A boosted chimp optimizer for numerical and engineering design optimization challenges

Ch. Leela Kumari1 · Vikram Kumar Kamboj1,2 · S. K. Bath3 · Suman Lata Tripathi1 · Megha Khatri1 · Shivani Sehgal1,4

Received: 6 February 2021 / Accepted: 12 November 2021 / Published online: 24 March 2022
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 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
mathematical formulation of search functions was designed
to offer good competitiveness to current existing meta-heu-
ristics optimizers. The intention to design this optimization
technique is motivated by individual intelligence and sen-
sual 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 algo-
rithm called Spotted Hyena Optimizer (SHO) 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 litera-
ture review on related algorithms in Sect. 2, and concepts
of improved chimp optimizer (ICHIMP) algorithm are dis-
cussed 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 simulta-
neously 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 Evolu-
tion (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 opti-
mization 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-heu-
ristics 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 com-
binatorial 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 programings
were performed on real-world problems. Harmony search
(HS) [35] is inspired using the music production cycle anal-
ogy. 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 trans-
verse 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 gravitation 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 (MOSH) [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. Bernstein-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. Pndhan | 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 |
| 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 |
| 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 Hatamlo | 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 stabilities 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 chimpanzees. 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 improvised 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

\[
\vec{D} = |\vec{C}Y_{\text{prey}}(\text{iteration}) - \xi \cdot \vec{Y}_{\text{chimp}}(\text{iteration})| 
\]

(1)

\[
\vec{Y}_{\text{chimp}}(\text{iteration} + 1) = \tilde{Q}_{\xi} + \tilde{Y}_{\text{prey}}(\text{iteration}) - \tilde{A} \cdot \vec{D}. 
\]

(2)

Here, \(\tilde{A}, \xi, \) 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 \(\tilde{A}, \xi, \) 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:
\[ \tilde{Y}_{\text{Chimp}}(\text{iteration} + 1) = \begin{cases} \tilde{Y}_{\text{Prey}}(\text{iteration}) - \tilde{A} \cdot \tilde{D} & \text{if } \xi > 0.5 \\ \text{Chaotic value} & \text{if } \xi < 0.5 \end{cases} \] (2-i)

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

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

where SAN represents the search agent number;

\[ \tilde{A} = 2\eta v_1 - \eta \] (3)

\[ \tilde{C} = 2v_2 \] (4)

\[ \xi = \text{chaotic vector} \] (5)

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

\[ \tilde{A} \] Non-linearly decreases from 2.5 to 0 in both the phases iteratively. The vectors \( v_1 \) and \( v_2 \) are ranged \([0, 1] \). \( \xi \) 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)

\[ \tilde{D}_{\text{Attacker}} = \text{abs} \left[ \tilde{C}_1 \tilde{Y}_{\text{Attacker}} - \tilde{Y} \right]. \] (7a)

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

\[ \tilde{D}_{\text{Attacker}} = \begin{cases} |	ilde{C} \tilde{Y}_{\text{Attacker}}(\text{iteration}) - \xi \tilde{Y}(\text{iteration})| : |A| < 1 \\ |	ilde{C} \tilde{Y}_{\text{Attacker}}(\text{ran}(1), \text{iteration}) - \xi \tilde{Y}(\text{ran}(3), \text{iteration})| : |A| > 1 \end{cases} \] (7.a-i)

\[ \tilde{D}_{\text{Barrier}} = \text{abs} \left[ \tilde{C}_2 \tilde{Y}_{\text{Barrier}} - \tilde{Y} \right]. \] (7b)

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

\[ \tilde{D}_{\text{Barrier}} = \begin{cases} |	ilde{C} \tilde{Y}_{\text{Barrier}}(\text{iteration}) - \xi \tilde{Y}(\text{iteration})| : |A| < 1 \\ |	ilde{C} \tilde{Y}_{\text{Barrier}}(\text{ran}(1), \text{iteration}) - \xi \tilde{Y}(\text{ran}(3), \text{iteration})| : |A| > 1 \end{cases} \] (7.b-i)

\[ \tilde{D}_{\text{Chaser}} = \text{abs} \left[ \tilde{C}_3 \tilde{Y}_{\text{Chaser}} - \tilde{Y} \right]. \] (7c)

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

\[ \tilde{D}_{\text{Chaser}} = \begin{cases} |	ilde{C} \tilde{Y}_{\text{Chaser}}(\text{iteration}) - \xi \tilde{Y}(\text{iteration})| : |A| < 1 \\ |	ilde{C} \tilde{Y}_{\text{Chaser}}(\text{ran}(1), \text{iteration}) - \xi \tilde{Y}(\text{ran}(3), \text{iteration})| : |A| > 1 \end{cases} \] (7.c-i)

\[ \tilde{D}_{\text{Driver}} = \text{abs} \left[ \tilde{C}_4 \tilde{Y}_{\text{Driver}} - \tilde{Y} \right]. \] (7d)

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

\[ \tilde{D}_{\text{Driver}} = \begin{cases} |	ilde{C} \tilde{Y}_{\text{Driver}}(\text{iteration}) - \xi \tilde{Y}(\text{iteration})| : |A| < 1 \\ |	ilde{C} \tilde{Y}_{\text{Driver}}(\text{ran}(1), \text{iteration}) - \xi \tilde{Y}(\text{ran}(3), \text{iteration})| : |A| > 1 \end{cases} \] (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

\[ \tilde{Y}_1 = \tilde{Y}_{\text{Attacker}} - \tilde{A}_1 \tilde{D}_{\text{Attacker}} \] (8a)

\[ \tilde{Y}_2 = \tilde{Y}_{\text{Barrier}} - \tilde{A}_2 \tilde{D}_{\text{Barrier}} \] (8b)

\[ \tilde{Y}_3 = \tilde{Y}_{\text{Chaser}} - \tilde{A}_3 \tilde{D}_{\text{Chaser}} \] (8c)

\[ \tilde{Y}_4 = \tilde{Y}_{\text{Driver}} - \tilde{A}_4 \tilde{D}_{\text{Driver}}. \] (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)

\[ \tilde{Y}(\text{iteration} + 1) = \frac{\tilde{Y}_1 + \tilde{Y}_2 + \tilde{Y}_3 + \tilde{Y}_4}{4}. \] (9)
To generate the initial arbitrary position of search agents, the below mathematical equation can be adopted

\[ \vec{Y}_{\text{rand}} = LB_i + \xi \times (UB_i - LB_i) \; ; \; i \in 1, 2, 3, ..., \text{Dim} \quad \text{(10)} \]

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

(a) 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;
```

(b) 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_Attacker=abs (C2*Y (rand_num (4), j)-Y (rand_num (1), j)
If rand>CR
D_Attacker=Y_Barrier(j)
end
end
Y2=Y_Barrier(j)-A2*D_Barrier;
```

(c) PSEUDO Code for Calculation of Y3

```
r1=rand ();
r2=rand ();
A3=2*a*r1-a;
C3=2*r2;
if abs (A3)<1
D_Attacker=abs (C3*Y_Attacker (j)-Y (i, j));
else
D_Attacker=abs (C3*Y (rand_num (3), j)-Y (rand_num (2),j)
If rand>CR
D_Attacker=Y_Attacker(j)
end
end
Y3=Y_Attacker(j)-A3*D_Attacker;
```

(d) PSEUDO Code for Calculation of Y4

```
r1=rand ();
r2=rand ();
A4=2*a*r1-a;
C4=2*r2;
if abs (A4)<1
D_Attacker=abs (C4*Y_Driver (j)-Y (i, j));
else
D_Attacker=abs (C4*Y (rand_num (3), j)-Y (rand_num (4), j)
If rand>CR
D_Attacker=Y_Driver(j)
end
end
Y4=Y_Driver(j)-A4*D_Driver;
```

