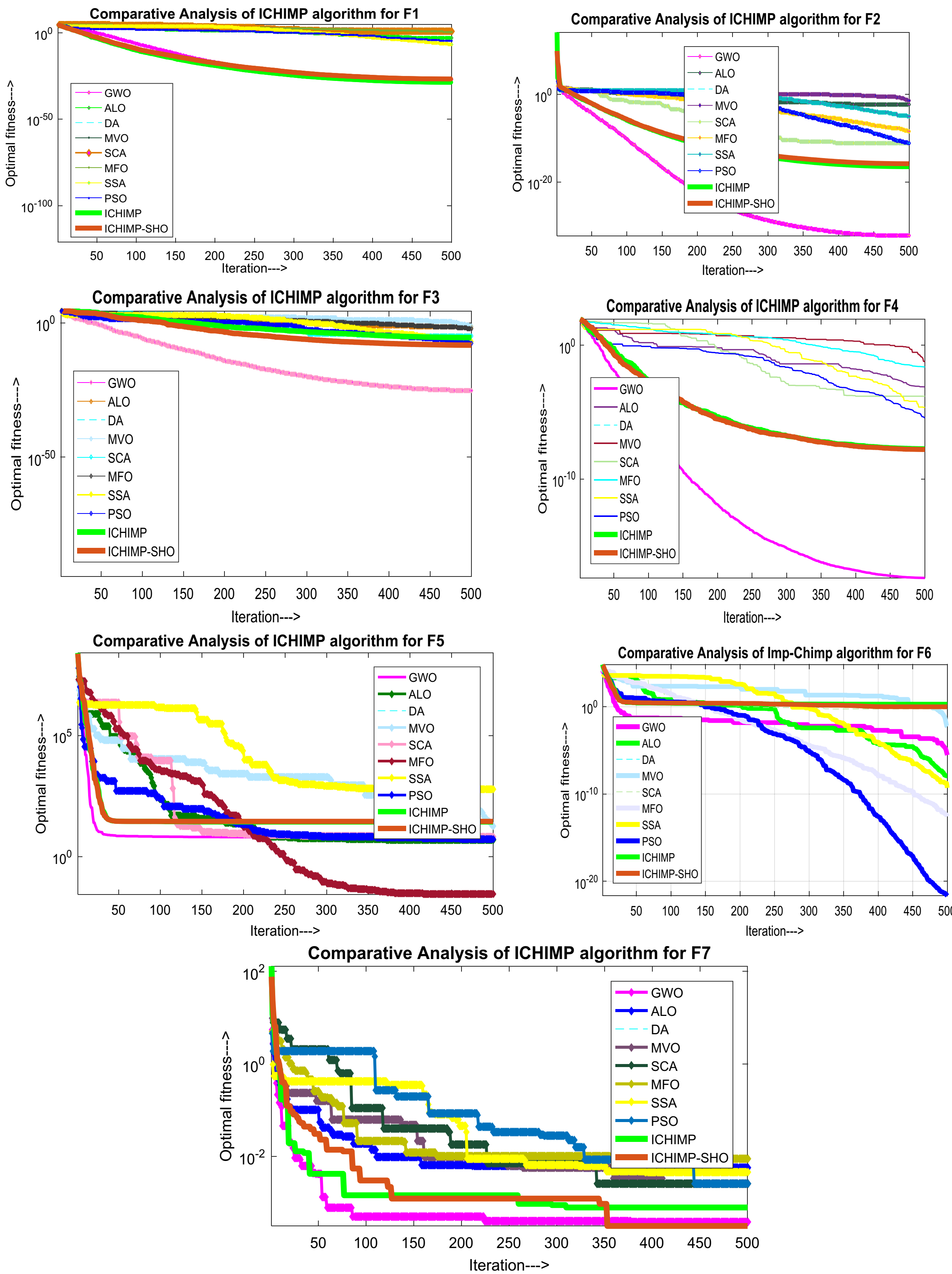


Fig. 5 Comparative curve of ICHIMP-SHO with GWO, DA, ALO, MVO, SSA, and PSO for UM standard bench mark functions

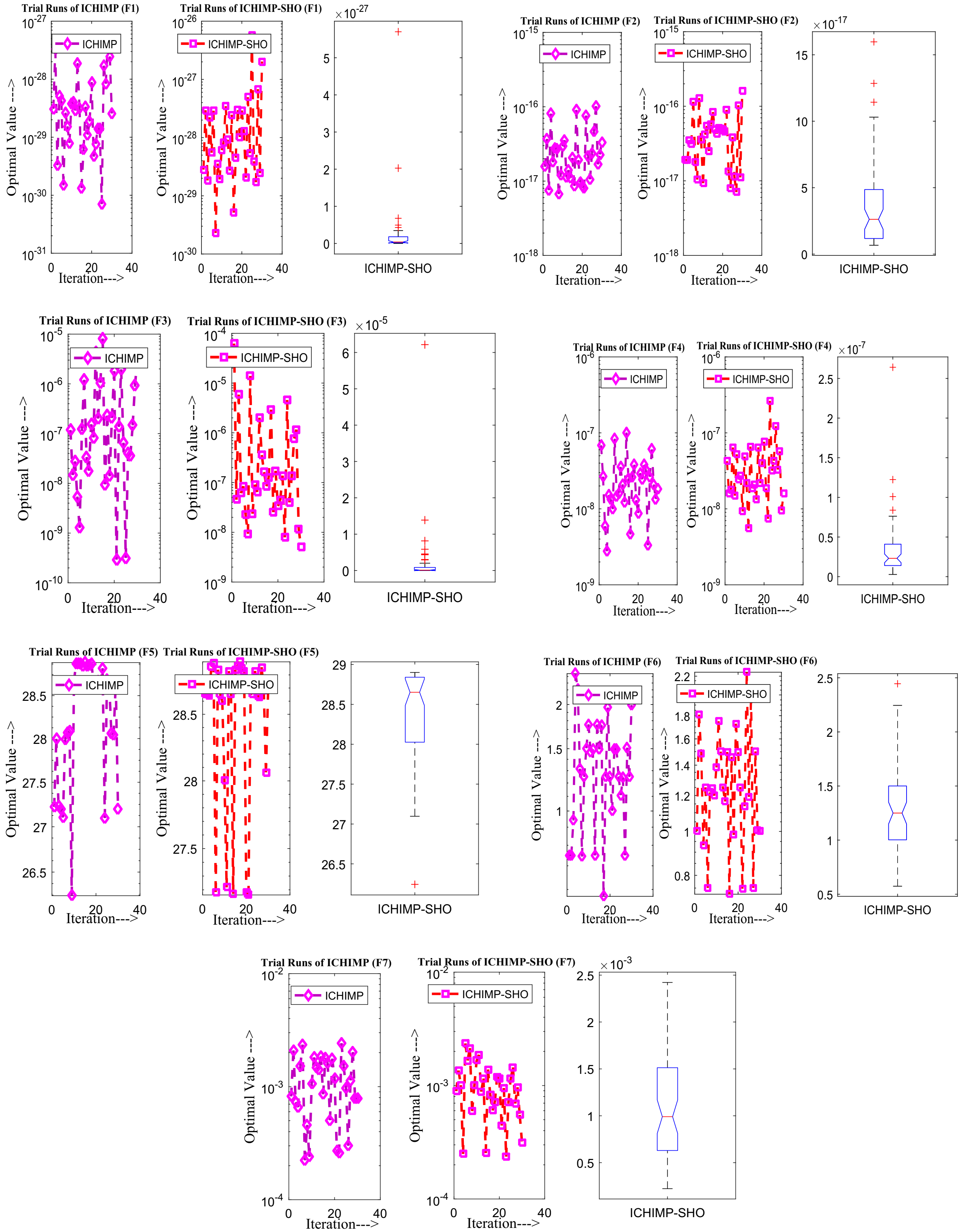


Fig. 6 Trial runs of ICHIMP and ICHIMP-SHO for UM standard bench mark functions

Table 9 Test results of multi-modal benchmark functions using ICHIMP-SHO algorithm

Function	Mean value	St. Deviation	Best fitness value	Worst fitness value	Median value	Wilcoxon Rank Sum Test	*t* Test	
						P value	*P* value	*h* value
F8	− 5231.965502	755.2916365	− 6835.710117	− 3547.406759	− 5099.515588	1.7344E−06	2.85762E−26	1
F9	7.95808E−14	5.29885E−14	0	2.27374E−13	5.68434E−14	2.89814E−06	4.53821E−09	1
F10	9.52719E−14	1.69917E−14	6.83897E−14	1.35891E−13	9.50351E−14	1.67736E−06	1.13726E−23	1
F11	0.001725278	0.004532246	0	0.015441836	0	0.125	0.045981511	1
F12	0.088180059	0.097309399	0.011803437	0.567716636	0.074592218	1.7344E−06	2.80855E−05	1
F13	1.911006715	0.317196258	1.325094518	2.527060925	1.866111264	1.7344E−06	1.49706E−24	1

Table 10 Execution time for multi-modal benchmark problems using ICHIMP-SHO algorithm

Function	Best time	Average time	Worst time
F8	1.359375	1.510416667	1.734375
F9	1.328125	1.479166667	1.703125
F10	1.34375	1.521875	1.8125
F11	1.40625	1.5203125	1.6875
F12	1.71875	1.8515625	2.015625
F13	1.65625	1.805729167	1.921875

8 Engineering-based optimization design problems

To validate the efficacy of the suggested ICHIMP-SHO algorithm, 11 types of engineering-based optimization designs are considered: pressure vessel, Speed reducer problem, Three-bar truss problem, welded beam, gear train design problem, belleville spring problem, cantilever beam design, rolling element bearing, (discrete variables), I-beam design, Multi-disk clutch break, and Tension/compression spring

Table 11 Comparison for multi-modal benchmark functions

Algorithm	Factors	(F8–F13) multi-modal benchmark functions					
		F8	F9	F10	F11	F12	F13
Lightning search algorithm (LSA) [62]	Avg	− 8001.3887	62.7618960	1.077446947	0.397887358	2.686199995	0.007241875
	St.Deviation	669.159310	14.9153021	0.337979509	1.68224E−16	0.910802774	0.006753356
Battle Royale Optimization algorithm (BRO) [133]	Mean	− 7035.2107	48.275350	0.350724	0.001373	0.369497	0.000004
	St.Deviation	712.33269	14.094585	0.688702	0.010796	0.601450	0.000020
Opposition based enhanced grey wolf optimization algorithm (OEGWO) [134]	Avg	-3.36×10^{3}	8.48×10^{-1}	9.41×10^{-15}	7.50×10^{-13}	9.36×10^{-02}	1.24E+00
	St.Deviation	3.53×10^{2}	4.65E+00	3.56×10^{-15}	4.11×10^{-12}	3.95×10^{-02}	2.09×10^{-1}
Photon search algorithm (PSA) [59]	Mean	11, 648.5512	7.3763	1.6766	0.5294	0.1716	1.5458
	St.Deviation	1230.4314	9.1989	0.9929	0.6102	0.2706	3.3136
Hybrid Harris Hawks Optimizer-Pattern Search algorithm (hHHO-PS) [74]	Avg	− 12, 332	00	8.88×10^{-6}	00	2.94×10^{-15}	1.16×10^{-13}
	St.Deviation	335.7988	0	0	0	3.52E−15	1.15E−13
Spotted Hyena Optimizer (SHO) [70]	Mean	$-1.16E\times10^{3}$	0.00E+00	2.48E+000	00	3.68×10^{-2}	9.29×10^{-1}
	St.Deviation	$2.72E\times10^{2}$	0.00E+00	1.41E+000	00	1.15×10^{-2}	9.52×10^{-2}
Harris Hawks Optimizer (HHO) [100]	Mean	− 12561.38	0	8.88×10^{-16}	0	8.92×10^{-6}	0.000101
	St.Deviation	40.82419	0	0	0	1.16×10^{-5}	0.000132
Enhanced Crow search algorithm (ECSA)[135]	Mean	− 2332.3867	0	8.88178E−16	0	0.11738407	0.444690657
	St.Deviation	223.93995	0	0	0	0.2849633	0.199081675
Transient Search Optimization (TSO) [136]	Avg	− 12, 569.5	00	8.88×10^{-16}	0	1.30×10^{-4}	7.55×10^{-4}
	St.Deviation	1.81×10^{-2}	00	0	0	1.67×10^{-4}	1.74×10^{-3}
ICHIMP− SHO (proposed algorithm)	Mean	− 5231.965502	7.95808E−14	9.52719E−14	0.001725278	0.088180059	1.911006715
	St.Deviation	755.2916365	5.29885E−14	1.69917E−14	0.004532246	0.097309399	0.317196258

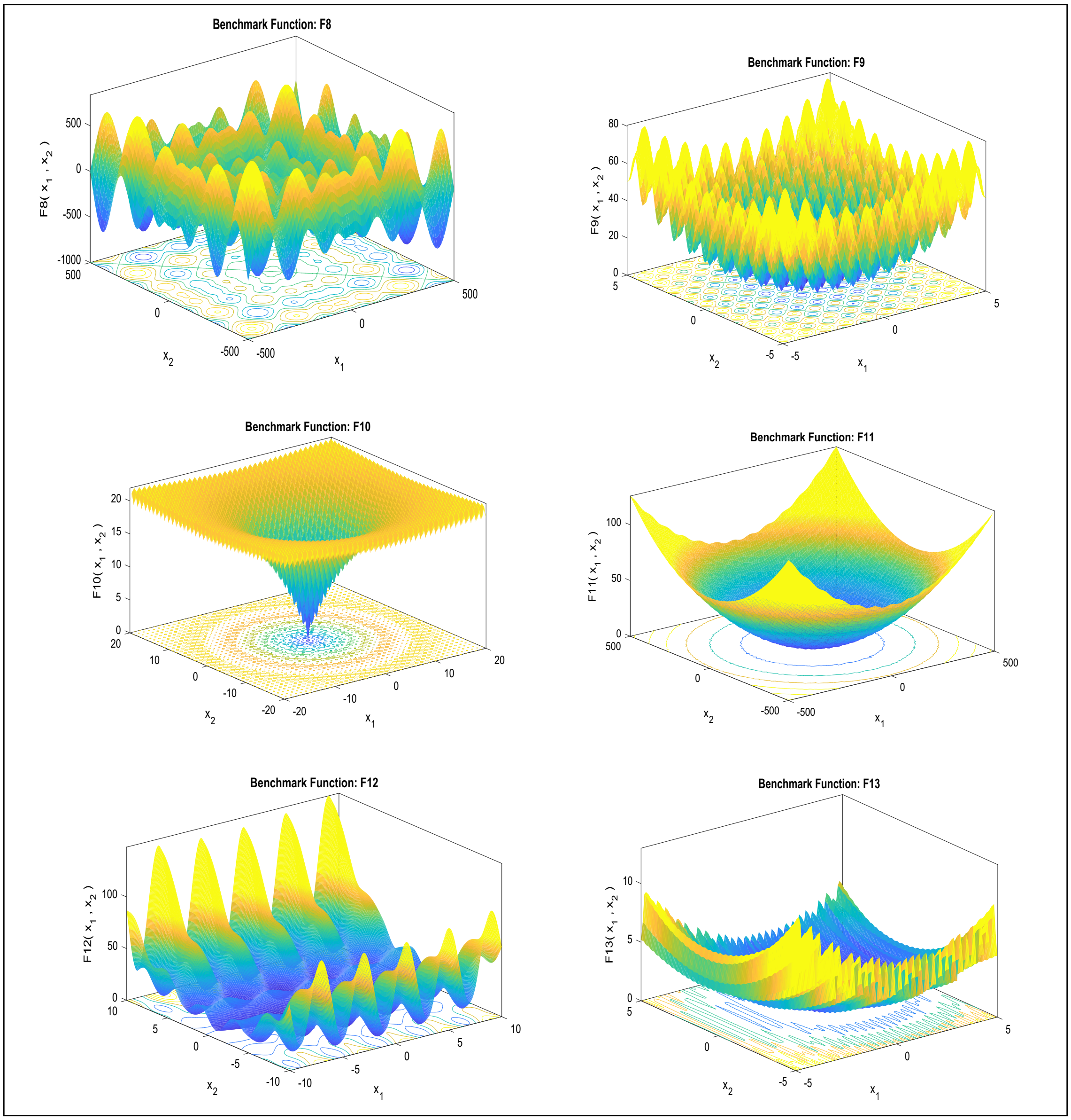


Fig. 7 3D view of multi-modal (MM) standard benchmark problem

design problem. The results for engineering-based optimization design issues were examined using several meta-heuristic optimizers, and convergence curves were compared to the standard CHIMP method, as shown in Fig. 24. Table 15 describes the engineering-based optimization design problems; Table 16 presents best values (Best fit), the average value (Ave), median value (Median), standard deviation (SD), and worst value (Worst fit); Table 17 shows Wilcoxon *P* value and *t* test values and the computation time of engineering-based optimization design problems is shown in Table 18.

