The Journal of Supercomputing
https://doi.org/10.1007/s11227-024-06291-7

Flood algorithm (FLA): an efficient inspired meta-heuristic for engineering optimization

Mojtaba Ghasemi[1] · Keyvan Golalipour[2] · Mohsen Zare[3] · Seyedali Mirjalili[4,5] · Pavel Trojovský[6] · Laith Abualigah[7,8,9,10] · Rasul Hemmati[11]

Accepted: 6 June 2024

Abstract

Introducing a novel meta-heuristic optimization algorithm, the Flood Algorithm (FLA) draws inspiration from the intricate movement and flow patterns of water masses during flooding events in river basins. FLA mathematically models key phenomena such as the movement of water toward slopes, flow rates over time, soil permeability effects, and periodic increases and decreases in water levels from precipitation and losses. Leveraging these models, the algorithm guides the movement and evolution of a population of potential solutions toward enhanced optimality. The algorithm endeavors to establish an appropriate correlation between the fundamental aspects of natural flood events and the optimization process. Its formulation and working mechanism are described in detail. It operates in two main phases—a regular movement phase, where the population moves naturally toward current best solutions, and a flooding phase, which introduces random disturbances to increase diversity. New solutions are periodically introduced while weaker ones are removed, mirroring the natural cycles of water levels. FLA's effectiveness is demonstrated through its application on well-known benchmark optimization problems and engineering design problems. Extensive comparisons have been carried out on CEC2005 functions using 16 algorithms in both basic and enhanced modes, as well as on CEC2014 functions with dimensions 30, 50, and 100 using a total of 20 other algorithms. These rigorous studies unequivocally confirm the robustness and strength of the proposed algorithm. Furthermore, the algorithm's performance on 12 constrained engineering problems demonstrates its ability to tackle real-world challenges. The FLA's source code is publicly available at https://www.optim-app.com/projects/fla.

Keywords Engineering optimization · Modern algorithms · Nature-inspired optimization algorithms · Flood Algorithm (FLA)

Extended author information available on the last page of the article

Published online: 01 July 2024

1 Introduction

In optimization problems, the main goal is to discover the best solution by considering the set of situations of the issue. First, the objective function must be defined according to the nature of the problem. In some cases, several objective functions are considered, and their effects are considered by various approaches that have been studied and analyzed in recent years [1, 2]. In a power plant, for example, an attempt is made to use the optimal method to maximize the amount of electricity produced in steam turbines by minimizing constraints such as cost, time, and pollution. Therefore, optimization is a critical process in design, which should ultimately minimize the energy required and maximize the desired profit [3, 4].

Optimization application in most sciences, such as physics, chemistry, and economics, is evident and has been analyzed in many problems. Examples of these problems in different fields include intrusion detection algorithms [5], energy perturbation calculations for drug discovery [6], path-planning techniques [7], shape and topology optimization [8], and operation and planning problems for DG units [9]. To achieve optimal production in massive designs and industrial research, the need for an efficient and robust optimization algorithm is understandable and necessary [10].

In general, in a typical issue, having the objective function ($f(x)$) of the problem, the aim is to discover the control parameter x so that the value of $f(x)$ becomes optimal, maximized, or minimized in some cases. After defining the problem control parameters, the problem is first considered as an objective function of them. Physical limits are then displayed as a kind of problem constraint. Then, the optimal solution is extracted by solving the resulting model inspired by optimization algorithms [10].

In general, an optimization problem is implemented as follows [10]:

$$(\text{minor max})z = f(X)\begin{cases} g_i(X) \leq 0; i = 1, 2, \dots, Ng \\ h_i(X) = 0; i = 1, 2, \dots, Nh \end{cases} X = \left[x_1, x_2, \dots, x_D\right] \tag{1}$$

$f(X)$ represents the objective function, and X is a D-dimension vector of control variables for the considered problem or algorithm input. Also, $h_i(X)$ and $g_i(X)$ are the inequality and quality constraints of the problem, respectively. The number of g and h constraints is N_g and N_h.

Although many optimization algorithms have been suggested in the recent two decades, as mentioned in the abstract, we are always looking for a simpler and more comprehensive algorithm to reach the optimal solution we expect in most optimization problems [11]. A glance at the previous algorithms shows some general shortcomings. In this paper, the modeling has been carried out to eliminate these shortcomings: First, previously published algorithms have exhibited an imbalance between local and global search. Most algorithms are optimized to a local solution and lose their optimization power. Secondly, they need considerable time and iterations to reach an acceptable solution. Third, some algorithms require complex calculations, so the user has difficulty using that algorithm. The proposed meta-heuristic approach outlined in this study represents a straightforward optimization method characterized by robustness and efficacy, striking a suitable balance between local and global exploration to effectively attain desired solutions within various optimization functions. The results obtained substantiate these assertions.

In recent decades, many optimization algorithms have been suggested to optimize various issues, most of which are based on accidental research. The algorithms above seek the feasible area parallel to several initial points. These algorithms are population based. By population, we mean a group consisting of simple, independent, and asynchronous agents that optimize, collaborate, and interact with each other to discover optimal solutions to an issue. In a population-based algorithm, each one of the members carries out a series of special operations and shares data with the swarm. Swarm intelligence is a phenomenon that results from the combined impact of relatively simple operations [12]. The operations entail localized interactions among individuals and factors, culminating in a global outcome that optimizes the system without necessitating a central controller. Typically, population-based algorithms share two common aspects:

1. Exploration: The ability to evaluate feasible solutions not in the current solution's neighborhood.
2. Exploitation: The ability to discover the optimal result in a feasible solution's neighborhood.

In an optimizer, the first part of iterations searches for a feasible area to find new solutions, and the second part repeats the operation. On the other hand, the population of this algorithm goes through the following three steps in each iteration to achieve the goals of exploitation and exploration:

1. **Self-adaptation**: Every population member enhances their capabilities during this phase.
2. **Cooperation**: During this phase, information is exchanged among the participants.
3. **Competition**: Better solutions are selected, while weaker ones are discarded at this point.

The execution of these three steps is usually random and based on a probability distribution function. For example, the process of the particle swarm optimization (PSO) algorithm is based on the collective behavior of birds in search of food in the wilderness. This algorithm follows three steps inspired by the birds' behavior. In this way, in the beginning, a swarm of particles is randomly formed in the feasible region, and this algorithm, by updating the position of these particles, tries to discover the optimal solution in each iteration. This is the competition stage. The position of each member, which can be compared to a bird's position in the wild, is updated at every step with the help of two data points. One of the data points is the best position the member (bird) has ever reached, which is known as self-adaptation. This position is recognized, stored, and displayed in the algorithm with P_{best}. The second data used by the algorithm are the best solution ever obtained by a population of the swarm and are called G_{best}. This is the cooperation stage. All inspired meta-heuristics generate satisfactory results, and no meta-heuristic optimizer can perform better in optimizing all objective functions. This study aims to present a novel meta-heuristic for solving the following optimization problem [13].

$$\min\{f(X) : X \in Z\} Z = \left\{ X \in \mathbb{R}^D : -\infty < \min_d \le x_d \le \max_d < +\infty, d = 1, 2, 3, \ldots, D \right\} \tag{2}$$

where $f : \mathbb{R}^D \to \mathbb{R}$ is a nonlinear function, and $\min_d$ and $\max_d$ are the lower and upper limits of the component x_d. It is clear that the maximization problems can turn into a minimization problem as follows:

$$\max f(X) = -\min(-f(X)) \tag{3}$$

In this problem, the position or solution of the ith member in the area Z is given by X_i $i = 1, 2, 3, \ldots$, and the algorithm population is N_{pop}. At this point, the $f(X)$ value is represented by $f(X_i) = f_i$ and is known as the fitness function.

Meta-heuristic algorithms attempt to replicate natural phenomena to discover the best solution as machine intelligence approaches [14–16]. The classification of optimization algorithms [17] is divided into four general categories depending on their source of inspiration: (i) algorithms based on swarm intelligence (SI), (ii) algorithms based on evolution (EB), (iii) algorithms based on physics (PB), and (iv) algorithms based on games (GB). SI models the behavior of biological beings, plants, and natural processes; EB models the genetic sciences; PB models the behavior of various physical laws; and GB models the behavior of different game rules. Table 1 summarizes some popular optimization methods of the background research.

Now, the question is, why all these new meta-heuristic algorithms? It is worth reiterating that, as stated in [33], different algorithms are required in different domains to address the optimization needs of diverse test functions effectively. Previous research has shown that each algorithm is capable of achieving satisfactory results for certain functions but may fail to produce optimal solutions for others. This need for algorithm diversity is a widely recognized fact in the field.

Evolutionary methods traverse solution spaces with exceptional exploration capability, uncovering diverse solutions. They are meticulously designed to discover global optima, skillfully avoiding local optima by fostering diversity in the population through mutation, selection, social interactions, or inspiration from natural phenomena. Moreover, these methods showcase remarkable adaptability, adeptly navigating dynamic environments and adjusting their search strategies based on feedback. Their versatility is a testament to their potential to find successful applications in various problem domains, including optimization, scheduling, machine learning, and data mining.

However, it is essential to acknowledge the challenges of evolutionary methods. They can be computationally demanding, especially for large-scale problems or high-dimensional search spaces, necessitating substantial computational resources for fitness function evaluation and candidate solution generation. These methods may not be the fastest in convergence, as the search process involves iterations and generations. Parameter tuning is complex, as evolutionary methods rely on multiple parameters such as population size, mutation rate, crossover probability, and specific

Table 1 Some widely used and popular meta-heuristic methods

Year	Meta-heuristic	Refs.	Class	Source of inspiration
1983	Simulated annealing (SA)	[18]	PB	The "SA" is attributed to the common origin of its usage with solid annealing, the process by which crystalline materials are heated and then cooled very slowly, eliminating flaws like crystals with other configurations
1989	Evolution strategy (ES)	[19]	EB	The biological method of evolution
1992	Genetic algorithm (GA)	[20]	EB	Mutations and selection are essential for neo-Darwinian evolution
1995	Particle swarm optimization (PSO)	[21]	SI	Using only simplistic social interactions instead of traditional cognitive capacities to create computational intelligence. Heppner and Grenander's [17] inspired the study on bird flocks seeking maize
1997	Differential evolution (DE)	[22]	EB	A relatively recent evolutionary algorithm, natural evolutionary phenomena, and the application of floating-point coding in place of binary integers
2006	Invasive weed optimization (IWO)	[23]	EB	Inspired by weeds being colonized. Weeds represent a severe danger to desirable cropland plants as a hazard to agriculture, with their strong, invasive growing tendencies
2007	Imperialist competitive algorithm (ICA)	[24]	PB	The imperialist struggle between empires, weak empires collapsing during this competition, and great empires taking their territories in possession
2008	Firefly algorithm (FA)	[25]	SI	Mimics fireflies' social behavior based on the features of fireflies' flashing and attraction
2008	Biogeography-based optimization (BBO)	[26]	EB	A bio-inspired meta-heuristic focused on island biogeography mathematics and species movement from one environment to another
2009	Gravitational search algorithm (GSA)	[27]	PB	The search agents are a group of masses following the rules of gravity and weight, and their connections are based on Newtonian gravitational and motion laws
2009	Cuckoo search (CS)	[28]	SI	CS is predicated in conjunction with the Levy flying movements of a few fruit flies and birds on obligatory parasitic brood behavior
2010	Bat-inspired algorithm (BA)	[29]	SI	BA uses the echolocation trend of bats to look for a place for design
2011	Teaching–learning Algorithm (TLBO)	[30]	SI	The learning and teaching behavior of students
2012	Flower pollination algorithm (FPA)	[31]	EB	FPA mimics flower pollination manners. Pollination helps plants mature, and it is related to pollinators like insects transferring pollen
2012	Water cycle algorithm (WCA)	[32]	PB	WCA is inspired by the formwork of the water flow and cycle on and under the earth's surface, ending in seawater

Table 1 (continued)

Year	Meta-heuristic	Refs.	Class	Source of inspiration
2014	Gray wolf optimizer (GWO)	[33]	SI	The mechanism gray wolves employ for hunting
2015	Moth-flame optimization (MFO)	[34]	SI	Moths' navigation mechanism, namely, transverse orientation
2015	competitive swarm optimizer (CSO)	[35]	SI	In animal competitiveness, the particles learn through randomly selected competition rather than from the best position globally or personally
2016	Whale optimization Algorithm (WOA)	[36]	SI	Mimics the bubble-net hunting mechanism of humpback whales
2016	artificial Infectious disease (AID)	[37]	EB	Using the SEIQR epidemic dynamic model, the algorithm assumes that humans live in an ecosystem, each with unique characteristics
2016	Lion optimization algorithm (LOA)	[38]	SI	The algorithm is based on lion behaviors like hunting, mating, and defense
2016	Water evaporation optimization (WEO)	[39]	PB	WEO reflects the well-known principles for water molecules' evaporation off a robust and weight-mounted surface. A WEO algorithm updates molecules in two sequential stages globally and locally, namely the single-layer and bulk evaporation phases
2016	Rooted tree optimization (RTO)	[40]	EB	It includes different roots from the main tree to start the search. It is the center of the major arbitrary collection. The next generation is obtained by assessing the primary range by the root nearest to maize and the fineness level
2017	Grasshopper optimization algorithm (GOA)	[41]	SI	Inspired by Grasshoppers' swarming food-search behavior, the grasshopper's life cycle is split into two main phases: larval and adult
2017	Weighted superposition attraction (WSA)	[42]	PB	This method consists of two primary mechanisms, the overlay and the attraction of agents, which may be seen in many systems
2017	Salp swarm agorithm (SSA)	[43]	SI	It works according to the swarm behavior of salps
2017	Collective decision optimization (CDO)	[44]	PB	The social treatment includes their decision-making qualities and replicates human social behavior, including experience, the group-thinking phase, and the leading and innovation-based phases
2018	Emperor penguin optimizer (EPO)	[45]	SI	That imitates the emperor penguins' huddled demeanor. The primary phases of EPO are to create the hood limit, calculate the heat surrounding the hood, calculate the interval, and identify the efficient mover
2018	Pity beetle algorithm (PBA)	[46]	SI	It was created based on the beetle Pityogenes chalcographus's tendency to accumulate resources for their nests and sustenance in sick trees

Table 1 (continued)

Year	Meta-heuristic	Refs.	Class	Source of inspiration
2018	CFA optimizer	[47]	PB	Follows Franklin's and Coulomb's laws applied to electric charge particles
2018	Tree growth algorithm (TGA)	[48]	EB	The intensification and diversification of the TGA follow two key phases. Plants fight for sunlight and nourishment. In the intensification phase, the food source is divided among the well-developed trees that have adapted to absorb more light. Later, the TGA allows other trees to compete for light absorption and moves to a virgin area to expand
2019	Henry gas solubility optimization (HGSO)	[49]	PB	Following Henry's formula when solving optimization challenges
2019	Dice game optimizer (DGO)	[50]	GB	The algorithm is based on an older game, and the searchers are a group of players. Each player has a unique number of guides assigned to them, based on the number of dice. At least one and up to six guide players control each player's movements
2019	Harris hawks optimization (HHO)	[51]	SI	The collaboration among Harris' hawks and their chasing behavior
2019	Sailfish optimizer (SO)	[52]	SI	SO was inspired by a group of sailfish hunters. This approach utilizes two populations: sailfish to intensify the search for the best and sardines to produce search diversity
2020	Marine predators algorithm (MPA)	[53]	SI	Ocean predators use a Lévy-Brownian movement-based foraging strategy alongside a policy for encounter rate optimization in biological interactions between predators and prey
2020	Shell game optimization (SGO)	[54]	GB	The goal of the SGO is to discover the ball buried under one of the three shells. The major benefit of SGO is that it has no control parameters, so there is no need to configure them
2020	Slime mold algorithm (SMA)	[55]	EB	SMA simulates the negative and positive feedback of the slime mould propagation wave
2020	Gradient-based optimizer (GBO)	[56]	PB	GBO explores search space by adopting two primary operators: local escaping operator (LEO) and gradient search rule (GSR), and a collection of solutions
2020	Turbulent flow of water-based Optimization (TFWO)	[57]	PB	I was inspired by a natural occurrence: whirlpools formed in chaotic water flow
2020	Mayfly algorithm (MA)	[58]	SI	The algorithm is derived from mayflies' flight behavior and reproductive processes, effectively integrating the benefits of swarm intelligence and evolutionary algorithms
2021	Hunger games search (HGS)	[59]	GB	In HGS, individual hunger drives collaborative connections
2021	Red fox optimization (RFO)	[60]	SI	The red fox life and hunting behavior mathematical model is the basis of RFO
2021	Wild geese algorithm (WGA)	[61]	EB	The life cycle of wild geese is followed
2021	Aquila optimizer (AO)	[62]	SI	The Aquila's behaviors in the prey hunting are Imitated

Table 1 (continued)

Year	Meta-heuristic	Refs.	Class	Source of inspiration
2021	Colony predation algorithm (CPA)	[63]	SI	CPA incorporates distinctive specifications into a regular model, which incorporates achievement rates for matching strategies and simulates selective release behavior observed in prey animals
2022	War strategy optimization (WSO)	[64]	GB	WSO portrays two fundamental tactics in warfare: defensive and offensive. The war strategy is envisioned as an optimization procedure, wherein any soldier dynamically progresses toward the best outcome, guided by the strategic movements of army troops during the conflict
2022	Snake optimizer (SO)	[65]	SI	The SO was inspired by snakes' specific mating behavior. Each snake engages in competitive interactions to secure the best partner when the available food is ample, and the temperature is low. SO models snake reproduction processes, behaviors, and search patterns
2022	Pelican optimization algorithm (POA)	[66]	SI	In the POA, search agents simulate the natural behavior of pelicans during hunting as they seek food sources
2022	Cone search (CS)	[67]	SI	Particularly in the initial iterations within its local search space, the term "Cone" is coined from a distinctive strategy where the shift from exploration to exploitation unfolds gradually. In the Cone Search (CS), the local search space undergoes a gradual linear reduction during iterations to enhance exploration
2022	Cheetah optimizer (CO)	[68]	SI	Drawing inspiration from the hunting tactics employed by cheetahs, the cheetah optimizer (CO) is proposed. Cheetahs, renowned for their hunting prowess, employ three primary strategies when pursuing prey: searching, sitting and waiting, and attacking
2023	Flying foxes optimizer (FFO)	[69]	SI	This is inspired by the survival strategies of flying foxes during heatwaves and utilizes Fuzzy Logic (FL) to determine parameters individually for each solution, resulting in a parameters-free optimization algorithm
2023	Growth optimizer (GO)	[70]	EB	The design inspiration originated from individuals' reflection and learning processes during their societal growth cycle. Reflection included assessing and adjusting learning strategies to enhance personal development. Learning, in this context, was the process through which individuals matured by acquiring information from the external environment
2023	Coronavirus mask protection algorithm (CMPA)	[71]	EB	CMPA is conceived by taking cues from the spread of the COVID-19 virus and human self-protective measures. The progression of immunity and infection within CMPA involves several stages: diffusion, infection, and the immune response

Table 1 (continued)

Year	Meta-heuristic	Refs.	Class	Source of inspiration
2023	Bedbug meta-heuristic algorithm (BMHA)	[72]	SI	BMHA draws its main inspiration from the dynamic and stationary behaviors observed in bedbugs within their natural habitat. It is designed to include two essential phases, exploitation, and exploration, by simulating the social interaction among bedbugs during their foraging activities
2023	Geyser inspired Algorithm (GEA)	[73]	PB	Derived from unusual events occurring in various regions worldwide, it involves the following steps: searching for channels and the eruptive process

algorithmic parameters. Selecting appropriate values for these parameters can be intricate and require manual tuning or optimization. Furthermore, there is a risk of premature convergence, where evolutionary methods can get ensnared in local optima or converge prematurely, leading to suboptimal solutions. Maintaining diversity within the population and balancing exploration and exploitation are pivotal to addressing this issue.

The FLA's equations are not merely abstract mathematical constructs but rather intimately correlated with the population's diversity, reflecting the dynamics of natural phenomena. These equations acknowledge the interdependence of population members and operate in a way that preserves their dynamics during the search, regardless of other members. This connection is particularly evident in the natural occurrence of a flood, where the volume of water embodies energy, a force that magnifies with the amplification of water volume.

By integrating fundamental concepts derived from natural phenomena, the proposed algorithm holds great potential for significantly enhancing optimization performance. The algorithm population, analogous to the water mass in nature, is crucial in finding the optimal solution to the objective function. The algorithm employs a crowd-based mutation mechanism to facilitate movement toward more optimal points, emulating the collective motion of water along slopes. Moreover, the algorithm introduces novel equations of mutation or movement within the population to expedite optimization speed and promote population diversity reminiscent of the natural flooding phenomenon. Furthermore, an equation based on the iteration count is incorporated to enable the possibility of running a new mutation, drawing an analogy to the rate of flow exiting a pond. Additionally, the algorithm augments the likelihood of new mutations by leveraging an equation tied to the cost of population issues, akin to the water impermeability coefficient of soil. The algorithm further strengthens the capacity for escaping local optima and enhancing population diversity by generating new populations, akin to replenishing water through processes like melting snow, rainfall, or groundwater. This concept is complemented by selectively eliminating weaker populations, mirroring the natural phenomenon of water loss and disappearance through diverse mechanisms. The iterative nature of the algorithm simulates the search process for improved solutions, offering more opportunities akin to the cyclic pattern and fluid flow observed in nature. By integrating these concepts, the proposed algorithm strives to augment optimization performance by emulating the behavior exhibited in natural phenomena. Consequently, it presents a promising prospect for advancing optimization algorithms and modeling natural phenomena.