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

\[
\vec{d}_h = |\vec{y} - \vec{Q}(s) - \vec{Q}(s)| \quad (11)
\]

\[
\vec{Q}(s + 1) = \vec{Q}(s) - \vec{z} \vec{d}_h, \quad (12)
\]

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

\[
\vec{y} = 2\vec{r}_1 \quad (13)
\]

\[
\vec{z} = 2\vec{H}\vec{r}_2 - \vec{H} \quad (14)
\]

---

**PSEUDO code of Chimp Algorithm**

**Algorithm 1: Chimp**

1. Initialize the Chimp population \(x_i = 1, 2, \ldots n\)
2. Initialize \(\eta, \xi, \hat{A}, \) and \(\hat{C}\)
3. Calculate the position of each chimp
4. Divide chims randomly into independent groups
5. Until stopping condition is satisfied
6. Calculate the fitness of each chimp
7. \(X_{Attacker}\) = the best search agent
8. \(X_{Chaser}\) = the second best search agent
9. \(X_{Barrier}\) = the third best search agent
10. \(X_{Driver}\) = the fourth best search agent

while \(t < \text{maximum number of iterations}\)

for each chimp:

1. Extract the chimp’s group
2. Use its group strategy to update \(\eta, \xi, \hat{A}, \) and \(\hat{C}\)
3. Use \(\eta, \xi, \hat{A}\) and \(\hat{C}\) to calculate \(\hat{A}\) and then \(\hat{D}\)
4. Calculate \(Y_1\) and \(Y_2\) using Pseudo Code of Fig.2 (a)
5. Calculate \(Y_3\) and \(Y_4\) using Pseudo Code of Fig.2 (b)
6. Update \(\eta, \xi, \hat{A}, \) and \(\hat{C}\)

\(X_{Attacker}, X_{Driver}, X_{Barrier}, X_{Chaser}\)

\(I = I + 1\)

end while

return \(X_{Attacker}\)
\[ H = 5 - (\text{Itr} \times (5/\text{Maxitr})) \]

where Itr = 1, 2, 3, ..., Maxitr.

Here, \( H \) 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

\[ \vec{d}_h = |\vec{y} - \vec{Q}_h - \vec{Q}_k| \]

(16)

\[ \vec{Q}_k = \vec{Q}_h - \vec{z} \vec{d}_h \]

(17)

\[ \vec{C}_h = \vec{Q}_k + \vec{Q}_{k+1} + \cdots + \vec{Q}_{k+N} \]

(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

\[ N = \text{count}_{\vec{Q}_h, \vec{Q}_{h+1}, \vec{Q}_{h+2}, \cdots (\vec{Q}_h + \vec{M})} \]

(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

\[ \vec{Q}(s + 1) = \frac{\vec{C}_h}{N} \]

(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.
Flow Chart of proposed ICHIMP-SHO Algorithm

START

Examine input data for every objective function i.e. Dim and Range

Log in the input factors for Chimp Algorithm i.e. search agent number, Dim, Attacker, Barrier, Chaser, Driver position and Score etc.

Place iteration counter = 0

Calculate the fitness value of each objective function while satisfying lower and upper boundaries of search space

YES

YES

YES

YES

IF fitness>AScore & fitness>BScore & fitness>CScore & fitness>DScore

IF fitness>AScore & fitness>BScore & fitness<CScore

IF fitness<AScore

Stop

Calculate A1 & C1 using equation (8.a) and (7.a) respectively

Update DAttacker and Y1 using equations (6.a) and (7.a) respectively

Calculate A2 & C2 using equation (7.b) and (6.b) respectively

Evaluate the new fitness value and compare it with previously obtained fitness to select the best one out of these two

Set AScore = fitness and Attacker position = Best position of search agent

Set Barrier score = fitness and Barrier position = Best position of search agent

Update DBarrier and Y2 using equations (7.b) & (8.b)

Calculate A3 & C3 using equations (8.c) and (7.c) respectively

Update DChaser, Y3, DDriver, Y4 using equations (7.c), (8.c), (7.d), (8.d) respectively

Evaluate mean position, Y=(Y1+Y2+Y3+Y4)/4 and also update the mean position of Chimps

Evaluate the new fitness value and compare it with previously obtained fitness to select the best one out of these two

Modify the locations of search agents by hunting strategy equations (1) & (20) respectively

Calculate A4 & C4 using equation (7.d) and (6.d) respectively

Update DChaser and Y4 using equation (6.a) and (7.a) respectively

Reduce ξ value from 2.5 to 0 non-linearly

Calculate A5 & C5 using equation (8.a) and (7.a) respectively

NO

IF fitness<AScore & fitness<BScore & fitness>CScore

IF fitness<AScore & fitness<BScore

IF fitness<AScore & fitness<BScore & fitness>CScore

IF fitness<AScore & fitness<BScore & fitness>CScore & fitness<DScore

Set AChimp = fitness and Chaser position = Best position of search agent

Update DChancer and Y1 using equation (6.a) and (7.a) respectively

Place Chaser score = fitness and Chaser position = Best position of search agent

Place Barrier score = fitness and Barrier position = Best position of search agent

Calculate AChimp score & fitness using equation (8.a) & fitness<DScore & fitness<DScore & fitness<DScore & fitness<DScore

NO

Calculate AChimp score & fitness using equation (8.a) & fitness<DScore & fitness<DScore & fitness<DScore & fitness<DScore

Update DChancer and Y1 using equation (6.a) and (7.a) respectively

Modify the locations of search agents by hunting strategy equations (1) & (21) respectively

Store Optimize fitness=AScore

Best position=Attacker position

Place iteration counter = iteration+1

Initialize the arbitrary position of every search agent using equation (10)

Store Optimize fitness=AScore

Best position=Attacker position

NO

NO

NO

NO

NO

NO

STOP

Fig. 3 (continued)
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

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

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
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 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

---

PSEUDO code of Improved Chimp-SHO Algorithm

Algorithm 3: Imp-Chimp-SHO

1. Initialize the Chimp population \( x_{i+1} \) (i=1, 2... n)
2. Initialize \( \eta, \xi, \tilde{A} \) and \( \tilde{C} \)
3. Calculate the position of each chimp
4. Divide chims randomly into independent groups
5. Until stopping condition is satisfied
6. Calculate the fitness of each chimp
7. \( X_{\text{Attacker}} = \) the best search agent
8. \( X_{\text{Chaser}} = \) the second best search agent
9. \( X_{\text{Barrier}} = \) the third best search agent
10. \( X_{\text{Driver}} = \) the fourth best search agent
11. While (I < maximum number of iterations)
12. For each chimp:
   a. Extract the chimp’s group
   b. Use its group strategy to update \( \eta, \xi, \tilde{A} \) and \( \tilde{C} \)
   c. Use \( \eta, \xi, \tilde{A} \) and \( \tilde{C} \) to calculate \( \tilde{A} \) and then \( \tilde{D} \)
13. Calculate \( Y_1 \) and \( Y_2 \) using Pseudo Code of Fig. 2(a)
14. Calculate \( Y_3 \) and \( Y_4 \) using Pseudo Code of Fig. 2(b)
15. Calculate \( Y = (Y_1+Y_2+Y_3+Y_4)/4 \)
16. Update Y further using SHO algorithm
17. Refer to pseudo code of SHO from algorithm 2
18. Update \( \eta, \xi, \tilde{A} \) and \( \tilde{C} \)
19. Update \( X_{\text{Attacker}}, X_{\text{Driver}}, X_{\text{Barrier}}, X_{\text{Chaser}} \)
20. \( I = I+1 \)
21. End while
22. Return \( X_{\text{Attacker}} \)
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 ($f_{\text{min}}$). 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 core i3 processor laptop with a 7th generation CPU and 8GB RAM.