8.1 Pressure vessel design

One of the multidisciplinary engineering optimization problems is depicted in Fig. 13, which is named Pressure Vessel design problem [137, 138]. The important aspect of this issue in engineering optimization design is to minimize or decrease the overall price, which includes material quality, welding, and the vessel's cylindrical form, as illustrated in Fig. 13. While, there are four types of factors utilized to create the pressure vessel issue (q1, q2, q3, and q4), such as shell thickness (T_s), head thickness (T_h), internal radius ®, and cylindrical unit length (L) which are taken into account. This vessel has end caps on either sides, and the structure's

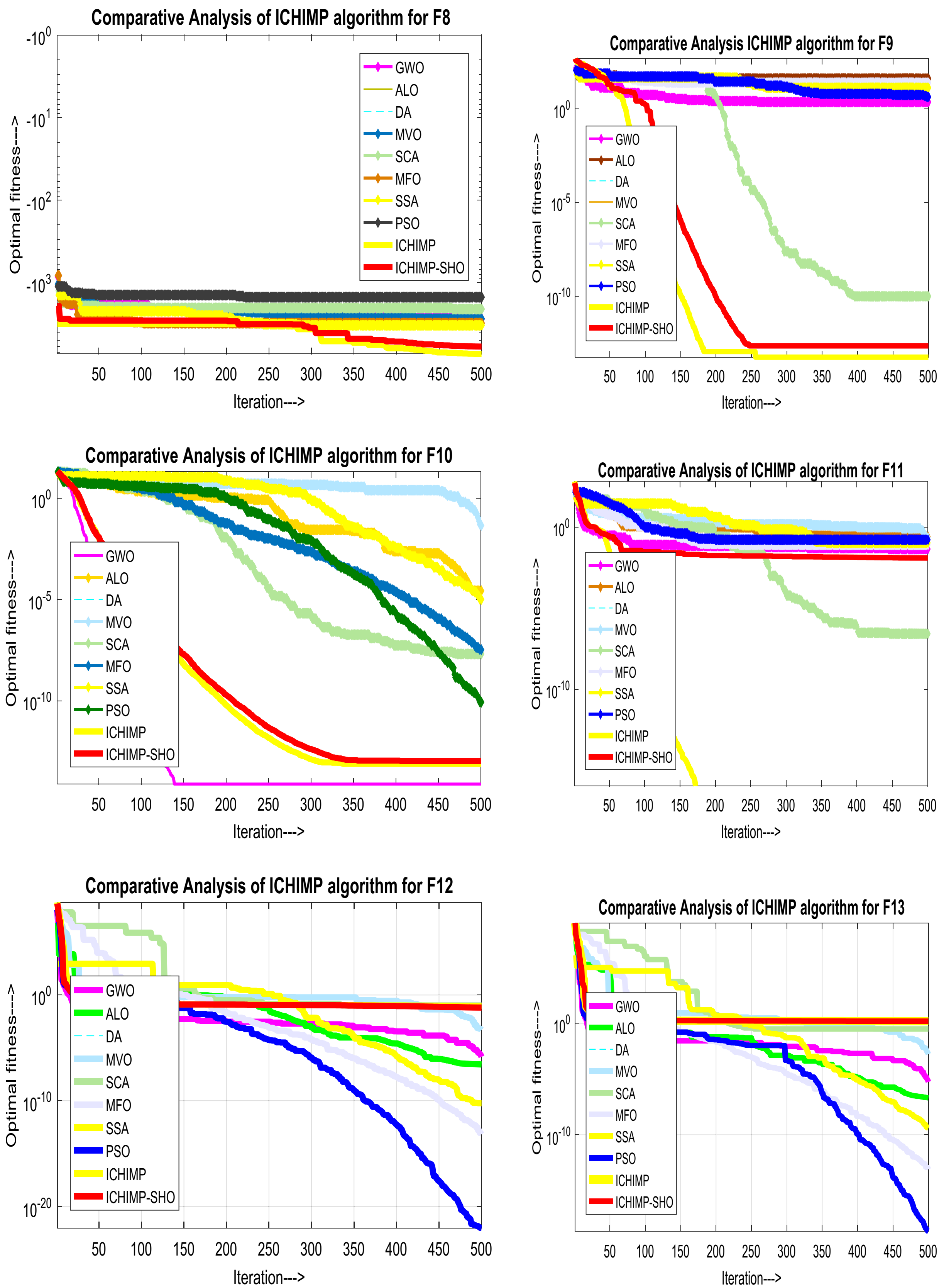


Fig. 8 Comparative curve of ICHIMP-SHO with GWO, DA, ALO, MVO, SSA, and PSO for MM standard bench mark functions

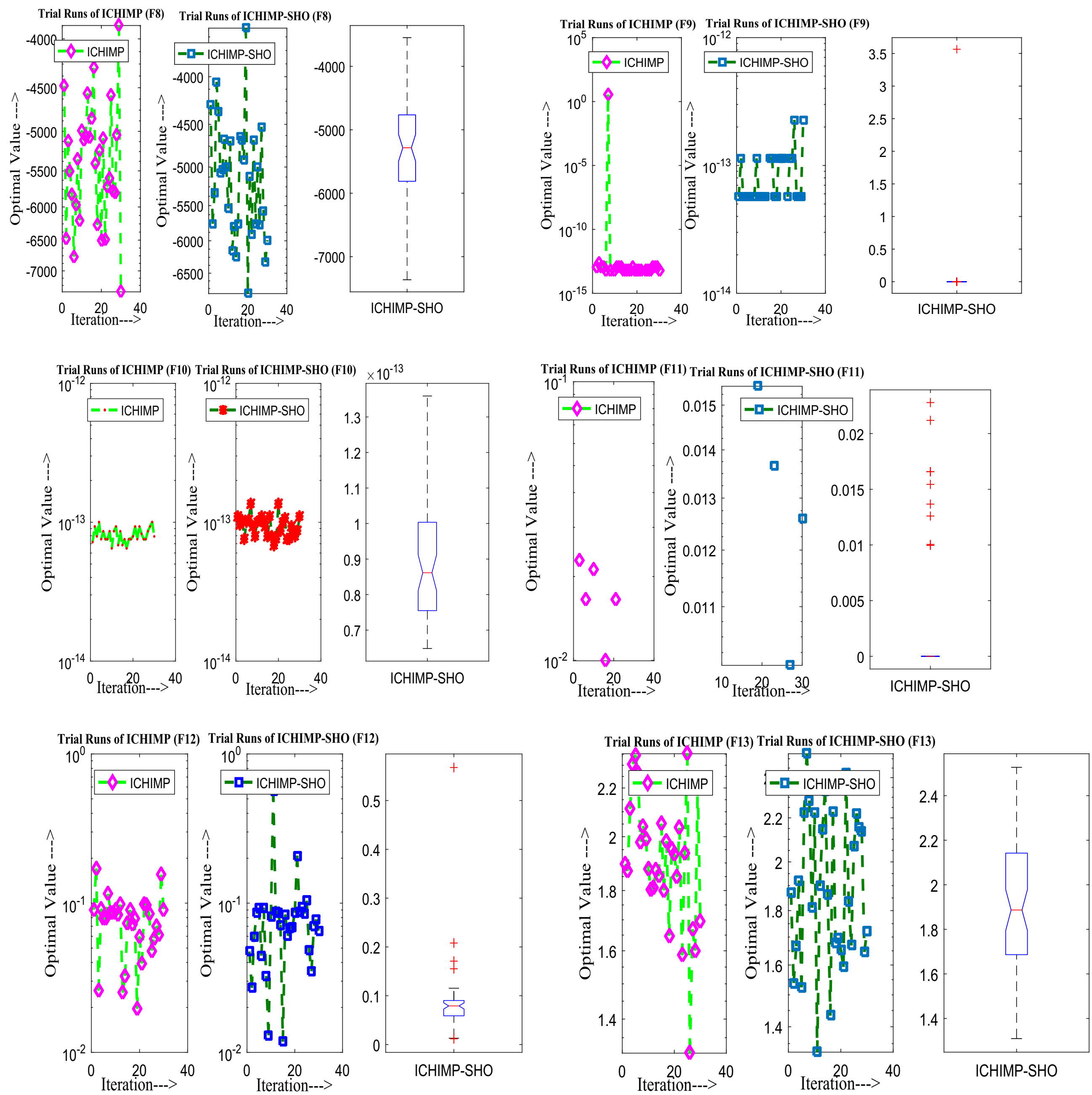


Fig. 9 Trial Runs of ICHIMP and ICHIMP-SHO for MM standard bench mark functions

head is hemispherical in form. The four types of constraints described above are the topic of a design problem, and the mathematical specification issue for the pressure vessel is represented in Eqs. (22)–(23d). Table 19 summarizes the conclusions of the analysis. The following are the results of ICHIMP-SHO compared with various algorithms.

We consider

$$\vec{q} = [q_1 q_2 q_3 q_4] = [T_s T_h R L_h]. \tag{22}$$

To minimize

$$f(\vec{q}) = 0.6224 q_1 q_3 q_4 + 1.7781 q_2 q_3^2 + 3.1661 q_1^2 q_4 + 19.84 q_1^2 q_3. \tag{23}$$

Here

$$g_1(\vec{q}) = -q_1 + 0.0193 q_3 \le 0 \tag{23a}$$

$$g_2(\vec{q}) = q_3 + 0.00954 q_3 \le 0 \tag{23b}$$

Table 12 Test observations for Fixed Dimensions Functions using ICHIMP-SHO algorithm

Function	Mean	STD	Best fitness	Worst fitness	Median	Wilcoxon Rank Sum Test	t Test	
						P value	P value	h Value
F14	5.923306745	4.529146785	0.998003838	12.67050581	2.982105157	1.7344E−06	2.85762E−26	1
F15	0.003199196	0.006885586	0.000307505	0.020678817	0.000509082	1.7344E−06	0.016514745	1
F16	− 1.031628421	2.91482E−08	− 1.031628453	− 1.031628341	− 1.031628427	1.7344E−06	1.0841E−220	1
F17	0.397889119	3.73497E−06	0.397887373	0.397907788	0.397888221	1.7344E−06	1.4346E−147	1
F18	3.000056878	7.82165E−05	3.000000224	3.000253679	3.000022152	1.7344E−06	1.053E−134	1
F19	− 3.861720787	0.002004746	− 3.862779317	− 3.855118521	− 3.862617975	1.7344E−06	4.9726E−97	1
F20	− 3.266961533	0.070877244	− 3.321992205	− 3.114124068	− 3.32196637	1.7344E−06	5.07294E−50	1
F21	− 9.054114924	2.269865564	− 10.15311737	− 2.630423301	− 10.15104629	1.7344E−06	1.47747E−19	1
F22	− 9.791774335	1.890396917	− 10.4026378	− 2.765188383	− 10.40030086	1.7344E−06	1.05403E−22	1
F23	− 10.1738343	1.371487294	− 10.53611947	− 5.128445026	− 10.53438125	1.7344E−06	4.06341E−27	1

Table 13 Execution time for fixed dimensions benchmark problems using ICHIMP-SHO algorithm

Function	Best time	Average time	Worst time
F14	1.359375	1.510416667	1.734375
F15	0.25	0.288541667	0.5
F16	0.140625	0.209375	0.359375
F17	0.140625	0.222395833	0.40625
F18	0.140625	0.195833333	0.40625
F19	0.265625	0.2953125	0.515625
F20	0.375	0.428125	0.625
F21	0.375	0.4484375	0.703125
F22	0.4375	0.509375	0.640625
F23	0.546875	0.609375	0.78125

$$q_3(\vec{p}) = -\pi q_3^2 q_4 - \frac{4}{3}\pi q_3^3 + 1296000 \le 0 \tag{23c}$$

$$g_4(\vec{q}) = q_4 - 240 \le 0; \tag{23d}$$

Variable range, $0 \le q_1 \le 99$

$0 \le q_2 \le 99$

$10 \le q_3 \le 200$

$10 \le q_4 \le 20.$

8.2 Speed reducer

As illustrated in Fig. 14 [110], this type of design issue has seven variables. It is made up of the face width × 1, the teeth module × 2, the pinion teeth number × 3, the first shaft length bearings × 4, the second shaft length bearings × 5, the 1st shaft diameter × 6, and the 2nd shaft diameter × 7. The weight of the velocity reducer must be reduced first which is the main aim of this issue. Figure 14 depicts the engineering design of a speed reducer. Table 20 summarizes the results of the analysis. GSA [61], hHHO-SCA [68], PSO [18], OBSCA, MFO [122], SCA, HS [31], and GA [10] are compared to the analytical findings of ICHIMP-SHO. Equations (24)–(24k) show the mathematical framework of the pressure vessel optimization design issue. The following is how the equations are written.

Minimizing

$$f(x) = 0.7854x_1x_2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2). \tag{24}$$

Subjected to

$$g_1(\vec{x}) = \frac{27}{x_1x_2^2x_3} - 1 \le 0 \tag{24a}$$

$$g_2(\vec{x}) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \le 0 \tag{24b}$$

$$g_3(\vec{x}) = \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \le 0 \tag{24c}$$

$$g_4(\vec{x}) = \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \le 0 \tag{24d}$$

Table 14 Comparison for fixed-dimension benchmark functions

Algorithm	Parameters	(F14–F23) fixed-dimension benchmark functions									
		F14	F15	F16	F17	F18	F19	F20	F21	F22	F23
Lightning search algorithm (LSA) [62]	Mean	0.358172550	0.024148546	0.000534843	− 1.031628453	3.000000000	− 3.862782148	− 3.272060061	− 7.027319823	− 7.136702131	− 7.910438367
	St. Deviation	0.743960008	0.047279168	0.000424113	0.000000000	3.34499E−15	0.000000000	0.059276470	3.156152099	3.514977671	3.596042666
Enhanced Crow search algorithm (ECSA) [135]	Mean	1.000269	0.000327	− 1.03161	0.397993	3.00003	− 3.86061	− 3.32066	− 10.1532	− 10.44028	− 10.5359
	St. Deviation	2.62E−03	1.24337E−05	2.20378E−05	1.16E−04	2.752E−05	4.53E−04	1.79E−03	8.75374E−05	1.611114E−04	4.62E−04
Transient Search Optimization (TSO) [136]	Mean	9.68E+000	9.01×10^{-4}	-1.06×10^{-1}	3.97×10^{-1}	3.00E+000	− 3.75E+000	− 3.01	− 10.1485	-10.3958	10.5267
	St. Deviation	3.29E+000	1.06×10–4	2.86×10^{-11}	2.46×10^{-1}	9.05E+000	$4.39E\times10^{-1}$	0.170990	$7.42\times10^{-1-3}$	1.43×10^{-2}	2.63×10^{-2}
Photon Search Algorithm (PSA) [59]	Mean	0.4802	0.0077	− 1.036	0.3979	3	− 3.8556	− 3.043	− 9.7302	− 9.8628	− 9.8189
	St. Deviation	0.1158	0.0224	2.33×10^{-7}	1.41×10^{-7}	1.36×10^{-5}	0.0153	0.1940	1.1347	1.2894	1.8027
Hybrid Harris Hawks Optimizer-Pattern Search algorithm (hHHO-PS) [74]	Mean	0.998004	0.000307	− 1.03163	0.397887	3	− 3.86278	− 3.322	− 10.1532	− 10.4029	− 10.5364
	St. Deviation	1.57×10^{-16}	1.65×10^{-13}	1.11×10^{-16}	00	2.63×10^{-15}	2.26×10^{-15}	4.35×10^{-15}	7.47×10^{-12}	7.74×10^{-15}	7.69×10^{-15}
Spotted Hyena Optimizer (SHO) [70]	Mean	1.130	2.70×10^{-3}	− 1.0316	0.398	3.000	− 3.89	− 1.44E+000	− 2.08E+000	1.61×10^{1}	− 1.68E+000
	St. Deviation	0.5659	5.43×10^{-3}	5.78×10^{-14}	1.26×10^{-14}	2.66×10^{-13}	1.13×10^{-11}	5.47×10^{-1}	3.80×10^{-1}	2.04×10^{-4}	2.64×10^{-1}

Table 14 (continued)

Algorithm	Parameters	(F14–F23) fixed-dimension benchmark functions									
		F14	F15	F16	F17	F18	F19	F20	F21	F22	F23
Harris Hawks Optimizer (HHO) [100]	Mean	1.361171	0.00035	− 1.03163	0.397895	3.000001225	− 3.8597664	− 3.06481	− 5.37397	− 5.08346	− 5.78398
	St. Deviation	0.95204	3.20×10^{-5}	1.86×10^{-9}	1.60×10^{-5}	4.94×10^{-6}	0.00519467	0.136148	1.227502	0.004672	1.712458
ICHIMP-SHO (proposed algorithm)	Mean	5.923306745	0.003199196	− 1.031628421	0.397889119	3.000056878	− 3.861720787	− 3.266961533	− 9.054114924	− 9.791774335	− 10.1738343
	St. Deviation	4.529146785	0.006885586	2.91482E−08	3.73497E−06	7.82165E−05	0.002004746	0.070877244	2.269865564	1.890396917	1.371487294

$$g_5(\vec{x}) = \frac{1}{110x_6^3}\sqrt{(\frac{745.0x_4}{x_2x_3})^2 + 16.9\times10^6} - 1 \le 0 \tag{24e}$$

$$g_6(\vec{x}) = \frac{1}{85x_7^3}\sqrt{(\frac{745.0x_5}{x_2x_3})^2 + 157.5\times10^6} - 1 \le 0 \tag{24f}$$

$$g_7(\vec{x}) = \frac{x_2x_3}{40} - 1 \le 0 \tag{24g}$$

$$g_8(\vec{x}) = \frac{5x_2}{x_1} - 1 \le 0 \tag{24h}$$

$$g_9(\vec{x}) = \frac{x_1}{12x_2} - 1 \le 0 \tag{24i}$$

$$g_{10}(\vec{x}) = \frac{1.5x_6 + 1.9}{12x_2} - 1 \le 0 \tag{24j}$$

$$g_{11}(\vec{x}) = \frac{1.1x_7 + 1.9}{x_5} - 1 \le 0. \tag{24k}$$

Here

$2.6 \le x_1 \le 3.6, 0.7 \le x_2 \le 0.8, 17 \le x_3 \le 28,$
$7.3 \le x_4 \le 8.3, 7.8 \le x_5 \le 8.3,$
$2.9 \le x_6 \le 3.9 and 5 \le x_7 \le 5.5.$