In general, the superiorities of the suggested natural and physical optimizer, FLA, can be summarized and expressed as follows:

1. The novel and powerful idea of the basic algorithm: as a novel meta-heuristic optimizer based on a physical phenomenon in nature
2. High ability to find optimal solutions for different functions compared to standard algorithms
3. Acceptable speed in finding the optimal solution with a suitable robustness

2 Flood algorithm (FLA)

In this part of the paper, the proposed meta-heuristic optimizer, which is designed via mathematical modeling of the flood phenomenon, will be explained, and the various components of this new optimization algorithm will be discussed.

2.1 Modeling of FLA

To design and create new inspirational meta-heuristic algorithms, we will explore various hypotheses and establish equivalences through mathematical modeling. For example, in the context of the genetic algorithm, the alignment of genetic and optimization processes relies on underlying assumptions. These assumptions manifest across various facets of the optimization algorithm, encompassing selection, crossover, and mutation. Furthermore, scientific concepts are articulated through straightforward mathematical models to introduce an optimization algorithm [74]. In the proposed FLA algorithm, as in most of the optimization meta-heuristics suggested in recent years, the FLA initially begins with a specified swarm, which is a model of the amount of water in a watershed that arises randomly. The position of water masses indicates the feasible solutions to the problem and is equivalent to particles in optimizing the particle group. River flow from this watershed is the equivalent of a possible population route to find a new location. The water masses exit this pond through the river with a general coordinated velocity and movement toward the slope, which is equivalent to the natural movement of a stream or particle mass. As mentioned, this basin is located in a watershed, so, in certain conditions that will be modeled, which depends on the amount of water flow out of the basin, a flood will occur, and the movement of water particles will be disturbed and increased. This motion is the core of FLA and causes the particle mass to move toward the optimum solution or the absolute minimum cost of the issue. As in the real world, if water infiltration in this pond into the soil is low, the possibility of flooding will increase. Two items that will increase the likelihood of floods in this watershed are melting snow and rain and adding water through other rivers or even springs, modeled in the suggested FLA so that it can escape from the optimal local solutions.

This section describes how to model the flood, as well as how to implement the proposed algorithm. Figure 1 shows a watershed area. A small water basin collects water from waterfalls and melting ice and snow. This basin consists of streams flowing into more extensive waterways such as rivers, lagoons, rivers, and seas.

2.2 Flooding phenomenon

Flooding is one of the most complex and destructive processes of natural events, which is often known as a destructive phenomenon and has important biological and economic effects. The melting of large volumes of snow trapped at elevations relative to the average water level in a river, or the interconnection of water of several rivers and springs, and many factors such as rainfall characteristics, topology, river

Fig. 1 Rhine river floodplain flood

morphology, area features, environmental structures, and human activities can be mentioned [75]. A wide variety of works have been performed in flood prediction in various watersheds in the field of flood hydrograph prediction [76–78]. Hydrological simulation models of precipitation-distributed and semi-distributed runoff along with the knowledge of characteristics of watersheds and their climate are some of the proposed solutions, such as how much water penetrates the soil in the area, which should be considered to identify flood areas and determine the nature of destructive floods in basins [79].

In fact, during floods, considerable amounts of water come out of the river and occupy new areas. In the proposed algorithm, we hypothesized that the flood would cause turbulence in the water mass or a swarm of the proposed algorithm, which would play a role in finding more efficient solutions so that the speed of the water search space or a swarm of the algorithm would increase dramatically. This flood randomly depends on the flow rate of water leaving the estuary, and the soil's low permeability increases the probability of flooding. Also, with the melting of snow or rain, some water is added to the river, as well as the effects of the sun, heat, and wind, and holes and ponds along the river cause water to evaporate or trap large volumes of water. It is better to consider these two opposing phases in a reciprocal phase so that this amount of new excess water volume replaces the amount of water loss. Hence, these two volumes or the number of populations are equal in the optimization algorithm. Mathematical modeling and inspiration from flood and water flow for the proposed algorithm are according to Table 2.

2.3 Proposed FLA

Mathematical modeling of the suggested FLA meta-heuristic for optimization is performed in the following main sections:

- Move the population toward the solution of the best cost of the objective function (move in the path with the highest slope to move the water)
- Creating floods and or disturbances in the movement of the population
- Increase the new population and reduce the weaker population (increase and loss of basin water)

Table 2 The suggested FLA meta-heuristic uses mathematical modeling and inspiration from flood and water flow

Phase (1) in nature	Equivalent phase (2) in the FLA algorithm	Executive effect (3)
Water mass	Algorithm population	Responsible to find the best solution to the objective function
The mass movement of water in the direction of the slope	Mutation in the crowd toward the best current solution	Move to a more optimal point
Flooding	Creating new equations of mutation or movement in the population	Increasing the speed of optimization and population diversity
Rate of flow leaving the pond	An equation in terms of iteration of the algorithm	Creating the possibility of running a new mutation
Water impermeability coefficient of the soil	An equation in terms of the cost of the issue of the population	Increasing the probability of a new mutation
Melting snow and rain or groundwater and adding water in any way possible	Creating new populations in the algorithm	Increasing the power of local optimal escape and population diversity
Losses and disappearance of the water in any way possible	Eliminate the weak population	Increasing the power of local optimal escape and population diversity (actually complements the previous section)
Nature cycle and water flow	The iterations of the algorithm	Searching and more opportunities to find better solutions

Phase I (Regular movement phase)

This phase includes population search in the problem dimension (natural movement of water toward the slope or better points), modeling population or water flow, soil impermeability coefficient, and its effect on flooding. As mentioned, the direction shows the best particle of the swarm or mass of water. S_{best}, in other words, shows the exact slope of the path for water to pass. For example, the S_i-th particle moves naturally to the slope and gains a value approximately equal to $S_{best,}$ and, in fact, in the science of optimization, it aims to reduce the distance between itself and the best member, that is, it moves and moves around S_{best} about $S_j - S_i$. This natural motion is shown in Fig. 2. This equation of general motion is given as follows:

$$S_i^{new} = S_{best} + rand \times \left(S_j - S_i\right) \tag{4}$$

where *rand* generates random values between 0 and 1 to the size of D. The problem has a dimension of $j = 1$: D. S_j is the jth randomly member of the population. Based on the above equation, it can be seen that the water mass tends to flow toward the slope of the path and moves naturally, or in other words, toward a better solution. Of course, the water moves in the direction of the river because there is pressure behind it, and it does not have to be a slop necessarily.

As mentioned above, with the increase of water flow from the river, for any reason, there may be floods and turbulence in the movement of the water volume. In modeling and inspirations, we have modeled this depletion coefficient or flow of water according to the iterations of the algorithm, which has been modeled in the following equation:

$$Pk = \frac{1.2}{Iter}\left[+ \frac{1}{\left(\frac{Iter_{max}}{4}\right)\times Iter} \times \ln\left(\sqrt{Iter_{max} \times Iter^2 + 1} + \frac{Iter_{max}}{4}\right)\sqrt{Iter_{max} \times Iter^2 + 1}\right]^{-\frac{2}{3}} \tag{5}$$

In the above equation, Pk is the water depletion coefficient, $Iter_{max}$ shows the maximum size of algorithm generations, and $Iter$ shows the current generation of the algorithm. Figures 3, 4, and 5 illustrate the sample changes curve for this depletion coefficient for $Iter_{max} = 5000$. In Fig. 3, this coefficient starts with a

Fig. 2 Natural move in the path with the highest slope to move the water (Rhine River)

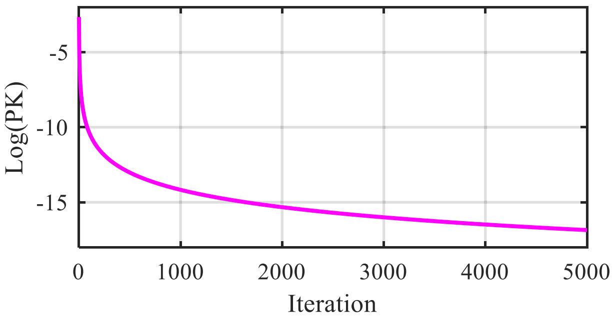


Fig. 3 The sample changes curve (log (Pk))

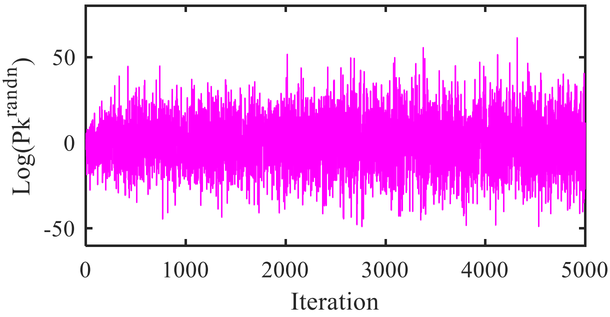


Fig. 4 The sample changes curve (log $((Pk)^{randn})$)

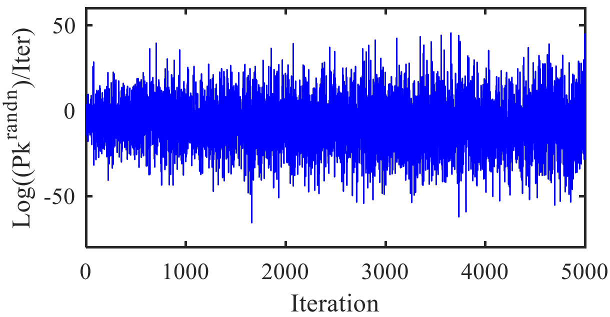


Fig. 5 The sample changes curve (log $((Pk)^{randn})$/Iter)

maximum value of 0.0702 and ends with a minimum value of 4.8000E-08 at the end of the algorithm.

However, as it turned out, the flood event will happen randomly with the probability of a random *rand* value, which during the flood will be the equation of motion of the water masses as in Eq. (6). As can be seen, the water masses rush randomly from the pond to this side and the other side of the area independently of each other and depending on the amount of water depletion coefficient.

$$S_i^{new} = S_i + \left(\frac{(Pk)^{randn}}{Iter} \right) \times \left(rand \times \left(S_{max} - S_{min} \right) + S_{min} \right) \tag{6}$$

In the above equation, *randn* will be a normal distribution whose value changes between infinite negative to infinite positive, which is one-dimensional. S_{max} and S_{min} are the maximum and minimum control variables of the problem in question, or in other words, the search space limits. In Fig. 4, a sample change of the coefficient Pk^{randn} is well shown, which causes disturbance in the movement of water masses during floods, and throws the particles around. This also prevents trapping in the local optimal solutions.

On the other hand, the effect of water permeability in the area's soil causes water to sink into the soil or get stuck in the holes and spread around or evaporate water, reducing the likelihood of flooding. We have modeled this effect based on the equation of the cost of the optimization function for any population, which can be a different value for each member. This equation shows that the lower or better the cost of the objective function of the mass value, the lower the soil permeability coefficient and the higher the probability of flooding. This effect is given in the following equation and is also shown for a member of the population in Fig. 6 in different iterations of the algorithm.

$$Pe_i = \left(\frac{f\left(S_i\right) - f_{min}}{f_{max} - f_{min}} \right)^2 \tag{7}$$

In this equation, $f_{\min}$ and $f_{\max}$ show the best and worst values of the optimization function found up to the current iteration of FLA. In FLA, this probability is reduced by the probability of flooding due to the depletion of water from the pond, and as a result, the probability of flooding will decrease. The following code illustrates this phase well for ith particle:

$$if rand > rand + Pe_i$$

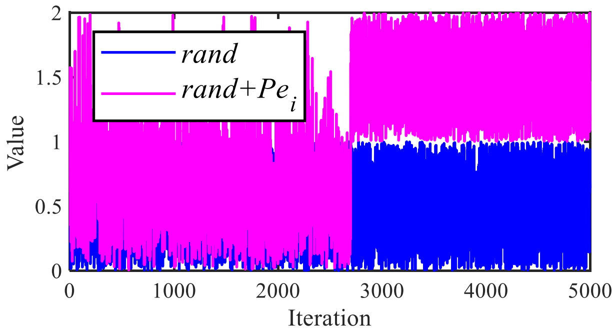


Fig. 6 The probability of flooding for a typical member decreases with increasing time

$$S_i^{new} = S_i + \left(\frac{(Pk)^{randn}}{Iter}\right) \times \left(rand \times \left(S_{max} - S_{min}\right) + S_{min}\right)$$

else

$$S_i^{new} = S_{best} + rand \times \left(S_j - S_i\right)$$

end

Now, there is a new position for the ith swarm called S_i^{new}, which is compared with the previous position and replaces it if it has a better value.

Phase II: an increase and decrease in the basin water or population in the algorithm

In nature, the amount of water in a pond is rejuvenated by rain and melting snow or water through springs. On the other hand, water sinking into the ground or getting stuck in holes or even evaporating causes some volume of water to evaporate, as shown in Fig. 7. As mentioned before, we considered the two to be reciprocal, meaning that *Ne* number of water particles evaporate, which are the weakest members. On the other hand, when *Ne* water particles are added, the total number of particles will remain constant. This process is shown in Fig. 5. Conversely, the probability of occurrence of these two opposite conditions (*Pt*) is assumed to be equal.

$$Pt = \left|\sin\left(\frac{rand}{Iter}\right)\right| \tag{8}$$

Figure 8 shows the probability of increasing and decreasing the basin water or population in FLA during its iterations, which decreases over time.

These new particles will replace the worst solutions and are added to the system. As can be seen from the equation below, the new particles will be around the best position:

Fig. 7 The example of increase and decrease of water in the basin

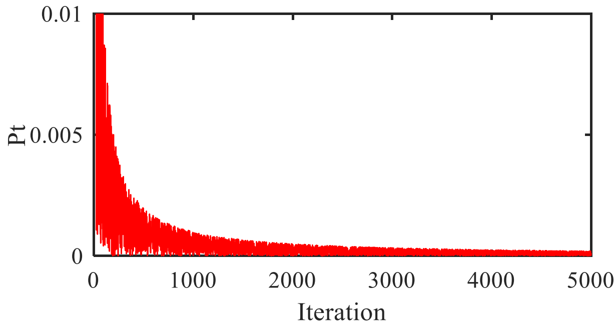


Fig. 8 The probability of increased and decreased water in the basin or population in the algorithm (Pt)

$$\begin{aligned} & S_e^{new} = S_{best} + rand \times \left(rand \times \left(S_{max} - S_{min}\right) + S_{min}\right) \\ & e = 1 : Ne \end{aligned} \tag{9}$$

Now, this cycle will continue until a desired number of iterations are executed or an acceptable optimal solution is achieved by the user. Figure 9 shows the flow chart of the proposed FLA, and the pseudo-code of this algorithm is shown in Fig. 10. Notably, only the differential terms are not responsible for the displacement of the population. Leveraging effective ideas can enhance the algorithm's efficiency. Such ideas are notably illustrated in Eqs. (6) and (9). These equations ensure that the population members can still be improved even if converging to a local optimum.

2.4 Computational complexity of FLA

Initialization, fitness evaluation, and population updating determine how complex or time-consuming the calculations of FLA will be. *Npop* is the size of members, and so the computational complexity of the initialization process and updating process are obtained as O ($Npop$) and O ($Npop \times Iter_{max}$) + O ($Npop \times Iter_{max} \times D$). The latter formulation includes researching the best solution. Consequently, the ultimate computational complexity of FLA will be O ($Npop \times (Iter_{max} + Iter_{max} \times D + 1)$).

3 A basic understanding of FLA performance as a new *meta*-heuristic optimizer

In this section, we get a basic understanding of how FLA works and compare its performance quality with some basic and well-known algorithms so that we can use this data to gain general knowledge of FLA in the continuation of this paper. The test cases are run by MATLAB utilizing a laptop computer with an Intel Core i7-7700 CPU with a clock rate of 3.6 GHz and 32 GB of RAM. Moreover, in this part, to show the efficiency of FLA for various real-world functions related to real-world problems, we have selected 14 test functions of CEC2005 and standard modern optimization functions, including various multimodal and unimodal minimization

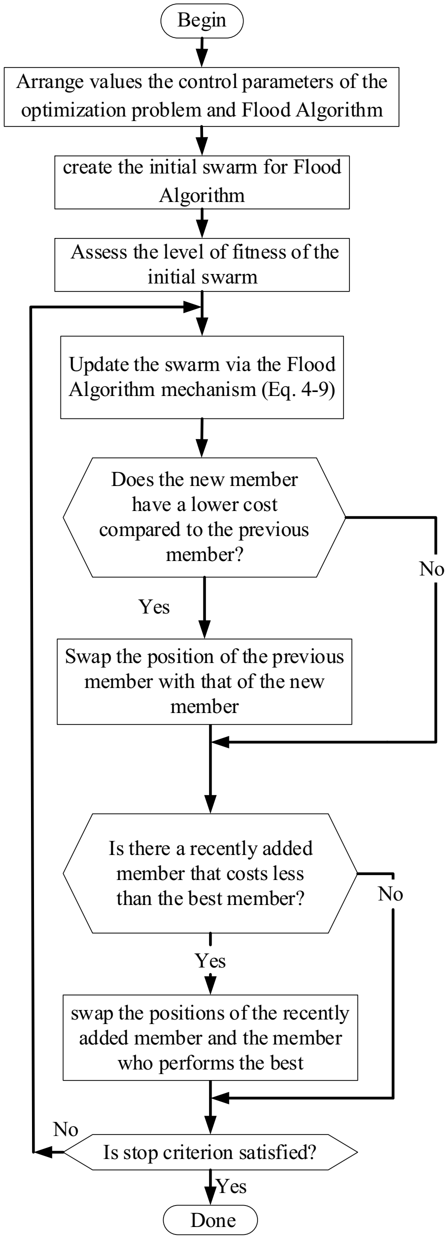


Fig. 9 The flow chart of the suggested FLA

functions, which have been used widely in the recent studies, as criteria to measure the robustness and capability of FLA. The information presented in this section outlines the top-performing solutions of each optimizer based on two distinct measures—mean and standard deviation (Std.) results. These values are computed across 30 independent runs for each optimizer.

Algorithm 1 FLA optimizer

Initialization: To arrange sizes of the initial parameters of FLA: the scaling factor Ne, iterations (generations) maximum number $Iter_{max}$, and the swarm size $Npop$ and setting the generations number $Iter = 0$ for population;

To create the accidental initial swarm $Npop(i = 1, 2, \ldots, Npop)$;
$S_i = S_{min} + rand \times (S_{max} - S_{min})$
To evaluate the fitness of the initial random swarm;
while The i **till** $Iter_{max}$ **do**
  To arrange the generations $Iter = Iter + 1$;
  for $i = 1$ to $Npop$ **do**
    $Pe_i = (\frac{f(S_i) - f_{min}}{f_{max} - f_{min}})^2$
    if $rand > rand + Pe_i$ **then**
      $S_i^{new} = S_i + \frac{Pk^{randn}}{Iter} \times (rand * (S_{max} - S_{min}) + S_{min}) \rightarrow$ (Flood flow: Chaotic and fast movement)
    else
      $S_i^{new} = S_{best} + rand * (S_j - S_i) \rightarrow$(Regular movement)
    end if
    if $f(S_i^{new}) < f(S_i)$ **then**
      $S_i = S_i^{new}$ and $f(S_{best}) = f(S_i)$;
    end if
    if $f(S_i) < f(S_{best})$ **then**
      $S_{best} = S_i$ and $f(S_{best}) = f(S_i)$;
    end if
  end for
  if $rand < Pt; Pt = |sin(\frac{rand}{Iter})|$ **then**
    for $e = 1$ to Ne **do**
      $S_e^{new} = S_{best} + rand \times (rand * (S_{max} - S_{min}) + S_{min}) \rightarrow$(The increase and decrease of water)
      if $f(S_e^{new}) < f(S_{best})$ **then**
        $S_{best} = S_e^{new}$ and $f(S_{best}) = f(S_e^{new})$;
      end if
    end for
  end if
end while

Output: Return the optimum solution that has been optimized by FLA: S_{best}

Fig. 10 The pseudo-code of FLA

3.1 CEC2005 test functions

Nowadays, designers, engineers, and researchers deal with various complex nonlinear and complex problems whose optimization functions are much more complex than traditional, conventional testing functions. Many real test functions have been proposed, such as the CEC2005 test functions [80], which have been used successfully in many papers [57]. For this reason, in this paper, only functions 1 through 14, respectively, are applied to show the efficiency of FLA, as shown in Table 3. Also, for example, Fig. 11 depicts the three-dimensional (3D) specifications with the convergence of FLA through a run for the F2, F5, and F11.

When FLA starts, the movements are abrupt and converge step by step. Search history and the path of the first search factor in the first dimension are plotted in Fig. 11 to illustrate the convergence behavior of FLA for the F2, F5, and F11. The second and third columns in Fig. 12 report the search history of the search factors and the convergence, respectively.

3.1.1 The effect of *Ne* on FLA

As previously stated, the FLA has a control parameter known as *Ne*, which can generate different values and outcomes when applied to different problems. The following analysis of this control parameter has been conducted based on a range of test functions. We have employed a system that runs 30 times for all tests for each function. The number of fitness evolutions is 300,000 [80], and all problems' dimensions are set at 30.