| Table 2 | Uni-modal (UM) standard benchmark functions |
|---------|---------------------------------------------|
| Functions | Dimensions | Range | $f_{\text{min}}$ |
| $F_1(U) = \sum_{m=1}^{2} U_m^2$ | 30 | $[-100, 100]$ | 0 |
| $F_2(U) = \sum_{m=1}^{2} |U_m| + \prod_{m=1}^{2} |U_m|$ | 30 | $[-10, 10]$ | 0 |
| $F_3(U) = \sum_{m=1}^{2} (\sum_{n=1}^{2} U_n)^2$ | 30 | $[-100, 100]$ | 0 |
| $F_4(U) = \max_{m}\{U_m, 1 \leq m \leq 2\}$ | 30 | $[-100, 100]$ | 0 |
| $F_5(U) = \sum_{m=1}^{2} [100(U_m - U_m)^2 + (U_m - 1)^2]$ | 30 | $[-38, 38]$ | 0 |
| $F_6(U) = \sum_{m=1}^{2} (U_m + 0.5)^2$ | 30 | $[-100, 100]$ | 0 |
| $F_7(U) = \sum_{m=1}^{2} mU_m^3 + \text{random}(0, 1)$ | 30 | $[-1.28, 1.28]$ | 0 |

| Table 3 | Multi-modal (MM) standard functions |
|---------|---------------------------------------------|
| Functions | Dim | Range | $f_{\text{min}}$ |
| $F_8(U) = \sum_{m=1}^{2} -U_m \sin(\sqrt{|U_m|})$ | 30 | $[-500, 500]$ | 418.98295 |
| $F_9(U) = \sum_{m=1}^{2} [U_m^2 - 10 \cos(2\pi U_m) + 10]$ | 30 | $[-5.12, 5.12]$ | 0 |
| $F_{10}(U) = -20 \exp(0.2 \sqrt{\left(\frac{1}{5} \sum_{m=1}^{2} U_m^2\right)} + \exp(\frac{1}{5} \sum_{m=1}^{2} \cos(2\pi U_m) + 20 + d)$ | 30 | $[-32, 32]$ | 0 |
| $F_{11}(U) = 1 + \sum_{m=1}^{2} \frac{U_m^2}{\sinh^2(U_m)} - \prod_{m=1}^{2} \cos(\frac{U_m}{\sinh(U_m)})$ | 30 | $[-600, 600]$ | 0 |
| $F_{12}(U) = \frac{x}{z} \left\{ 10 \sin(x \tau_1) + \sum_{m=1}^{2} (\tau_m - 1)^2 [1 + 10 \sin^2(x \tau_{m+1})] + (\tau_x - 1)^2 \right\}$ | 30 | $[-50, 50]$ | 0 |
| + $\sum_{m=1}^{2} g(U_m, 10, 100, 4)$ | | |
| Where, $\tau_m = 1 + \frac{\sqrt{x^2 + 1}}{4} \left\{ \right.$ | | |
| $\left\{ \begin{array}{ll} \sqrt{x^2 + 1} & (U_m - b)U_m \geq b \\
| \end{array} \right.$ | \fi
| $x(U_m - b)U_m < b$ | $0$ | $x(U_m - b)U_m < b$ | $b$ |
| $F_{13}(U) = 0.1 \left\{ \sin^2(3\pi U_m) + \sum_{m=1}^{2} (U_m - 1)^2 [1 + \sin^2(3\pi U_m + 1)] + (x_1 - 1)^2 [1 + \sin^2] \right\}$ | 30 | $[-50, 50]$ | 0 |
Table 4 Fixed-Dimension (FD) standard functions

| Fixed-modal (FD) (F14–F23) standard benchmark functions | Dimension | Range | f_{\text{min}} |
|----------------------------------------------------------|-----------|-------|---------------|
| F_{14}(U) = \left[\frac{1}{500} + \sum_{m=1}^{2} \frac{1}{20000} \sum_{n=1}^{2} (U_m - q_{mn})^2\right]^{-1} | 2        | [-65.536, 65.536] | 1              |
| F_{15}(U) = \sum_{m=1}^{11} \left[ b_m - \frac{1}{U_m + q_{m1} + q_{m2}} \right]^2 | 4        | [-5, 5] | 0.00030       |
| F_{16}(U) = 4U_1^2 - 2.1U_1^2 + \frac{1}{5}U_1^2 + U_1U_2 - 4U_2^2 + 4U_2^4 | 2        | [-5, 5] | -1.0316       |
| F_{17}(U) = (U_2 - \frac{5.1}{4\pi} U_1^2 + \frac{1}{4\pi} U_1 - 6)^2 + 10(1 - \frac{1}{\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)]^2 | 2        | [-2, 2] | 3             |
| x[30 + (2U_1 - 3U_2)^2x(18 - 32U_1 + 12U_1^2 + 48U_2 - 36U_1U_2 + 27U_2^2)] | 7        | [1, 3] | -3.32         |
| F_{19}(U) = -\sum_{m=1}^{3} d_m \exp(-\sum_{m=1}^{3} U_m(U_m - q_{mn})^2) | 3        | [0, 1] | -3.32         |
| F_{20}(U) = -\sum_{m=1}^{4} d_m \exp(-\sum_{m=1}^{4} U_m(U_m - q_{mn})^2) | 6        | [0, 10] | -10.1532      |
| F_{21}(U) = -\sum_{m=1}^{4} [U - b_m(U - b_m)^2 + d_m]^{-1} | 4        | [0, 10] | -10.4028      |
| F_{22}(U) = -\sum_{m=1}^{4} [(U - b_m)(U - b_m)^2 + d_m]^{-1} | 4        | [0, 10] | -10.5363      |
| F_{23}(U) = -\sum_{m=1}^{4} [(U - b_m)(U - b_m)^2 + 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 P value | t test P value | h value |
|----------|------|---------------|--------------------|---------------------|-------|-------------------------------|---------------|--------|
| F1       | 3.91443E–28 | 1.07214E–27 | 2.2954E–27 | 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 | 1.38683E–05 | 1.3094E–07 | 1.7344E–06 | 0.077611649 | 0 |
| F4       | 3.64228E–08 | 3.08114E–08 | 2.77594E–09 | 1.26518E–07 | 3.2531E–08 | 1.7344E–06 | 4.36866E–07 | 1 |
| F5       | 28.38318365 | 0.671703255 | 26.23392716 | 28.89070125 | 28.65306199 | 1.7344E–06 | 6.30357E–49 | 1 |
| F6       | 1.481698857 | 0.405139111 | 0.742524222 | 2.257134021 | 1.732292069 | 1.7344E–06 | 1.57295E–18 | 1 |
| F7       | 0.001127419 | 0.00056314 | 0.00275088 | 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 Chimpanzee-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.
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 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     |