8.3 Three-bar truss engineering design problem

To test the suggested ICHIMP-SHO algorithm output, this engineering design is considered which is figured in Fig. 15. The idea is to reduce the fitness value of the weight. It is imbibed with three constraints, namely, deflection constraint, buckling constraint, and stress constraint. Equations (25–26c) expose the three-bar truss problem numerically and its comparison results are tabulated in Table 21.

$$\text{Consider } \vec{x} = [x_1, x_2] = [A_1, A_2] \tag{25}$$

$$\text{Minimize } f(\vec{x}) = (2\sqrt{2}x_1 + x_2) * l \tag{26}$$

$$\text{Subject to } g_1(\vec{x}) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \le 0 \tag{26a}$$

$$g_2(\vec{x}) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \le 0 \tag{26b}$$

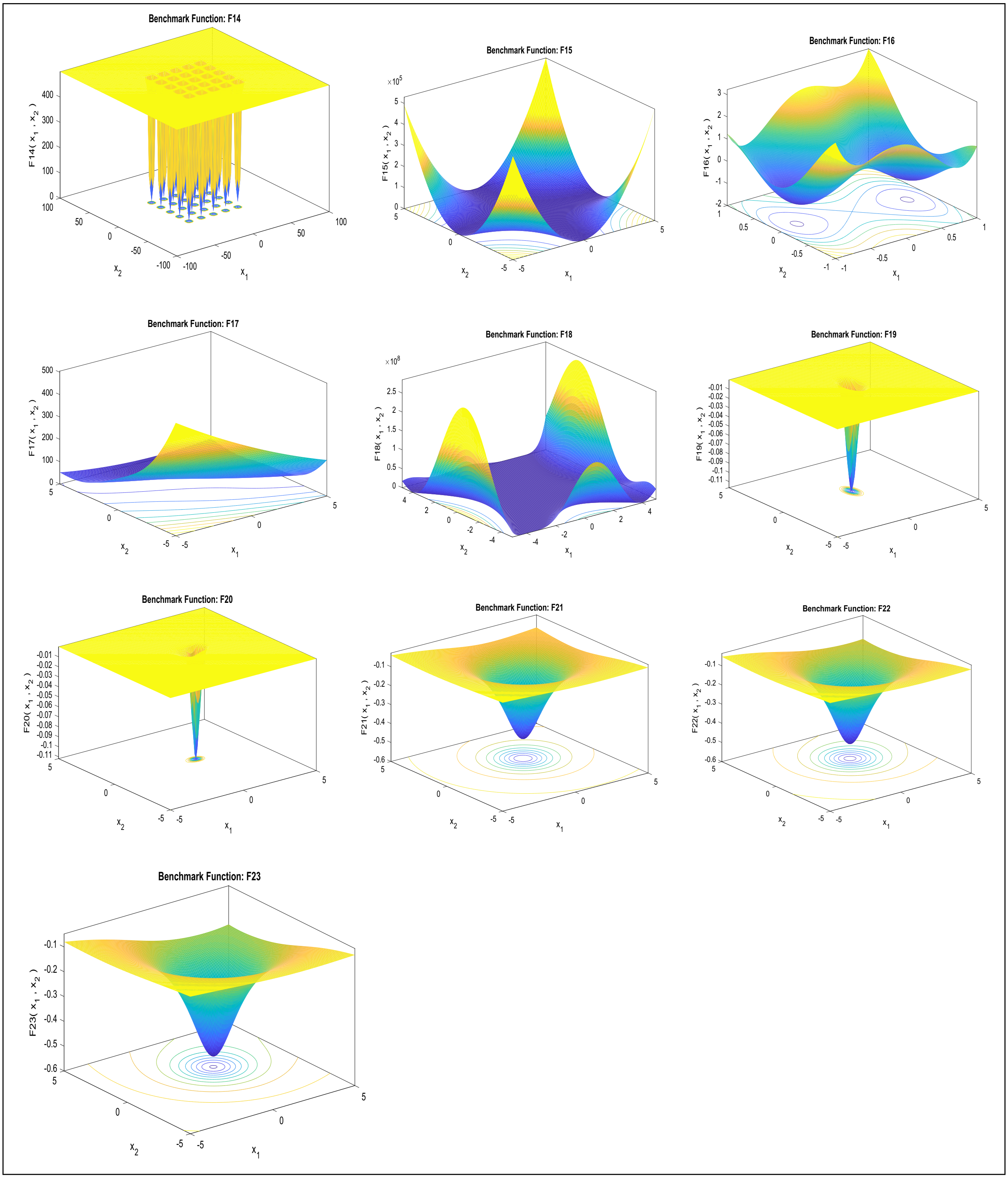


Fig. 10 3D view of fixed-dimension (FD) modal standard benchmark functions

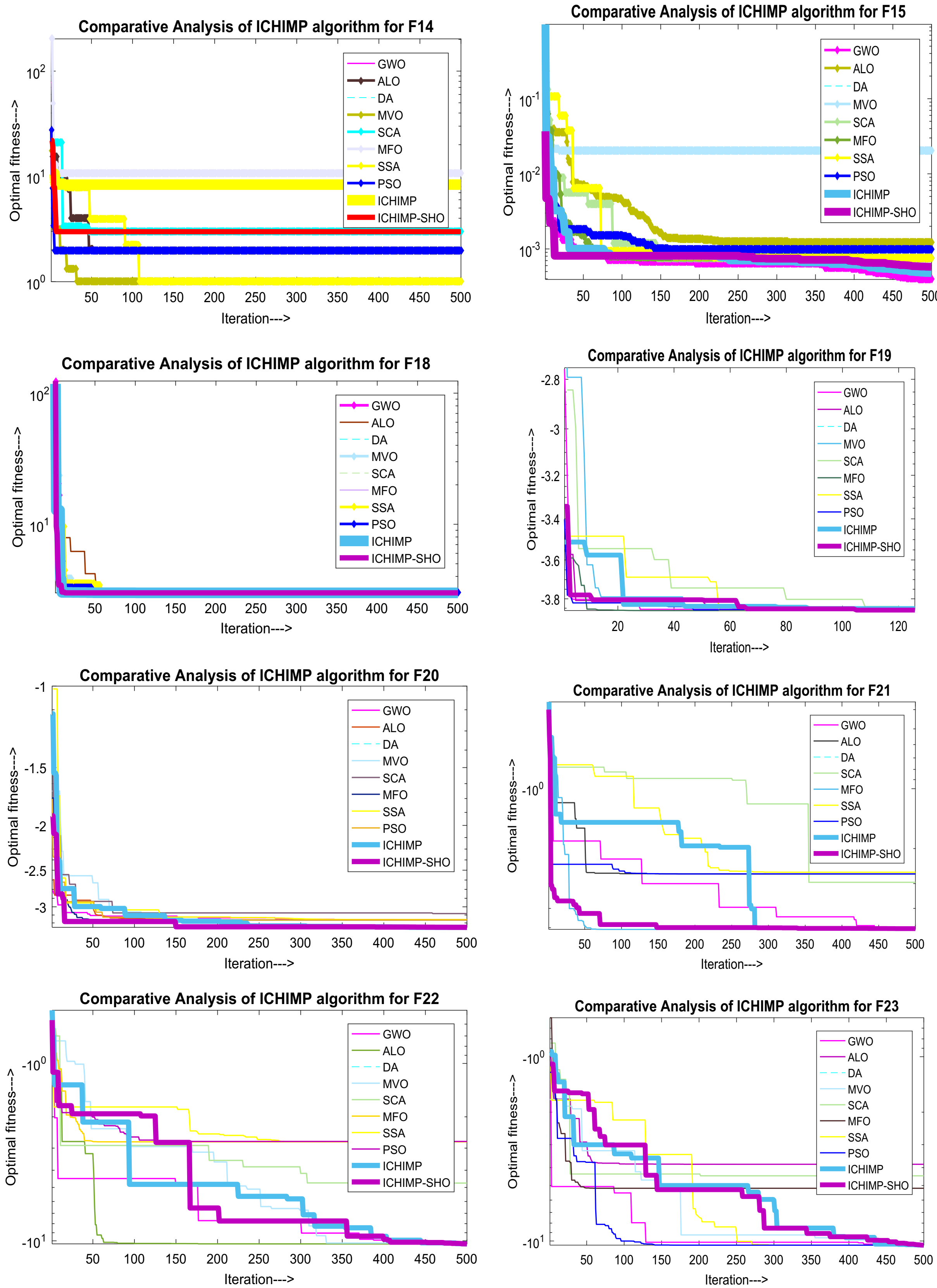


Fig. 11 Comparative curve of ICHIMP-SHO with GWO, DA, ALO, MVO, SSA, and PSO for fixed standard

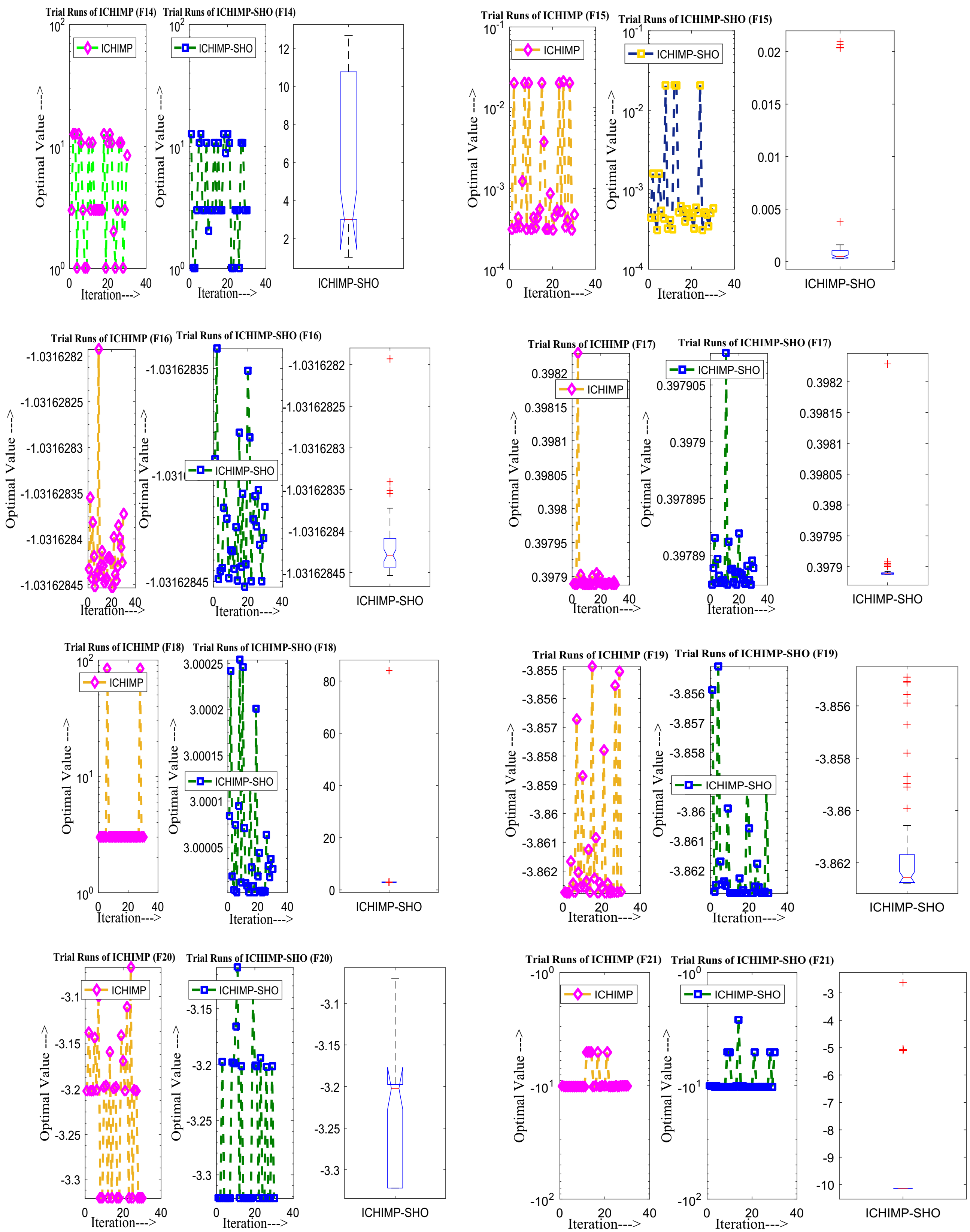


Fig. 12 Trial Runs of ICHIMP and ICHIMP-SHO for fixed-dimension standard bench mark functions

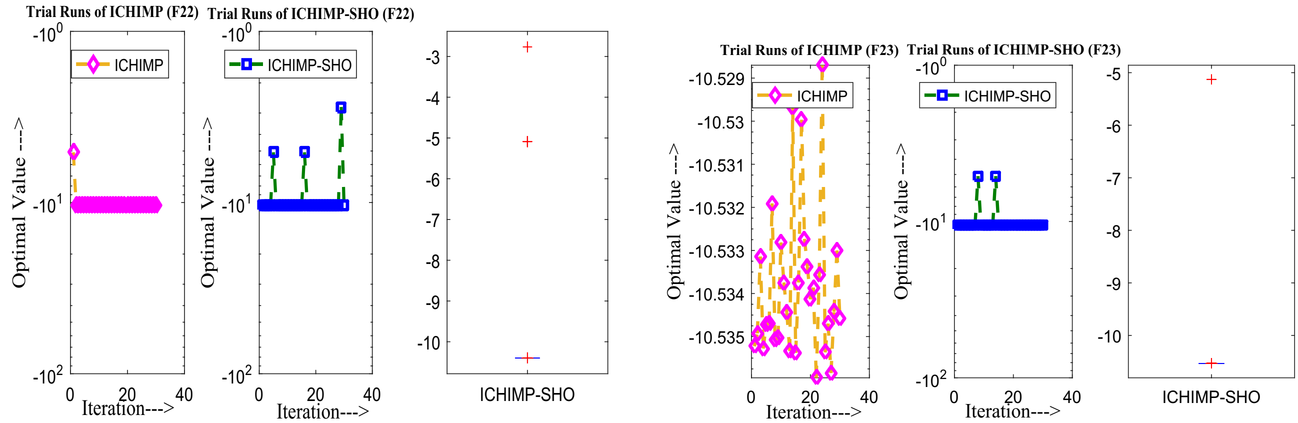


Fig. 12 (continued)

$$g_3(\vec{x}) = \frac{1}{\sqrt{2}x_2 + x_1}P - \sigma \le 0. \tag{26c}$$

Variable range $0 \le x_1, x_2 \le 1$.
Here, l = 100 cm, P = 2KN/cm^2, and σ = 2KN/cm^2.

8.4 Welded beam

In Fig. 16 [110, 111], this problem is depicted. The main focus is on lowering the welded beam's manufacturing costs: (i) bar height (h), (ii) weld thickness (h), (iii) bar length (l), and (iv) bar thickness (b) are the four variables which are all constrained by things like Buckling bar (Pc), End beam deflection (d), Side restrictions and shear stress (s), and Bending beam stress (h). The welded beam optimization design equations are presented in Eqs. (27)–(29f). In Table 22, the results of ICHIMP-SHO are compared to those of hHHO-SCA [68] and other algorithms.