Under the same conditions, various values are taken into account for the *Ne* parameter in the initial test. Optimization is carried out for standard real-parameter functions for these values. The different values chosen for *Ne* are 1, 5, 10, and 20. The algorithm's population number for this section is 60. As shown in Table 4, it is clear that $Ne = 5$ can be a good choice for all functions, which we have used in this paper to do a comparative study. Note that the symbol *Mr* represents the average rank across all functions. *Nb* refers to the number of times the best rank is achieved, while *Nw* denotes the number of times the worst rank is obtained in all simulations. For a coefficient of $Ne = 5$, the FLA has a minimum value of $Mr = 2$. Moreover, this value has the lowest *Nw* when compared to other *Ne* values, making it the optimal limit.

3.1.2 FLA with the various population sizes

One of the parameters that needs to be determined by the user in FLA is the number of population, N_{pop}, which is similar to other algorithms. To find the best value for N_{pop}, we tested FLA with three different values of 30, 60, and 90 on the functions introduced above. The results of the test are presented in Table 5, and it can be concluded that choosing 60 as the population size is a suitable value for most of the functions that need to be optimized using FLA. Therefore, we set the population size to 60 in this study. By using this value, the proposed algorithm achieved a minimum value of $Mr = 1.64$. Moreover, *Nw*, which is the desirable limit, was also lower in the algorithm with a population size of 60 compared to the other tested population sizes.

In general, to show the value of any proposed new algorithm, it should be compared with standard, widely used basic algorithms in equal and fair conditions to determine the efficiency of FLA better. In this study, simulation and optimization are performed compared with several standard popular algorithms, including ABC, BBO, DE/rand/1, MPSO (modified PSO), FA, RAO1, WOA,

Table 3 The selected CEC2005 functions with $f_{\min}=0$ [57, 80, 81]

Unimodal	Limits
$F1=\sum_{i=1}^{D} y_i^2$, $o=[o_1, o_2, \ldots, o_D]$ (the shifted global optimum) for any function. And $y=x-o$ for any shifted function and $y=(x-o)*M$ for any shifted rotated function, M (orthogonal matrix)	$[-100, 100]^D$
$F2=\sum_{i=1}^{D}\left(\sum_{j=1}^{i} y_j\right)^2$	
$F3=\sum_{i=1}^{D}\left(10^6\right)^{\frac{i-1}{D-1}} y_i^2$(shifted rotated)	
$F4=\left(\sum_{i=1}^{D}\left(\sum_{j=1}^{i} y_j\right)^2\right) * (0.4\lvert N(0,1)\rvert+1)$	
$F5=\max\left\{\lvert A_i x-B_i\rvert\right\}$, A is a D*D matrix, A_i is ith row of A. And $B_i=A_i * o$. o is a D*1 vector, o_i are random number in the range [-100,100]	
Multimodal	
$F6=\sum_{i=1}^{D-1}(100(y_i^2-y_{i+1})^2+(y_i-1)^2)$, $y=x-o+1$.	
$F7=\frac{1}{4000}\sum_{i=1}^{D}\left(y_i\right)^2-\prod_{i=1}^{D}\cos\left(\frac{y_i}{\sqrt{i}}\right)+1$, $y=(x-o)*M$, $M=M'(1+0.3\lvert N(0,1)\rvert)$. M': linear transformation matrix	$[-600, 600]^D$
$F8=-20\exp(-0.2\sqrt{D^{-1}\sum_{i=1}^{D} y_i^2})-\exp\left(D^{-1}\sum_{i-1}^{D}\cos\left(2\pi y_i\right)\right)+20+e$, the $y=(x-o)*M$	$[-32, 32]^D$
$F9=\sum_{i=1}^{D}(y_i^2-10\cos(2\pi y_i)+10)$, $y=x-o$	$[-5, 5]^D$
$F10=\sum_{i=1}^{D}(y_i^2-10\cos(2\pi y_i)+10)$, $y=(x-o)*M$	
$F11=\sum_{i=1}^{D}\left(\sum_{k=0}^{k\max}\left[a^k\cos\left(2\pi b^k\left(y_i+0.5\right)\right)\right]\right)-D\sum_{k=0}^{k\max}\left[a^k\cos\left(2\pi b^k\right)\right]$, $a=0.5$ $b=3$ $k\max=20$	$[-0.5, 0.5]^D$
$F12=\sum_{i=1}^{D}\left(A_i-B_i(x)\right)^2$, $A_i=\sum_{j=1}^{D}\left(a_{ij}\sin\alpha_j+b_{ij}\cos\alpha_j\right)$, $B_i(x)=\sum_{j=1}^{D}\left(a_{ij}\sin x_j+b_{ij}\cos x_j\right)$, . a_{ij}, b_{ij} are integer random numbers in the range [-100,100], α are random numbers in the range $[-\pi,\pi]$	$[-\pi,\pi]^D$
Expanded multimodal	
$F13=F8\left(F2\left(y_1,y_2\right)\right)+\ldots+F8\left(F2\left(y_{D-1},y_D\right)\right)+F8\left(F2\left(y_D,y_1\right)\right)$, $y=x-o+1$.	$[-3, 1]^D$
$F14=F6\left(y_1,y_2\right)+\ldots+F6\left(y_{D-1},y_D\right)+F6\left(y_D,y_1\right)$, $y=(x-o)*M$.	$[-100, 100]^D$

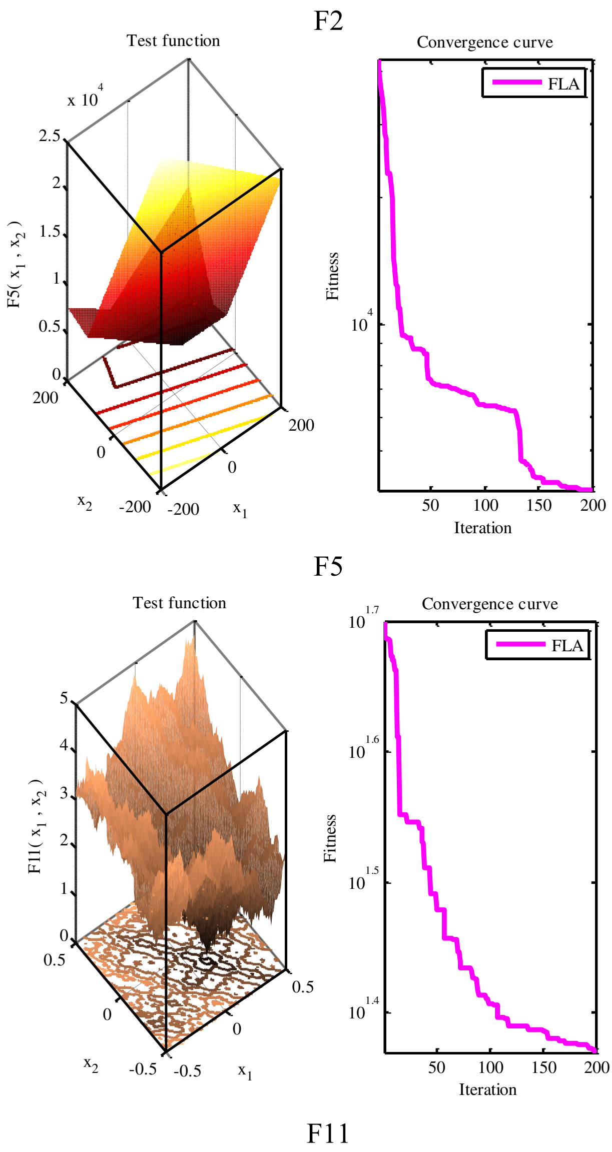


Fig. 11 A 3D trend of CEC2005 functions with the convergence of FLA for a sample run for the F2, F5, and F11

and PSO, in the first phase of recognizing FLA. We have selected the population number and variables of each algorithm based on the primary reference of the algorithm, as shown in Table 6. We use the same system for any method with 30 runs for any function with $D = 30$.

The results of optimizers are presented in Table 7, which are based on two criteria: mean and *Std.* values over 30 independent runs for each function. The

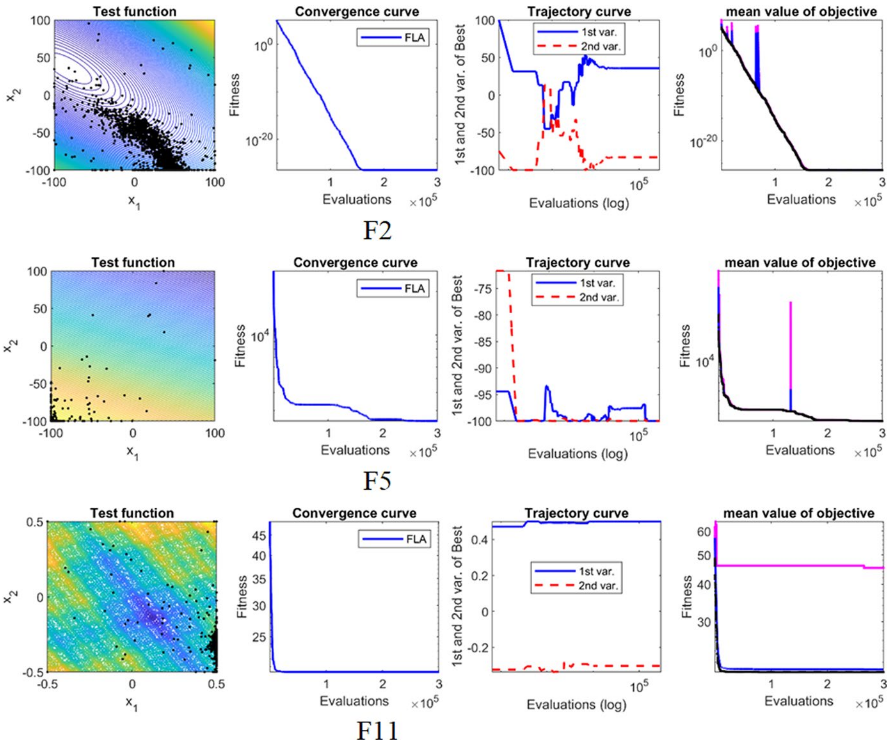


Fig. 12 Search trajectory and history in the first dimension by the first population

proposed algorithm seems to provide better optimal results than other algorithms, given the abovementioned criteria. Additionally, the last part of the table includes a comparison of different algorithms, where "+" denotes a better result than FLA, "−" represents a weaker solution, and "=" indicates an equal value. The proposed algorithm outperforms standard and basic meta-heuristic algorithms in optimizing real-parameter functions under similar conditions.

The results indicate FLA's potential to be considered a robust new emerging algorithm. FLA has achieved the best solution among the algorithms for 6 out of 14 functions and has also won over ABC, BBO, DE/rand/1, MPSO (modified PSO), FA, RAO1, WOA, and PSO algorithms for 11, 10, 9, 10, 8, 12, 14, and 14 functions, respectively.

To enhance clarity, we conducted a Friedman's test [88, 89] on the dataset presented in Table 8. This statistical analysis yielded valuable insights, including Friedman's rank (*Fr*), mean Fr (positioned one line before the final line), and the final rank (found in the last line). Notably, FLA emerged with the highest Friedman average rank, underscoring its remarkable efficacy compared to the other algorithms. The proposed algorithm, FLA, outperformed established techniques, securing its position as a leading optimization solution. Following closely, the FA and MPSO algorithms claimed the second and third ranks, respectively.

Table 4 The simulation results by FLA for any *Ne*

Functions		*Ne* = 1	*Ne* = 5	*Ne* = 10	*Ne* = 20
		Mean ± Std Rank	Mean ± Std Rank	Mean ± Std Rank	Mean ± Std Rank
Unimodal functions	F1	1.05E−28 ± 1.00E−28 4	4.63E−29 ± 7.85E−29 2	1.00E−28 ± 1.00E−28 3	**3.36E−29 ± 8.2E−29** **1**
	F2	3.73E−27 ± 1.8E−27 4	2.62E−27 ± 2.87E−27 2	3.68E−27 ± 1.37E−27 3	**2.13E−27 ± 1.27E−27** **1**
	F3	160,338 ± 106,321 4	**77,947 ± 36,867** **1**	160,325 ± 106,314 3	114,323 ± 37,128 2
	F4	**0.05377 ± 0.08234** **1**	0.05376 ± 0.1084 2	0.05382 ± 0.08058 3	0.1756 ± 0.235 4
	F5	3685 ± 1489 3	**2805 ± 1083** **1**	4162 ± 1505 4	3630 ± 1084 2
Basic multimodal functions	F6	0.6644 ± 1.6275 4	9.97E−3 ± 2.35E−1 2	0.65 ± 1.6275 3	**4.48E−03 ± 1.02E−01** **1**
	F7	0.0266 ± 0.0205 4	0.020 ± 0.0156 3	0.0176 ± 0.0132 2	**0.00862 ± 0.0051** **1**
	F8	**20.5 ± 0.1317** **1**	20.56 ± 0.1207 3	20.53 ± 0.083 2	20.59 ± 0.0622 4
	F9	124.3 ± 39.3 3	117 ± 13.19 2	128 ± 28.38 4	**107 ± 23.82** **1**
	F10	**143.6 ± 57.30** **1**	199 ± 76.07 4	185 ± 33.03 2	189 ± 35.64 3
	F11	28.09 ± 5.58 3	**23.8 ± 2.87** **1**	31.6 ± 4.00 4	27.19 ± 6.63 2
	F12	120,865 ± 183,393 3	**54,190 ± 23,674** **1**	80,854 ± 88,137 2	140,141 ± 138,535 4

Table 4 (continued)

Functions		$Ne=1$	$Ne=5$	$Ne=10$	$Ne=20$
Expanded multimodal Functions	F13	**5.56 ± 1.24**	6.0 ± 2.36	7.39 ± 2.26	8.27 ± 6.42
		1	2	3	4
	F14	**12.16 ± 1.0**	12.22 ± 0.604	12.43 ± 0.372	12.56 ± 0.5157
		1	2	3	4
Nb/ Nw/Mr	5/5/2.64	4/1/2.0	0/3/2.93	5/5/2.43	

The bold numbers indicate the best solutions for each function

Table 5 The optimal results for the CEC2005 functions by FLA with $Ne = 5$

Functions		*Npop*		
		30 Mean ± Std Rank	60 Mean ± Std Rank	90 Mean ± Std Rank
Unimodal Functions	F1	7.36E−28 ± 3.83E−28 3	4.63E−29 ± 7.85E−29 2	**0.0 ± 0.0** **1**
	F2	2.97E−26 ± 2.42E−26 3	2.62E−27 ± 2.87E−27 2	**2.21E−27 ± 2.97E−27** **1**
	F3	**67,835 ± 24,752** **1**	77,947 ± 36,867 2	122,690 ± 54,264 3
	F4	18.98 ± 21.37 3	0.05376 ± 0.1084 2	**0.01764 ± 0.0258** **1**
	F5	5386 ± 1926 3	2805 ± 1083 2	**2692 ± 1277** **1**
Basic Multimodal Functions	F6	0.6644 ± 1.63 2	**9.97E−03 ± 2.35E−01** **1**	1.32 ± 2.05 3
	F7	**0.0188 ± 0.016** **1**	0.020 ± 0.0156 2	0.0233 ± 0.0226 3
	F8	**20.44 ± 0.126** **1**	20.56 ± 0.1207 2	20.61 ± 0.1197 3
	F9	122 ± 34.9 3	117 ± 13.19 2	**85.8 ± 35.00** **1**
	F10	238 ± 32.5 3	**199 ± 76.07** **1**	200 ± 101 2
	F11	27.20 ± 4.06 2	**23.8 ± 2.87** **1**	27.6 ± 4.40 3
	F12	193,608 ± 258,652 3	**54,190 ± 23,674** **1**	71,109 ± 58,647 2
Expanded Multimodal Functions	F13	13.53 ± 8.02 3	6.0 ± 2.36 2	**5.76 ± 2.16** **1**
	F14	12.48 ± 0.922 2	**12.22 ± 0.604** **1**	12.85 ± 0.6913 3
Nb/ Nw/Mr		3/8/2.36	5/0/1.64	6/6/2.0

The bold numbers indicate the best solutions for each function

Conversely, the PSO and WOA algorithms displayed weaker performance, receiving the lowest ranks based on Friedman's analysis for the specified problems. Furthermore, the convergence trends of these optimizers were visualized in three figures, focusing on sample functions 2, 5, and 11 (Fig. 13). Noteworthy is FLA's commendable and acceptable convergence trend, showcasing its notable capability to navigate away from optimal local solutions. This outcome reinforces FLA's standing as a promising algorithm with significant potential in optimization scenarios.

The outcomes presented in the table, coupled with an exploration of convergence traits across various functions, underscore FLA's suitable and commendable superiority over foundational algorithms like ABC, BBO, and PSO. FLA distinguishes itself by orchestrating floods and turbulence within the particle mass, significantly elevating its optimization prowess. This innovative approach involves augmenting local search capabilities, removing entrapment in suboptimal local solutions, and demonstrating a favorable convergence speed. The distinctiveness of FLA's methodology positions it as a noteworthy advancement compared to essential algorithms such as ABC, BBO, DE/rand/1, MPSO (modified PSO), FA, RAO1, WOA, and PSO. This recognition is a testament to FLA's efficacy and potential to outperform established algorithms, marking it as a promising and innovative addition to the optimization field.

3.2 Comparing FLA with those of the recent algorithms

We have also done some other tests to compare the FLA and other methods based on efficiency and robustness, as shown in Table 9. The list of these new algorithms includes real-coded genetic algorithm (RCGA or GL-25) [90, 91], comprehensive learning particle swarm optimization (PSO) (CLPSO) [90, 92], self-adaptive differential evolution (DE) (SADE) [93, 94], the ensemble of parameters and mutation strategies in DE (EPSDE) [95, 96], adaptive unified DE (AUDE2) [93], fully informed PSO (FIPSO) [97, 98], orthogonal learning PSO (OLPSO) [97, 99], and static heterogeneous PSO (SHPSO) [97, 100], which are tested on CEC2005 benchmark.

According to the results, the most remarkable accuracy is observed for EPSDE and FLA in unimodal functions. The best result of the FLA is achieved on F2, F3, F4, F6, and F11, while it ranked second for functions F1, F8, and F14. Moreover, as per Table 9, CLPSO, EPSDE, and OLPSO are equally potent in giving the mean values of function F9.

Based on Table 10, FLA is superior to the rest of the compared algorithms when the average ranking-based Friedman test is adopted. OLPSO and EPSDE take the next rankings, followed by CLPSO and GL-25. Even though OLPSO and EPSDE show significant performance in some functions, they are not as efficient as the proposed FLA method.