### Table 8 Evaluation for (F1–F7) problems

| Algorithm | Parameters | (F1–F7) Uni-modal benchmark functions |
|-----------|------------|---------------------------------------|
| Lightning Search algorithm (LSA) [62] | Mean: 3.340000000 | F1: 0.024079674 F2: 0.000046 F3: 54.865255 F4: 0.036806544 F5: 1.156232033 F6: 8.2692958 F7: 64.28160301 |
| Battle Royale Optimization algorithm (BRO) [133] | Avg: 54.865255 St.Deviation: 29.92194448 | F1: 0.518757 F2: 99.936848 F3: 2.8731E-08 F4: 0.000368 F5: 0.000094 |
| Opposition based enhanced grey wolf optimization algorithm (OEGWO) [134] | Avg: 1.01 × 10^-1 St.Deviation: 2.72 × 10^1 | F1: 1.90 × 10^-5 F2: 7.27 × 10^1 F3: 1.40 × 10^0 F4: 3.63 × 10^-4 F5: 0.000368 |
| Photon Search Algorithm (PSA) [59] | Mean: 3978.0837 St.Deviation: 705.1589 | F1: 332.6410 F2: 19.8667 F3: 0.0237 F4: 0.000115 F5: 0.000158 |
| Hybrid Harris Hawks Optimizer-Pattern Search algorithm (hHHO-PS) [74] | Avg: 4.90 × 10^-25 St.Deviation: 4.63 × 10^-25 | F1: 1.1947 F2: 3.27 × 10^-1 F3: 3.68 × 10^-1 F4: 3.61 × 10^-14 F5: 0.001193 |
| Spotted Hyena Optimizer (SHO) [70] | Mean: 0.000046 St.Deviation: 5.53E-01 | F1: 332.6410 F2: 19.8667 F3: 0.0237 F4: 0.000115 F5: 0.000158 |
| Harris Hawks Optimizer (HHO) [100] | Mean: 6.92 × 10^-51 St.Deviation: 6.20 × 10^-54 | F1: 332.6410 F2: 19.8667 F3: 0.0237 F4: 0.000115 F5: 0.000158 |
| Enhanced Crow search algorithm (ECSA) [135] | Mean: 5.03 × 10^-20 St.Deviation: 2.18 × 10^-9 | F1: 332.6410 F2: 19.8667 F3: 0.0237 F4: 0.000115 F5: 0.000158 |
| Transient Search Optimization (TSO) [136] | Mean: 1.25 × 10^-80 St.Deviation: 1.70 × 10^-47 | F1: 332.6410 F2: 19.8667 F3: 0.0237 F4: 0.000115 F5: 0.000158 |
| ICHIMP-SHO (Proposed algorithm) | Mean: 3.91443E-28 St.Deviation: 6.82 × 10^-3 | F1: 332.6410 F2: 19.8667 F3: 0.0237 F4: 0.000115 F5: 0.000158 |
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 algorithms.

Hence, the test results for UM, MM, and FD benchmarks problems are tabulated in Tables 6, 7, 8, 9, 10, 11, 12, 13, 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. 4 3D view of uni-modal (UM) standard benchmark problems
Comparative Analysis of ICHIMP algorithm for F1

Comparative Analysis of ICHIMP algorithm for F2

Comparative Analysis of ICHIMP algorithm for F3

Comparative Analysis of ICHIMP algorithm for F4

Comparative Analysis of ICHIMP algorithm for F5

Comparative Analysis of Imp-Chimp algorithm for F6

Comparative Analysis of ICHIMP algorithm for F7

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
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 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 | 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           | 2.27374E–13        | 5.68434E–14         | 2.89814E–06 | 4.53821E–09            | 1.13726E–23 | 1      |
| F10      | 9.52719E–14           | 1.69917E–14           | 1.35891E–13        | 9.50351E–14         | 1.67736E–06 | 1.13726E–23            | 2.80855E–05 | 1      |
| F11      | 0.001725278           | 0.0045322460          | 0.01541836         | 0                   | 0.125        | 0.045981511            | 1.49706E–24 | 1      |
| F12      | 0.088180059           | 0.0973093990.011803437| 0.567716636        | 0.074592218         | 1.7344E–06  | 2.85762E–26            | 1.13726E–23 | 1      |
| F13      | 1.911006715           | 0.3171962581.325094518| 2.527069025        | 1.86611264          | 1.7344E–06  | 2.85762E–26            | 1.13726E–23 | 1      |

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

| Function | Best time | Average time | Worst time |
|----------|-----------|--------------|------------|
| F8       | 1.359375  | 1.510416677  | 1.734375   |
| F9       | 1.328125  | 1.479166677  | 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   |

### Table 11 Comparison for multi-modal benchmark functions

| Algorithm | Factors | (F8–F13) multi-modal benchmark functions |
|-----------|---------|-----------------------------------------|
| Lightning search algorithm (LSA) [62] | Avg | 62.7618960 | 1.077446947 | 0.397887358 | 2.686199985 | 0.007241875 |
| Battle Royale Optimization algorithm (BRO) [133] | Mean | 48.275350  | 0.350724    | 0.001373    | 0.369497    | 0.000004    |
| Opposition based enhanced grey wolf optimization algorithm (OEGWO) [134] | Avg | 4.84E+10   | 9.41E+15    | 5.70E+13    | 9.36E+02    | 1.24E+00    |
| Photon search algorithm (PSA) [59] | Mean | 7.7367     | 1.6766      | 0.5294      | 0.1716      | 1.5458       |
| Hybrid Harris Hawks Optimizer-Pattern Search algorithm (hHHO-PS) [74] | Avg | 0.00E+00   | 8.88E+06    | 0.00E+00    | 2.94E+15    | 1.16E+13     |
| Spotted Hyena Optimizer (SHO) [70] | Mean | 1.16E+03   | 0.00E+00    | 2.48E+00    | 0.00E+00    | 3.68E+02    | 9.29E+1  |
| Harris Hawks Optimizer (HHO) [100] | Mean | 1.256138   | 0.00E+00    | 8.88E+16    | 0.00E+00    | 8.92E+06    | 0.000101   |
| Enhanced Crow search algorithm (ECSA) [135] | Mean | 2332.3867  | 0.00E+00    | 8.88E+16    | 0.00E+00    | 1.16E+10    | 9.52E+2   |
| Transient Search Optimization (TSO) [136] | Mean | 12.569.5   | 0.00E+00    | 0.00E+00    | 0.00E+00    | 0.2849633   | 0.19901675 |
| ICHIMP−SHO (proposed algorithm) | Mean | 5231.965502 | 9.52719E-14 | 1.69917E-14 | 0.001725278 | 0.088180059 | 1.911006715 |

---
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 $\Phi$, and cylindrical unit length ($L$) which are taken into account.

This vessel has end caps on either sides, and the structure’s
Comparative Analysis of ICHIMP algorithm for F8

Comparative Analysis of ICHIMP algorithm for F9

Comparative Analysis of ICHIMP algorithm for F10

Comparative Analysis of ICHIMP algorithm for F11

Comparative Analysis of ICHIMP algorithm for F12

Comparative Analysis of ICHIMP algorithm for F13

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

$$\vec{q} = [q_1, q_2, q_3, q_4] = [T_sT_hRL]$$.

To minimize

$$f(\vec{q}) = 0.6224q_1q_3 + 1.7781q_2q_3^2 + 3.1661q_1^2q_4 + 19.84q_1^2q_3^2.$$

Here

$$g_1(\vec{q}) = -q_1 + 0.0193q_3 \leq 0$$

(23a)

$$g_2(\vec{q}) = q_3 + 0.00954q_3 \leq 0$$

(23b)
Variable range, $0 \leq q_1 \leq 99$

Speed reducer

As illustrated in Fig. 14 [110], this type of design issue has seven variables. It is made up of the face width $\times 1$, the teeth module $\times 2$, the pinion teeth number $\times 3$, the first shaft length bearings $\times 4$, the second shaft length bearings $\times 5$, the first shaft diameter $\times 6$, and the second shaft diameter $\times 7$. 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.