Let us consider

$$\vec{z} = [z_1 z_2 z_3 z_4] = [hltb] \tag{27}$$

$$f(\vec{z}) = 1.10471 z_1^2 z_2 + 0.04811 z_3 z_4 (14.0 + z_2). \tag{28}$$

By addressing

$$g_1(\vec{z}) = \tau(\vec{z}) - \tau_{maxi} \le 0, \tag{28a}$$

$$g_2(\vec{z}) = \sigma(\vec{z}) - \sigma_{maxi} \le 0 \tag{28b}$$

$$g_3(\vec{z}) = \delta(\vec{z}) - \delta_{maxi} \le 0 \tag{28c}$$

$$g_4(\vec{z}) = z_1 - z_4 \le 0 \tag{28d}$$

$$g_5(\vec{z}) = P_i - P_c(\vec{z}) \le 0 \tag{28e}$$

$$g_6(\vec{z}) = 0.125 - z_1 \le 0 \tag{28f}$$

$$g_7(\vec{z}) = 1.10471 z_1^2 + 0.04811 z_3 z_4 (14.0 + z_2) - 5.0 \le 0. \tag{28g}$$

Range of variables: $0.1 \le z_1 \le 2, 0.1 \le z_2 \le 10, 0.1 \le z_3 \le 10, 0.1 \le z_4 \le 2$.
Here

$$\tau(\vec{z}) = \sqrt{(\tau/)^2 + 2\tau/\tau//\frac{z_2}{2R} + (\tau//)^2}, \tag{29a}$$

$$\tau/ = \frac{P_i}{\sqrt{2}z_1 z_2}, \tau// = \frac{MR}{J}, M = P_i\left(L + \frac{z_2}{2}\right), \tag{29b}$$

$$R = \sqrt{\frac{z_2^2}{4} + \left(\frac{z_1 + z_3}{2}\right)^2} \tag{29c}$$

$$J = 2\left\{\sqrt{2}z_1 z_2\left[\frac{z_2^2}{4} + \left(\frac{z_1 + z_3}{2}\right)^2\right]\right\} \tag{29d}$$

$$\sigma(\vec{y}) = \frac{6P_i L}{z_4 z_3^2}, \delta(\vec{y}) = \frac{6P_i L^3}{E z_2^2 z_4} \tag{29e}$$

$$P_c(\vec{z}) = \frac{4.013E\frac{\sqrt{z_3^2 z_4^6}}{36}}{L^2}\left(1 - \frac{z_3}{2L}\sqrt{\frac{E}{4G}}\right) \tag{29f}$$

Table 15 Basic information of (SPECIAL1—SPECIAL11) engineering-based designs

Engineering functions and their description	Name of the function	Type of objective	No. of discrete variables	Count of constraint
Special 1	Pressure Vessel	Minimize cost	4	4
Special 2	Speed reducer	Minimize weight	7	11
Special 3	Three-wbar truss	Minimize weight	–	3
Special 4	Welded Beam	Minimize cost	4	7
Special 5	Gear train	Minimize gear ratio	4	1
Special 6	Belleville spring	Minimize weight	–	5
Special 7	Cantilever Beam	Minimize weight	5	1
Special 8	Rolling element bearing	Maximize dynamic load	10	9
Special 9	I beam	Minimize vertical deflection	4	4
Special 10	Spring design	Minimize weight	3	4
Special 11	Multiple disk clutch brake	Minimize weight	5	8

$L = 14in, \delta_{\max i} = 0.25in, E = 30 \times 1^6 psi, G = 12 \times 10^6 psi,$
$\tau_{\text{maxi}} = 13600psi, \sigma_{\text{maxi}} = 3000psi, P = 6000lb.$

8.5 Gear train design

Another form of engineering-based design optimization issue is the Gear Train Design problem, which includes four parameter categories, as shown in Fig. 17 [110]. The general objective of the architectural design is to minimize the scalar value of the gears and the teeth ratio. As a result, the teeth of each gear are considered in the decision variable. For the comparative study of ICHIMP-SHO, the analytical data are given in Table 23. The model for the relevant formulae is as follows:

Let us consider

$$\vec{Ge} = \left[Ge_1 Ge_2 Ge_3 Ge_4\right] = \left[M_A M_B M_C M_D\right]. \tag{30}$$

To minimize

$$f(\overrightarrow{Ge}) = \left(\frac{1}{6.931} - \frac{Ge_3 Ge_4}{Ge_1 Ge_4}\right)^2; \tag{30a}$$

subjected to

$$12 \le Ge_1, Ge_2, Ge_3, Ge_4 \le 60. \tag{30b}$$

8.6 Belleville spring

This issue is depicted in Fig. 18. This is a technique used to reduce the problem by selecting a parameter that exists already in the constraints to the designed variable ratios. Belleville spring is designed with minimum weight in such a way to suit many designed variables, such as spring height (S_H), external part diameter (DIM_E), internal part diameter (DIM_I), and Belleville spring (S_T) thickness. Table 24 presents the comparison results. The constraints when subjected will be affected in deflection, deflection height, the internal and external portion of diameter, compressive types of stresses, and slope. The below equations are the mathematical expressions

$$\text{Minimizing}; f(w) = 0.07075\pi(DIM_E^2 - DIM_I^2)t; \tag{31}$$

$$\text{subjected to; } b_1(w) = G - \frac{4P\lambda_{\max}}{(1-\delta^2)\alpha DIM_E}\left[\delta(S_H - \frac{\lambda_{\max}}{2}) + \mu t\right] \ge 0 \tag{32}$$

$$b_2(w) = (\frac{4P\lambda_{\max}}{(1-\delta^2)\alpha DIM_E}\left[(S_H - \frac{\lambda}{2})(S_H - \lambda)t + t^3\right])\lambda_{\max} - P_{MAX} \ge 0 \tag{32a}$$

$$b_3(w) = \lambda_1 - \lambda_{\max} \ge 0 \tag{32b}$$

Table 16 ICHIMP-SHO results for engineering design issues

Name of design	Mean	Standard deviation	Best	Worst	Median
Pressure vessel	6060.428	268.201	5908.0551	6990.7201	5963.8967
Speed reducer problem	3012.077	4.325343	3003.6315	3020.8707	3012.2509
Three−bar truss problem	263.9036	0.006451	263.89701	263.92124	263.90085
Welded beam	1.729785	0.002854	1.726576	1.7404808	1.7291861
Gear train	3.91E−12	6.49E−12	6.38E−16	2.49E−11	1.10E−12
Belleville spring	1.995626	0.009945	1.9817655	2.0314199	1.9928986
Cantilever beam design	1.303427	0.000113	1.3032958	1.3038244	1.3034114
Rolling element bearing	− 85150.7	88.71365	− 85383.949	− 84905.165	− 85152.953
I-beam design	0.006626	3.62E−08	0.006626	0.0066261	0.006626
Spring design	0.012801	0.000117	0.0126915	0.0131369	0.0127433
Multiple disk clutch brake (discrete variables)	0.39118	0.00101	0.3900536	0.3946036	0.3910156

Table 17 Parametric results using proposed ICHIMP-SHO Algorithm

Name of design	*P* value	*t* Value	*h* Value
Pressure vessel	1.73E−06	4.73E−41	1
Speed reducer problem	1.73E−06	3.24E−84	1
Three-bar truss problem	1.73E−06	1.63E−135	1
Welded beam	1.73E−06	1.83E−82	1
Gear train	1.73E−06	0.002571225	0
Belleville spring	1.73E−06	1.52E−68	1
Cantilever beam design	1.73E−06	1.35E−119	1
Rolling element bearing	1.73E−06	2.95E−88	1
I-beam design	1.73E−06	2.27E−154	1
Spring design	1.73E−06	5.91E−61	1
Multiple disk clutch brake (Discrete variables)	1.73E−06	7.92E−77	1

Table 18 Results of computational time using proposed ICHIMP-SHO algorithm

Name of design	Best time	Mean time	Worst time
Pressure vessel	0.15625	0.284375	0.375
Speed reducer problem	0.390625	0.475520833	0.65625
Three-bar truss problem	0.15625	0.199479167	0.328125
Welded beam	0.234375	0.310416667	0.484375
Gear train	0.203125	0.270833333	0.390625
Belleville spring	0.25	0.302083333	0.421875
Cantilever beam design	0.28125	0.338020833	0.453125
Rolling element bearing	0.515625	0.6328125	0.875
I-beam design	0.21875	0.283333333	0.390625
Spring design	0.1875	0.252083333	0.40625
Multiple disk clutch brake (discrete variables)	0.296875	0.365625	0.5625

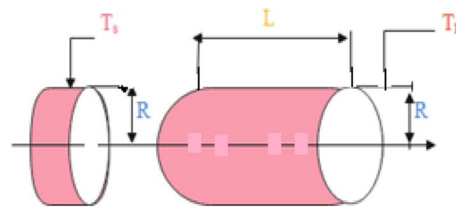


Fig. 13 Design of pressure vessel

$$b_4(w) = H - S_H - t \geq 0 \tag{32c}$$

$$b_5(w) = DIM_{MAX} - DIM_E \geq 0 \tag{32d}$$

$$b_6(w) = DIM_E - DIM_I \geq 0 \tag{32e}$$

$$b_7(w) = 0.3 - \frac{S_H}{DIM_E - DIM_I} \geq 0, \tag{32f}$$

where

$$\alpha = \frac{6}{\pi \ln J}\left(\frac{J-1}{\ln J} - 1\right)^2$$

$$\delta = \frac{6}{\pi \ln J}\left(\frac{J-1}{\ln J} - 1\right)$$

$$\mu = \frac{6}{\pi \ln J}\left(\frac{J-1}{2}\right)$$

$P_{MAX} = 5400\ lb.$
$P = 30e6$ psi, $\lambda_{max} = 0.2$ in, $\delta = 0.3$, $G = 200$ Kpsi,
$H = 2$ in, $DIM_{MAX} = 12.01$ in, $J = \frac{DIM_E}{DIM_I}$, $\lambda_1 = f(a)a$, $a = \frac{S_H}{t}$.

Table 19 Comparative observations of ICHIMP-SHO for pressure vessel optimisation design issue with other algorithms

Comparative algorithms	Optimal values for variables				Optimum cost
	T_s	T_h	R	L	
Proposed ICHIMP-SHO	0.781785	0.390768	40.48638	197.7438	5908.0551
hHHO-SCA [75]	0.945909	0.447138	46.8513	125.4684	6393.092794
BCMO [139]	0.7789243362	0.3850096372	40.3556904385	199.5028780967	6059.714
SMA [89]	0.7931	0.3932	40.6711	196.2178	5994.1857
ACO [140]	0.8125	0.4375	42.1036	176.5727	6059.0888
GWO [19]	0.8125	0.4345	42.0892	176.7587	6051.564
AIS-GA [141]	0.8125	0.4375	42.098411	176.67972	6060.138
GSA [106]	1.125	0.625	55.9887	84.4542	8538.84
DELC [142]	0.8125	0.4375	42.0984455	176.636595	6059.7143
SiC-PSO [143]	0.8125	0.4375	42.098446	176.636596	6059.714335
G-QPSO [144]	0.8125	0.4375	42.0984	176.6372	6059.7208
NPGA [145]	0.8125	0.437500	42.097398	176.654047	6059.946341
CDE [146]	0.8125	0.437500	42.098411	176.637690	6059.7340
HHO [100]	0.8125	0.4375	42.098445	176.636596	6000.46259
CLPSO [147]	0.8125	0.4375	42.0984	176.6366	6059.7143
GeneAs [148]	0.9375	0.5000	48.3290	112.6790	6410.3811
MFO [37]	0.8125	0.4375	42.0981	176.641	6059.7143
ACO	0.8125	0.4375	42.1036	176.5727	6059.089
MVO [63]	0.8125	0.4375	42.0907382	176.738690	6060.8066
SCA	0.817577	0.417932	41.74939	183.57270	6137.3724
HS [35]	1.099523	0.906579	44.456397	176.65887	6550.0230
Lagrangian multiplier	1.125	0.625	58.291	43.69	7198.043
Branch-bound	1.125	0.625	47.7	117.701	8129.1
ChOA [76]	1.043	0.548	53.236	77.330	6.854

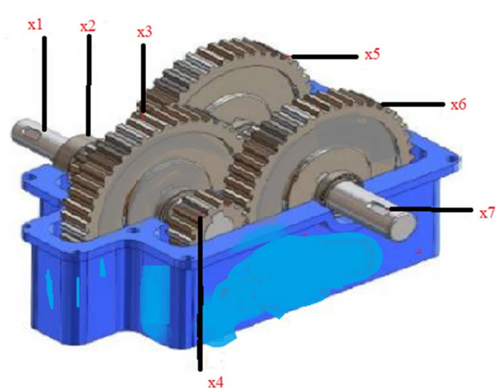


Fig. 14 Speed reducer design of engineering problem

8.7 Cantilever beam design

As shown in Fig. 19, the goal of this civil-based engineering problem is to reduce beam weight. This is made up of five different sorts of shapes [111]. The final goal is to minimize the weight of the beam, as illustrated in Fig. 19. It is also granted upon by any single variable, and the entire design configuration comprises structural characteristics of five types, with the beam thickness being kept constant. To avoid infringing on Eqs. (33)–(34) for the design of the final optimum solution, the location of the vertical constraint should be calculated throughout the design procedure confront. Table 25 compares the results to those of other techniques. ICHIMP-SHO observations fared better than other algorithms. The following is the design formula:

Let us consider $\vec{L} = [L_1 L_2 L_3 L_4]$

$$f(\vec{L}) = 0.6224(L_1 + L_2 + L_3 + L_4 + L_5). \tag{33}$$

By addressing

$$g(\vec{L}) = \frac{61}{L_1^3} + \frac{37}{L_2^3} + \frac{19}{L_3^3} + \frac{7}{L_4^3} + \frac{1}{L_5^3} \le 1. \tag{34}$$

Ranges of variables are $0.01 \le L_1, L_2, L_3, L_4, L_5 \le 100$.

Table 20 Comparative results of ICHIMP-SHO for speed reducer optimisation design issue with other algorithms

Comparative algorithms	Optimal values for variables							Optimum fitness
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
Proposed ICHIMP-SHO	3.506012	0.7	17	7.470045	7.89015	3.350829	5.288704	3003.6315
GSA [106]	3.600000	0.7	17	8.3	7.8	3.369658	5.289224	3051.120
hHHO-SCA [75]	3.506119	0.7	17	7.3	7.99141	3.452569	5.286749	3029.873076
PSO [149]	3.500019	0.7	17	8.3	7.8	3.352412	5.286715	3005.763
OBSCA	3.0879	0.7550	26.4738	7.3650	7.9577	3.4950	5.2312	3056.3122
MFO [37]	3.507524	0.7	17	7.302397	7.802364	3.323541	5.287524	3009.571
SCA	3.508755	0.7	17	7.3	7.8	3.461020	5.289213	3030.563
HS [35]	3.520124	0.7	17	8.37	7.8	3.366970	5.288719	3029.002
GA [30]	3.510253	0.7	17	8.35	7.8	3.362201	5.287723	3067.561

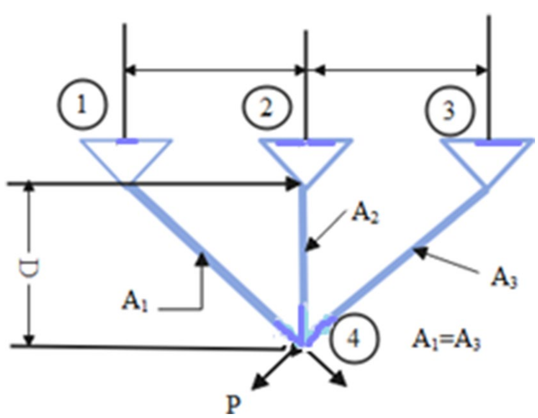


Fig. 15 Three-bar truss engineering design issue

8.8 Rolling element bearing

The main aim of this design issue is to improve the rolling part's dynamic bearing ability, as shown in Fig. 20 [110, 147]. This problem in engineering design has ten choice variable numbers: (i) pitch diameter (DIM_P), (ii) ball diameter (DIM_B), (iii) ball numbers (N_b), (iv) outer raceway curvature coefficient, and (v) inner raceway curvature coefficient. The following five variables (KD_{min}, KD_{max}, Ɛ, e, and f), which are only evaluated for discrete integers, have an impact on the interior section of the geometry. On kinematic circumstances and specifications, a total of nine nonlinear restrictions are challenged. Table 26 compares the results of ICHIMP-SHO with other known methods for the rolling

Table 21 Comparative observations of ICHIMP-SHO for three-bar truss optimisation design issue with other algorithms

Comparative algorithms	Optimal values for variables		Optimum weight
	X_1	X_2	
Proposed ICHIMP-SHO	0.788595	0.408486	263.89701
Hernandez	0.788	0.408	263.9
Ray and Saini [150]	0.795	0.398	264.3
Gandomi [151]	0.78867	0.40902	263.9716
EEGWO [76]	0.790761722154339	0.402632303723429	2.63924420788787771E+02
GWO-SA [152]	0.789	0.408	263.896
WDE [76]	0.515535107819326	0.0156341500434795	2.639297829829848E+02
ALO	0.789	0.408	263.896
CS [153]	0.789	0.409	263.972
hHHO-SCA [75]	0.788498	0.40875	263.8958665
DEDS [154]	0.789	0.408	263.896
CSA [155]	0.788638976	0.408350573	263.895844337
MBA [151]	0.789	0.409	263.896
Ray and Liew [156]	0.788621037	0.408401334	263.8958466
Raj et al.	0.789764410	0.405176050	263.89671

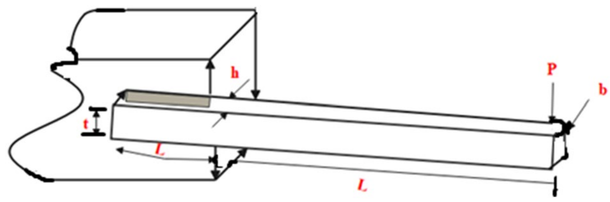


Fig. 16 Welded mechanical beam model

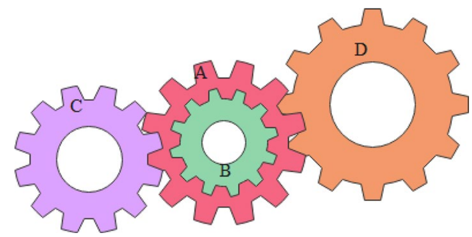


Fig. 17 Design of gear train optimization design

bearing design problem. From Eqs. (35a) through (35c), the mathematical formulation for the tendered engineering design is shown.

For maximizing

$$C_D = f_c N^{2/3} \mathrm{DIM}_B^{1.8}. \tag{35a}$$

If DIM ≤ 25.4 mm

$$C_D = 3.647 f_C N^{2/3} \mathrm{DIM}_B^{1.4}. \tag{35b}$$

If DIM ≥ 25.4 mm.