3.3 Convergence of inspired meta-heuristics

Over the last 20 years, there has been significant progress in the performance and efficiency of meta-heuristic algorithms. However, an important question is whether these algorithms converge to the optimal solution. To address this, Gutjahr [101] has studied the convergence of these algorithms to a proposed solution X_{best}^{Iter}, which represents the best approximation found so far for the optimal solution in a given iteration Iter. Suppose the optimal value of the optimization function in the problem is as Eq. (2), f^*, assuming that the Z region is finite. In that case, we show that $f\left(X_{best}^{Iter}\right)$ converges to f^* when $Iter \to \infty$. To achieve this goal, we introduce the index function I, which is used in the following equation:

Table 6 The parameter settings of comparison algorithms on CEC2005

Algorithm	Parameter	Value
BBO [26]	The selecting of initial parameters as in [22] (http://academic.csuohio.edu/simond/bbo/);	
ABC [82]	The selecting of initial parameters as in [76] (http://mf.erciyes.edu.tr/abc/);	
PSO [83]	The selecting of initial parameters as in [78]	
DE/rand/1 [22]	Crossover constant	$CR=0.5$
	Weighting factor	$F=0.5$
	Population	$Npop=60$
MPSO [84]	Inertia weight	$w_{initial}=1$
	Inertia weight damping ratio	$w_{damp}=0.99$, $w_{Iter+1} = w_{damp} \times w_{Iter}$
	Global learning coefficient	$c_2=2.0$
	Personal learning coefficient	$c_1=1.5$
	Population	$Npop=60$
FA [85]	Uniform mutation range	$\text{delta}=0.05\times(X_{max}\text{-}X_{min})$
	Mutation coefficient damping ratio	$\text{alpha}_{\text{damp}}=0.98$ $alpha_{Iter+1} = alpha_{damp} * alpha_{Iter}$
	Population	$Npop=30$
	Mutation coefficient	$\text{alpha}_{\text{initial}}=0.2$
	Light absorption coefficient	$\text{gamma}=1$
		$m=2$
	Attraction coefficient base value	$\text{beta}_{\text{initial}}=2$
RAO1 [86]	Population	$Npop=30$
WOA [36, 87]	Scaling factor l	$[-1, 1]$
	Population	$Npop=60$
	Scaling factor a	Linear reduction from 2 to 0
	Scaling factor b	1

Table 7 Optimal results of the algorithms on CEC2005 with $D=30$

Functions		DE/rand/1	ABC	WOA	MPSO	BBO	RAO1	FA	PSO	FLA ($Npop=60$)
		Mean ± Std Winer	Mean ± Std Rank, Winner	Mean ± Std Winner	Mean ± Std Winner	Mean ± Std Winner	Mean ± Std Winner	Mean ± Std Winner	Mean ± Std Winner	Mean ± Std
Unimodal Functions	F1	**0.00E+00** **0.00E+00** +	5.79E−16 1.4E−16 –	3.87E−01 1.89E−01 –	3.57E−23 9.57E−23 –	0.0231 0.0064 –	5.05E−29 9.52E−29 –	4.87E−11 5.08E−12 –	605.4440 224.4084 –	4.63E−29 7.85E−29
	F2	1.87E+03 7.82E+02 –	3700 911 –	5.20E+04 6.48E+03 –	4.03E−18 5.39E−18 –	11.03 2.17 –	2.59E−10 4.65E−10 –	4.86E−05 5.11E−05 –	2305 440 –	**2.62E−27** **2.87E−27**
	F3	1.14E+08 2.81E+07 –	11,840,888 2,287,545 –	2.30E+07 9.84E+06 –	1.04E+06 3.16E+05 –	2,552,456 851,496 –	1.73E+07 6.65E+06 –	1.30E+06 4.00E+05 –	16,119,112 10,393,277 –	**77,947** **36,867**
	F4	4.99E+03 1.33E+03 –	45,272 4571 –	1.24E+05 3.16E+04 –	2.05E+02 2.62E+02 –	45.34 7.57 -	1.42E+02 2.38E+02 –	1.30E+02 8.02E+01 –	11,059 4010 –	**0.05376** **0.1084**
	F5	**2.45E+02** **1.97E+02** +	9348 1252 –	1.73E+04 3.92E+03 –	4.18E+03 1.33E+03 –	4662 590 –	1.67E+03 1.43E+03 +	3.11E+03 1.02E+03 –	6621 2252 –	2805 1083
Basic Multi-modal Functions	F6	2.42E+01 1.73E+01 –	4.445 2.48 –	6.53E+03 4.99E+03 –	4.60E+01 5.37E+01 –	169 31.68 –	1.95E+00 3.70E+00 –	6.94E+01 7.29E+01 –	21,371,486 23,818,856 –	**9.97E−03** **2.35E−01**
	F7	**1.84E−10** **2.66E−10** +	0.0214 0.012 –	1.66E+00 3.27E−01 –	3.07E−02 2.11E−02 –	0.9050 0.1058 –	2.36E−02 1.65E−02 –	4.18E−03 7.34E−03 +	137 79.15 –	0.020 0.0156
	F8	2.09E+01 6.38E−02 –	20.92 0.038 –	2.08E+01 1.14E−01 –	2.09E+01 4.65E−02 –	20.61 0.125 –	2.09E+01 5.81E−02 –	2.10E+01 4.22E−02 –	20.67 0.1027 –	**20.56** **0.1207**

Table 7 (continued)

Functions		DE/rand/1	ABC	WOA	MPSO	BBO	RAO1	FA	PSO	FLA ($Npop = 60$)
	F9	8.26E+01 6.87E+00 +	**4.44E−15** **4.59E−15** **+**	2.53E+02 4.30E+01 –	1.32E+02 3.22E+01 –	34.95 11.06 +	1.73E+02 2.08E+01 –	7.77E+01 1.76E+01 +	164.66 37.25 –	117 13.19
	F10	1.79E+02 9.00E+00 +	268 53.75 –	4.23E+02 8.25E+01 –	1.19E+02 3.95E+01 +	75.17 10.82 +	2.37E+02 1.73E+01 –	6.34E+01 1.22E+01 +	206.83 68.23 –	199 76.07
	F11	3.97E+01 5.18E−01 –	27.78 1.57 –	3.49E+01 3.16E+00 –	2.04E+01 2.82E+00 +	24.74 4.12 –	3.91E+01 2.26E+00 –	**1.52E+01** **3.84E+00** **+**	26.75 4.034 –	23.8 2.87
	F12	9.17E+04 6.34E+04 –	12,945 4066 +	1.58E+05 9.70E+04 –	**7.21E+03** **7.41E+03** **+**	8956 6976 +	4.88E+04 4.90E+04 +	1.87E+04 2.25E+04 +	89,249 39,012 –	54,190 23,674
Expanded Multimodal Functions	F13	1.21E+01 8.21E−01 –	**0.444** **0.059** **+**	2.05E+01 5.99E+00 –	4.02E+00 1.80E+00 +	2.37 0.246 +	1.67E+01 1.68E+00 –	3.57E+00 9.55E−01 +	10.79 1.64 –	6.0 2.36
	F14	1.35E+01 1.47E−01 –	13.01 0.191 –	1.32E+01 4.23E−01 –	1.27E+01 3.96E−01 –	12.65 0.997 –	1.34E+01 2.23E−01 –	1.24E+01 6.77E−01 –	13.16 0.105 -	**12.22** **0.604**
±/=		5/**9**/0	3/**11**/0	0/**14**/0	4/**10**/0	4/**10**/0	2/**12**/0	6/**8**/0	0/**14**/0	––
Nb/Nw		3/3	2/**0**	0/7	1/**0**	0/**0**	0/**0**	2/1	0/3	**6/0**

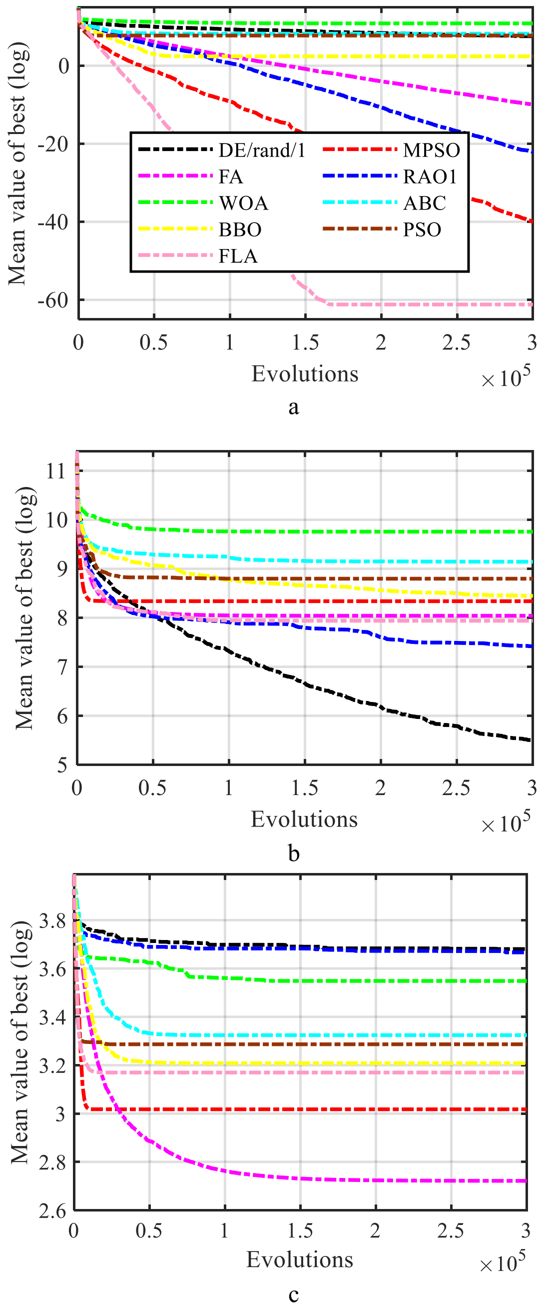
The bold numbers indicate the best solutions for each function

Table 8 The results for the Friedman' test on CEC2005

F		DE/rand/1	ABC	WOA	MPSO	BBO	RAO1	FA	PSO	FLA
		Fr	Fr	Fr	Fr	Fr	Fr	Fr	Fr	Fr
Unimodal Functions	f_1	1	5	8	4	7	3	6	9	2
	f_2	6	8	9	2	5	3	4	7	1
	f_3	9	5	8	2	4	7	3	6	1
	f_4	6	8	9	5	2	4	3	7	1
	f_5	1	8	9	5	6	2	4	7	3
Basic Multimodal Functions	f_6	4	3	8	5	7	2	6	9	1
	f_7	1	4	8	6	7	5	2	9	3
	f_8	6.5	6.5	4	6.5	2	6.5	9	3	1
	f_9	4	1	9	6	2	8	3	7	5
	f_{10}	4	8	9	3	2	7	1	6	5
	f_{11}	9	6	7	2	4	8	1	5	3
	f_{12}	8	3	9	1	2	5	4	7	6
Expanded Multimodal Functions	f_{13}	7	1	9	4	2	8	3	6	5
	f_{14}	9	5	7	3.5	3.5	8	2	6	1
Mean Fr		5.3929	5.1071	8.0714	3.9286	3.9643	5.4643	3.6429	6.7143	**2.7143**
Final rank		6	5	9	3	4	7	2	8	**1**

The bold numbers indicate the best solutions for each function

Fig. 13 Convergence trends of optimizers for the **a** second (F2), **b** fifth (F5), and **c** eleventh test benchmarks (F11)



$$F^{Iter} = I\left(f\left(X_{best}^{Iter}\right)\right) = \begin{cases} 1\text{if } f\left(X_{best}^{Iter}\right) = f^* \\ 0\text{otherwise} \end{cases} \tag{10}$$

If the optimal solution is found in iteration *Iter*, then $F^{Iter}=1$; and $F^{Iter}=0$ otherwise. Also, the time of the first hit is shown by t_1 and is defined as follows:

$$t_1 = \min\left\{Iter \geq 1 : F^{Iter} = 1\right\} \tag{11}$$

If it is assumed that the mathematical expectation is represented by E and the probability function is represented by P_r, then:

$$k_{Iter} = E\left(F^{Iter}\right) = P_r\left(F^{Iter} = 1\right) = P_r\left(t_1 \leq Iter\right), Iter = 1, 2, ..., Iter_{max} \tag{12}$$

So, $f\left(X_{best}^{Iter}\right)$ converges to f^* only and only if $k_{Iter} \to 1$ when $Iter \to \infty$. Assuming a simplified random search and if the probability of occurrence of an optimal solution in an iteration of the algorithm is $P_{Iter}\left(0 \leq P_{Iter} \leq 1\right)$, and regarding the independency and standard geometric distribution, we have:

$$\begin{aligned} &P_r\left(F^{Iter} = 0\right) = \left(1 - P_{Iter}\right)^{Iter} \\ &\lim_{iter \to +\infty} P_r\left(F^{Iter} = 0\right) = 0 \end{aligned} \tag{13}$$

Therefore, $k_{Iter} \to 1$ when $Iter \to \infty$. To put it another way, convergence to the optimal value when $Iter \to \infty$ occurs with a probability of 1. Hence, according to what has been said before, when the algorithm includes only random search, its convergence when $Iter \to \infty$ occurs with a probability of 1. When examining the entire area of the optimization problem through the use of Fluid-Particle Algorithms (FLA) with either regular or chaotic water particle movements, the value of P_{Iter}, i.e., the probability of finding an optimal solution within a single iteration, either increases or remains the same when compared to the probability of finding an optimal solution through random search. Based on this observation and similar reasoning, it can be concluded that the proposed FLA has a convergence. Simply put, when $Iter \to \infty$ the FLA will converge to the optimal solution of the problem with a probability of 1.

4 A comparative and comprehensive study of the FLA

In this part, the efficiency of FLA is compared with modern inspired optimizers through the real-parameter functions of CEC2014 and by applying them to twelve common optimization problems in engineering.

4.1 Real-parameter optimization on the CEC2014 test functions

The CEC2014 suite is used to analyze the robustness and capability of the FLA in terms of exploration, exploitation, and local optimum avoidance. Four classes are

Table 9 The simulation results for the functions for the different modified methods on CEC2005

F	GL-25	sHPSO	OLPSO	FIPS	AuDE2	CLPSO	jDE	EPSDE	FLA
	Mean Std Winner								Mean Std
F1	5.60E−27	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.29E+01	**0.00E+00**	5.38E−12	**0.00E+00**	4.63E−29
	1.76E−26	**0.00E+00**	**0.00E+00**	**0.00E+00**	5.41E+01	**0.00E+00** +	2.12E−12	**0.00E+00**	7.85E−29
	-	+	+	+	–		–	+	
F2	4.04E+01	1.44E−02	1.38E+01	7.79E+01	1.27E+04	8.40E+02	1.23E+02	4.25E−10	**2.62E−27**
	6.28E+01	7.10E−02	8.33E+00	2.71E+01	3.24E+03	1.90E+02	2.00E+01	2.12E−09	**2.87E−27**
	-	–	–	–	–	–	–	–	
F3	2.19E+06	8.75E+05	1.60E+07	2.45E+07	4.31E+07	1.42E+07	4.21E+06	1.37E+06	**7.79E+04**
	1.08E+06	5.34E+05	7.04E+06	6.29E+06	1.81E+07	4.19E+06	9.02E+05	4.97E+06	**3.69E+04**
	–	–	–	–	–	–	–	–	
F4	9.07E+02	2.02E+04	2.18E+03	1.15E+03	1.81E+04	6.99E+03	5.99E+03	2.30E+05	**5.38E−02**
	4.25E+02	9.94E+03	1.09E+03	3.73E+02	4.17E+03	1.73E+03	1.36E+03	6.26E+05	**1.08E−01**
	-	–	–	–	–	–	–	–	
F5	2.51E+03	6.94E+03	3.30E+03	2.22E+03	1.28E+04	3.86E+03	5.14E+03	**9.15E+02**	2.81E+03
	1.96E+02	1.43E+03	3.75E+02	5.14E+02	1.92E+03	4.35E+02	7.36E+02	**5.45E+02**	1.08E+03
	-	–	–	+	–	–	–	+	
F6	2.15E+01	1.15E+02	2.07E+01	3.77E+01	7.94E+06	4.16E+00	3.57E+01	3.18E−01	**9.97E−03**
	1.17E+00	2.29E+02	2.50E+01	3.50E+01	1.27E+07	3.48E+00	3.46E+00	1.10E+00	**2.35E−01**
	–	–	–	–	–	–	–	–	
F7	2.78E − 02	4.00E−02	**1.00E−02**	3.00E−02	9.94E+01	4.51E−01	6.22E−02	1.22E−02	2.00E−02
	3.62E − 02	4.00E−02	**1.00E−02**	2.00E−02	4.79E+01	8.47E−02	1.97E−02	1.34E−02	1.56E−02
	–	–	+	–	–	–	–	+	
F8	2.09E+01	**2.02E+01**	2.10E+01	2.09E+01	2.09E+01	2.09E+01	2.09E+01	2.09E+01	2.06E+01
	5.94E − 02	**1.90E−01**	8.00E−02	6.00E−02	5.72E−02	4.41E−02	4.59E−02	6.24E−02	1.21E−01
	-	+	–	–	–	–	–	–	
F9	2.45E+01	8.25E+01	**0.00E+00**	5.71E+01	9.14E+01	**0.00E+00**	4.34E+01	**0.00E+00**	1.17E+02
	7.35E+00	2.44E+01	**0.00E+00**	1.46E+01	2.08E+01	**0.00E+00**	6.04E+00	**0.00E+00**	1.32E+01
	+	+	+	+	+	+	+	+	

Table 9 (continued)

F	GL-25	sHPSO	OLPSO	FIPS	AuDE2	CLPSO	jDE	EPSDE	FLA
F10	1.42E+02	2.43E+02	1.09E+02	1.78E+02	1.66E+02	1.04E+02	1.92E+02	**6.14E+01**	1.99E+02
	6.45E+01	8.92E+01	1.89E+01	9.25E+00	3.00E+01	1.53E+01	1.35E+01	**9.02E+00**	7.61E+01
	+	–	+	+	+	+	+	**+**	
F11	3.27E+01	3.23E+01	2.51E+01	3.84E+01	2.98E+01	2.60E+01	2.79E+01	3.57E+01	**2.38E+01**
	7.79E+00	3.84E+00	3.14E+00	1.52E+00	2.95E+00	1.63E+00	1.18E+00	2.87E+00	**2.87E+00**
	–	–	–	–	–	–	–	–	
F12	6.53E+04	2.43E+04	**1.22E+04**	5.62E+04	9.49E+04	1.79E+04	3.78E+04	5.75E+04	5.42E+04
	4.69E+04	2.66E+04	**5.42E+03**	2.00E+04	4.07E+04	5.24E+03	5.63E+03	1.05E+04	2.37E+04
	–	+	**+**	–	–	+	+	–	
F13	6.23E+00	6.14E+00	**1.86E+00**	1.25E+01	6.14E+00	2.06E+00	6.04E+00	2.52E+00	6.00E+00
	4.88E+00	2.31E+00	**2.80E−01**	9.60E−01	1.72E+00	2.15E−01	5.82E−01	1.77E−01	2.36E+00
	-	–	**+**	–	–	+	–	+	
F14	1.31E+011.84E − -01	1.31E+01	1.31E+01	1.31E+01	**1.19E+01**	1.28E+01	1.29E+01	1.34E+01	1.22E+01
	-	3.90E−01	2.00E−01	2.10E−01	**3.99E−01**	2.48E−01	1.56E−01	1.98E−01	6.04E−01
		–	–	–	**+**	–	–	–	
$\pm/=$	2/12/0	4/10/0	6/8/0	4/10/0	3/11/0	5/9/0	11/3/0	6/8/0	–
Nb	0	2	5	1	1	2	0	4	5

The bold numbers indicate the best solutions for each function

Table 10 Mean rankings computed by Friedman test on CEC2005

F	GL-25 *Fr*	sHPSO *Fr*	OLPSO *Fr*	FIPS *Fr*	AuDE2 *Fr*	CLPSO *Fr*	jDE *Fr*	EPSDE *Fr*	FLA *Fr*
F1	7	**3**	**3**	**3**	9	**3**	8	**3**	6
F2	5	3	4	6	9	8	7	2	**1**
F3	4	2	7	8	9	6	5	3	**1**
F4	2	8	4	3	7	6	5	9	**1**
F5	3	8	5	2	9	6	7	**1**	4
F6	5	8	4	7	9	3	6	2	**1**
F7	4	6	**1**	5	9	8	7	2	3
F8	5.50	**1**	9	5.50	5.50	5.50	5.50	5.50	2
F9	4	7	**2**	6	8	**2**	5	**2**	9
F10	4	9	3	6	5	2	7	**1**	8
F11	7	6	2	9	5	3	4	8	**1**
F12	8	3	**1**	6	9	2	4	7	5
F13	8	6.5	**1**	9	6.5	2	5	3	4
F14	6.5	6.5	6.5	6.5	**1**	3	4	9	2
Fr	5.2143	5.5000	3.7500	5.8571	7.2143	4.2500	5.6786	4.1071	**3.4286**
Total rank	5	6	2	8	9	4	7	3	**1**

The bold numbers indicate the best solutions for each function

discussed for 30 CEC2014 test functions [102] as real-time functions. The same is followed in this paper so that a comparison between the FLA and some other algorithms is made.

4.1.1 Effect of population size on FLA

The FLA is free from control parameters, but the sizes of iterations and swarm are set manually. Four populations of sizes 30, 45, 60, and 90 are used to test the FLA. The tests are performed using only one system, and the program is run 30 times, and as any function has 30 dimensions, the number of evolutions is 300,000.

Mean, best, and Std. quantities have been reported in Tables 11, 12, and 13; *Nb*, *Nw*, and Mean are presented in the tables. It is also observed that when the size of the population is small, more suitable unimodal functions are found. On the other side, populations with large sizes provide more acceptable multimodal functions. However, a compromised solution is obtained if the population size is between 30 and 90. As the tables report, the paper adopts a population size = 90.

As is observed, FLA succeeds in discovering the optimal location without even trapping in the local optimum points, and its convergence speed is highly acceptable when applied to functions of CEC2014. So, one can trust this algorithm's reliability and robustness. Table 10 and Fig. 14 provide the convergence data for 5-example test functions.

4.1.2 A comparative study on the CEC2014

The selected test functions with dimensions of 30 are run 30 times, so each function evolved 300,000 times. *Npop* for FLA has been chosen 90, while for the rest of the algorithms has been chosen 60.

The control parameters used for each algorithm are selected based on the original version of the algorithm as well as its modified versions. The value of these control parameters for any method is briefly described in Table 14.

Table 15 lists the obtained results, while Table 16 reports the values of Friedman mean ranks and final ranks. As shown in these tables, the FLA performs on the test functions and is superior to its counterparts in most cases. As the tables report, the algorithms are not the same, and their successful performance on test functions differs. The results also highlight the unusual behavior of the FLA in terms of exploration and exploitation potential.

The results of Table 16 prove that the FLA is robust and trustable when compared with its counterparts and provides the lowest rank of Friedman mean. Consequently, the FLA is ranked first as it never provides the worst result in comparison to its meta-heuristic.

4.1.3 Comparison with the recent optimization methods on CEC2014

Table 17 compares FLA, and some recent optimization methods with a dimension of 30 applied to the CEC2014 functions. The list of algorithms includes CMA-ES [105, 106], LJA (Jaya with Levy flight) [107], MFO [34, 108], CS [109, 110], TOGPEAe (Gray Prediction Evolutionary Algorithm based on the TOBL) [111], SOS [112, 113], MG-SCA (a Memory-Guided Sine–Cosine Algorithm (SCA)) [108], m-SCA (a new hybrid SCA) [114], NIWTLBO (a TLBO nonlinear control parameters) [115], mTLBO (a modified TLBO) [115, 116], and FIPS-URing [98, 117].

The signs +, −, and = in Table 17 show the better, worse, and similar behavior or result of the FLA compared to other algorithms. According to the results, the FLA's significant performance for the first four functions is noticeable.

Tables 18 and 19 list the results of various optimization algorithms, including the proposed FLA. One can easily judge the competency of the provided algorithms based on the results in these tables. As is observed, the FLA can be assumed as the most robust and trustable algorithm. However, there is close competition between the FLA and TOGPEAe.

The results of Table 17 show that the optimum solutions are obtained via FLA in seven functions. While, according to the results, the novel CMA-ES and FIPS-URing present the best results in some cases, they are not as competent as the suggested algorithm. As per the findings, LJA, MFO, SOS, and MG-SCA are the weakest algorithms in this article's described optimization methods.