Minimizing

Subjected to

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

$0 \leq q_2 \leq 99$

$10 \leq q_3 \leq 200$

$10 \leq q_4 \leq 20$.

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 $\times 1$, the
Table 14 Comparison for fixed-dimension benchmark functions

| Algorithm | Parameters | (F14–F23) fixed-dimension benchmark functions |
|-----------|------------|-----------------------------------------------|
| Lightning search algorithm (LSA) [62] | Mean: 0.358172550, St. Deviation: 0.743960008 | F14: 0.000534843, F15: 1.031628453, F16: 3.000000000, F17: 3.862782148, F18: -3.272060061, F19: -7.027319823, F20: -7.136702131, F21: -7.910438367 |
| Enhanced Crow search algorithm (ECSA) [135] | Mean: 1.000269, St. Deviation: 2.62E−03 | F14: 0.000327, F15: -1.03161, F16: 0.397993, F17: 3.00003, F18: -3.86061, F19: -3.32066, F20: 1.79E−03, F21: 8.75374E−05 |
| Transient Search Optimization (TSO) [136] | Mean: 9.68E+000, St. Deviation: 3.29E+000 | F14: 9.01 × 10−4, F15: 1.06 × 10−4, F16: 3.97 × 10−1, F17: 3.00E+000, F18: -3.75E+000, F19: -3.01, F20: 7.42 × 10−1−3, F21: 2.63 × 10−2 |
| Photon Search Algorithm (PSA) [59] | Mean: 0.4802, St. Deviation: 0.1158 | F14: 0.0077, F15: -1.036, F16: 0.3979, F17: 3, F18: -3.8556, F19: -3.043, F20: 9.7302, F21: 9.8628 |
| Hybrid Harris Hawks Optimizer-Pattern Search algorithm (hHHO-PS) [74] | Mean: 0.998004, St. Deviation: 1.57 × 10−16 | F14: 0.00307, F15: -1.03163, F16: 0.397887, F17: 3, F18: -3.86278, F19: -3.322, F20: 10.1532, F21: 10.4029 |
| Spotted Hyena Optimizer (SHO) [70] | Mean: 1.130, St. Deviation: 0.5659 | F14: 2.70 × 10−3, F15: 1.0316, F16: 0.398, F17: 3.00, F18: -3.89, F19: -1.44E+000, F20: 7.47 × 10−12, F21: 7.69 × 10−15 |
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.

Consider \( \vec{x} = [x_1, x_2] = [A_1, A_2] \) \hspace{1cm} (25)

Minimize \( f(\vec{x}) = (2\sqrt{2}x_1 + x_2) \ast l \) \hspace{1cm} (26)

Subject to \( g_1(\vec{x}) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} - p - \sigma \leq 0 \) \hspace{1cm} (26a)

\( g_2(\vec{x}) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} - p - \sigma \leq 0 \) \hspace{1cm} (26b)

Here

\( 2.6 \leq x_1 \leq 3.6, 0.7 \leq x_2 \leq 0.8, 17 \leq x_1 \leq 28, \)

\( 7.3 \leq x_4 \leq 8.3, 7.8 \leq x_5 \leq 8.3, \)

\( 2.9 \leq x_6 \leq 3.9 \) and \( 5 \leq x_7 \leq 5.5. \)

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.
Fig. 10 3D view of fixed-dimension (FD) modal standard benchmark functions
Comparative Analysis of ICHIMP algorithm for F14

Comparative Analysis of ICHIMP algorithm for F15

Comparative Analysis of ICHIMP algorithm for F18

Comparative Analysis of ICHIMP algorithm for F19

Comparative Analysis of ICHIMP algorithm for F20

Comparative Analysis of ICHIMP algorithm for F21

Comparative Analysis of ICHIMP algorithm for F22

Comparative Analysis of ICHIMP algorithm for F23

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 benchmark functions
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

\[
\vec{z} = \left[ z_1z_2z_3z_4 \right] = \left[ hlhb \right]
\]

\[
f(\vec{z}) = 1.10471z_1^2z_2 + 0.04811z_3z_4(14.0 + z_2).
\]

By addressing

\[
g_1(\vec{z}) = \tau(\vec{z}) - \tau_{\text{maxi}} \leq 0,
\]

\[
g_2(\vec{z}) = \sigma(\vec{z}) - \sigma_{\text{maxi}} \leq 0
\]

\[
g_3(\vec{z}) = \delta(\vec{z}) - \delta_{\text{maxi}} \leq 0
\]

\[
g_4(\vec{z}) = z_1 - z_4 \leq 0
\]

(27)

\[
g_5(\vec{z}) = P_i - P_c(\vec{z}) \leq 0
\]

(28a)

\[
g_6(\vec{z}) = 0.125 - z_1 \leq 0
\]

(28b)

\[
g_7(\vec{z}) = 1.10471z_1^2 + 0.04811z_3z_4(14.0 + z_2) - 5.0 \leq 0.
\]

(28c)

\[
\tau(\vec{z}) = \sqrt{(\tau/)^2 + 2\tau/\tau//^2 + (\tau//)^2},
\]

(29a)

\[
\tau// = \frac{P_i}{\sqrt{2z_1z_2}}, \quad \tau// = \frac{MR}{J}, M = P_i \left( L + \frac{z_2}{2} \right),
\]

(29b)

\[
R = \sqrt{\frac{z_2^2}{4} + \left( \frac{z_1 + z_3}{2} \right)^2}
\]

(29c)

\[
J = 2 \left\{ \sqrt{2z_1z_2} \left[ \frac{z_2^2}{4} + \left( \frac{z_1 + z_3}{2} \right)^2 \right] \right\}
\]

(29d)

\[
\sigma(\vec{y}) = \frac{6P_jL}{z_4z_3}, \quad \delta(\vec{y}) = \frac{6P_jL^3}{Ez_4z_3}
\]

(29e)

\[
P_c(\vec{z}) = \frac{4.013Ez_4}{L^2} \left[ 1 - \frac{z_3}{2L} \sqrt{\frac{E}{4G}} \right].
\]

(29f)
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

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

### 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 ($DIME$), internal part diameter ($DIMI$), 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

Minimizing

$$ f(w) = 0.07075\pi(DIME_E^2 - DIMI_I^2)t; $$  

subjected to:

\begin{align*}
  b_1(w) &= G - \frac{4P\lambda_{\text{max}}}{(1 - \delta^2)aDIME_E} \left[ \delta(S_H - \frac{\lambda_{\text{max}}}{2}) + \mu t \right] \geq 0 \\
  b_2(w) &= \frac{4P\lambda_{\text{max}}}{(1 - \delta^2)aDIME_E} \left[ (S_H - \frac{\lambda_{\text{max}}}{2})(S_H - \lambda)t + \lambda^2 \right] \lambda_{\text{max}} - P_{\text{MAX}} \geq 0 \\
  b_3(w) &= \lambda_1 - \lambda_{\text{max}} \geq 0
\end{align*}

| Table 15 | Basic information of (SPECIAL1—SPECIAL11) engineering-based designs |
|---|---|
| Name of the function | Type of objective | No. of discrete variables | Count of constraint |
| Pressure Vessel | Minimize weight | 4 | 4 |
| Speed reducer | Minimize weight | 7 | 7 |
| Three-bar truss | Minimize weight | 4 | 4 |
| Welded Beam | Minimize weight | 7 | 7 |
| Gear train | Minimize gear ratio | 4 | 4 |
| Belleville spring | Minimize weight | 11 | 11 |