Addressing

$$r_1(y) = \frac{\theta_0}{2\sin^{-1}\left(\frac{DIM_B}{DIM_{MAX}}\right)} - N + 1 \geq 0 \tag{36}$$

$$r_2(y) = 2DIM_B - K_{DIM_{MIN}}(DIM - \mathrm{dim}) \geq 0 \tag{36a}$$

$$r_3(y) = K_{DIM_{MAX}}(DIM - \mathrm{dim}) \geq 0 \tag{36b}$$

$$r_4(y) = \beta B_W - DIM_B \leq 0 \tag{36c}$$

$$r_5(y) = DIM_{MAX} - 0.5(DIM + \mathrm{dim}) \geq 0 \tag{36d}$$

Table 22 Comparative observations of ICHIMP-SHO for welded beam optimisation design issue with other algorithms

Comparative algorithms	Optimal values for variables				Optimum cost
	h	*l*	*t*	*b*	
Proposed ICHIMP-SHO	0.205735	3.474456	9.040229	0.205802	1.726576
Coello (GA-based technique) [157]	0.2088	3.4205	8.9975	0.21	1.748309
hHHO-SCA [75]	0.190086	3.696496	9.386343	0.204157	1.779032249
GA [30]	0.2489	6.1730	8.1789	0.2533	2.4331
GSA [106]	0.1821	3.857	10	0.2024	1.88
Coello and Montes (NPGA) [145]	0.205986	3.471328	9.020224	0.205706	1.728226
Random	0.4575	4.7313	5.0853	0.6600	4.1185
CDE [146]	0.203137	3.542998	9.033498	0.206179	1.733462
(PSOStr)[158]	0.2015	3.526	9.041398	0.205706	1.731186
Simplex	0.2792	5.6256	7.7512	0.2796	2.5307
PSO [149]	0.197411	3.315061	10.00000	0.201395	1.820395
He and Wang (CPSO) [159]	0.202369	3.544214	9.04821	0.205723	1.728024
David	0.2434	6.2552	8.2915	0.2444	2.3841
MFO [104]	0.203567	3.443025	9.230278	0.212359	1.732541
Gandomi et al. (FA) [160]	0.2015	3.562	9.0414	0.2057	1.73121
Approx	0.2444	6.2189	8.2189	0.2444	2.3815
SCA	0.204695	3.536291	9.004290	0.210025	1.759173
HS [35]	0.2442	6.2231	8.2915	0.2443	2.3807

Table 23 Comparative observations of ICHIMP-SHO for gear train optimisation design issue with other algorithms

Comparative algorithms	Optimal values for variables				Gear ratio	Optimum fitness
	$x_1(T_d)$	$x_2(T_b)$	$x_3(T_a)$	$x_4(T_f)$		
Proposed ICHIMP-SHO	28.41056	13.14701	44.79595	57.79147	NA	6.38E−16
IMFO [161]	19	14	34	50	NA	3.0498E−13
ALO [162]	19.00	16.00	43.00	49.00	NA	2.7009E−012
CSA [153]	19.000	16.000	43.000	49.000	NA	2.7008571489E−12
ISA [163]	19	16	43	49	NA	2.701E−12
MP [164]	18	22	45	60	0.1467	5.712E−06
ALM (Kramer) [165]	33	15	13	41	0.1441	2.1246E−08
IDCNLP [166]	14	29	47	59	0.146411	4.5E−06
MIBBSQP [167]	18	22	45	60	0.146666	5.7E−06
MINSLIP [167]	19	16	42	50	NA	2.33E−07
SA [167]	30	15	52	60	0.14423	2.36E−09
MVEP (evolutionary programming) [168]	30	15	52	60	0.14423	2.36E−09
GeneAS[148]	17	14	33	50	0.144242	1.362E−09
MARS [169]	19	16	43	49	0.1442	2.7E−12
cGA [170]	13	20	53	34	NA	2.31E−11
HGA [170]	15	21	59	37	NA	3.07E−10
Ahga1 [170]	13	24	47	46	NA	9.92E−10
Ahga2 [170]	13	20	53	34	NA	2.31E−11
Flc-Ahga [170]	16	19	43	49	NA	2.70E−12
CAPSO [151]	16	19	49	43	0.1442	2.701E−12
MBA [151]	16	19	49	43	0.1442	2.7005E−0.12

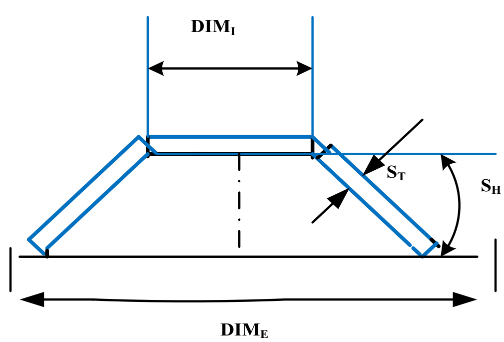


Fig. 18 Belleville spring engineering design

$$r_6(y) = DIM_{MAX} - 0.5(DIM + \mathrm{dim}) \geq 0 \quad (36e)$$

$$r_7(y) = (0.5 + re)(DIM + \mathrm{dim}) \geq 0 \quad (36f)$$

$$r_8(y) = 0.5(DIM - DIM_{MAX} - DIM_B) - \alpha DIM_B \geq 0 \quad (36g)$$

$$r_9(y) = f_I \geq 0.515 \quad (36h)$$

$$r_{10}(y) = f_0 \geq 0.515. \quad (36i)$$

Table 24 Comparative results of ICHIMP-SHO for Belleville spring optimisation design problem with other algorithms

Comparative algorithms	Optimal values for variables				Optimum fitness
	W_1	W_2	W_3	W_4	
Proposed ICHIMP-SHO	12.01	10.0292	0.204239	0.2	1.9817655
hHHO-SCA [75]	11.98603	10.0002	0.204206	0.2	1.98170396
TLBO [26]	12.01	10.03047	0.204143	0.2	0.198966
MBA [151]	12.01	10.030473	0.204143	0.2	0.198965

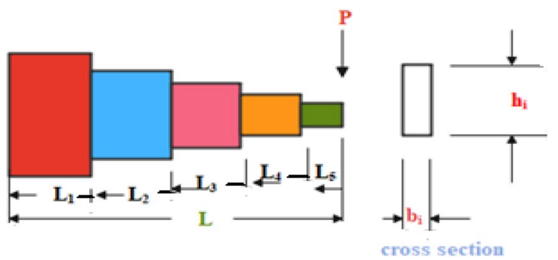


Fig. 19 Design of cantilever beam design

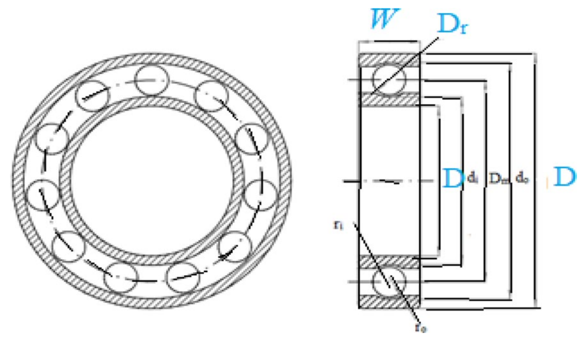


Fig. 20 Problem of rolling bearing design

Here

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

$$\theta_0 = 2\pi - 2\cos^{-1}\left(\frac{\left[\{(DIM-\dim)/2-3(t/4)\}^2+\left(DIM/2-t/4-DIM_B\right)^2-\{\dim/2+t/4\}^2\right]}{2\{(DIM-\dim)/2-3(t/4)\}\left\{D/2-t/4-DIM_B\right\}}\right)$$

$$\varepsilon = \frac{DIM_B}{DIM_{MAX}}, f_I = \frac{R_I}{DIM_B}, f_0 = \frac{R_0}{DIM_B}, t = DIM - \dim - 2DIM_B$$

$$DIM = 160, \dim = 90, B_W = 30, R_I = R_0 = 11.033$$

$$0.5(DIM+\dim) \le DIM_{MAX} \le 0.6(DIM+\dim),$$
$$0.15(DIM-\dim) \le DIM_B \le 0.45(DIM-\dim),$$
$$4 \le N \le 50$$

$$0.515 \le f_I \text{ and } f_0 \le 0.6$$

Table 25 Comparative results of ICHIMP-SHO for cantilever beam optimisation design issue with other algorithms

Comparative algorithms	Optimal values for variables					Optimum weight
	L_1	L_2	L_3	L_4	L_5	
Proposed ICHIMP-SHO	5.969898	4.872735	4.471633	3.487723	2.137855	1.3032958
IMFO [161]	5.97822	4.87623	4.46610	3.47945	2.13912	1.30660
SMA [89]	6.017757	5.310892	4.493758	3.501106	2.150159	1.339957
GCA_I [63]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
MMA [171]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
hGWO-SA [152]	5.9854	4.87	4.4493	3.5172	2.1187	1.3033
MVO [63]	6.02394022154	5.30301123355	4.4950113234	3.4960223242	2.15272617	1.3399595
GCA_II [63]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
CS [172]	6.0089	5.3049	4.5023	3.5077	2.1504	1.33999
ALO [162]	6.01812	5.31142	4.48836	3.49751	2.158329	1.33995
SOS [173]	6.01878	5.30344	4.49587	3.49896	2.15564	1.33996
hHHO-PS [74]	5.978829	4.876628	4.464572	3.479744	2.139358	1.303251
hHHO-SCA [75]	5.937725	4.85041	4.622404	3.45347	2.089114	1.30412236

Table 26 Comparative results of ICHIMP-SHO for rolling element beam optimisation design problem with other algorithms

Comparative algorithms	Optimal values for variables										Optimum fitness
	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}	
Proposed ICHIMP-SHO	125.6093	21.4035	10.9986	0.515	0.515	0.4226	0.6230	0.3014	0.0350	0.6123	− 85383.949
SHO [70]	125	21.4073	10.9326	0.515	0.515	0.4	0.7	0.3	0.2	0.6	85054.532
HHO [100]	125.00	21.00	11.0920	0.5150	0.5150	0.4000	0.6000	0.3000	0.0504	0.600	83011.883
WCA [174]	125.7211	21.4230	1.00103	0.5150	0.5150	0.4015	0.6590	0.3000	0.0400	0.6000	85538.48
PVS [175]	125.719060	21.425590	11.000000	0.515000	0.515000	0.400430	0.680160	0.300000	0.079990	0.700000	81859.741210
SCA [153]	125	21.0328	10.9657	0.515	0.515	0.5	0.7	0.3	0.0277	0.6291	83431.117
MFO [104]	125	21.0328	10.9657	0.515	0.5150	0.5	0.6758	0.3002	0.0239	0.6100	84002.524
MVO [63]	125.6002	21.3225	10.9733	0.515	0.5150	0.5	0.6878	0.3019	0.0361	0.6106	84491.266

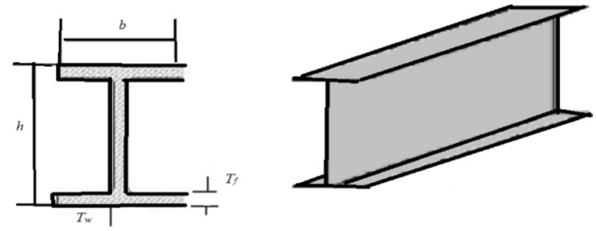


Fig. 21 I beam design and structure

$$0.4 \le K_{DIM_{MIN}} \le 0.5, 0.6 \le K_{DIM_{MAX}} \le 0.7,$$
$$0.3 \le re \le 0.1, 0.02 \le re \le 0.1, 0.6 \le \beta \le 0.85.$$

8.9 I-beam design

By altering the four parameters of the vertical I-beam, this engineering issue attempts to minimize vertical I-beam deviation. The four parameters b, h, t_w, t_f are shown in Fig. 21. In [150], it is stated that to obtain the dimensions of the beam shown in the figure, it has to satisfy geometric and strength constraints to optimize with the criteria: (1) cross-section of beam reduces its volume for given length; (2) static deflection to be noted when the beam is displaced on applying force. The mathematical formulations are given in Eqs. (37–39). Table 27 compares the analytical findings of ICHIMP-SHO with those of other well-known techniques.

Consider

$$\vec{x} = [\, x_1 \;\; x_2 \;\; x_3 \;\;\; x_4 \;\; x_5] = [\, b \;\; h \;\; t_w \;\;\; t_f \,], \tag{37}$$

$$\text{minimize } \; f(\vec{x}) = \frac{5000}{\frac{t_w(h-2t_f)^3}{12} + \frac{bt_f^3}{6} + 2bt_f(\frac{h-t_f}{2})^2}, \tag{38}$$

$$\text{subjected to } g(x) = 2bt_w + t_w(h - 2t_f) \le 0, \tag{39}$$

variable range $10 \le x_1 \le 50, 10 \le x_2 \le 80, 0.9 \le x_3 \le 5,$
$0.9 \le x_4 \le 5.$

8.10 Tension/compression spring design problem

This is a component of the mechanical engineering problem [110, 111], and is one of the engineering designs constraints shown in Fig. 22. The proposal's main characteristic is that it reduces the spring weight. To solve the Spring Model Tension/Compression problem, three types of variable designs are needed: wire diameter (dwr), mean coil diameter (D_c), and active coil number (N). The amount of the surge, the minimal variance, and the limitations centered on the shear stress all play a role in the design. Equations (40)–(41d) show the numerical equations for the suggested engineering

Table 27 Comparative results of ICHIMP-SHO for I-beam optimisation design problem with other algorithms

Comparative algorithms	Optimal values for variables				Optimum fitness
	$(b)\times 1$	$(h)\times 2$	$(T_w)\times 3$	$(T_f)\times 4$	
Proposed ICHIMP-SHO	50	80	1.76467	5	0.006626
BWOA [176]	50.00	80.00	1.76470588	5.00	0.00625958
SMA [89]	49.998845	79.994327	1.764747	4.999742	0.006627
hHHO-PS [74]	50.00	80.00	1.764706	5.00	0.006626
CS [172]	50.0000	80.0000	0.9000	2.3217	0.0131
MFO [104]	50.000	80.000	1.7647	5.000	0.0066259
SOS [173]	50.0000	80.0000	0.9000	2.3218	0.0131
CSA [153]	49.99999	80	0.9	2.3217923	0.013074119
ARMS [177]	37.05	80	1.71	2.31	0.131
Improved ARMS [177]	48.42	79.99	0.9	2.4	0.131

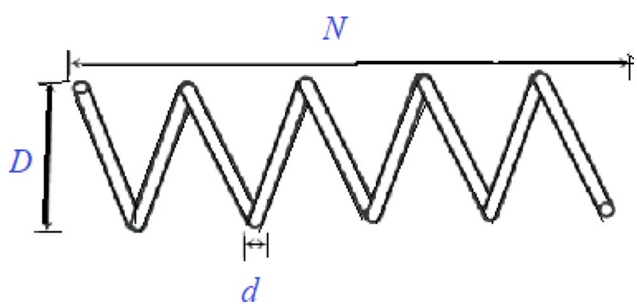


Fig. 22 The spring engineering tension/compression problem

optimization design issue. The results of ICHIMP-SHO are compared to those of other techniques, as shown in Table 28.

Let us consider

$$\vec{S} = [S_1 S_2 S_3] = [dwr D_c N]. \tag{40}$$

But to minimize

Table 28 Comparative results of ICHIMP-SHO for the spring engineering tension/compression problem with other algorithms

Comparative algorithms	Optimal values for variables			Optimum weight
	dwr	D_c	N	
Proposed ICHIMP-SHO	0.051324	0.347614	11.86041	0.0126915
GA [30]	0.05010	0.310111	14.0000	0.013036251
PSO [149]	0.05000	0.3140414	15.0000	0.013192580
IMFO [161]	0.051688973	0.356715627	11.289089342	0.012665233
HS [35]	0.05025	0.316351	15.23960	0.012776352
hHHO-SCA [75]	0.054693	0.433378	7.891402	0.012822904
GSA [106]	0.05000	0.317312	14.22867	0.012873881
BCMO [139]	0.0516597413	0.3560124935	11.3304429494	0.012665
SCA [153]	0.050780	0.334779	12.72269	0.012709667
MALO [178]	0.051759	0.358411	11.191500	0.0126660
MVO [63]	0.05000	0.315956	14.22623	0.012816930
hHHO-PS [74]	0.051682	0.356552	11.29867	0.012665
MFO [104]	0.05000	0.313501	14.03279	0.012753902
VCS [64]	0.051685684299756	0.356636508703361	11.29372966824506	0.012665222962643
AIS-GA	0.0516608	0.3560323	11.329555	0.0126666
BRGA	0.05167471	0.35637260	11.3092294	0.012665237
CDE [146]	0.051609	0.354714	11.410831	0.0126702
WCA [174]	0.051680	0.356522	11.300410	0.012665
DELC [142]	0.051689061	0.356717741	11.28896566	0.012665233
MBA [151]	0.051656	0.355940	11.344665	0.012665
HEAA	0.0516895376	0.3567292035	11.288293703	0.012665233
G-QPSO [144]	0.051515	0.352529	11.538862	0.012665

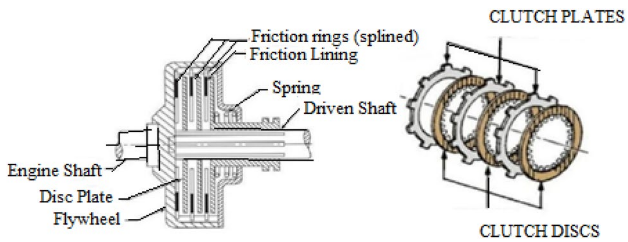


Fig. 23 Multiple clutch break design

$$f(\vec{S}) = (S_3 + 2)S_2S_1^2 \tag{41}$$

$$g_1(\vec{S}) = 1 - \frac{S_2^3 S_3}{71785 S_1^4} \le 0 \tag{41a}$$

$$g_2(\vec{S}) = \frac{4S_2^2 - S_1S_2}{12566(S_2S_1^3 - S_1^4)} + \frac{1}{5108S_1^2} \le 0 \tag{41b}$$

$$g_3(\vec{S}) = 1 - \frac{140.4S_1}{S_2^2 S_3} \le 0 \tag{41c}$$

$$g_4(\vec{S}) = \frac{S_1 + S_2}{1.5} - 1 \le 0. \tag{41d}$$

Ranges of variables are $0.005 \le S_1 \le 2.00, 0.25 \le S_2 \le 1.3, 2.00 \le S_3 \le 1.$

8.11 Multi-disk clutch break (discrete variables)

The multi-disk clutch brake design challenge [179] is one of the most critical technical difficulties highlighted in Fig. 23. The technique of optimization's main purpose is to reduce or increase weight; however, it is made up of five discrete variables: friction surface number (S_{fn}), disk thickness (T_h), outer surface radius (O_{sr}), actuating force form (F_{ac}), and inner surface radius (I_{sr}). From Eqs. (42)–(43g), the mathematical formulas for this design are shown. Table 29 compares the findings of ICHIMP-SHO with those of other techniques.