In case Wilcoxon's test is adopted for comparison purposes, Table 19 compares *p* values of various optimization algorithms discussed here. According to Table 18, FLA gives more acceptable results and ranks first. Its closest rival is NIWTLBO, although the latter is not as strong as the FLA.

Table 11 Mean optimal results (Mean) of FLA on the CEC2014

Function		*Npop* = 30 *Fr*	*Npop* = 45 *Fr*	*Npop* = 60 *Fr*	*Npop* = 90 *Fr*
F1	Unimodal	**1.20E+04** **1**	1.86E+04 2	2.66E+04 3	3.26E+04 4
F2		**7.58E−15** **1**	1.71E−14 2	2.94E−14 3	1.46E−13 4
F3		**8.70E−13** **1**	1.26E−08 2	1.67E−06 3	2.07E−04 4
F4	Simple Multimodal	1.37E+01 4	**6.35E+00** **1**	1.06E+01 3	8.46E+00 2
F5		**2.02E+01** **1**	2.03E+01 2	2.04E+01 3.5	2.04E+01 3.5
F6		2.80E+01 4	2.48E+01 3	2.28E+01 2	**2.16E+01** **1**
F7		2.46E−02 4	1.99E−02 2	2.01E−02 3	**1.14E−02** **1**
F8		1.28E+02 4	1.14E+02 3	1.07E+02 2	**9.10E+01** **1**
F9		1.58E+02 4	1.39E+02 3	1.24E+02 2	**1.14E+02** **1**
F10		3.61E+03 4	3.07E+03 3	2.99E+03 2	**2.71E+03** **1**
F11		3.80E+03 4	3.94E+03 3	3.72E+03 2	**3.45E+03** **1**
F12		**6.38E−01** **1**	7.18E−01 2	9.03E−01 4	7.70E−01 3
F13		5.44E−01 4	5.05E−01 3	4.37E−01 2	**3.92E−01** **1**
F14		5.52E−01 4	5.35E−01 2.5	**5.12E−01** **1**	5.35E−01 2.5
F15		2.91E+01 4	1.72E+01 3	1.34E+01 2	**7.83E+00** **1**
F16		1.23E+01 4	1.15E+01 2.5	1.15E+01 2.5	**1.14E+01** **1**
F17	Hybrid	**4.62E+03** **1**	5.25E+03 2	6.29E+03 3	7.83E+03 4
F18		**7.93E+03** **1**	1.19E+04 4	1.09E+04 3	9.29E+03 2
F19		1.81E+01 4	**1.39E+01** **1**	1.53E+01 3	1.41E+01 2
F20		**3.69E+02** **1**	3.81E+02 2	3.99E+02 4	3.87E+02 3
F21		4.91E+03 2	5.22E+03 3	**4.82E+03** **1**	6.59E+03 4
F22		6.51E+02 4	**6.07E+02** **1**	6.15E+02 2	6.28E+02 3

Table 11 (continued)

Function		*Npop* = 30 *Fr*	*Npop* = 45 *Fr*	*Npop* = 60 *Fr*	*Npop* = 90 *Fr*
F23	Composition	3.15E+02 2.5	3.15E+02 2.5	3.15E+02 2.5	3.15E+02 2.5
F24		2.53E+02 4	2.50E+02 3	2.47E+02 2	**2.45E+02** **1**
F25		2.08E+02 3	2.07E+02 2	2.09E+02 4	**2.06E+02** **1**
F26		1.01E+02 3.5	1.01E+02 3.5	1.00E+02 1.5	1.00E+02 1.5
F27		1.14E+03 4	1.06E+03 3	**9.15E+02** **1**	9.71E+02 2
F28		1.97E+03 4	1.68E+03 2	1.70E+03 3	**1.60E+03** **1**
F29		2.56E+07 4	1.74E+07 3	**1.03E+07** **1**	1.39E+07 2
F30		9.74E+03 4	4.01E+03 2	4.46E+03 3	**3.85E+03** **1**
Nb/*Nw*/Mean		8/18/3.0333	3/**1**/2.4333	4/3/2.4667	**13**/5/**2.0667**
Final rank		4	2	3	**1**

The bold numbers indicate the best solutions for each function

4.1.4 The effect of population size on FLA performance with $D = 50$

We have conducted tests to explore the impact of different populations on our algorithm, utilizing five distinct populations of 30, 45, 60, 75, and 90. Tables 20 and 21 contain the simulation results for this section. The most favorable results were achieved with populations of 75 and 90, with the former slightly outperforming the latter. Conversely, the least favorable outcomes were observed with a population of 30. However, it is noteworthy that, overall, the results across different populations exhibit marginal variations, especially between populations of 75 and 90. Populations of 30, 45, and 60 also demonstrated relatively similar outcomes across the majority of test functions.

Subsequently, we have compiled the optimal solutions obtained from the best-performing approaches in Table 22. Notably, populations of 90, achieving the best solutions 14 times, and populations of 30, with nine instances of superior outcomes, stand out as the leading performers in this section of the paper. Although the specifics of these results are outlined in Table 21, it is important to emphasize that, overall, the differences in outcomes across various populations remain statistically insignificant, particularly concerning the best solutions for the majority of test functions. These findings underscore the robustness and consistency of our algorithm across diverse population sizes.

Furthermore, Fig. 15 illustrates a comparison of the convergence rates for the test functions, offering insights into the impact of varying population sizes in FLA with $D = 50$ and NFEs = 5.00E+05. In general, it can be asserted that the

Table 12 Best statistical results (Best) of FLA on the CEC2014

Function		*Npop* = 30 *Fr*	*Npop* = 45 *Fr*	*Npop* = 60 *Fr*	*Npop* = 90 *Fr*
*F*1	Unimodal	**2.49E+03** **1**	3.28E+03 2	5.59E+03 3	6.24E+03 4
F2		0.00E+00 2	0.00E+00 2	0.00E+00 2	2.84E−14 4
F3		**5.68E−14** **1**	1.14E−13 2	7.77E−11 3	1.59E−08 4
F4	Simple Multimodal	2.76E−03 4	3.88E−04 3	4.80E−05 2	**5.69E−06** **1**
F5		2.00E+01 2.5	2.00E+01 2.5	2.00E+01 2.5	2.00E+01 2.5
F6		2.03E+01 4	1.58E+01 2	1.68E+01 3	**1.37E+01** **1**
F7		0.00E+00 2.5	0.00E+00 2.5	0.00E+00 2.5	0.00E+00 2.5
F8		7.86E+01 4	5.87E+01 2	6.87E+01 3	**5.21E+01** **1**
F9		1.03E+02 4	9.95E+01 3	6.77E+01 2	**5.57E+01** **1**
F10		2.20E+03 4	1.54E+03 3	**1.02E+03** **1**	1.43E+03 2
F11		2.71E+03 4	2.35E+03 2	2.48E+03 3	**2.26E+03** **1**
F12		1.74E−01 2	1.83E−01 3	4.12E−01 4	**1.25E−01** **1**
F13		3.08E−01 4	2.37E−01 3	2.21E−01 2	**2.11E−01** **1**
F14		2.01E−01 3	**1.86E−01** **1**	2.14E−01 4	1.96E−01 2
F15		8.99E+00 4	7.19E+00 2	8.94E+00 3	**3.98E+00** **1**
F16		1.09E+01 4	**9.78E+00** **1**	1.03E+01 3	1.01E+01 2
F17	Hybrid	**1.35E+03** **1**	1.52E+03 2	1.73E+03 3	2.44E+03 4
F18		**1.62E+02** **1**	2.89E+02 2	6.09E+02 4	3.22E+02 3
F19		9.18E+00 4	7.61E+00 3	7.55E+00 2	**7.35E+00** **1**
F20		**1.22E+02** **1**	1.48E+02 2	1.54E+02 3	1.91E+02 4
F21		1.09E+03 2	**1.08E+03** **1**	1.25E+03 3	1.99E+03 4
F22		1.47E+02 2	2.27E+02 4	**6.32E+01** **1**	1.97E+02 3

Table 12 (continued)

Function		*Npop* = 30 *Fr*	*Npop* = 45 *Fr*	*Npop* = 60 *Fr*	*Npop* = 90 *Fr*
F23	Composition	3.15E+02 2.5	3.15E+02 2.5	3.15E+02 2.5	3.15E+02 2.5
F24		2.43E+02 4	2.37E+02 3	2.29E+02 2	**2.28E+02** **1**
F25		2.03E+02 2.5	2.03E+02 2.5	2.03E+02 2.5	2.03E+02 2.5
F26		1.00E+02 2.5	1.00E+02 2.5	1.00E+02 2.5	1.00E+02 2.5
F27		4.01E+02 2.5	4.01E+02 2.5	4.01E+02 2.5	4.01E+02 2.5
F28		1.26E+03 4	1.01E+03 1.5	1.07E+03 3	1.01E+03 1.5
F29		1.12E+03 1.5	1.12E+03 1.5	1.13E+03 3	1.27E+03 4
F30		1.26E+03 3	1.17E+03 2	1.47E+03 4	**8.55E+02** **1**
Nb		5	3	2	11

The bold numbers indicate the best solutions for each function

proposed algorithm exhibits commendable and acceptable convergence characteristics for the majority of test functions. However, it is essential to note that our proposed algorithm is a novel and pioneering approach, leaving room for further enhancements. The algorithm demonstrates promising convergence properties across various test functions, though continuous refinement efforts can be directed toward its continuous improvement.

In this section, we will evaluate the performance of the proposed FLA optimizer for dimensions of 50. We have scrutinized the efficacy of FLA in comparison to the obtained optimal results of the popular algorithms in the recent studies such as ICA [118], BA [119], FPA [119], FIPSO [81], HHO [118], SAP-Rao [118], GWO [118], and DE/rand/1 [120]. The coordinates and parameter settings of these algorithms for this segment are presented in the main reference [81, 118–120]. A succinct summary of the results, including the mean values, Std., and the Friedman test, is encapsulated in Table 23.

Upon examining the results of this table, the optimization prowess and competitive performance of the proposed FLA optimizer are convincingly demonstrated in comparison to eight well-known and widely used algorithms. The FLA optimizer excels over all counterparts across a majority of test functions with high dimensions. Notably, this comparison shows that the FIPSO and FPA algorithms emerge as the strongest competitors following the FLA optimizer. They have outperformed the proposed optimization method for 12 and 10 test functions, respectively, although each has fallen short for 18 and 17 test functions, respectively. Additionally, other algorithms have succumbed between 22 to 28 instances (for DE/rand/1 and BA, respectively) to the proposed method, showcasing the consistent superiority of the proposed FLA optimizer.

Table 13 Standard deviation (Std.) statistical results of FLA on the CEC2014

Function		*Npop* = 30 *Fr*	*Npop* = 45 *Fr*	*Npop* = 60 *Fr*	*Npop* = 90 *Fr*
F1	Unimodal	5.56E+03 1	1.19E+04 2	1.39E+04 3	2.45E+04 4
F2		1.48E−14 3	1.42E−14 2	1.18E−14 1	3.20E−13 4
F3		1.72E−12 1	4.84E−08 2	4.57E−06 3	8.97E−04 4
F4	Simple Multimodal	3.28E+01 4	1.93E+01 1	2.41E+01 3	2.19E+01 2
F5		1.70E−01 3	1.48E−01 2	2.13E−01 4	1.37E−01 1
F6		4.61E+00 4	4.16E+00 2	3.68E+00 1	4.37E+00 3
F7		2.06E−02 3	3.40E−02 4	1.78E−02 2	1.18E−02 1
F8		3.50E+01 4	2.95E+01 3	2.14E+01 2	2.12E+01 1
F9		4.02E+01 4	3.33E+01 3	3.05E+01 1	3.30E+01 2
F10		7.55E+02 3	7.14E+02 2	9.13E+02 4	6.47E+02 1
F11		6.39E+02 2	9.19E+02 4	7.28E+02 3	5.92E+02 1
F12		3.39E−01 3	3.36E−01 2	2.99E−01 1	4.51E−01 4
F13		1.46E−01 4	1.36E−01 3	1.33E−01 2	1.12E−01 1
F14		3.03E−01 4	2.32E−01 1	2.72E−01 3	2.38E−01 2
F15		1.54E+01 4	7.67E+00 3	4.01E+00 2	3.12E+00 1
F16		6.80E−01 2.5	7.94E−01 4	6.80E−01 2.5	6.76E−01 1
F17	Hybrid	2.57E+03 1	2.60E+03 2	3.78E+03 3	4.12E+03 4
F18		7.93E+03 1	9.36E+03 3	9.94E+03 4	9.11E+03 2
F19		2.21E+01 4	1.10E+01 1	1.54E+01 2	1.94E+01 3
F20		1.26E+02 2	1.52E+02 4	1.44E+02 3	1.22E+02 1
F21		4.16E+03 2	4.40E+03 3	2.71E+03 1	4.77E+03 4
F22		2.15E+02 1	2.17E+02 3	2.49E+02 4	2.16E+02 2

Table 13 (continued)

Function		*Npop* = 30 *Fr*	*Npop* = 45 *Fr*	*Npop* = 60 *Fr*	*Npop* = 90 *Fr*
F23	Composition	1.11E−13 4	5.78E−14 2	5.78E−14 2	5.78E−14 2
F24		1.04E+01 4	7.18E+00 3	6.72E+00 2	4.79E+00 1
F25		4.26E+00 3	3.91E+00 2	8.91E+00 4	2.90E+00 1
F26		1.51E−01 4	1.38E−01 3	1.10E−01 1	1.14E−01 2
F27		2.32E+02 2	2.29E+02 1	3.54E+02 4	2.33E+02 3
F28		4.17E+02 2	3.46E+02 1	4.60E+02 4	4.27E+02 3
F29		3.21E+07 4	1.81E+07 3	1.77E+07 2	1.65E+07 1
F30		2.36E+04 4	1.84E+03 1	2.20E+03 3	2.16E+03 2
Mean		2.9167	2.4000	2.5500	**2.1333**

The bold numbers indicate the best solutions for each function

Through an examination of the average Friedman rank for each individual test function, once again, the proposed method has secured the top and most favorable rank, while GWO has obtained the least favorable and final rank. It is noteworthy that GWO, renowned as one of the most popular and widely used modern algorithms in recent years, has achieved the poorest performance among the algorithms considered.

4.1.5 The effect of population size on FLA performance with $D = 100$

In this segment of the simulation, we have undertaken optimization efforts to enhance the efficiency of FLA across three distinct populations: 30, 60, and 90, with a focus on large-scale and real-dimensional scenarios involving 100 measurements. Subsequently, we have synthesized the optimal solutions derived from different populations, presenting them comprehensively in Table 24. Noteworthy achievements include populations of 60, showcasing the best solutions 15 times, and populations of 90, exhibiting superior outcomes in 12 instances, emerging as the leading performers in this research section. While the detailed results are outlined in the table, it is essential to emphasize that, overall, the observed differences in outcomes between the various populations of 60 and 90 remain statistically insignificant, particularly concerning the optimal solutions for the majority of test functions. These findings underscore the resilience and consistency of our algorithm across diverse population sizes within the realm of engineering and optimization.

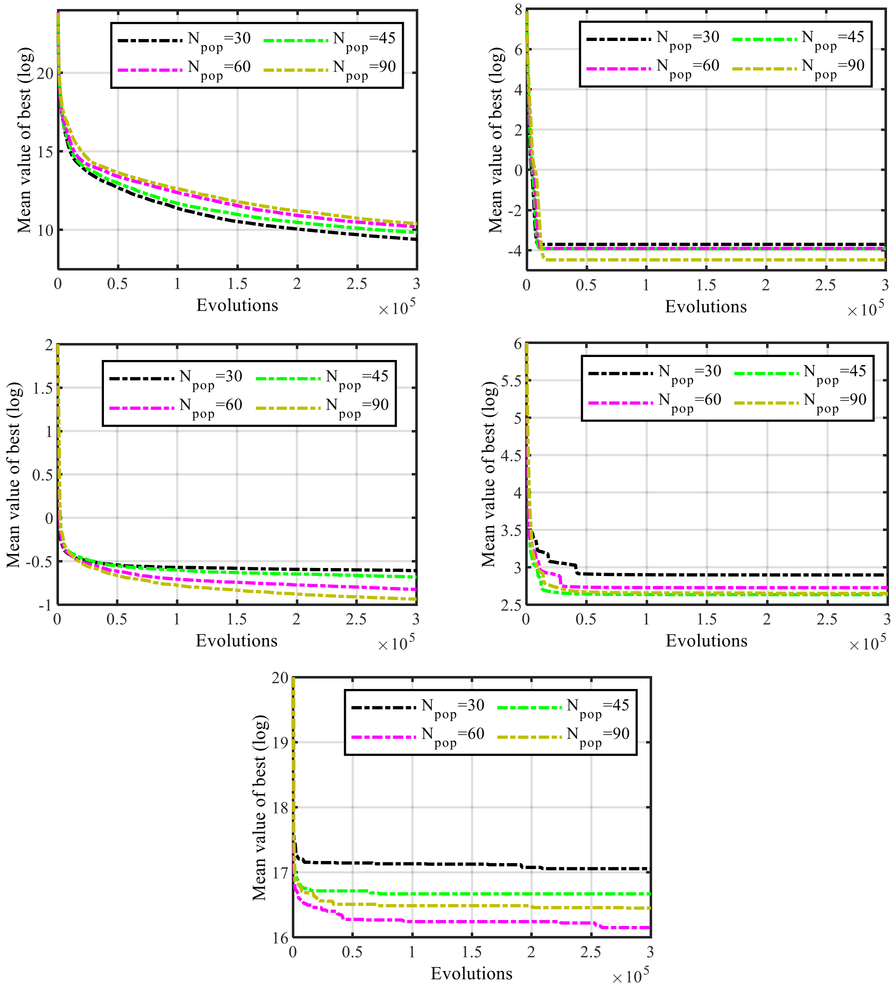


Fig. 14 Convergence trends for the best result via FLA for some functions of the CEC2014 (F1, F7, F13, F19, and F29 from left to right)

5 Application of FLA in the engineering problems

One approach to highlight the high performance of the FLA is to test its applicability to engineering problems, as described below. To this end, *Npop* has been set at 90, and the program is run 30 times. To ensure the results, the data of the optimal results are extracted from the mentioned literature. This study effectively addresses the constraints of the optimization problem by employing the static penalty technique [121, 122]:

Table 14 The parameter settings of comparison algorithms on CEC2014

Algorithm	Parameter	Value
SMA [55]	Vc factor	Decreases linearly from 1 to 0
	Z factor	0.03
WOA [36]	Scaling factor l	[− 1, 1]
	Scaling factor b	1
	Scaling factor a	Linear reduction from 2 to 0
GWO [33]	Convergence parameter a	[2 0]
Modified PSO (MPSO) [84]	Inertia weight ω	Decreases linearly from 0.9 to 0.4
	Acceleration control parameter c_1	2.0
	Acceleration control parameter c_2	2.0
Arithmetic optimization algorithm (AOA) [103]	Parameter α is a crucial factor that determines the accuracy of exploitation over multiple iterations	5
	μ is a control parameter to adjust the search process	0.5
Lévy flight distribution (LFD) [104]	Index β	1.5
	Scalar CSV	0.5
	Threshold	2
	Random number a_1	10
	Random number a_2	0.00005
	Random number a_3	0.005
	Parameter δ_1	0.9
	Parameter δ_2	0.1
RAO1 [86]	Without any control parameter	––

Table 14 (continued)

Algorithm	Parameter	Value
Emperor penguin optimizer (EPO) [45]	Parameter L	[1.5, 2]
	Parameter f	[2, 3]
	Parameter M	2
	Function $S()$	[0, 1.5]
	A Constant	[−1.5, 1.5]
	Temperature Profile (T')	[1, 1000]
BAT [29]	Frequency maximum	2
	Frequency minimum	0.0
	r (Pulse rate)	0.5
	A (Loudness)	0.5
FLA (the proposed)	Factor Ne	5