Table 16  ICHIMP-SHO results for engineering design issues

| Name of design          | Mean      | Standard deviation | Best      | Worst      | Median    |
|-------------------------|-----------|--------------------|-----------|------------|-----------|
| Pressure vessel         | 6060.428  | 268.201            | 5908.055  | 6990.7201  | 5963.8967 |
| Speed reducer problem   | 3012.077  | 4.32543            | 3003.6315 | 3020.8707  | 3012.2509 |
| Three-bar truss problem | 263.9036  | 0.06451            | 263.8970  | 263.9214   | 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           | 1.981765  | 2.031499   | 1.9928986 |
| Cantilever beam design  | 1.303427  | 0.000113           | 1.303295  | 1.3038244  | 1.3034144 |
| Rolling element bearing | 85150.7   | 88.71365           | 85383.949 | 84905.165  | 85152.953 |
| I-beam design           | 0.006626  | 3.62E−08           | 0.006626  | 0.006626   | 0.006626  |
| Spring design           | 0.012801  | 0.000117           | 0.0126915 | 0.006626   | 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              | 0.2033125 | 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

\[ b_4(w) = H - S_H - t \geq 0 \]  

\[ b_5(w) = DIM_{\text{MAX}} - DIM_E \geq 0 \]  

\[ b_6(w) = DIM_E - DIM_I \geq 0 \]  

\[ b_7(w) = 0.3 - \frac{S_H}{DIM_E - DIM_I} \geq 0 \]  

where

\[ a = \frac{6}{\pi \ln J} \left( \frac{J - 1}{\ln J} - 1 \right)^2 \]  

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

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

\[ P_{\text{MAX}} = 5400 \text{ lb.} \]  

\[ P = 306 \text{ psi}, \quad \lambda_{\text{MAX}} = 0.2 \text{ in}, \quad \delta = 0.3, \quad G = 200 \text{ Kpsi}, \]  

\[ H = 2 \text{ in}, \quad DIM_{\text{MAX}} = 12.01 \text{ in}, \quad J = \frac{DIM_{\text{MAX}}}{DIM_I}, \quad \lambda = f(a)a, \quad a = \frac{S_H}{t} \]
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 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 | 0.385096372 | 40.3556904385 | 199.5028789067 | 6059.714 |
| SMA [89]               | 0.7931 | 0.3932 | 40.6711 | 196.2178 | 5994.1837 |
| 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 | 42.098455 | 176.636595 | 6059.7143 |
| SiC-PSO [143]          | 0.8125 | 0.4375 | 42.098446 | 176.636596 | 6059.71435 |
| G-QPSO [144]           | 0.8125 | 0.4375 | 42.0984 | 176.6372 | 6059.7208 |
| NPGA [145]             | 0.8125 | 0.4375 | 42.097398 | 176.654047 | 6059.946341 |
| CDE [146]              | 0.8125 | 0.4375 | 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 | 6059.0898 |
| MVO [63]               | 0.8125 | 0.4375 | 42.0907382 | 176.738690 | 6060.8066 |
| SCA                    | 0.8177577 | 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 |

The following is the design formula:

Let us consider $L = [L_1, L_2, L_3, L_4]$

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

By addressing

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

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

---

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

**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 $L = [L_1, L_2, L_3, L_4]$

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

By addressing

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

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

---

**Fig. 14** Speed reducer design of engineering problem
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 (DIMp), (ii) ball diameter (DIMb), (iii) ball numbers (Nb), (iv) outer raceway curvature coefficient, and (v) inner raceway curvature coefficient. The following five variables (KDmin, KDmax, E, 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

| Comparative algorithms | Optimal values for variables | Optimum fitness |
|------------------------|-----------------------------|-----------------|
| 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 5.287819 | 3029.002 |
| GA [30]                | 3.510253 0.7 17 8.35 7.8 3.362201 5.287723 | 3067.561 |

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 (DIMp), (ii) ball diameter (DIMb), (iii) ball numbers (Nb), (iv) outer raceway curvature coefficient, and (v) inner raceway curvature coefficient. The following five variables (KDmin, KDmax, E, 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

| Comparative algorithms | Optimal values for variables | Optimum fitness |
|------------------------|-----------------------------|-----------------|
| 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 5.287819 | 3029.002 |
| GA [30]                | 3.510253 0.7 17 8.35 7.8 3.362201 5.287723 | 3067.561 |
bearing design problem. From Eqs. (35a) through (35c), the mathematical formulation for the tendered engineering design is shown.

For maximizing

$$C_D = f_c N^{2/3} \text{DIM}_B^{1.8}. \quad \text{(35a)}$$

If $\text{DIM} \leq 25.4$ mm

$$C_D = 3.647 f_c N^{2/3} \text{DIM}_B^{1.4}. \quad \text{(35b)}$$

If $\text{DIM} \geq 25.4$ mm.

Addressing

$$r_1(y) = \frac{\theta_0}{2 \sin^{-1} \left( \frac{\text{DIM}}{\text{DIM}_{\text{MAX}}} \right)} - N + 1 \geq 0 \quad \text{(36)}$$

$$r_2(y) = 2\text{DIM}_B - K_{\text{DIM}_{\text{MIN}}} (\text{DIM} - \text{dim}) \geq 0 \quad \text{(36a)}$$

$$r_3(y) = K_{\text{DIM}_{\text{MAX}}} (\text{DIM} - \text{dim}) \geq 0 \quad \text{(36b)}$$

$$r_4(y) = \beta B_w - \text{DIM}_B \leq 0 \quad \text{(36c)}$$

$$r_5(y) = \text{DIM}_{\text{MAX}} - 0.5(\text{DIM} + \text{dim}) \geq 0 \quad \text{(36d)}$$

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

| Comparative algorithms                              | Optimal values for variables | Optimum cost |
|-----------------------------------------------------|-----------------------------|--------------|
| Proposed ICHIMP-SHO                                 |                             |              |
| Coello (GA-based technique) [157]                   | $h = 0.205735$, $l = 3.474456$, $t = 9.040229$, $b = 0.205802$ | $0.726576$  |
| HHHO-SCA [75]                                        |                             |              |
| GA [30]                                              | $h = 0.2088$, $l = 3.4205$, $t = 8.9975$, $b = 0.21$ | $1.748309$  |
| GSA [106]                                            | $h = 0.2489$, $l = 6.1730$, $t = 8.1789$, $b = 0.2533$ | $2.4331$    |
| Coello and Montes (NPGA) [145]                      | $h = 0.205986$, $l = 3.471328$, $t = 9.020224$, $b = 0.205706$ | $1.728226$  |
| Random                                              | $h = 0.4575$, $l = 4.7313$, $t = 5.0853$, $b = 0.6600$ | $4.1185$    |
| CDE [146]                                            | $h = 0.203137$, $l = 3.542998$, $t = 9.033498$, $b = 0.206179$ | $1.733462$  |
| (PSOStr) [158]                                       | $h = 0.2015$, $l = 3.526$, $t = 9.041398$, $b = 0.205706$ | $1.731186$  |
| Simplex                                              | $h = 0.2792$, $l = 5.6256$, $t = 7.7512$, $b = 0.2796$ | $2.5307$    |
| PSO [149]                                            | $h = 0.197411$, $l = 3.315061$, $t = 10.00000$, $b = 0.201395$ | $1.820395$  |
| He and Wang (CPSO) [159]                             | $h = 0.202369$, $l = 3.544214$, $t = 9.04821$, $b = 0.205723$ | $1.728024$  |
| David                                                | $h = 0.2434$, $l = 6.2552$, $t = 8.2915$, $b = 0.2444$ | $2.3841$    |
| MFO [104]                                            | $h = 0.203567$, $l = 3.443025$, $t = 9.230728$, $b = 0.212359$ | $1.732541$  |
| Gandomi et al. (FA) [160]                            | $h = 0.2015$, $l = 3.562$, $t = 9.0414$, $b = 0.2057$ | $1.73121$   |
| Approx                                               | $h = 0.2444$, $l = 6.2189$, $t = 8.2189$, $b = 0.2444$ | $2.3815$    |
| SCA                                                  | $h = 0.204695$, $l = 3.536291$, $t = 9.004290$, $b = 0.210025$ | $1.759173$  |
| HS [35]                                               | $h = 0.2442$, $l = 6.2231$, $t = 8.2915$, $b = 0.2443$ | $2.3807$    |