Mathematical formulas for optimization design are provided below as follows:

$$f(O_{sr}, I_{sr}, S_{fn}, T_h) = \pi Th\gamma (O_{sr}^2 - I_{sr}^2)(S_{fn} + 1), \tag{42}$$

where,

$I_{sr} \in 60, 61, 62....80; O_{sr} \in 90, 91, ...110;$
$T_h \in 1, 1.5, 2, 2.5, 3; F_{ac} \in 600, 610, 620, 1000;$
$S_{fn} \in 2, 3, 4, 5, 6, 7, 8, 9,$

subjected to

$$cb_1 = D_0 - D_{in} - \Delta D \ge 0 \tag{43}$$

$$cb_2 = L_{MAX} - (S_f + 1)(Th + \alpha) \ge 0 \tag{43a}$$

$$cb_3 = PM_{MAX} - PM_\pi \ge 0 \tag{43b}$$

$$cb_4 = PM_{MAX}Z_{MAX} + PM_\pi Z_{SR} \ge 0 \tag{43c}$$

$$cb_5 = Z_{SR_{MAX}} - Z_{SR} \ge 0 \tag{43d}$$

$$cb_6 = t_{MAX} - t \ge 0 \tag{43e}$$

$$cb_7 = RC_h - RC_f \ge 0 \tag{43f}$$

$$cb_8 = t \ge 0 \tag{43g}$$

Table 29 Comparative observations of ICHIMP-SHO for multiple clutch optimisation design problem with other algorithms

Comparative algorithms	Optimal values for variables					Optimum fitness
	×1	×2	×3	×4	×5	
Proposed ICHIMP-SHO	69.99315	90	15	1000	2.31519	0.3900536
HHO [100]	69.999999	90.00	1.00	1000.00	2.312781994	0.259768993
WCA [174]	70.00	90.00	1.00	910.000	3.00	0.313656
MBFPA [180]	70	90	1	600	2	0.235242457900804
PVS [175]	70	90	1	980	3	0.31366
hHHO-SCA [75]	70	90	2.312785	1000	1.5	0.389653842
NSGA-II	70	90	3	1000	1.5	0.4704
TLBO [74]	70	90	3	810	1	0.3136566
MADE [75]	70.00	90	3	810	1	0.3136566
hHHO-PS [74]	76.594	96.59401	1.5	1000	2.13829	0.389653

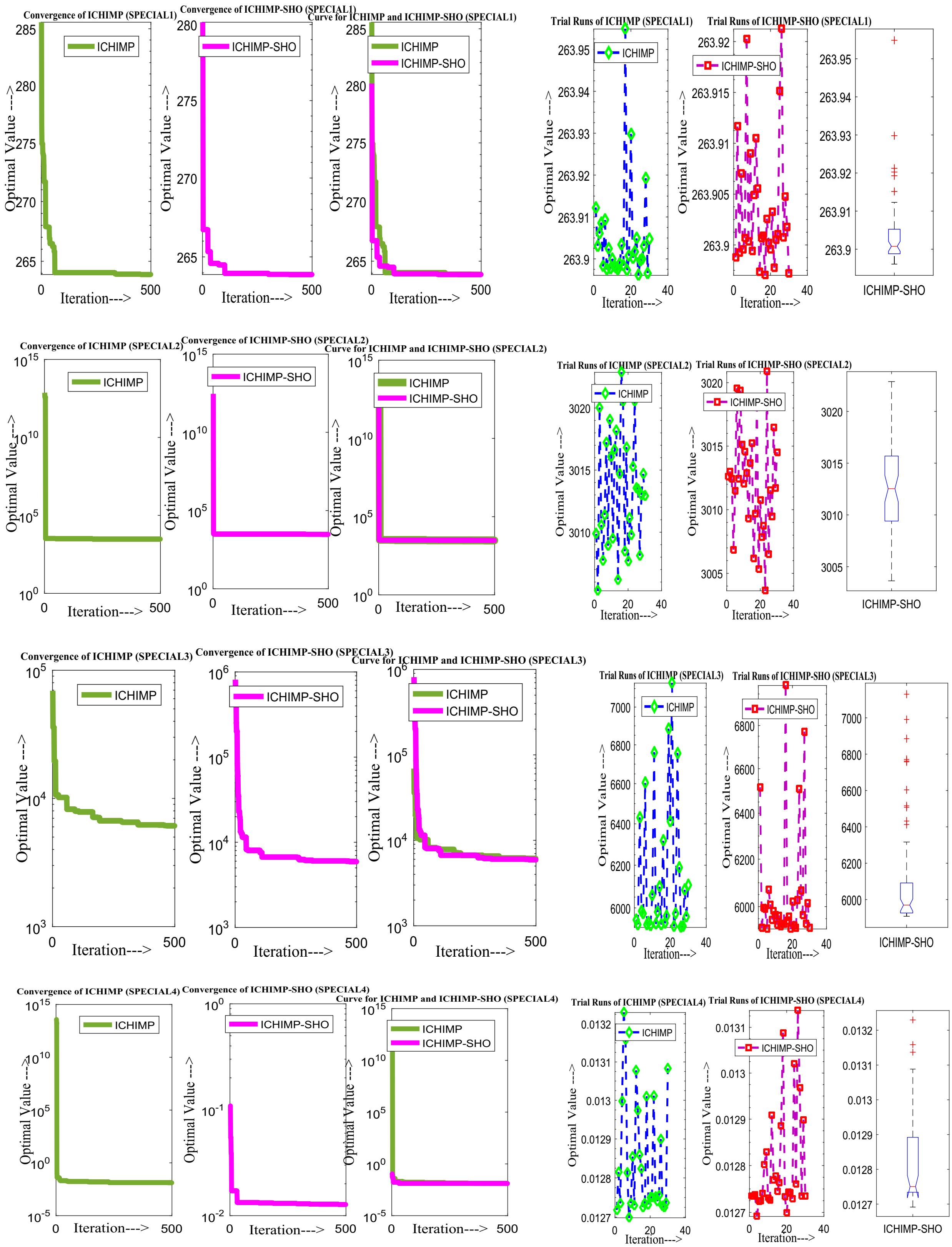


Fig. 24 Convergence curve and Trial runs for multidisciplinary engineering design problem with ICHIMP and ICHIMP-SHO

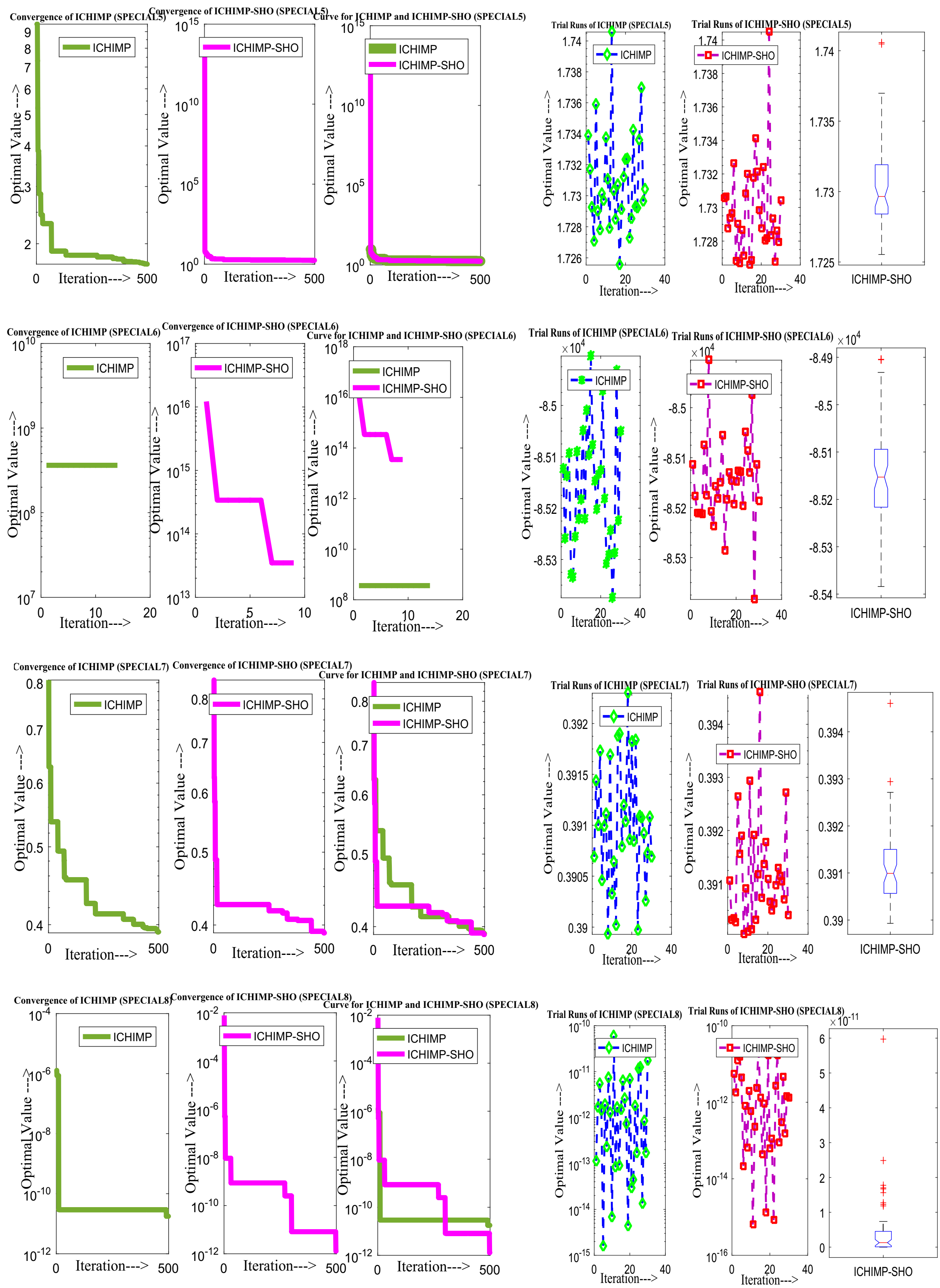


Fig. 24 (continued)

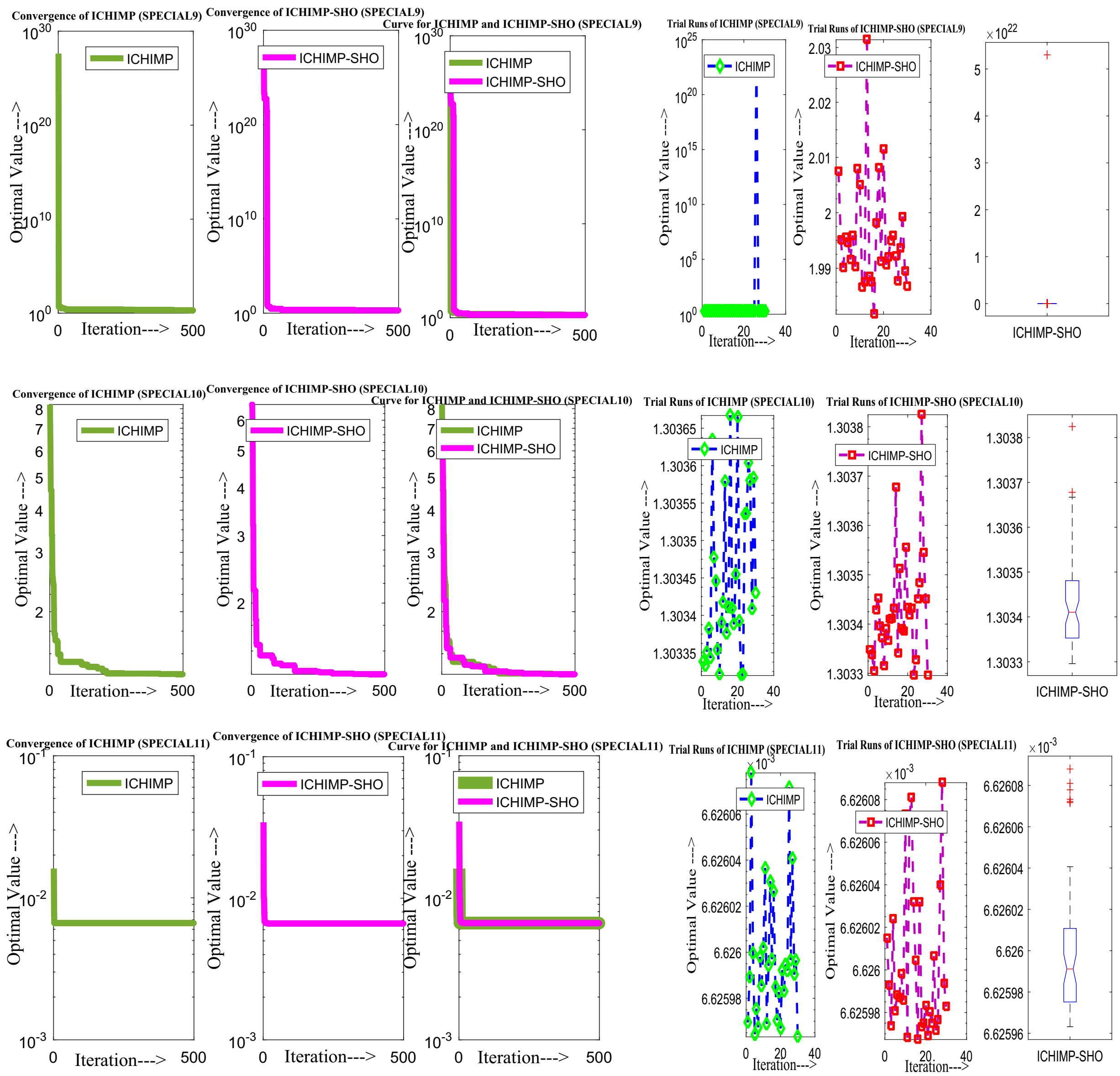


Fig. 24 (continued)

Here,

$$PM_{\pi} = \frac{F_{ac}}{\Pi(D_0^2 - D_{in}^2)}$$

$$Z_{SR} = \frac{2\pi n\left(D_0^3 - D_{in}^3\right)}{90\left(D_0^2 - D_{in}^2\right)}$$

$$t = \frac{i_x \pi n}{30\left(RC_h + RC_f\right)}.$$

9 Conclusion

In the proposed research, two hybrid variants of chimp optimizers have been successfully developed and named as Imp-Chimp and Imp-Chimp-SHO, which are based on a wholesome attitude roused by amazing thinking and hunting ability with a sensual movement for finding the optimal solution in the global search region. The newly developed improved variant of Chimp optimizer has been successfully tested for various engineering design and standard benchmark optimization problems, which includes uni-modal, multi-modal, and fixed dimensions benchmark problems. After validating the efficiency of the proposed optimizers for standard benchmarks and engineering design problems, it has been experimentally observed that both the variants are competitive for finding the solution within the global search space. Based on experimental results and comparative analysis with other methodologies, it has been recommended that the proposed hybrid variants can be universally accepted to solve any of the hard engineering design challenges in the global search space. However, while dealing with these two variants as compared to the standard ChoA, both the algorithms are slow with respect to computational complexity due to sequential hybridized nature of the algorithm (Fig. 24).

Furthermore, these hybrid variants can be applied to solve the single and multi-area economic load dispatch problem with renewable energy sources, charging and discharging of PEVs/BEVs, storage strategies, automatic generation, and monitoring functions of the realistic power system. Furthermore, the developed hybrid algorithm versions will aid various academics and upcoming analysts working on new population-based approaches, unique optimization strategies, and the development of hybrid optimization algorithms.

Acknowledgements The authors wish to thank Dr. M. Khishe and Dr. M. R. Mosavi for providing free access to MATLAB code of Chimp Optimizer algorithm. The corresponding author (Dr. Vikram Kumar Kamboj) wishes to thank Dr. O.P. Malik, Professor Emeritus, Department of Electrical and Computer Engineering, Schulich School of Engineering, University of Calgary, Calgary, Alberta, CANADA for his continuous support, guidance, encouragement and for providing advance research facilities for post-doctoral research at the University of Calgary, Alberta, CANADA. Further, Dr. Vikram Kumar Kamboj wishes to thank Mitacs Elevate for providing Financial Support for Post Doctoral Fellowship (PDF) Research activity at University of Calgary, Alberta, Canada and Simon Fraser University, British Columbia, Canada.