Table 15 The optimal results for different algorithms on CEC2014

F	SMA	GWO	WOA	MPSO	AOA Mean Std Winner	LFD	RAO1	EPO	BA	FLA Mean Std
F1	3.64E+06 2.25E+06 –	4.60E+07 3.13E+07 –	3.05E+07 1.83E+07 –	3.20E+06 2.47E+06 –	8.26E+06 5.25E+06 -	8.19E+06 9.74E+05 -	1.83E+07 6.92E+06 -	2.17E+07 1.18E+07 -	3.83E+07 3.19E+07 -	**3.26E+04** **2.45E+04**
F2	1.49E+03 1.86E+03 –	1.04E+09 1.60E+09 –	4.93E+06 5.79E+06 –	4.95E+03 3.00E+03 –	5.00E+03 3.71E+03 –	2.75E+03 1.34E+03 –	7.59E+03 7.12E+03 –	1.72E+04 1.02E+04 -	2.63E+07 5.95E+06 -	**1.46E−13** **3.20E−13**
F3	9.76E+03 3.69E+03 –	2.95E+04 1.08E+04 –	3.65E+04 3.01E+04 –	8.27E+03 5.94E+03 –	2.78E+04 6.45E+03 –	3.47E+04 2.06E+04 –	2.98E+04 8.37E+03 –	1.24E+04 1.69E+04 -	7.18E+04 2.36E+04 -	**2.07E−04** **8.97E−04**
F4	1.89E+02 4.08E+02 –	1.86E+02 5.16E+01 –	1.82E+02 7.09E+01 –	1.95E+02 2.01E+02 –	3.17E+02 1.76E+02 –	9.38E+02 6.55E+02 –	1.12E+02 6.35E+01 –	3.77E+02 3.54E+02 -	2.45E+02 5.34E+01 -	**8.46E+00** **2.19E+01**
F5	2.12E+01 8.14E−02 –	2.09E+01 5.87E−02 –	2.05E+01 2.05E−01 –	2.11E+01 6.92E−02 –	2.10E+01 7.38E−01 –	2.14E+01 8.25E−01 –	2.09E+01 6.11E−02 –	2.14E+01 3.68E−01 -	2.11E+01 1.00E−01 -	**2.04E+01** **1.37E−01**
F6	1.92E+01 5.61E+00 +	2.22E+01 2.72E+00 –	3.47E+01 5.18E+00 –	2.31E+01 1.46E+01 –	4.07E+01 2.46E+01 –	**8.26E+00** **6.45E+00** +	2.79E+01 6.28E+00 –	2.19E+01 1.03E+01 -	3.72E+01 6.23E+00 -	2.16E+01 4.37E+00
F7	7.46E−02 1.88E−02 –	1.14E+01 1.33E+01 –	9.14E−01 3.49E−01 –	**5.10E−03** **5.28E−04** +	4.47E−02 3.51E−02 –	5.68E−02 7.21E−02 –	7.01E−02 2.42E−01 –	2.25E−01 3.14E−01 -	2.93E+00 9.30E−01 -	1.14E−02 1.18E−02
F8	1.98E+02 4.27E+01 –	9.26E+01 1.58E+01 –	1.88E+02 2.95E+01 –	1.05E+02 6.52E+01 –	3.92E+02 1.45E+02 –	3.45E+02 6.89E+01 –	1.81E+02 3.01E+01 –	3.24E+02 2.48E+02 -	2.10E+02 4.29E+01 -	**9.10E+01** **2.12E+01**
F9	2.07E+02 1.92E+02 –	**8.87E+01** **1.91E+01** +	2.59E+02 4.73E+01 –	4.18E+02 7.34E+01 –	3.82E+02 1.27E+02 –	7.24E+02 9.76E+01 –	2.09E+02 2.20E+01 –	5.83E+02 3.92E+02 -	2.56E+02 7.20E+01 -	1.14E+02 3.30E+01

Table 15 (continued)

F	SMA	GWO	WOA	MPSO	AOA	LFD	RAO1	EPO	BA	FLA
					Mean					Mean
					Std					Std
					Winner					
F10	6.75E+03	2.79E+03	3.87E+03	8.68E+03	8.56E+03	7.00E+03	6.32E+03	9.27E+03	6.98E+03	**2.71E+03**
	2.63E+03	5.21E+02	4.15E+02	7.82E+03	7.90E+02	2.07E+03	5.71E+02	4.29E+03	5.15E+02	**6.47E+02**
	–	–	–	–	–	–	–	-	-	
F11	5.28E+03	3.70E+03	5.01E+03	7.11E+03	9.24E+03	8.19E+03	7.01E+03	1.37E+04	9.18E+03	**3.45E+03**
	7.00E+02	6.16E+02	7.19E+02	1.19E+03	2.06E+03	2.48E+03	3.41E+02	1.42E+04	6.25E+02	**5.92E+02**
	–	–	–	–	–	–	–	-	-	
F12	3.19E+00	1.44E+00	1.70E+00	4.87E+00	3.00E+00	3.67E+00	2.25E+00	6.11E+00	1.25E+00	**7.70E−01**
	5.96E−01	1.18E+00	6.14E−01	6.05E−01	1.27E−01	5.12E−01	4.00E−01	9.72E−01	8.14E−01	**4.51E−01**
	–	–	–	–	–	–	-	-	-	
F13	4.83E−01	3.97E−01	4.98E−01	3.48E−01	4.75E−01	4.22E−01	4.75E−01	**3.28E−01**	5.32E−01	3.92E−01
	2.47E−01	7.67E−02	1.90E−01	9.12E−02	6.46E−02	9.49E−02	6.19E−02	**9.16E−02**	1.47E−01	1.12E−01
	–	–	–	+	–	-	-	**+**	-	
F14	5.48E−01	2.35E+00	5.60E−01	6.46E−01	8.87E−01	8.47E−01	5.47E−01	6.39E−01	6.24E−01	**5.35E−01**
	3.91E−01	4.22E+00	5.85E−02	3.47E−01	3.24E−01	6.60E−01	3.20E−01	9.54E−01	7.72E−02	**2.38E−01**
	–	–	–	–	–	–	–	-	-	
F15	2.45E+01	1.97E+01	8.95E+01	2.65E+01	1.76E+01	1.85E+01	1.82E+01	5.25E+01	4.24E+01	**7.83E+00**
	1.16E+00	1.24E+01	4.10E+01	2.35E+00	8.63E−01	7.19E+00	4.12E+00	3.98E+01	1.75E+01	**3.12E+00**
	–	–	–	–	–	–	–	-	-	
F16	1.59E+01	1.27E+01	1.25E+01	**1.06E+01**	1.65E+01	1.44E+01	1.31E+01	1.85E+01	1.30E+01	1.14E+01
	7.28E−01	7.31E−01	5.00E−01	**3.99E−01**	4.90E−01	3.80E−01	3.14E−01	8.09E−01	7.22E−01	6.76E−01
	–	–	–	+	–	–	–	-	-	
F17	1.24E+06	1.94E+06	4.02E+06	7.26E+05	2.03E+06	9.63E+05	1.33E+06	7.56E+06	3.19E+06	**7.83E+03**
	5.46E+05	1.74E+06	1.31E+06	4.17E+05	5.46E+05	9.25E+04	5.12E+05	2.94E+06	7.45E+05	**4.12E+03**
	–	–	–	–	–	–	–	-	-	

Table 15 (continued)

F	SMA	GWO	WOA	MPSO	AOA	LFD	RAO1	EPO	BA	FLA
					Mean Std Winner					Mean Std
F18	2.79E+04 6.05E+03 –	7.16E+06 2.26E+07 –	2.49E+04 8.48E+04 –	2.29E+04 8.76E+03 –	2.01E+04 1.14E+04 –	5.48E+04 4.27E+04 –	4.70E+06 5.19E+06 –	3.00E+04 2.91E+04 -	2.25E+04 8.29E+03 -	**9.29E+03** **9.11E+03**
F19	1.20E+01 2.37E+01 +	3.88E+01 2.14E+01 –	4.36E+01 3.92E+01 –	**8.85E+00** **5.09E+00** +	1.75E+01 4.96E+00 –	5.01E+01 2.25E+01 –	9.28E+00 3.40E+00 +	3.28E+01 1.93E+01 –	3.49E+01 1.53E+01 –	1.41E+01 1.94E+01
F20	4.38E+03 2.52E+03 –	1.40E+04 7.48E+03 –	2.01E+04 1.32E+04 –	3.64E+03 9.28E+02 –	3.47E+03 1.57E+03 –	8.28E+03 4.53E+03 –	6.24E+03 2.54E+03 –	6.58E+04 5.97E+04 –	2.51E+04 2.30E+04 –	**3.87E+02** **1.22E+02**
F21	4.56E+05 3.79E+05 –	2.32E+05 2.30E+05 –	9.59E+06 7.96E+05 –	2.95E+05 3.62E+05 –	3.26E+05 2.30E+05 –	3.12E+06 1.35E+06 –	4.99E+05 2.16E+05 –	2.05E+06 1.84E+06 –	5.40E+05 9.21E+04 –	**6.59E+03** **4.77E+03**
F22	5.47E+02 1.70E+02 +	**2.60E+02** **1.26E+02** +	7.15E+02 3.60E+02 –	4.21E+02 9.37E+01 +	8.29E+02 8.00E+01 –	5.99E+02 8.56E+01 +	7.34E+02 3.47E+02 –	1.53E+03 1.07E+03 –	7.25E+02 4.81E+02 –	6.28E+02 2.16E+02
F23	3.16E+02 1.24E−01 –	3.28E+02 4.13E+00 –	3.31E+02 6.43E+00 –	3.16E+02 3.25E+00 –	3.17E+02 3.37E−03 –	3.16E+02 3.15E−04 –	**3.15E+02** **6.26E−13** =	3.17E+02 2.45E−02 –	3.45E+02 6.14E+00 –	**3.15E+02** **5.78E−14**
F24	2.54E+02 3.60E+00 –	**2.00E+02** **5.91E−04** +	2.12E+02 5.02E+00 +	2.73E+02 6.30E+00 –	2.82E+02 7.92E−01 –	3.23E+02 6.05E+01 –	2.48E+02 7.49E+00 –	3.15E+02 1.16E+01 –	2.56E+02 5.26E+00 –	2.45E+02 4.79E+00
F25	2.08E+02 1.05E+00 -	2.11E+02 1.88E+00 -	2.23E+02 1.93E+01 -	2.09E+02 4.17E+00 -	2.16E+02 6.11E+00 -	2.17E+02 5.39E+00 -	2.09E+02 3.10E+00 -	2.15E+02 8.73E+00 -	2.13E+02 2.41E+01 -	**2.06E+02** **2.90E+00**

Table 15 (continued)

F	SMA	GWO	WOA	MPSO	AOA	LFD	RAO1	EPO	BA	FLA
					Mean					Mean
					Std					Std
					Winner					
F26	1.18E+02	1.10E+02	1.01E+02	1.08E+02	1.13E+02	1.09E+02	1.01E+02	1.20E+02	1.17E+02	**1.00E+02**
	3.16E+01	3.15E+01	1.23E−01	6.29E − 02	1.31E+01	4.14E+01	2.34E−01	1.30E − 01	3.10E−01	**1.14E−01**
	–	–	–	–	–	–	–	–	–	
F27	**5.32E+02**	6.02E+02	9.90E+02	6.36E+02	8.29E+02	2.04E+03	9.92E+02	2.94E+03	1.02E+03	9.71E+02
	9.75E+01	9.77E+01	4.14E+02	7.75E+01	8.04E+01	6.47E+01	2.57E+02	1.15E+03	5.85E+02	2.33E+02
	+	+	–	+	+	–	–	–	–	
F28	1.75E+03	**8.89E+02**	2.29E+03	1.15E+03	1.70E+03	2.95E+03	1.73E+03	3.92E+03	2.26E+03	1.60E+03
	9.48E+02	**5.31E+01**	5.72E+02	9.96E+01	8.17E+02	9.74E+02	5.10E+02	7.48E+02	6.14E+02	4.27E+02
	–	+	–	+	–	–	–	–	–	
F29	1.92E+07	**9.92E+04**	1.50E+07	5.18E+06	6.26E+06	4.75E+06	2.47E+06	1.53E+07	2.99E+07	1.39E+07
	1.23E+07	**1.61E+05**	1.20E+07	1.84E+06	3.75E+06	8.26E+06	4.53E+06	6.94E+06	2.34E+07	1.65E+07
	–	+	–	+	+	+	+	–	–	
F30	4.61E+03	3.72E+04	8.98E+04	4.19E+03	3.95E+03	4.06E+03	4.55E+03	4.75E+03	8.27E+04	**3.85E+03**
	2.39E+03	3.51E+04	9.59E+04	2.52E+03	2.15E+03	6.91E+02	3.97E+03	3.82E+03	1.53E+04	**2.16E+03**
	–	–	–	–	–	–	–	–	–	
±/=	4/26/0	6/24/0	1/29/0	8/22/0	2/28/0	3/27/0	2/27/1	1/29/0	0/30/0	—
p values	3.1128E−7	4.0615E−5	6.584E−9	2.3056E−5	8.981E−9	8.3358E−9	1.6607E−7	2.308E−10	1.486E−10	--

The bold numbers indicate the best solutions for each function

Table 16 The mean rankings via Friedman test (*Fr*) with $D=30$ on CEC2014 for the studied algorithms

F	SMA *Fr*	GWO *Fr*	WOA *Fr*	MPSO *Fr*	AOA *Fr*	LFD *Fr*	RAO1 *Fr*	EPO *Fr*	BA *Fr*	FLA *Fr*
F1	3	10	8	2	5	4	6	7	9	**1**
F2	2	10	8	4	5	3	6	7	9	**1**
F3	3	6	9	2	5	8	7	4	10	**1**
F4	5	4	3	6	8	10	2	9	7	**1**
F5	8	3.5	2	6.5	5	9.5	3.5	9.5	6.5	**1**
F6	2	5	8	6	10	**1**	7	4	9	3
F7	6	10	8	**1**	3	4	5	7	9	2
F8	6	2	5	3	10	9	4	8	7	**1**
F9	3	**1**	6	8	7	10	4	9	5	2
F10	5	2	3	9	8	7	4	10	6	**1**
F11	4	2	3	6	9	7	5	10	8	**1**
F12	7	3	4	9	6	8	5	10	2	**1**
F13	8	4	9	2	6.5	5	6.5	**1**	10	3
F14	3	10	4	7	9	8	2	6	5	**1**
F15	6	5	10	7	2	4	3	9	8	**1**
F16	8	4	3	**1**	9	7	6	10	5	2
F17	4	6	9	2	7	3	5	10	8	**1**
F18	6	10	5	4	2	8	9	7	3	**1**
F19	3	8	9	**1**	5	10	2	6	7	4
F20	4	7	8	3	2	6	5	10	9	**1**
F21	5	2	10	3	4	9	6	8	7	**1**
F22	3	**1**	6	2	9	4	8	10	7	5
F23	4	8	9	4	6.5	4	**1.5**	6.5	10	**1.5**
F24	5	**1**	2	7	8	10	4	9	6	3
F25	2	5	10	3.5	8	9	3.5	7	6	**1**
F26	9	6	2.5	4	7	5	2.5	10	8	**1**
F27	**1**	2	6	3	4	9	7	10	8	5
F28	6	**1**	8	2	4	9	5	10	7	3
F29	9	**1**	7	4	5	3	2	8	10	6
F30	6	8	10	4	2	3	5	7	9	**1**
Mean Fr	4.8667	4.9167	6.4833	4.200	6.0333	6.550	4.7167	7.9667	7.3500	**1.9167**
Final rank	4	5	7	2	6	8	3	10	9	**1**

The bold numbers indicate the best solutions for each function

$$\psi(\mathbf{X}) = f(\mathbf{X}) \pm \left[\sum_{i=1}^{m} \mathrm{K}_i \cdot \max\left(0, UEQ_i(\mathbf{X})\right)^{\phi} + \sum_{j=1}^{n} \tilde{\mathrm{K}}_j \cdot \left| EQ_j(\mathbf{X}) \right|^{\chi} \right] \tag{14}$$

In the context of this constraint handling approach, the symbol $\psi(\mathbf{X})$ denotes the objective function, while K_i and $\tilde{K}_j$ refer to two significant penalty coefficients. The un-equality and equality constraints of the problem are, respectively,

Table 17 A comparison of FLA with modern optimizers on CEC2014

F	CMA-ES	LJA	MFO	CS	TOGPEAe	SOS	MG-SCA	m-SCA	NIWTLBO	mTLBO	FIPS-URing	FLA
	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner
F1	9.42E+04	6.31E+07	7.59E+07	3.50E+07	6.54E+06	6.95E+07	2.92E+07	2.26E+07	4.95E+05	6.46E+07	6.27E+07	**3.26E+04**
	7.89E+04	1.87E+07	9.77E+07	2.49E+07	3.49E+06	2.92E+07	2.07E+07	6.35E+06	3.70E+05	4.03E+07	2.08E+07	2.45E+04
	–	–	–	–	–	–	–	–	–	–	–	
F2	2.55E+10	4.77E+09	1.36E+10	1.95E+07	1.79E+07	3.61E+09	2.26E+09	8.11E+07	1.78E+02	1.18E+09	4.05E+04	**1.46E−13**
	3.85E+09	6.03E+08	8.42E+09	5.49E+07	2.26E+07	7.76E+08	1.69E+09	5.59E+07	2.49E+02	1.52E+09	3.30E+04	3.20E−13
	–	–	–	–	–	–	–	–	–	–	–	
F3	1.45E+04	6.91E+04	8.99E+04	3.10E+04	5.47E+03	2.38E+04	1.77E+04	2.59E+04	2.52E+01	4.68E+00	5.47E+04	**2.07E−04**
	5.66E+03	1.07E+04	4.98E+04	1.36E+04	4.07E+03	1.35E+04	6.63E+03	6.43E+03	5.73E+01	1.73E+01	2.46E+04	8.97E−04
	–	–	–	–	–	–	–	–	–	–	–	
F4	2.52E+03	4.08E+02	1.14E+03	2.03E+02	1.44E+02	4.10E+02	2.76E+02	1.83E+02	9.32E+01	3.85E+02	2.66E+01	**8.46E+00**
	5.36E+02	5.38E+01	1.13E+03	6.69E+01	4.37E+01	6.58E+01	6.55E+01	2.58E+01	3.17E+01	1.17E+02	4.98E−01	**2.19E+01**
	–	–	–	–	–	–	–	–	–	–	–	
F5	**2.00E+01**	2.09E+01	2.04E+01	**2.00E+01**	2.05E+01	2.06E+01	2.04E+01	2.09E+01	2.09E+01	2.09E+01	2.10E+01	2.04E+01
	2.63E−05	4.97E–02	1.75E−01	**2.28E−03**	3.56E−01	6.32E−02	1.44E−01	7.07E−02	4.68E–02	5.60E–02	5.21E−02	1.37E−01
	–	–	=	+	–	–	=	–	–	–	–	
F6	4.09E+01	3.39E+01	2.40E+01	3.23E+01	2.16E+01	2.33E+01	1.94E+01	1.46E+01	2.88E+01	2.42E+01	3.81E+01	2.16E+01
	2.13E+00	1.29E+00	3.33E+00	3.27E+00	3.46E+00	1.55E+00	2.89E+00	3.35E+00	3.01E+00	2.27E+00	1.60E+00	4.37E+00
	–	–	–	–	=	–	+	+	–	–	–	
F7	2.31E+02	1.58E+01	1.17E+02	1.79E+00	1.40E+00	3.33E+01	1.99E+01	2.01E+00	3.15E−01	6.00E+01	**8.99E−05**	1.14E−02
	2.83E+01	2.80E+00	6.91E+01	2.19E+00	3.12E−01	6.92E+00	1.18E+01	4.62E−01	1.42E+00	2.36E+01	2.95E−04	1.18E−02
	–	–	–	–	–	–	–	–	–	–	+	
F8	2.83E+02	2.24E+02	1.43E+02	1.71E+02	**5.89E+01**	8.16E+01	1.07E+02	1.10E+02	1.42E+02	1.16E+02	8.75E+01	9.10E+01
	2.21E+01	9.93E+00	3.81E+01	3.46E+01	**1.80E+01**	6.84E+00	2.14E+01	1.24E+01	1.96E+01	2.46E+01	1.21E+01	2.12E+01
	–	–	–	–	+	+	–	–	–	–	+	
F9	3.28E+02	2.61E+02	2.23E+02	2.80E+02	**7.22E+01**	1.69E+02	1.39E+02	1.31E+02	1.72E+02	1.25E+02	1.66E+02	1.14E+02
	7.65E+01	1.47E+01	6.06E+01	5.16E+01	**2.17E+01**	1.75E+01	2.56E+01	9.81E+00	2.15E+01	2.89E+01	1.14E+01	3.30E+01
	–	–	–	–	+	–	–	–	–	–	–	

Table 17 (continued)