**Fig. 16** Welded mechanical beam model

**Fig. 17** Design of gear train optimization design
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 |
|------------------------|-----------------------------|-----------|-----------------|
| x₁(Tₐ)                 | x₂(T₂)                      | x₃(T₉)    | x₄(T₄)         |
| 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

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

| Comparative algorithms | Optimal values for variables | Optimum fitness |
|------------------------|-----------------------------|-----------------|
| W₁                    | W₂                         | W₃              | W₄              |
| 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     |
Here

\[
f_c = 37.91 \left[ 1 + \left\{ 1.04 \frac{(1 - \varepsilon)}{\varepsilon} \left( \frac{f_I (2f_0 - 1)}{f_0 (2f_I - 1)} \right)^{0.41} \right\}^{10/3} \right]^{-0.3} \times \left[ \frac{(1 - \varepsilon)^{0.39}}{\varepsilon^{1.39}} \right] \left[ \frac{2f_I}{2f_I - 1} \right]^{0.41}
\]

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

\[
\varepsilon = \frac{\text{DIM}_B}{\text{DIM}_\text{MAX}} \cdot f_I, f_0 = \frac{R_I}{\text{DIM}_B}, \quad t = \text{DIM} - \text{dim} - 2\text{DIM}_B
\]

\[
0.5(\text{DIM} + \text{dim}) \leq \text{DIM}_\text{MAX} \leq 0.6(\text{DIM} + \text{dim}), \quad 0.15(\text{DIM} - \text{dim}) \leq \text{DIM}_B \leq 0.45(\text{DIM} - \text{dim}),
\]

\[
\text{DIM} = 160, \text{dim} = 90, B_W = 30, R_I = R_0 = 11.033
\]

\[
4 \leq N \leq 50
\]

\[
0.515 \leq f_I \text{ and } f_0 \leq 0.6
\]

**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 | 1.33957 |
| 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 | 4.4950112324 | 3.4960223242 | 2.15272617 | 1.339595 |
| 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 | 1.3399 |
| ALO [162]              | 6.01812 | 5.31142 | 4.48836 | 3.49751 | 2.158329 | 1.3395 |
| SOS [173]              | 6.01878 | 5.30344 | 4.49587 | 3.49896 | 2.15564 | 1.3396 |
| 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 |
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

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

subjected to

\[
g(x) = 2bt_w + t_n(h - 2t_f) \leq 0.
\]

Variable range \( 10 \leq x_1 \leq 50, \ 10 \leq x_2 \leq 80, \ 0.9 \leq x_3 \leq 5, \ 0.9 \leq x_4 \leq 5. \)

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

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

But to minimize

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

| Comparative algorithms | Optimal values for variables | Optimum fitness |
|------------------------|-----------------------------|-----------------|
| Proposed ICHIMP-SHO    |                            |                |
| 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 2.3217923 | 0.0131        |
| MFO [104]              | 50.00 80.00 1.7647 5.000 | 0.0066259      |
| SOS [173]              | 50.0000 80.0000 0.9000 2.3218 | 0.0131        |
| CSA [153]              | 49.99999 80.0000 0.9 2.3217923 | 0.013074119    |
| ARMS [177]             | 37.05 79.99 0.9 2.31 | 0.131      |
| Improved ARMS [177]    | 48.42 79.99 0.9 2.4 | 0.131      |

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 |
|------------------------|-----------------------------|----------------|
| Proposed ICHIMP-SHO    |                            |                |
| 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]               | 0.05168568429975 0.35663508703361 11.2937296824506 | 0.01266522292643 |
| AIS-GA                 | 0.0516608 0.3560323 11.329555 | 0.012666       |
| 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      |
Ranges of variables are $0.005 \leq S_1 \leq 2.00, 0.25 \leq S_2 \leq 1.3, 2.00 \leq S_3 \leq 1.8$.

### 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:

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

$$g_1(\vec{S}) = 1 - \frac{S_3^2S_2}{71785S_1^4} \leq 0$$

$$g_2(\vec{S}) = \frac{48S_2^2 - S_1S_2}{12566(S_2S_1^3 - S_1^4)} + \frac{1}{5108S_1^2} \leq 0$$

$$g_3(\vec{S}) = 1 - \frac{140S_1S_2}{S_3^2S_2} \leq 0$$

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

subjected to

$$c_{b1} = D_0 - D_{in} - \Delta D \geq 0$$

$$c_{b2} = L_{MAX} - (S_f + 1)(Th + \alpha) \geq 0$$

$$c_{b3} = PM_{MAX} - PM_x \geq 0$$

$$c_{b4} = PM_{MAX}Z_{MAX} + PM_xZ_{SR} \geq 0$$

$$c_{b5} = Z_{SR_{MAX}} - Z_{SR} \geq 0$$

$$c_{b6} = t_{MAX} - t \geq 0$$

$$c_{b7} = RC_h - RC_f \geq 0$$

$$c_{b8} = t \geq 0.$$ (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 |
|------------------------|-----------------------------|-----------------|
| Proposed ICHIMP-SHO    | $x_1: 69.99315$ $x_2: 90$ $x_3: 15$ $x_4: 1000$ $x_5: 2.31519$ | $0.3900536$ |
| HHO [100]              | $x_1: 69.999999$ $x_2: 90.00$ $x_3: 1.00$ $x_4: 1000.00$ $x_5: 2.312781994$ | $0.259768993$ |
| WCA [174]              | $x_1: 70.00$ $x_2: 90.00$ $x_3: 1.00$ $x_4: 910.000$ $x_5: 3.00$ | $0.313656$ |
| MBFPA [180]            | $x_1: 70$ $x_2: 90$ $x_3: 1$ $x_4: 600$ $x_5: 2$ | $0.235242457900804$ |
| PVS [175]              | $x_1: 70$ $x_2: 90$ $x_3: 1$ $x_4: 980$ $x_5: 3$ | $0.31366$ |
| hHHO-SCA [75]          | $x_1: 70$ $x_2: 90$ $x_3: 2.312785$ $x_4: 1000$ $x_5: 1.5$ | $0.389653842$ |
| NSGA-II                | $x_1: 70$ $x_2: 90$ $x_3: 3$ $x_4: 1000$ $x_5: 1.5$ | $0.4704$ |
| TLBO [74]              | $x_1: 70$ $x_2: 90$ $x_3: 3$ $x_4: 810$ $x_5: 1$ | $0.3136566$ |
| MADE [75]              | $x_1: 70.00$ $x_2: 90$ $x_3: 3$ $x_4: 810$ $x_5: 1$ | $0.3136566$ |
| hHHO-PS [74]           | $x_1: 76.594$ $x_2: 96.59401$ $x_3: 1.5$ $x_4: 1000$ $x_5: 2.13829$ | $0.389653$ |
Fig. 24 Convergence curve and Trial runs for multidisciplinary engineering design problem with ICHIMP and ICHIMP-SHO
Fig. 24 (continued)
Here,

\[ PM_x = \frac{F_x}{H(D_0^3 - D_{in}^3)} \]

\[ Z_{SR} = \frac{2\pi n(D_0^3 - D_{in}^3)}{90(D_0^3 - D_{in}^3)} \]

\[ t = \frac{i_x \pi n}{30(RC_h + RC_f)} \]