References

1. Abbassi R, Abbassi A, Heidari AA, Mirjalili S (2019) An efficient salp swarm-inspired algorithm for parameters identification of photovoltaic cell models. Energy Convers Manag 179(January):362–372. https://doi.org/10.1016/j.enconman.2018.10.069
2. Faris H et al (2019) An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks. Inf Fus 48(August):67–83. https://doi.org/10.1016/j.inffus.2018.08.002
3. &NA (2000) Rapid Communications. JAIDS J Acquir Immun Defic Syn 23(5):374. https://doi.org/10.1097/00126334-200004150-00002
4. Wu G (2016) Across neighborhood search for numerical optimization. Inf Sci (NY) 329(61563016):597–618. https://doi.org/10.1016/j.ins.2015.09.051
5. Wu G, Pedrycz W, Suganthan PN, Mallipeddi R (2015) A variable reduction strategy for evolutionary algorithms handling equality constraints. Appl Soft Comput J 37:774–786. https://doi.org/10.1016/j.asoc.2015.09.007
6. Khishe M, Mosavi MR (2020) Chimp optimization algorithm. Expert Syst Appl 149:113338. https://doi.org/10.1016/j.eswa.2020.113338
7. Saremi S, Mirjalili S, Lewis A (2014) Biogeography-based optimisation with chaos. Neural Comput Appl 25(5):1077–1097. https://doi.org/10.1007/s00521-014-1597-x
8. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim 39(3):459–471. https://doi.org/10.1007/s10898-007-9149-x
9. Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim. https://doi.org/10.1023/A:1008202821328
10. Elsayed SM, Sarker RA, Essam DL (2014) A new genetic algorithm for solving optimization problems. Eng Appl Artif Intell 27:57–69. https://doi.org/10.1016/j.engappai.2013.09.013
11. Gandomi AH, Yang XS, Alavi AH (2013) Erratum: Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems (Engineering with Computers DOI: 10.1007/s00366-011-0241-y). Eng Comput 29(2):245. https://doi.org/10.1007/s00366-012-0308-4
12. Das S, Biswas A, Dasgupta S, Abraham A (2009) Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications. Stud Comput Intell 203:23–55. https://doi.org/10.1007/978-3-642-01085-9_2
13. Y. Xin-She (2012) Flower pollination algorithm for global optimization. In: Unconventional computation and natural computation, Springer, pp 240–49
14. Lam AYS, Li VOK (2012) Chemical reaction optimization: a tutorial. Memet Comput 4(1):3–17. https://doi.org/10.1007/s12293-012-0075-1
15. Yang XS (2010) Firefly algorithm, stochastic test functions and design optimization. Int J Bio Inspired Comput 2(2):78–84. https://doi.org/10.1504/IJBIC.2010.032124
16. Zong-Yuan M (2002) "5 1063 1" pp. 1–5. https://doi.org/10.1006/rwgn.2001.0729
17. Rao RV, Savsani VJ, Balic J (2012) Teaching-learning-based optimization algorithm for unconstrained and constrained real-parameter optimization problems. Eng Optim 44(12):1447–1462. https://doi.org/10.1080/0305215X.2011.652103
18. Kim TH, Maruta I, Sugie T (2010) A simple and efficient constrained particle swarm optimization and its application to engineering design problems. Proc Inst Mech Eng Part C J Mech Eng Sci 224(2):389–400. https://doi.org/10.1243/09544062JMES1732
19. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey Wolf Optimizer, vol. 69. Elsevier Ltd.
20. Cuevas E, Cienfuegos M (2014) A new algorithm inspired in the behavior of the social-spider for constrained optimization. Expert Syst Appl 41(2):412–425. https://doi.org/10.1016/j.eswa.2013.07.067
21. Yadav A, Deep K (2013) Constrained optimization using gravitational search algorithm. Natl Acad Sci Lett 36(5):527–534. https://doi.org/10.1007/s40009-013-0165-8
22. Gandomi AH, Yang XS, Alavi AH, Talatahari S (2013) Bat algorithm for constrained optimization tasks. Neural Comput Appl 22(6):1239–1255. https://doi.org/10.1007/s00521-012-1028-9
23. Whitley D (2001) An overview of evolutionary algorithms: practical issues and common pitfalls. Inf Softw Technol 43(14):817–831. https://doi.org/10.1016/S0950-5849(01)00188-4
24. Calvet L, De Armas J, Masip D, Juan AA (2017) Learnheuristics: hybridizing metaheuristics with machine learning for optimization with dynamic inputs. Open Math 15(1):261–280. https://doi.org/10.1515/math-2017-0029
25. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications. Fut Gen Comput Syst. https://doi.org/10.1016/j.future.2019.02.028

26. Rao RV, Savsani VJ, Vakharia DP (2012) Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems. Inf Sci (NY) 183(1):1–15. https://doi.org/10.1016/j.ins.2011.08.006
27. Hansen P, Mladenović N, Moreno Pérez JA (2010) Variable neighbourhood search: methods and applications. Ann Oper Res 175(1):367–407. https://doi.org/10.1007/s10479-009-0657-6
28. Doǫan B, Ölmez T (2015) A new metaheuristic for numerical function optimization: vortex Search algorithm. Inf Sci (NY) 293(August):125–145. https://doi.org/10.1016/j.ins.2014.08.053
29. Takeang C, Aurasopon A (2019) Multiple of hybrid lambda iteration and simulated annealing algorithm to solve economic dispatch problem with ramp rate limit and prohibited operating zones. J Electr Eng Technol 14(1):111–120. https://doi.org/10.1007/s42835-018-00001-z
30. Yalcinoz T, Altun H, Uzam M (2001) Economic dispatch solution using a genetic algorithm based on arithmetic crossover. 2001 IEEE Porto Power Tech Proc 2(4):153–156. https://doi.org/10.1109/PTC.2001.964734
31. Naama B, Bouzeboudja H, Allali A (2013) Solving the economic dispatch problem by using Tabu Search algorithm. Energy Procedia 36:694–701. https://doi.org/10.1016/j.egypro.2013.07.080
32. Nguyen TT, Vo DN (2015) The application of one rank cuckoo search algorithm for solving economic load dispatch problems. Appl Soft Comput J 37:763–773. https://doi.org/10.1016/j.asoc.2015.09.010
33. Swain RK, Sahu NC, Hota PK (2012) Gravitational search algorithm for optimal economic dispatch. Procedia Technol 6:411–419. https://doi.org/10.1016/j.protcy.2012.10.049
34. Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster y 1 introduction. IEEE Trans Evol Comput 3(July):82–102
35. Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. Simulation 76(2):60–68. https://doi.org/10.1177/003754970107600201
36. Ghaemi M, Feizi-Derakhshi MR (2014) Forest optimization algorithm, vol. 41. Elsevier Ltd
37. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature−inspired heuristic paradigm. Knowl-Based Syst 89:228–249. https://doi.org/10.1016/j.knosys.2015.07.006
38. Salimi H (2015) Stochastic fractal search: a powerful metaheuristic algorithm, vol 75. Elsevier B.V.
39. Xu J, Zhang J (2014) Exploration-exploitation tradeoffs in metaheuristics: Survey and analysis. In: Proc. 33rd Chinese Control Conf. CCC 2014, pp. 8633–8638. https://doi.org/10.1109/ChiCC.2014.6896450.
40. Yang XS, Deb S, Fong S (2014) Metaheuristic algorithms: optimal balance of intensification and diversification. Appl Math Inf Sci 8(3):977–983. https://doi.org/10.12785/amis/080306
41. Nematollahi AF, Rahiminejad A, Vahidi B (2017) A novel physical based meta-heuristic optimization method known as lightning attachment procedure optimization. Appl Soft Comput J 59:596–621. https://doi.org/10.1016/j.asoc.2017.06.033
42. Rechenberg I (1989) Evolution strategy: nature's way of optimization, pp 106–126. https://doi.org/10.1007/978-3-642-83814-9_6
43. Biswas A, Mishra KK, Tiwari S, Misra AK (2013) Physics-inspired optimization algorithms: a survey. J Optim 2013:1–16. https://doi.org/10.1155/2013/438152
44. Formato RA, Engineers E (2014) Central force optimization algorithm. Intell Syst Ref Libr 62(November):333–337. https://doi.org/10.1007/978-3-319-03404-1_19
45. Kaveh A, Talatahari S (2010) A novel heuristic optimization method: charged system search. Acta Mech 213(3–4):267–289. https://doi.org/10.1007/s00707-009-0270-4
46. Ali MM, Golalikhani M (2010) An electromagnetism-like method for nonlinearly constrained global optimization. Comput Math Appl 60(8):2279–2285. https://doi.org/10.1016/j.camwa.2010.08.018
47. Erol OK, Eksin I (2006) A new optimization method: big bang-big crunch. Adv Eng Softw 37(2):106–111. https://doi.org/10.1016/j.advengsoft.2005.04.005
48. Mirjalili S, Lewis A (2014) Adaptive gbest-guided gravitational search algorithm. Neural Comput Appl 25(7–8):1569–1584. https://doi.org/10.1007/s00521-014-1640-y
49. Hongye L, Atashpaz-Gargari E, Lucas C (2007) Imperialistic competitive algorithm ICA IEEE CEC 2007 inspired by imperialistic competition
50. Kumar M, Kulkarni AJ, Satapathy SC (2018) Socio evolution and learning optimization algorithm: a socio-inspired optimization methodology. Fut Gen Comput Syst 81:252–272. https://doi.org/10.1016/j.future.2017.10.052
51. Ruiz-Vanoye JA, Díaz-Parra O, Cocón F, Soto A (2021) Meta-heuristics algorithms based on the grouping of animals by social behavior for the traveling salesman problem. Int J Comb Optim Prob Inform
52. Faris H, Aljarah I, Mirjalili S (2018) Improved monarch butterfly optimization for unconstrained global search and neural network training. Appl Intell 48(2):445–464. https://doi.org/10.1007/s10489-017-0967-3
53. Gandomi AH, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simul 17(12):4831–4845. https://doi.org/10.1016/j.cnsns.2012.05.010
54. Mirjalili SZ, Mirjalili S, Saremi S, Faris H, Aljarah I (2018) Grasshopper optimization algorithm for multi-objective optimization problems. Appl Intell 48(4):805–820. https://doi.org/10.1007/s10489-017-1019-8
55. Rizk-Allah RM, Hassanien AE, Elhoseny M, Gunasekaran M (2019) A new binary salp swarm algorithm: development and application for optimization tasks. Neural Comput Appl 31(5):1641–1663. https://doi.org/10.1007/s00521-018-3613-z
56. Khishe M, Safari A (2019) Classification of sonar targets using an mlp neural network trained by dragonfly algorithm. Wirel Pers Commun 108(4):2241–2260. https://doi.org/10.1007/s11277-019-06520-w
57. Khishe M, Mosavi MR (2019) Improved whale trainer for sonar datasets classification using neural network. Appl Acoust 154:176–192. https://doi.org/10.1016/j.apacoust.2019.05.006
58. Hashim FA, Houssein EH, Mabrouk MS, Al-atabany W (2019) Henry gas solubility optimization: a novel physics-based algorithm. Fut Gen Comput Syst 101:646–667. https://doi.org/10.1016/j.future.2019.07.015
59. Liu Y, Li R (2020) PSA: a photon search algorithm 16(2):478–493
60. Wang GG, Guo L, Gandomi AH, Hao GS, Wang H (2014) Chaotic Krill Herd algorithm. Inf Sci (NY) 274(January):17–34. https://doi.org/10.1016/j.ins.2014.02.123
61. Meng XB, Gao XZ, Lu L, Liu Y, Zhang H (2016) A new bio-inspired optimisation algorithm: bird Swarm Algorithm. J Exp Theor Artif Intell 28(4):673–687. https://doi.org/10.1080/0952813X.2015.1042530
62. Shareef H, Ibrahim AA, Mutlag AH (2015) Lightning search algorithm. Appl Soft Comput J 36:315–333. https://doi.org/10.1016/j.asoc.2015.07.028
63. Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl 27(2):495–513. https://doi.org/10.1007/s00521-015-1870-7
64. Li MD, Zhao H, Weng XW, Han T (2016) A novel nature-inspired algorithm for optimization: virus colony search. Adv Eng Softw 92:65–88. https://doi.org/10.1016/j.advengsoft.2015.11.004

65. Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. Adv Eng Softw 105:30–47. https://doi.org/10.1016/j.advengsoft.2017.01.004
66. Deb S, Gao XZ, Tammi K, Kalita K, Mahanta P (2020) Recent studies on chicken swarm optimization algorithm: a review (2014–2018). Artif Intell Rev 53(3):1737–1765. https://doi.org/10.1007/s10462-019-09718-3
67. Singh N, Singh SB (2017) A novel hybrid GWO-SCA approach for optimization problems. Eng Sci Technol Int J 20(6):1586–1601. https://doi.org/10.1016/j.jestch.2017.11.001
68. Huang KW, Wu ZX (2018) CPO: a crow particle optimization algorithm. Int J Comput Intell Syst 12(1):426–435. https://doi.org/10.2991/ijcis.2018.125905658
69. Aala Kalananda VKR, Komanapalli VLN (2021) A combinatorial social group whale optimization algorithm for numerical and engineering optimization problems. Appl Soft Comput 99:106903. https://doi.org/10.1016/j.asoc.2020.106903
70. Dhiman G, Kumar V (2017) Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications. Adv Eng Softw 114:48–70. https://doi.org/10.1016/j.advengsoft.2017.05.014
71. Dhiman G, Kumar V (2018) Multi-objective spotted hyena optimizer: a Multi-objective optimization algorithm for engineering problems. Knowl Based Syst 150(March):175–197. https://doi.org/10.1016/j.knosys.2018.03.011
72. Hu K, Jiang H, Ji CG, Pan Z (2020) A modified butterfly optimization algorithm: an adaptive algorithm for global optimization and the support vector machine. Expert Syst. https://doi.org/10.1111/exsy.12642
73. Kumar V, Kaur A (2020) Binary spotted hyena optimizer and its application to feature selection. J Ambient Intell Humaniz Comput 11(7):2625–2645. https://doi.org/10.1007/s12652-019-01324-z
74. Krishna AB, Saxena S, Kamboj VK (2021) A novel statistical approach to numerical and multidisciplinary design optimization problems using pattern search inspired Harris hawks optimizer. Springer, London
75. Kamboj VK, Nandi A, Bhadoria A, Sehgal S (2020) An intensify Harris Hawks optimizer for numerical and engineering optimization problems. Appl Soft Comput J 89:106018. https://doi.org/10.1016/j.asoc.2019.106018
76. Zamani H, Nadimi-shahraki MH (2020) Enhancement of bernstain-search differential evolution algorithm to solve constrained engineering problems. Int J Comput Sci Eng 9(6):386–396
77. Meng Z, Li G, Wang X, Sait SM, Yıldız AR (2020) A comparative study of metaheuristic algorithms for reliability-based design optimization problems. Arch Comput Methods Eng. https://doi.org/10.1007/s11831-020-09443-z
78. Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH (2021) The arithmetic optimization algorithm. Comput Methods Appl Mech Eng 376:113609. https://doi.org/10.1016/j.cma.2020.113609
79. Hashim FA, Hussain K, Houssein EH, Mabrouk MS, Al-Atabany W (2021) Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. Appl Intell 51(3):1531–1551. https://doi.org/10.1007/s10489-020-01893-z
80. Bala Krishna A, Saxena S, Kamboj VK (2021) hSMA-PS: a novel memetic approach for numerical and engineering design challenges, no. 0123456789. Springer, London
81. Abualigah L, Yousri D, Abd Elaziz M, Ewees AA, Al-qaness MAA, Gandomi AH (2021) Aquila optimizer: a novel meta-heuristic optimization algorithm. Comput Ind Eng 157:107250. https://doi.org/10.1016/j.cie.2021.107250
82. Xu Z et al (2021) Spiral motion mode embedded grasshopper optimization algorithm: design and analysis. IEEE Access 9:71104–71132. https://doi.org/10.1109/access.2021.3077616
83. Neshat M et al (2021) Wind turbine power output prediction using a new hybrid neuro-evolutionary method. Energy 229:120617
84. Kaur A, Singh L, Dhillon JS (2021) Modified Krill Herd Algorithm for constrained economic load dispatch problem. J Ambient Energy Int. https://doi.org/10.1080/01430750.2021.1888798
85. Nandi A, Kamboj VK (2021) A meliorated Harris Hawks optimizer for combinatorial unit commitment problem with photovoltaic applications. J Electr Syst Inf Technol. https://doi.org/10.1186/s43067-020-00026-3
86. Yang Y, Chen H, Heidari AA, Gandomi AH (2021) Hunger games search: visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. Expert Syst Appl 177:114864. https://doi.org/10.1016/j.eswa.2021.114864
87. Osaba E, Yang X-S (2021) Soccer-inspired metaheuristics: systematic review of recent research and applications. Appl Optim Swarm Intell. https://doi.org/10.1007/978-981-16-0662-5_5
88. Barshandeh S, Piri F, Sangani SR (2020) HMPA: an innovative hybrid multi-population algorithm based on artificial ecosystem-based and Harris Hawks optimization algorithms for engineering problems. Springer, London
89. Li S, Chen H, Wang M, Heidari AA, Mirjalili S (2020) Slime mould algorithm: a new method for stochastic optimization. Fut Gen Comput Syst 111:300–323. https://doi.org/10.1016/j.future.2020.03.055
90. Faramarzi A, Heidarinejad M, Mirjalili S, Gandomi AH (2020) Marine predators algorithm: a nature-inspired metaheuristic. Expert Syst Appl 152:113377. https://doi.org/10.1016/j.eswa.2020.113377
91. Abdel-Basset M, Chang V, Mohamed R (2020) HSMA_WOA: a hybrid novel Slime mould algorithm with whale optimization algorithm for tackling the image segmentation problem of chest X-ray images. Appl Soft Comput J 95:106642. https://doi.org/10.1016/j.asoc.2020.106642
92. Chen Z, Liu W (2020) An efficient parameter adaptive support vector regression using K-Means clustering and chaotic slime mould algorithm. IEEE Access 8:156851–156862. https://doi.org/10.1109/ACCESS.2020.3018866
93. Premkumar M, Jangir P, Sowmya R, Alhelou HH, Heidari AA, Chen H (2021) MOSMA: multi-objective slime mould algorithm based on elitist non-dominated sorting. IEEE Access 9:3229–3248. https://doi.org/10.1109/ACCESS.2020.3047936
94. Zhao J, Gao ZM (2020) The chaotic slime mould algorithm with chebyshev map. J Phys Conf Ser. https://doi.org/10.1088/1742-6596/1631/1/012071
95. Majhi SK, Mishra A, Pradhan R (2019) A chaotic salp swarm algorithm based on quadratic integrate and fire neural model for function optimization. Prog Artif Intell 8(3):343–358. https://doi.org/10.1007/s13748-019-00184-0
96. Li Y, Han M, Guo Q (2020) Modified whale optimization algorithm based on tent chaotic mapping and its application in structural optimization. KSCE J Civ Eng 24(12):3703–3713. https://doi.org/10.1007/s12205-020-0504-5
97. Ji Y et al (2020) An adaptive chaotic sine cosine algorithm for constrained and unconstrained optimization. Complexity. https://doi.org/10.1155/2020/6084917
98. Paul C, Roy PK, Mukherjee V (2020) Chaotic whale optimization algorithm for optimal solution of combined heat and power economic dispatch problem incorporating wind, vol 35. Elsevier Ltd
99. Dhiman G, Kaur A (2019) A hybrid algorithm based on particle swarm and spotted hyena optimizer for global optimization, vol 816. Springer, Singapore
100. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications.