F	CMA-ES Mean Std Winner	LJA Mean Std Winner	MFO Mean Std Winner	CS Mean Std Winner	TOGPEAe Mean Std Winner	SOS Mean Std Winner	MG-SCA Mean Std Winner	m-SCA Mean Std Winner	NIWTLBO Mean Std Winner	mTLBO Mean Std Winner	FIPS-URing Mean Std Winner	FLA Mean Std Winner
F10	**2.61E+02** **1.06E+02** +	5.68E+03 3.95E+02 –	3.47E+03 8.85E+02 –	2.66E+03 5.34E+02 +	3.79E+03 1.91E+03 –	1.38E+03 1.96E+02 +	2.82E+03 6.83E+02 –	3.58E+03 4.72E+02 –	3.15E+03 4.38E+02 –	2.87E+03 6.57E+02 –	6.57E+03 2.44E+02 –	2.71E+03 6.47E+02
F11	**1.69E+02** **1.98E+02** +	6.88E+03 3.12E+02 –	4.15E+03 6.90E+02 –	4.13E+03 5.35E+02 –	4.53E+03 1.45E+03 –	4.48E+03 4.10E+02 –	3.30E+03 6.26E+02 +	4.93E+03 4.71E+02 –	3.05E+03 7.21E+02 +	3.28E+03 5.48E+02 +	7.68E+03 2.66E+02 –	3.45E+03 5.92E+02
F12	**3.03E−01** **2.18E+00** +	2.49E+00 2.73E–01 –	4.33E−01 2.64E−01 +	5.11E−01 2.56E−01 +	7.74E−01 1.10E+00 –	8.29E−01 1.44E−01 –	6.33E−01 3.36E−01 +	1.76E+00 2.89E−01 –	1.93E+00 6.19E–01 –	2.50E+00 2.66E−01 –	2.41E+00 2.21E−01 –	7.70E–01 4.51E–01
F13	5.51E+00 3.07E–01 –	1.08E+00 1.19E−01 –	2.21E+00 1.34E+00 –	4.81E−01 1.17E−01 –	4.93E−01 1.15E−01 –	7.79E−01 1.48E−01 –	5.51E−01 8.94E−02 –	3.86E−01 5.92E−02 +	5.93E−01 1.44E−01 –	1.77E+00 9.83E−01 –	**3.74E−01** **5.25E−02** +	3.92E–01 1.12E–01
F14	7.53E+01 8.08E+00 –	4.33E+00 1.70E+00 –	3.54E+01 2.47E+01 –	3.08E−01 5.64E−02 +	**2.63E−01** **5.02E−02** +	9.76E+00 3.87E+00 –	2.34E+00 3.31E+00 –	2.65E−01 2.94E−02 +	2.91E−01 1.67E−01 +	2.04E+01 8.51E+00 –	3.14E−01 4.46E−02 +	5.35E–01 2.38E–01
F15	1.02E+04 3.24E+04 –	5.05E+01 9.36E+00 –	2.23E+05 5.77E+05 –	9.80E+01 3.02E+01 –	2.43E+01 7.54E+00 –	2.50E+02 2.05E+02 –	8.72E+01 1.01E+02 –	1.52E+01 1.47E+00 –	1.74E+02 2.96E+02 –	1.20E+03 1.46E+03 –	1.55E+01 8.97E−01 –	**7.83E+00** **3.12E+00**
F16	1.38E+01 5.31E−01 –	1.28E+01 1.78E−01 –	1.27E+01 5.33E−01 –	1.27E+01 5.01E−01 –	1.23E+01 6.54E−01 –	1.13E+01 3.70E−01 +	1.16E+01 6.91E−01 –	1.20E+01 2.89E−01 –	1.16E+01 4.92E–01 –	**1.11E+01** **7.48E−01** +	1.33E+01 1.51E−01 –	1.14E+01 6.76E–01
F17	**5.49E+03** **3.62E+03** +	2.63E+06 9.76E+05 –	3.39E+06 4.07E+06 –	1.48E+06 1.21E+06 –	7.12E+04 5.59E+04 –	5.68E+06 3.70E+06 –	9.56E+05 7.62E+05 –	5.99E+05 3.59E+05 –	2.52E+04 3.08E+04 –	6.56E+05 1.24E+06 –	1.62E+06 3.98E+05 –	7.83E+03 4.12E+03

Table 17 (continued)

F	CMA-ES	LJA	MFO	CS	TOGPEAe	SOS	MG-SCA	m-SCA	NIWTLBO	mTLBO	FIPS-URing	FLA
	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner
F18	1.52E+09 3.93E+08 –	1.26E+07 1.06E+07 –	5.19E+06 3.61E+07 –	7.67E+03 6.70E+03 +	3.07E+03 3.05E+03 +	4.95E+05 2.26E+05 –	1.48E+05 9.00E+05 –	1.61E+05 8.25E+04 –	**2.06E+03** **2.34E+03** +	1.92E+04 8.10E+04 –	1.62E+05 1.29E+05 –	9.29E+03 9.11E+03
F19	2.98E+02 4.25E+01 –	3.78E+01 3.45E+01 –	7.36E+01 5.32E+01 –	5.33E+01 3.63E+01 –	1.60E+01 1.80E+01 –	3.80E+01 2.93E+01 –	2.28E+01 1.43E+01 –	1.93E+01 5.95E+00 –	2.30E+01 1.40E+01 –	8.02E+01 3.50E+01 –	1.75E+01 7.88E−01 –	**1.41E+01** **1.94E+01**
F20	4.61E+03 3.88E+03 –	9.92E+03 3.69E+03 –	5.67E+04 4.34E+04 –	3.93E+04 2.20E+04 –	3.31E+03 3.67E+03 –	1.59E+04 1.09E+04 –	4.24E+03 3.82E+03 –	1.22E+04 3.70E+03 –	3.75E+02 1.56E+02 +	**2.72E+02** **1.21E+02** +	2.68E+04 7.75E+03 –	3.87E+02 1.22E+02
F21	6.86E+03 2.76E+03 –	6.94E+05 2.03E+05 –	7.83E+05 1.18E+06 –	3.54E+05 3.48E+05 –	1.73E+04 1.74E+04 –	7.86E+05 6.12E+05 –	2.35E+05 2.39E+05 –	1.10E+05 5.66E+04 –	1.44E+04 8.91E+03 –	3.18E+04 2.86E+04 –	6.46E+05 2.68E+05 –	**6.59E+03** **4.77E+03**
F22	1.61E+03 2.92E+02 –	5.47E+02 1.05E+02 +	8.67E+02 2.29E+02 –	9.47E+02 3.31E+02 –	6.87E+02 1.65E+02 –	5.36E+02 1.74E+02 +	3.39E+02 1.78E+02 +	**2.57E+02** **5.71E+01** +	6.41E+02 2.93E+02 –	5.32E+02 2.09E+02 +	4.53E+02 1.29E+02 +	6.28E+02 2.16E+02
F23	5.79E+02 4.94E+01 –	3.43E+02 3.41E+00 –	3.71E+02 3.98E+01 –	3.29E+02 7.51E+00 –	2.16E+01 3.46E+00 –	3.32E+02 4.05E+00 –	3.29E+02 4.03E+00 –	3.21E+02 1.58E+00 –	**2.00E+02** **0.00E+00** +	3.53E+02 1.98E+01 –	3.14E+02 1.18E−02 +	3.15E+02 5.78E−14
F24	2.12E+02 7.49E+00 +	2.57E+02 4.04E+00 –	2.76E+02 2.73E+01 –	2.78E+02 3.11E+01 –	3.17E+02 1.93E+00 –	2.66E+02 4.05E+00 –	**2.00E+02** **1.56E−03** +	**2.00E+02** **4.29E−02** +	**2.00E+02** **1.99E−05** +	**2.00E+02** **7.93E−04** +	2.24E+02 6.49E−01 +	2.45E+02 4.79E+00
F25	2.12E+02 2.97E+00 –	2.16E+02 2.58E+00 –	2.14E+02 7.65E+00 –	2.23E+02 9.39E+00 –	2.09E+02 7.21E+00 –	2.16E+02 2.60E+00 –	2.11E+02 2.82E+00 –	2.01E+02 3.19E+00 +	**2.00E+02** **0.00E+00** +	2.04E+02 7.51E+00 +	2.10E+02 2.65E+00 –	2.06E+02 2.90E+00

Table 17 (continued)

F	CMA-ES	LJA	MFO	CS	TOGPEAe	SOS	MG-SCA	m-SCA	NIWTLBO	mTLBO	FIPS-URing	FLA
	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner	Mean Std Winner
F26	1.25E+02	1.01E+02	1.03E+02	**1.00E+02**	1.04E+02	1.01E+02	1.01E+02	**1.00E+02**	1.90E+02	1.41E+02	**1.00E+02**	**1.00E+02**
	5.51E+01	1.02E−01	1.50E+00	**1.63E−01**	1.83E+01	1.22E−01	1.53E−01	**5.40E−02**	3.03E+01	4.93E+01	**4.44E−02**	**1.14E−01**
	–	–	–	=	–	–	–	=	–	–	=	
F27	1.07E+03	9.86E+02	9.21E+02	4.27E+02	9.26E+02	5.03E+02	8.19E+02	4.33E+02	5.24E+02	9.40E+02	1.29E+03	9.71E+02
	2.30E+02	2.48E+02	2.23E+02	1.96E+01	8.92E+01	1.48E+02	9.17E+01	2.01E+01	2.76E+02	2.48E+02	3.39E+01	2.33E+02
	–	–	+	+	+	+	+	+	+	+	–	
F28	2.79E+03	1.13E+03	1.12E+03	3.49E+03	1.58E+03	1.33E+03	9.68E+02	1.05E+03	1.88E+03	2.36E+03	**3.98E+02**	1.60E+03
	5.92E+02	6.63E+01	1.57E+02	5.48E+02	2.22E+02	1.36E+02	1.06E+02	2.74E+02	3.98E+02	5.34E+02	**2.88E+00**	4.27E+02
	–	+	+	–	+	+	+	+	–	–	+	
F29	3.52E+04	9.82E+05	3.06E+06	5.44E+05	2.09E+03	1.11E+05	1.19E+06	4.44E+04	9.18E+05	5.45E+06	**2.25E+02**	1.39E+07
	5.34E+03	2.07E+06	3.62E+06	2.61E+06	5.16E+02	1.14E+05	3.25E+06	1.77E+04	3.58E+06	1.21E+07	**1.77E+00**	1.65E+07
	+	+	+	+	+	+	+	+	+	+	+	
F30	6.48E+05	1.09E+04	5.89E+04	2.49E+04	1.09E+04	3.73E+04	1.92E+04	3.48E+04	2.93E+03	8.61E+04	**8.88E+02**	3.85E+03
	1.31E+05	4.24E+03	5.40E+04	2.26E+04	6.80E+03	2.12E+04	8.25E+03	1.05E+04	7.84E+02	6.81E+04	**2.09E+02**	2.16E+03
	–	–	–	–	–	–	–	–	+	–	+	

The bold numbers indicate the best solutions for each function

Table 18 The mean rankings via Friedman test for the modern algorithms with $D=30$ on CEC2014

F	CMA-ES	LJA	MFO	CS	TOGPEAe	SOS	MG-SCA	m-SCA	NIWTLBO	mTLBO	FIPS-URing	FLA
F1	2	9	12	7	4	11	6	5	3	10	8	**1**
F2	12	10	11	5	4	9	8	6	2	7	3	**1**
F3	5	11	12	9	4	7	6	8	3	2	10	**1**
F4	12	9	11	6	4	10	7	5	3	8	2	**1**
F5	1.5	9.5	4	1.5	6	7	4	9.5	9.5	9.5	12	4
F6	12	10	6	9	3.5	5	2	**1**	8	7	11	3.5
F7	12	7	11	5	4	9	8	6	3	10	**1**	2
F8	12	11	9	10	**1**	2	5	6	8	7	3	4
F9	12	10	9	11	**1**	7	5	4	8	3	6	2
F10	**1**	11	8	3	10	2	5	9	7	6	12	4
F11	**1**	11	7	6	9	8	4	10	2	3	12	5
F12	**1**	11	2	3	6	7	4	8	9	12	10	5
F13	12	9	11	4	5	8	6	2	7	10	**1**	3
F14	12	8	11	4	**1**	9	7	2	3	10	5	6
F15	11	5	12	7	4	9	6	2	8	10	3	**1**
F16	12	10	8.5	8.5	7	2	4.5	6	4.5	**1**	11	3
F17	**1**	10	11	8	4	12	7	5	3	6	9	2
F18	12	11	10	3	2	9	6	7	**1**	5	8	4
F19	12	7	10	9	2	8	5	4	6	11	3	**1**
F20	6	7	12	11	4	9	5	8	2	**1**	10	3
F21	2	10	11	8	4	12	7	6	3	5	9	**1**
F22	12	6	10	11	9	5	2	**1**	7	4	3	8
F23	12	9	10	6.5	2	8	6.5	5	**1**	11	3	4
F24	5	8	10	11	12	9	2.5	2.5	2.5	2.5	6	7
F25	8	10.5	9	12	5	10.5	7	2	**1**	3	6	4
F26	10	6	8	2.5	9	6	6	2.5	12	11	2.5	2.5

Table 18 (continued)

F	CMA-ES	LJA	MFO	CS	TOGPEAe	SOS	MG-SCA	m-SCA	NIWTLBO	mTLBO	FIPS-URing	FLA
F27	11	10	6	**1**	7	3	5	2	4	8	12	9
F28	11	5	4	12	7	6	2	3	9	10	**1**	8
F29	3	8	10	6	2	5	9	4	7	11	**1**	12
F30	12	4.5	10	7	4.5	9	6	8	2	11	**1**	3
Fr	8.25	8.7833	9.1833	6.9	4.9	7.45	5.45	4.9833	4.95	7.1667	6.15	**3.8333**
Rank	10	11	12	7	2	9	5	4	3	8	6	**1**
Nb	4	0	0	1	3	0	0	2	3	**2**	5	**7**

The bold numbers indicate the best solutions for each function

Table 19 Wilcoxon's test and efficiency of FLA versus the other studies

Corresponding optimizer	The efficiency of FLA is			FLA versus
	similar with	better than	worse than	p values
CMA-ES	0	23	7	6.7987E−04
LJA	0	27	3	3.1191E−08
MFO	1	25	4	7.6259E−08
CS	1	22	7	3.6693E−04
TOGPEAe	1	22	7	0.0376
SOS	0	23	7	1.6442E−05
MG-SCA	1	21	8	0.0008
m-SCA	1	20	9	0.0022
NIWTLBO	0	20	10	0.0037
mTLBO	0	22	8	4.4340E−04
FIPS-URing	1	19	10	0.0031

represented by $UEQ_i(\mathbf{X})$ and $EQ_j(\mathbf{X})$. The $EQ_j(\mathbf{X})$ is chosen from the set {1, 2}. In situations involving integer variables or specific multiples of a number, employing the rounding method in conjunction with penalty coefficients has effectively addressed the issue.

5.1 Welded beam optimal design

Figure 16 reveals the formwork of a welded beam. The control variables $\{x_1(h), x_2(l), x_3(t), x_4(b)\}$ or $\{s_1, s_2, s_3, s_4\}$ are characteristics of the weld thickness, the clamped bar length, the bar height, and the bar thickness, respectively [123]. The objective is to search the best optimal design factors to optimize beam fabrication costs. This issue must do is address several restrictions such as end deflection (δ), bending stress (σ), buckling load (Pc), and shear stress (τ) [97, 98]. The formulation of this problem can be found in Appendix A [124].

There have been some studies to optimize this issue. Additional methods are found in Table 25. The data compare the FLA's obtained results with studied methods described in recent studies that reveal FLA can achieve the top competitive optimal results. Table 26 reveals that FLA is an ideal algorithm for this issue.

5.2 ThreE−bar truss optimization

The constraint problem is well-known for researchers who aim to optimize the volume while maximizing strength [148]. Figure 17 reveals the structure of this design problem. These two control variables (x_1 and x_2 (or s_1 and s_2)), representing the cross-sectional zones, can be tuned by considering three inequality design

Table 20 Summary of the results for FLA with $D=50$ and NFEs = 5.00E+05 on CEC2014

Function		$Npop=30$	$Npop=45$	$Npop=60$	$Npop=75$	$Npop=90$
F1	Unimodal	8.69E+04 3.90E+04	1.14E+05 5.07E+04	2.21E+05 1.04E+05	1.14E+05 3.57E+04	2.06E+05 7.68E+04
F2		1.07E+04 1.05E+04	6.56E+03 9.14E+03	1.07E+04 1.20E+04	8.10E+03 1.01E+04	7.64E+03 9.78E+03
F3		2.20E−04 3.06E−04	1.73E−02 1.26E−02	3.90E−01 8.84E−01	2.73E+00 3.66E+00	1.00E+01 9.08E+00
F4	Simple Multimodal	2.18E+01 3.75E+01	3.87E+01 5.67E+01	4.15E+01 4.85E+01	2.26E+01 4.09E+01	4.25E+01 6.69E+01
F5		2.03E+01 1.89E−01	2.04E+01 1.14E−01	2.05E+01 1.24E−01	2.06E+01 6.45E−02	2.05E+01 1.40E−01
F6		5.51E+01 5.65E+00	4.60E+01 8.22E+00	5.18E+01 9.75E+00	4.15E+01 7.24E+00	4.20E+01 3.69E+00
F7		1.55E−02 2.53E−02	2.21E−02 1.85E−02	1.09E−02 1.38E−02	1.23E−02 1.63E−02	3.56E−03 5.52E−03
F8		2.87E+02 5.96E+01	2.48E+02 2.41E+01	2.22E+02 3.22E+01	1.98E+02 2.16E+01	2.01E+02 2.75E+01
F9		3.55E+02 4.46E+01	3.33E+02 4.80E+01	3.02E+02 4.07E+01	2.66E+02 4.27E+01	2.48E+02 3.59E+01
F10		6.97E+03 9.74E+02	6.09E+03 4.32E+02	6.12E+03 1.03E+03	5.75E+03 7.26E+02	5.76E+03 1.17E+03
F11		7.22E+03 7.94E+02	7.09E+03 1.19E+03	7.81E+03 2.00E+03	7.08E+03 1.89E+03	6.77E+03 2.09E+03
F12		1.19E+00 7.34E−01	1.33E+00 5.72E−01	1.23E+00 4.79E−01	1.17E+00 3.70E−01	1.22E+00 4.54E−01
F13		5.38E−01 8.19E−02	5.57E−01 9.07E−02	5.57E−01 1.04E−01	4.80E−01 8.25E−02	4.96E−01 9.63E−02
F14		3.76E−01 1.60E−01	4.90E−01 2.95E−01	4.87E−01 3.44E−01	4.66E−01 3.09E−01	4.10E−01 2.49E−01
F15		9.53E+01 3.28E+01	6.45E+01 2.15E+01	6.26E+01 3.66E+01	4.86E+01 1.99E+01	3.63E+01 1.86E+01
F16		2.13E+01 1.31E+00	2.05E+01 5.28E−01	2.08E+01 8.03E−01	2.04E+01 7.87E−01	2.05E+01 6.20E−01
F17	Hybrid	1.58E+04 5.49E+03	2.07E+04 1.07E+04	2.29E+04 8.48E+03	1.92E+04 5.33E+03	2.43E+04 8.67E+03
F18		3.91E+03 1.87E+03	3.00E+03 2.49E+03	3.57E+03 2.43E+03	3.47E+03 2.37E+03	3.72E+03 2.64E+03
F19		2.38E+01 3.57E+00	2.02E+01 5.19E+00	2.22E+01 2.76E+00	1.95E+01 3.50E+00	1.84E+01 4.03E+00
F20		6.16E+02 2.29E+02	6.08E+02 2.18E+02	6.74E+02 1.95E+02	5.84E+02 2.68E+02	7.79E+02 2.53E+02
F21		1.61E+04 9.87E+03	1.10E+04 7.80E+03	2.14E+04 1.28E+04	1.67E+04 9.02E+03	2.43E+04 1.32E+04
F22		1.45E+03 3.94E+02	1.56E+03 2.42E+02	1.39E+03 5.67E+02	1.17E+03 3.39E+02	1.35E+03 5.33E+02

Table 20 (continued)

Function		*Npop* = 30	*Npop* = 45	*Npop* = 60	*Npop* = 75	*Npop* = 90
F23	Composition	3.44E+02 0.00E+00	3.44E+02 0.00E+00	3.44E+02 1.52E−13	3.44E+02 1.52E−13	3.44E+02 1.52E−13
F24		3.09E+02 1.39E+01	3.14E+02 8.97E+00	3.00E+02 9.07E+00	2.99E+02 1.13E+01	2.94E+02 5.89E+00
F25		2.18E+02 1.41E+01	2.16E+02 1.52E+01	2.11E+02 2.96E+00	2.12E+02 5.52E+00	2.10E+02 3.86E+00
F26		1.01E+02 1.07E−01	1.01E+02 1.12E−01	1.01E+02 1.02E−01	1.01E+02 1.39E−01	1.01E+02 1.13E−01
F27		2.00E+03 1.61E+02	1.83E+03 1.86E+02	1.70E+03 1.57E+02	1.69E+03 1.44E+02	1.76E+03 1.08E+02
F28		3.52E+03 7.55E+02	3.45E+03 7.95E+02	3.69E+03 7.95E+02	3.22E+03 8.65E+02	2.78E+03 6.94E+02
F29		2.53E+08 1.78E+08	1.98E+08 1.15E+08	1.23E+08 8.82E+07	1.62E+08 1.22E+08	9.43E+07 4.94E+07
F30		1.81E+04 4.96E+03	1.50E+04 3.67E+03	1.41E+04 1.99E+03	1.43E+04 4.51E+03	1.27E+04 2.92E+03

constraints [149, 150]. These control variables have been bounded as $0 < s_1$ and $s_2 < 1$; the formulation of this problem can be found in appendix A [149, 150]:

Table 27 shows the optimal results for this issue, created by FLA and compared with studied algorithms in recent papers. Based on information acquired from this optimization problem, FLA achieved higher optimization performance than the competing algorithms. Table 28 shows the best result for this issue via FLA.

5.3 Cantilever beam optimization

This issue has been defined to minimize the weight of a cantilever beam with applies 5 optimal hollow square blocks [159]. The fifth block has been located under straight stress, which is a solution to this problem being tightly powered through the previous block, as has been revealed in Fig. 18. Dimensions of the profile of the cubes are described via $\{s_1, s_2, s_3, s_4$, and $s_5\}$ (which are the control parameters of the cubes) [137].

The results of the optimizers explored for this optimization issue have been shown in Table 29, in which FLA has been won by more other optimizers by using optimum designs. It is also clear from this table that FLA is the most robust optimizer for this engineering problem. The FLA solutions achieve a lower standard deviation than the studied optimizers. Table 30 reveals the best solutions for solving this optimization issue by FLA.