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 Fusi 48(August):67–83. https://doi.org/10.1016/j.inffus.2018.08.002
3. &NA (2000) Rapid Communications. J AID S J Acquir Immun Defic Syn 23(5):374. https://doi.org/10.1007/s10263-0000-00002
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 optimization 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:100820281328
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/s10897-012-9142-2
15. Yang XS (2010) Firefly algorithm, stochastic test functions and design optimization. Int J Bio Inspired Comput 2(2):78–84. https://doi.org/10.1540/IJBIC.2010.01214
16. Zong-Yuan M (2002) “5 1063 1” pp. 1–5. https://doi.org/10.1006/rwn.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 Armis 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, Aljarih M, 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 nonlinear large scale problems. Inf Sci (NY) 183(1):1–15. https://doi.org/10.1016/j.ins.2011.08.006
27. Hansen P, Mladenovic 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. Doanj B, Olmez 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 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
33. Swain RK, Sahu NC, Hota PK (2012) Gravitational search algorithm. Inf Sci (NY) 293(August):125–145. https://doi.org/10.1016/j.ins.2014.08.053
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/10.1177/0037964107010600201
36. Salimi H (2012) Stochastic fractal search: a powerful method for optimal economic dispatch. Procedia Technol 6:411–419. https://doi.org/10.1016/j.protcy.2012.10.049
37. Nguyen TT, Vo DN (2015) The application of one rank cuckoo 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
38. Xu J, Zhang J (2014) Exploration-exploitation tradeoffs in multi-verse optimizers. Appl Math Inf Sci 8(3):977–983. https://doi.org/10.12785/amis/080306
39. Nematollahi AF, Rahiminejad A, Vahidi B (2017) A novel parallel based meta-heuristic optimization method known as light−neighbourhood search: methods and applications. Ann Oper Res 251(1):367–407. https://doi.org/10.1007/s10479-009-0657-6
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. Ruiz-Vanoye JA, Díaz-Parra O, Cocón F, Soto A (2021) Metaheuristics: Survey and analysis. In: Proc. 33rd Chinese Conf. CCC 2014, pp. 8633–8638. https://doi.org/10.1016/j.ins.2014.02.123
42. Rechenberg I (1989) Evolution strategy: nature’s way of optimization, vol 41. Elsevier Ltd
43. Rizk-Allah RM, Hassanien AE, Elhoseny M, Gunasekaran A (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
44. 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
45. 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
46. 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
47. Hongye L, Atlazap-Gargari E, Lucas C (2007) Imperialistic competitive algorithm ICA IEEE CEC 2007 inspired by imperialistic competition
48. 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
49. Ruiz-Vanoye JA, Díaz-Parra O, Cocón F, Soto A (2021) Metaheuristics algorithms based on the grouping of animals by social behavior for the traveling salesman problem. Int J Comb Optim Prob Inform
50. 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
51. 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
52. Faris H, Aljarah I, 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
53. Rizk-Allah RM, Hassanien AE, Elhoseny M, Gunasekaran A (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
54. 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
55. 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
56. Hashim FA, Houssine EH, Mabrouk MS, Al-atabany W (2019) Henry gas solubility optimization: a novel physics-based algorithm. J Opt: Theory and Application 181(2):881–896. https://doi.org/10.1007/S00040-018-0348-5
57. Lee JY, Park Y, Han S (2015) A new hybrid optimization algorithm: charged system search. Acta Mech 213(3–4):267–289. https://doi.org/10.1007/s00707-009-0270-4
58. 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
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/10.1080/17508120.2017.1310011
68. Huang KW, Wu ZX (2018) CFO: a crow particle optimization algorithm. Int J Comput Intell Syst 12(1):426–435. https://doi.org/10.1080/17512058.2018.1359056
69. Aala Kalananda VK, 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/10.1116/essy.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 bernstein-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, Yildz AR (2020) A comparative study of metaheuristic algorithms for reliability-based design optimization problems. Arch Comput Methods Eng. https://doi.org/10.1007/s11831-020-00944-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) Archmedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. Appl Intell 51(3):1531–1551. https://doi.org/10.1007/s10462-020-01893-z
80. Bala Krishna A, Saxena S, Kamboj VK (2021) hSMA-PS: a novel memetic algorithm 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 metaheuristic 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/10.1007/s13748-019-00184-0
85. 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
86. Dhiman G, Kaur A (2019) A hybrid algorithm based on particle swarm and spotted hyena optimizer for global optimization, vol 816. Springer, Singapore
87. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications.
101. 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

102. 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

103. Dhiman G (2021) SSC: A hybrid nature-inspired meta-heuristic paradigm. Knowl Based Syst. https://doi.org/10.1016/j.knosys.2015.07.006

104. Mirjalili S (2015) Moth–flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl. Based Syst. https://doi.org/10.1016/j.knosys.2014.07.007

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, a nature-inspired heuristic paradigm. Knowl Based Syst. https://doi.org/10.1016/j.knosys.2015.07.006

108. Yao Xin, Liu Yong, Lin G (1999) Evolutionary programming: a basis for design. Eng Comput. https://doi.org/10.1007/s10489-020-01727-y

109. 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

110. Dhiman G, Kaur A (2019) A hybrid algorithm based on particle swarm and spotted hyena optimizer for global optimization, vol 816. Springer, Singapore

111. Kaur S, Awasthi LK, Sangal AL (2021) HMOSSHA: a hybrid meta-heuristic approach for solving constrained optimization problems, vol 37. Springer, London

112. 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

113. 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

114. 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

115. 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

116. 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

117. 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

118. Wang J, Khishe M, Kaveh M, Mohammadi H (2021) Binary chimp optimization algorithm (BChOA): a new binary meta-heuristic for solving optimization problems. Cognit Comput. https://doi.org/10.1007/s12559-021-09933-7

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) HMOSSHSA: 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. Safiari A, Zahiri SH, Khishe M, Seyed 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. Yazed 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

128. 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

129. 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

130. Wang J, Khishe M, Kaveh M, Mohammadi H (2021) Binary chimp optimization algorithm (BChOA): a new binary meta-heuristic for solving optimization problems. Cognit Comput. https://doi.org/10.1007/s12559-021-09933-7

131. 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/e3xconf/202018401069.

132. 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

133. Sherif BV (2021) Detection and isolation of sel sh nodes in MANET using collaborative contact-based watchdog with. https://doi.org/10.1109/jetsconf.2020.8401069.

134. Rahkar Farshi T (2020) Opposition-based enhanced grey wolf optimization algorithm for AVR optimization, vol 24. Springer, Berlin, Heidelberg

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, Alghuwaimeen 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/s10458-018-1325-9

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 HIC, 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, CoelloCAC (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. 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

147. 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

148. Pelusi D, Masrella 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

149. Mirjalili S (2015) The ant lion optimizer, vol. 83. Elsevier Ltd.

150. 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

151. 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

152. Kanan (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

153. 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.

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.2003.814002

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/S0165-3615(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. Litinetski VV, Abramzon BM (1998) Mars—a multistart adaptive random search method for global constrained optimization. Eng Optim 32:319–326. https://doi.org/10.1016/j.compstruc.2011.08.002

161. 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

162. 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

163. 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.

164. 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

165. 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/10.1080/030521508940941

166. 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

167. 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

168. 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

169. Zolghadr-Asli B, Bozorg-Haddad O, Chu X (2018) Crow search algorithm (CSA). In: Studies in Computational Intelligence
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.