Fut Gen Comput Syst. https://doi.org/10.1016/j.future.2019.02.028
101. Chen X, Tianfield H, Li K (2019) SC. Swarm Evol Comput BASE Data. https://doi.org/10.1016/j.swevo.2019.01.003
102. Shadravan S, Naji HR, Bardsiri VK (2019) The sailfish optimizer: a novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. Eng Appl Artif Intell 80:20–34. https://doi.org/10.1016/j.engappai.2019.01.001
103. Verma C, Illes Z, Stoffova V (2019) Age group predictive models for the real time prediction of the university students using machine learning: preliminary results. In: Proc. 2019 3rd IEEE Int. Conf. Electr. Comput. Commun. Technol. ICECCT 2019. https://doi.org/10.1109/ICECCT.2019.8869136
104. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature−inspired heuristic paradigm. Knowl Based Syst. https://doi.org/10.1016/j.knosys.2015.07.006
105. Mirjalili S, Wang GG, dos Coelho LS (2014) Binary optimization using hybrid particle swarm optimization and gravitational search algorithm. Neural Comput Appl 25(6):1423–1435. https://doi.org/10.1007/s00521-014-1629-6
106. Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci (NY) 179(13):2232–2248. https://doi.org/10.1016/j.ins.2009.03.004
107. Simon D (2008) Biogeography-based optimization,
108. Yao Xin, Liu Yong, Lin G (1999) Evolutionary programming made faster. IEEE Trans Evol Computat. https://doi.org/10.1109/4235.771163
109. Glover F (1989) Tabu search—part I. Orsa J Comput 1(3):190–206
110. Hu T, Khishe M, Mohammadi M, Parvizi GR, Taher Karim SH, Rashid TA (2021) Real-time COVID-19 diagnosis from X-ray images using deep CNN and extreme learning machines stabilized by chimp optimization algorithm. Biomed Signal Process Control 68:102764. https://doi.org/10.1016/j.bspc.2021.102764
111. Zayed ME et al (2021) Predicting the performance of solar dish Stirling power plant using a hybrid random vector functional link/chimp optimization model. Sol Energy 222(March):1–17. https://doi.org/10.1016/j.solener.2021.03.087
112. Dhiman G (2021) SSC: A hybrid nature-inspired meta-heuristic optimization algorithm for engineering applications. Knowledge-Based Syst 222:106926. https://doi.org/10.1016/j.knosys.2021.106926
113. Kaur M, Kaur R, Singh N, Dhiman G (2021) SChoA: an newly fusion of sine and cosine with chimp optimization algorithm for HLS of datapaths in digital filters and engineering applications. Eng Comput. https://doi.org/10.1007/s00366-020-01233-2
114. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82. https://doi.org/10.1109/4235.585893
115. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2010) BGSA: binary gravitational search algorithm. Nat Comput 9(3):727–745. https://doi.org/10.1007/s11047-009-9175-3
116. Emary E, Zawbaa HM, Hassanien AE (2016) Binary grey wolf optimization approaches for feature selection. Neurocomputing 172:371–381. https://doi.org/10.1016/j.neucom.2015.06.083
117. Nakamura RYM, Pereira LAM, Costa KA, Rodrigues D, Papa JP, Yang XS (2012) BBA: a binary bat algorithm for feature selection. Braz Symp Comput Graph Image Process. https://doi.org/10.1109/SIBGRAPI.2012.47
118. Nakamura RYM, Pereira LAM, Rodrigues D, Costa KAP, Papa JP, Yang XS (2013) Binary bat algorithm for feature selection. Swarm Intell Bio-Inspired Comput 2010:225–237. https://doi.org/10.1016/B978-0-12-405163-8.00009-0
119. Kennedy J, Eberhart RC (1997) Discrete binary version of the particle swarm algorithm. Proc IEEE Int Conf Syst Man Cybern 5:4104–4108. https://doi.org/10.1109/icsmc.1997.637339
120. Dhiman G, Kaur A (2019) A hybrid algorithm based on particle swarm and spotted hyena optimizer for global optimization, vol 816. Springer, Singapore
121. Kaur S, Awasthi LK, Sangal AL (2021) HMOSHSSA: a hybrid meta-heuristic approach for solving constrained optimization problems, vol 37. Springer, London
122. Dhiman G (2020) MOSHEPO: a hybrid multi-objective approach to solve economic load dispatch and micro grid problems. Appl Intell 50(1):119–137. https://doi.org/10.1007/s10489-019-01522-4
123. Guo Z, Moayedi H, Foong LK, Bahiraei M (2020) Optimal modification of heating, ventilation, and air conditioning system performances in residential buildings using the integration of metaheuristic optimization and neural computing. Energy Build 214:109866. https://doi.org/10.1016/j.enbuild.2020.109866
124. Roth G, Dicke U (2005) Evolution of the brain and intelligence. Trends Cogn Sci 9(5):250–257. https://doi.org/10.1016/j.tics.2005.03.005
125. Khishe M, Mosavi MR (2020) Classification of underwater acoustical dataset using neural network trained by Chimp Optimization Algorithm. Appl Acoust 157:107005. https://doi.org/10.1016/j.apacoust.2019.107005
126. Saffari A, Zahiri SH, Khishe M, Seyyed Mohammadreza Mosavi (2020) Design of a fuzzy model of control parameters of chimp algorithm optimization for automatic sonar targets recognition. Iran J Mar Technol. [Online]. Available at http://ijmt.iranjournals.ir/article_241126.html.
127. Jia H, Sun K, Zhang W, Leng X (2021) An enhanced chimp optimization algorithm for continuous optimization domains. Syst Complex Intell. https://doi.org/10.1007/s40747-021-00346-5
128. Wang J, Khishe M, Kaveh M, Mohammadi H (2021) Binary chimp optimization algorithm (BChOA): a new binary metaheuristic for solving optimization problems. Cognit Comput. https://doi.org/10.1007/s12559-021-09933-7
129. Kumari CL, Kamboj VK (2020) An effective solution to single-area dynamic dispatch using improved chimp optimizer. E3S Web Conf. 184: 1–9. https://doi.org/10.1051/e3sconf/202018401069.
130. Abdul Jabbar NM, Mitras BA (2021) Modified chimp optimization algorithm based on classical conjugate gradient methods. J Phys Conf Ser. https://doi.org/10.1088/1742-6596/1963/1/012027
131. Houssein EH, Emam MM, Ali AA (2021) An efficient multilevel thresholding segmentation method for thermography breast cancer imaging based on improved chimp optimization algorithm. Expert Syst Appl 185(January):115651. https://doi.org/10.1016/j.eswa.2021.115651
132. Sherif BV (2021) Detection and isolation of sel sh nodes in MANET using collaborative contact-based watchdog with.
133. Rahkar Farshi T (2020) Battle royale optimization algorithm. Neural Comput Appl. https://doi.org/10.1007/s00521-020-05004-4
134. Hans R, Kaur H (2020) Opposition-based enhanced grey wolf optimization algorithm for feature selection in breast density classification. Int J Mach Learn Comput 10(3):458–464. https://doi.org/10.18178/ijmlc.2020.10.3.957
135. Bhullar AK, Kaur R, Sondhi S (2020) Enhanced crow search algorithm for AVR optimization, vol 24. Springer, Berlin, Heidelberg
136. Qais MH, Hasanien HM, Alghuwainem S (2020) Transient search optimization: a new meta-heuristic optimization algorithm. Appl Intell 50(11):3926–3941. https://doi.org/10.1007/s10489-020-01727-y

137. Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2013) Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. Appl Soft Comput 13(5):2592–2612
138. Bhadoria A, Kamboj VK (2018) Optimal generation scheduling and dispatch of thermal generating units considering impact of wind penetration using hGWO-RES algorithm. Appl Intell. https://doi.org/10.1007/s10489-018-1325-9
139. LE−Duc T, Nguyen QH, Nguyen-Xuan H (2020) Balancing composite motion optimization. Inf Sci (NY) 520:250–270. https://doi.org/10.1016/j.ins.2020.02.013
140. Kaveh A, Talatahari S (2010) An improved ant colony optimization for constrained engineering design problems. Eng Comput (Swansea Wales). https://doi.org/10.1108/02644401011008577
141. Bernardino HS, Barbosa HJC, Lemonge ACC (2007) A hybrid genetic algorithm for constrained optimization problems in mechanical engineering. In: 2007 IEEE Congr. Evol. Comput. CEC 2007, no. September, pp. 646–653. https://doi.org/10.1109/CEC.2007.4424532
142. Wang L, Li LP (2010) An effective differential evolution with level comparison for constrained engineering design. Struct Multidiscip Optim 41(6):947–963. https://doi.org/10.1007/s00158-009-0454-5
143. Cagnina LC, Esquivel SC, Nacional U, Luis DS, Luis S, Coello CAC (2008) Solving engineering optimization problems with the simple constrained particle swarm optimizer 1 introduction 2 literature review 3 our proposed approach : SiC-PSO. Eng Optim 32:319–326
144. dos Coelho LS (2010) Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. Expert Syst Appl 37(2):1676–1683. https://doi.org/10.1016/j.eswa.2009.06.044
145. Coello Coello CA, Montes EM (2002) Constraint-handling in genetic algorithms through the use of dominancE−based tournament selection. Adv Eng Inform 16(3):193–203. https://doi.org/10.1016/S1474-0346(02)00011-3
146. Zhuo Huang F, Wang L, He Q (2007) An effective co-evolutionary differential evolution for constrained optimization. Appl Math Comput 186(1):340–356. https://doi.org/10.1016/j.amc.2006.07.105
147. Economics R (2010) Comprehensive learning particle swarm optimizer for constrained mixed-variable optimization problems 3(6): 832–842. Received : 07–05–2010 Accepted : 05–10–2010
148. Deb K, Goyal M (1996) A combined genetic adaptive search (GeneAS) for engineering design. Comput Sci Inform 26(1): 30–45 [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.27.767%5Cnhttp://repository.ias.ac.in/82723/.
149. Victoire TAA, Jeyakumar AE (2004) Hybrid PSO–SQP for economic dispatch with valve-point effect. Electr Power Syst Res 71(1):51–59. https://doi.org/10.1016/J.EPSR.2003.12.017
150. Ray T, Saini P (2001) Engineering design optimization using a swarm with an intelligent information sharing among individuals. Eng Optim 33(6):735–748. https://doi.org/10.1080/03052150108940941
151. Gandomi AH, Yun GJ, Yang XS, Talatahari S (2013) Chaos-enhanced accelerated particle swarm optimization. Commun Nonlinear Sci Numer Simul 18(2):327–340. https://doi.org/10.1016/j.cnsns.2012.07.017
152. Bhadoria A, Marwaha S, Kamboj VK (2020) A solution to statistical and multidisciplinary design optimization problems using hGWO-SA algorithm. Neural Comput Appl. https://doi.org/10.1007/s00521-020-05229-3
153. Zolghadr-Asli B, Bozorg-Haddad O, Chu X (2018) Crow search algorithm (CSA). In: Studies in Computational Intelligence
154. Zhang M, Luo W, Wang X (2008) Differential evolution with dynamic stochastic selection for constrained optimization. Inf Sci (NY) 178(15):3043–3074. https://doi.org/10.1016/j.ins.2008.02.014
155. Kai Feng Z, Jing Niu W, Liu S (2021) Cooperation search algorithm: a novel metaheuristic evolutionary intelligence algorithm for numerical optimization and engineering optimization problems. Appl Soft Comput 98:106734. https://doi.org/10.1016/j.asoc.2020.106734
156. Ray T, Liew KM (2003) Society and civilization: an optimization algorithm based on the simulation of social behavior. IEEE Trans Evol Comput 7(4):386–396. https://doi.org/10.1109/TEVC.2003.814902
157. Coello Coello CA (2000) Use of a self-adaptive penalty approach for engineering optimization problems. Comput Ind 41(2):113–127. https://doi.org/10.1016/S0166-3615(99)00046-9
158. Dimopoulos GG (2007) Mixed-variable engineering optimization based on evolutionary and social metaphors. Comput Methods Appl Mech Eng 196(4–6):803–817. https://doi.org/10.1016/j.cma.2006.06.010
159. He Q, Wang L (2007) An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng Appl Artif Intell 20(1):89–99. https://doi.org/10.1016/j.engappai.2006.03.003
160. Gandomi AH, Yang XS, Alavi AH (2011) Mixed variable structural optimization using Firefly Algorithm. Comput Struct 89(23–24):2325–2336. https://doi.org/10.1016/j.compstruc.2011.08.002
161. Pelusi D, Mascella R, Tallini L, Nayak J, Naik B, Deng Y (2020) An improved moth-flame optimization algorithm with hybrid search phase. Knowl Based Syst. https://doi.org/10.1016/j.knosys.2019.105277
162. Mirjalili S (2015) The ant lion optimizer, vol. 83. Elsevier Ltd.
163. Gandomi AH (2014) Interior search algorithm (ISA): a novel approach for global optimization. ISA Trans 53(4):1168–1183. https://doi.org/10.1016/j.isatra.2014.03.018
164. Sandgren E (1990) Nonlinear integer and discrete programming in mechanical design optimization. J Mech Des Trans ASME 112(2):223–229. https://doi.org/10.1115/1.2912596
165. Kannan (1994) An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. J Mech Design Mech Des 1(June 1994): 405–411
166. Fu JF, Fenton RG, Cleghorn WL (1991) A mixed integer-discrete-continuous programming method and its application to engineering design optimization. Eng Optim 17(4):263–280. https://doi.org/10.1080/03052159108941075
167. Zhang C, Wang HP (1993) Mixed-discrete nonlinear optimization with simulated annealing. Eng Optim 21(4):277–291. https://doi.org/10.1080/03052159308940980
168. Cao YJ, Wu QH (1997) Mechanical design optimization by mixed-variable evolutionary programming. In: Proc. IEEE Conf. Evol. Comput. ICEC, pp. 443–446
169. Litinetski VV, Abramzon BM (1998) Mars—a multistart adaptive random search method for global constrained optimization in engineering applications. Eng Optim 30(2):125–154. https://doi.org/10.1080/03052159808941241
170. Yun Y (2005) Study on adaptive hybrid genetic algorithm and its applications to engineering design problems study on adaptive hybrid genetic algorithm and its applications to engineering design problems
171. Chlckermane H, Gea HC (1996) Structural optimization using a new local approximation method. Int J Numer Methods Eng. https://doi.org/10.1002/(sici)1097-0207(19960315)39:5%3c829::aid-nme884%3e3.0.co;2-u

172. Gandomi AH, Yang X-S, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 29(1):17–35. https://doi.org/10.1007/s00366-011-0241-y
173. Cheng MY, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. Comput Struct 139:98–112. https://doi.org/10.1016/j.compstruc.2014.03.007
174. Eskandar H, Sadollah A, Bahreininejad A, Hamdi M (2012) Water cycle algorithm—a novel metaheuristic optimization method for solving constrained engineering optimization problems, vol. 110–111. Elsevier Ltd
175. Savsani P, Savsani V (2016) Passing vehicle search (PVS): a novel metaheuristic algorithm. Appl Math Model 40(5–6):3951–3978. https://doi.org/10.1016/j.apm.2015.10.040
176. Chen H, Xu Y, Wang M, Zhao X (2019) A balanced whale optimization algorithm for constrained engineering design problems. Appl Math Model 71:45–59. https://doi.org/10.1016/j.apm.2019.02.004
177. Wang GG (2003) Adaptive response surface method using inherited Latin hypercube design points. J Mech Des Trans ASME 125(2):210–220. https://doi.org/10.1115/1.1561044
178. Wang M, Heidari AA, Chen M, Chen H, Zhao X, Cai X (2020) Exploratory differential ant lion-based optimization. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2020.113548
179. Abderazek H, Ferhat D, Ivana A (2016) Adaptive mixed differential evolution algorithm for bi-objective tooth profile spur gear optimization. Int J Adv Manuf Technol. https://doi.org/10.1007/s00170-016-9523-2
180. Wang Z, Luo Q, Zhou Y (2020) Hybrid metaheuristic algorithm using butterfly and flower pollination base on mutualism mechanism for global optimization problems, no. 0123456789. Springer, London

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.