Table 21 Friedman test (Fr) for FLA with D = 50 and NFEs = 5.00E+05 on CEC2014

Function		*Npop* = 30	*Npop* = 45	*Npop* = 60	*Npop* = 75	*Npop* = 90
F1	Unimodal	1	2.5	5	2.5	4
F2		4.5	1	4.5	3	2
F3		1	2	3	4	5
F4	Simple Multimodal	1	3	4	2	5
F5		1	2	3.5	5	3.5
F6		5	3	4	1	2
F7		4	5	2	3	1
F8		5	4	3	1	2
F9		5	4	3	2	1
F10		5	3	4	1.5	1.5
F11		4	2.5	5	2.5	1
F12		2	5	3.5	1	3.5
F13		3	4.5	4.5	1	2
F14		1	5	4	3	2
F15		5	4	3	2	1
F16		5	2	4	2	2
F17	Hybrid	1	3	4	2	5
F18		5	1	3	2	4
F19		5	3	4	2	1
F20		3	2	4	1	5
F21		2.5	1	4	2.5	5
F22		4	5	2.5	1	2.5
F23	Composition	3	3	3	3	3
F24		4	5	2.5	2.5	1
F25		4.5	4.5	2	2	2
F26		3	3	3	3	3
F27		5	4	1.5	1.5	3
F28		4	4	4	2	1
F29		5	4	2	3	1
F30		5	4	2.5	2.5	1
Mean *Fr*	3.5500	3.3000	3.4000	**2.2167**	2.5333	
Final rank	5	3	4	**1**	2	

The bold numbers indicate the best solutions for each function

5.4 Gear train optimization

Figure 19 reveals this optimization problem. The problem aims at reducing the gear ratio's cost. Only the control variables are limited on the gear train acceptable design, where n_A (s_1), n_B (s_2), n_D (s_3), and n_F (s_4) are discrete control parameters, and the number of teeth for a gear must be an integer as follows [157]:

In Table 31, FLA's solutions have been compared to some studied optimizers worked in recent papers, including mGWO, CSA, UPSO, MBA, BES, GOA, CS, and

Table 22 Best statistical results of FLA with $D=50$ and NFEs = 5.00E+05 on CEC2014

Function		*Npop* = 30	*Npop* = 45	*Npop* = 60	*Npop* = 75	*Npop* = 90
F1	Unimodal	**3.69E+04**	3.98E+04	1.15E+05	6.53E+04	7.53E+04
F2		3.49E+02	2.97E+02	1.78E+01	7.28E+00	**6.72E−02**
F3		**3.85E−07**	2.34E−03	1.76E−02	3.68E−02	6.94E−01
F4	Simple Multimodal	2.71E−01	3.28E−02	4.54E−01	3.68E−02	**2.90E−02**
F5		**2.01E+01**	2.03E+01	2.02E+01	2.05E+01	2.03E+01
F6		4.60E+01	3.49E+01	3.97E+01	**3.13E+01**	3.83E+01
F7		**0.00E+00**	1.14E−13	1.14E−13	**0.00E+00**	**0.00E+00**
F8		2.28E+02	2.13E+02	1.64E+02	1.69E+02	**1.56E+02**
F9		3.09E+02	2.55E+02	2.50E+02	1.90E+02	**1.85E+02**
F10		6.17E+03	5.51E+03	4.35E+03	4.52E+03	**4.23E+03**
F11		6.08E+03	5.01E+03	5.19E+03	4.41E+03	**3.94E+03**
F12		5.60E−01	5.75E−01	**5.17E−01**	7.09E−01	7.50E−01
F13		3.83E−01	4.62E−01	4.07E−01	**3.25E−01**	3.52E−01
F14		2.85E−01	2.20E−01	**2.13E−01**	2.53E−01	2.53E−01
F15		4.31E+01	2.57E+01	2.62E+01	2.95E+01	**1.98E+01**
F16		**1.94E+01**	1.96E+01	1.98E+01	**1.94E+01**	1.96E+01
F17	Hybrid	8.47E+03	**6.99E+03**	8.70E+03	1.25E+04	1.36E+04
F18		1.86E+03	4.81E+02	**4.20E+02**	5.93E+02	7.36E+02
F19		1.92E+01	1.46E+01	1.89E+01	1.38E+01	**1.30E+01**
F20		3.07E+02	3.16E+02	3.81E+02	**2.42E+02**	5.27E+02
F21		**2.12E+03**	3.81E+03	9.35E+03	5.94E+03	9.16E+03
F22		9.19E+02	1.22E+03	6.19E+02	5.27E+02	**3.41E+02**
F23	Composition	**3.44E+02**	**3.44E+02**	**3.44E+02**	**3.44E+02**	**3.44E+02**
F24		2.94E+02	3.05E+02	**2.86E+02**	2.88E+02	2.88E+02
F25		**2.05E+02**	**2.05E+02**	2.07E+02	**2.05E+02**	2.06E+02
F26		**1.00E+02**	**1.00E+02**	1.01E+02	**1.00E+02**	**1.00E+02**
F27		1.82E+03	1.55E+03	**1.44E+03**	1.50E+03	1.59E+03
F28		2.11E+03	2.57E+03	2.68E+03	2.13E+03	**1.80E+03**
F29		1.90E+03	1.11E+08	**1.61E+03**	5.87E+07	1.98E+03
F30		1.32E+04	1.15E+04	1.16E+04	1.09E+04	**1.04E+04**
Nb		9	4	7	8	**14**

The bold numbers indicate the best solutions for each function

EBS. According to these results, FLA gives an optimum value of 2.700857E−12. Also, FLA shows the most robust and stable optimizer for tackling the objective function and, in addition, is an effective optimizer for optimizing so. Also, Table 32 gives the best optimal solutions to the issue.

5.5 Tension/compression spring optimization

Figure 20 reveals how this problem is solved. Control parameters, $\{s_1(d), s_2(D), s_3(N)\}$, respectively, denote the wire diameter, the value of active coils, and the

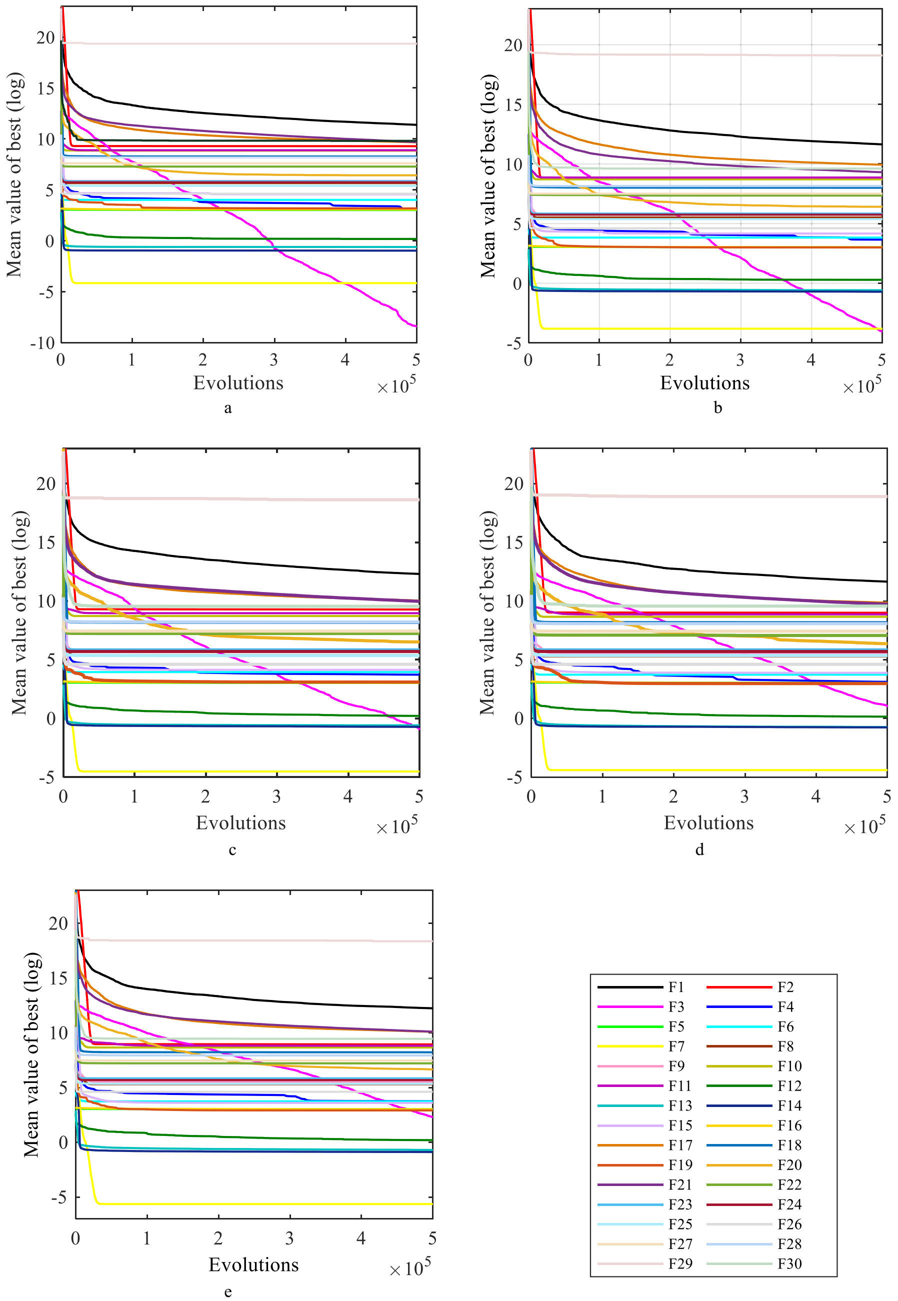


Fig. 15 Convergence trends for FLA with different ***N*pop** such as **a** 30, **b** 45, **c** 60, **d** 75, and **e** 90 with $D = 50$ and NFEs = 5.00E+05 (from left to right)

Table 23 The simulation results for CEC2014 with $D = 50$ and NFEs = 5.00E+05 on CEC2014

Function		ICA Mean Std *Fr*/Winner	BA Mean Std *Fr*/Winner	FPA Mean Std *Fr*/Winner	FIPSO Mean Std *Fr*/Winner	HHO Mean Std *Fr*/Winner	SAP-Rao Mean Std *Fr*/Winner	GWO Mean Std *Fr*/Winner	DE/rand/1 Mean Std *Fr*/Winner	FLA-75 Mean Std *Fr*
F1	Unimodal	3.17E+07 4.82E+06 5/–	1.12E+09 8.20E+08 9/–	2.66E+06 1.38E+06 2/–	3.43E+07 9.04E+06 6/–	1.05E+07 1.14E+07 3/–	1.75E+07 2.64E+06 4/–	8.14E+08 7.29E+06 8/–	3.86E+08 4.71E+07 7/–	**1.14E+05** **3.57E+04** **1**
F2		**4.91E+03** **2.57E+03** **1/+**	6.50E+10 2.85E+10 9/–	3.86E+07 1.30E+08 7/–	2.97E+04 3.10E+04 3/–	2.62E+06 2.47E+05 6/–	8.19E+04 4.25E+03 4/-	3.27E+09 1.68E+09 8/–	8.50E+05 1.48E+06 5/–	8.10E+03 1.01E+04 2
F3		6.29E+04 1.12E+04 6/–	7.81E+05 1.40E+06 9/–	1.01E+04 3.95E+03 4/–	1.15E+04 2.39E+03 5/–	4.58E+03 5.93E+02 2/–	9.48E+03 7.53E+02 3/–	8.34E+04 5.14E+03 8/–	6.52E+04 9.07E+03 7/–	**2.73E+00** **3.66E+00** **1**
F4	Simple Multimodal	1.35E+02 7.28E+01 5/–	1.81E+04 1.21E+04 9/–	1.87E+02 7.65E+01 6/–	2.68E+02 3.70E+01 7/–	8.75E+01 3.29E+01 2/–	9.20E+01 6.43E+01 3/–	6.78E+02 5.13E+01 8/–	1.14E+02 1.40E+01 4/–	**2.26E+01** **4.09E+01** **1**
F5		2.08E+01 4.19E−03 3/–	**2.00E+01** **9.25E−06** **1/+**	2.11E+01 4.95E−02 6.5/–	2.11E+01 3.65E−02 6.5/–	2.10E+01 3.82E−02 4/–	2.13E+01 5.69E−02 9/–	2.11E+01 5.25E−02 6.5/–	2.11E+01 3.90E−02 6.5/–	2.06E+01 6.45E−02 2
F6		3.15E+01 9.73E+00 3/+	8.04E+01 2.56E+00 9/–	4.48E+01 3.15E+00 7/–	2.02E+01 5.56E+00 2/+	**8.37E−01** **7.59E−01** **1/+**	4.33E+01 6.51E+00 6/–	3.65E+01 7.24E+00 4/+	5.95E+01 1.55E+00 8/–	4.15E+01 7.24E+00 5
F7		9.48E−02 6.64E−03 5/–	7.87E+02 3.06E+02 9/–	7.69E−01 4.49E−01 7/–	**6.07E−05** **1.72E−04** **1/+**	3.66E−03 9.72E−04 2/+	1.80E−01 5.36E−02 6/–	3.21E+01 9.18E+00 8/–	4.21E−02 1.52E−02 4/–	1.23E−02 1.63E−02 3
F8		**2.70E+01** **2.18E+01** **1/+**	4.68E+02 1.01E+02 9/-	1.48E+02 2.42E+01 5/+	1.60E+02 1.70E+01 6/+	4.03E+01 1.78E+01 2/+	5.74E+01 6.77E+01 3/+	1.42E+02 3.55E+01 4/+	2.95E+02 1.20E+01 8/-	1.98E+02 2.16E+01 7

Table 23 (continued)

Function	ICA Mean Std *Fr*/Winner	BA Mean Std *Fr*/Winner	FPA Mean Std *Fr*/Winner	FIPSO Mean Std *Fr*/Winner	HHO Mean Std *Fr*/Winner	SAP-Rao Mean Std *Fr*/Winner	GWO Mean Std *Fr*/Winner	DE/rand/1 Mean Std *Fr*/Winner	FLA-75 Mean Std *Fr*
F9	8.93E+02	6.27E+02	2.54E+02	3.25E+02	5.94E+02	5.39E+02	**2.27E+02**	4.13E+02	2.66E+02
	5.34E+01	1.52E+02	2.77E+01	1.80E+01	8.64E+01	1.45E+02	**3.69E+01**	1.89E+01	4.27E+01
	9/-	8/-	2/+	4/-	7/-	6/-	**1/+**	5/-	3
F10	7.19E+04	8.19E+03	**4.81E+03**	6.63E+03	3.60E+04	3.27E+05	8.27E+03	9.24E+03	5.75E+03
	5.16E+03	9.54E+02	**6.91E+02**	5.84E+02	6.73E+02	9.50E+04	6.96E+02	2.71E+02	7.26E+02
	8/-	4/-	**1/+**	3/-	7/-	9/-	5/-	6/-	2
F11	5.35E+04	8.22E+03	7.02E+03	1.26E+04	2.75E+04	2.48E+04	6.42E+04	1.31E+04	7.08E+03
	1.98E+03	9.96E+02	6.84E+02	3.03E+02	1.15E+03	1.35E+03	8.14E+02	1.96E+02	1.89E+03
	8/-	3/-	1.5/=	4/-	7/-	6/-	9/-	5/-	1.5
F12	**8.35E−01**	2.74E+00	1.62E+00	3.36E+00	4.01E+00	3.86E+00	3.40E+00	3.18E+00	1.17E+00
	9.47E−02	7.06E−01	3.36E−01	2.69E−01	8.49E−01	5.34E−01	4.00E−01	1.91E−01	3.70E−01
	1/+	4/-	3/-	6/-	9/-	8/-	7/-	5/-	2
F13	6.31E−01	5.68E+00	6.17E−01	**4.41E−01**	4.85E−01	4.90E−01	5.12E−01	6.60E−01	4.80E−01
	1.96E−01	1.16E+00	9.11E−02	**4.78E−02**	6.98E−02	5.93E−02	4.62E−02	5.11E−02	8.25E−02
	7/-	9/-	6/-	**1/+**	3/-	4/-	5/-	8/-	2
F14	5.80E−01	2.05E+02	3.53E−01	3.48E−01	3.72E−01	3.75E−01	5.18E−01	**3.00E−01**	4.66E−01
	3.14E−01	8.38E+01	1.71E−01	3.81E−02	9.61E−02	9.43E−02	9.02E−02	**3.89E−02**	3.09E−01
	8/-	9/-	3/+	2/+	4.5/+	4.5/+	7/-	**1/+**	6
F15	9.81E+01	8.56E+05	1.05E+02	3.22E+01	6.35E+01	2.26E+01	1.83E+02	4.03E+01	4.86E+01
	2.20E+01	1.09E+06	3.97E+01	1.41E+00	7.20E+00	1.87E+01	5.37E+01	1.25E+00	1.99E+01
	6/-	9/-	7/-	2/+	5/-	1/+	8/-	3/+	4
F16	2.15E+01	2.33E+01	2.16E+01	2.15E+01	3.09E+01	2.16E+01	2.18E+01	2.23E+01	**2.04E+01**
	7.13E−01	4.75E−01	4.90E−01	3.10E−01	4.41E−01	5.82E−01	6.92E−01	2.04E−01	**7.87E−01**
	4/-	8/-	4/-	4/-	9/-	4/-	4/-	7/-	**1**

Table 23 (continued)

Function		ICA Mean Std *Fr*/Winner	BA Mean Std *Fr*/Winner	FPA Mean Std *Fr*/Winner	FIPSO Mean Std *Fr*/Winner	HHO Mean Std *Fr*/Winner	SAP-Rao Mean Std *Fr*/Winner	GWO Mean Std *Fr*/Winner	DE/rand/1 Mean Std *Fr*/Winner	FLA-75 Mean Std *Fr*
F17	Hybrid	8.20E+06 3.45E+06 6/-	8.78E+07 9.96E+07 9/-	2.65E+04 1.34E+04 2/-	1.78E+06 7.18E+05 4/-	5.43E+06 4.95E+06 5/-	1.57E+06 1.93E+06 3/-	5.24E+07 8.75E+06 8/-	1.57E+07 2.64E+06 7/-	**1.92E+04** **5.33E+03** **1**
F18		5.97E+03 1.63E+03 7/-	1.48E+09 2.43E+09 9/-	1.84E+03 1.54E+03 4/+	1.18E+03 9.53E+02 3/+	**3.52E+02** **2.84E+02** **1/+**	3.72E+02 7.40E+02 2/+	3.99E+03 6.74E+02 6/-	7.31E+03 6.22E+03 8/-	3.47E+03 2.37E+03 5
F19		7.86E+01 2.32E+00 8/-	6.29E+02 2.97E+02 5.5/-	4.53E+01 2.82E+01 4/-	6.51E+01 1.91E+01 7/-	2.85E+01 6.55E+00 3/-	6.25E+01 4.79E+00 5.5/-	9.26E+01 4.35E+01 9/-	2.49E+01 2.59E+00 2/-	**1.95E+01** **3.50E+00** **1**
F20		8.44E+04 6.29E+04 8/-	1.25E+05 1.03E+05 9/-	1.52E+03 8.96E+02 2/-	1.86E+03 6.23E+02 3/-	1.29E+04 5.63E+03 5/-	5.38E+04 4.42E+03 7/-	2.19E+03 1.05E+03 4/-	1.53E+04 5.39E+03 6/-	**5.84E+02** **2.68E+02** **1**
F21		1.87E+06 6.10E+05 5/-	1.43E+07 2.15E+07 8/-	**7.83E+03** **4.45E+03** **1/+**	1.19E+06 4.72E+05 4/-	2.46E+06 4.80E+06 6/-	3.20E+05 8.35E+05 3/-	2.79E+07 3.51E+05 9/-	5.40E+06 9.74E+05 7/-	1.67E+04 9.02E+03 2
F22		1.75E+03 5.58E+02 6/-	7.68E+03 5.95E+03 8/-	**9.12E+02** **2.33E+02** **1/+**	1.02E+03 2.84E+02 2.5/+	2.04E+03 7.49E+02 7/-	9.18E+03 9.42E+02 9/-	1.00E+03 3.13E+02 2.5/+	1.31E+03 1.09E+02 5/-	1.17E+03 3.39E+02 4
F23	Composition	3.50E+02 5.21E−04 6.5/-	8.84E+02 1.97E+02 9/-	3.44E+02 3.42E−01 2.5/=	3.51E+02 1.26E+00 6.5/-	3.50E+02 6.54E−10 6.5/-	3.52E+02 3.47E−02 6.5/-	3.44E+02 3.18E−10 2.5/=	3.44E+02 1.03E−05 2.5/=	3.44E+02 1.52E−13 2.5
F24		2.90E+02 6.47E+00 6/=	5.78E+02 7.75E+01 9/-	2.90E+02 1.25E+01 6/=	2.58E+02 4.02E+00 1.5/+	3.16E+02 5.98E+00 8/-	2.80E+02 3.11E+00 3.5/+	2.84E+02 5.11E+00 3.5/+	2.59E+02 9.79E−01 1.5/+	2.99E+02 1.13E+01 6

Table 23 (continued)

Function	ICA Mean Std *Fr*/Winner	BA Mean Std *Fr*/Winner	FPA Mean Std *Fr*/Winner	FIPSO Mean Std *Fr*/Winner	HHO Mean Std *Fr*/Winner	SAP-Rao Mean Std *Fr*/Winner	GWO Mean Std *Fr*/Winner	DE/rand/1 Mean Std *Fr*/Winner	FLA-75 Mean Std *Fr*
F25	2.30E+02	2.92E+02	2.14E+02	2.21E+02	2.20E+02	2.18E+02	2.25E+02	2.81E+02	**2.12E+02**
	9.19E+00	3.59E+01	1.01E+01	2.16E+00	8.93E−01	6.72E+00	6.92E+00	7.51E+00	**5.52E+00**
	7/-	9/-	2/-	4.5/-	4.5/-	3/-	6/-	8/-	**1**
F26	1.68E+02	3.52E+02	1.07E+02	1.39E+02	1.86E+02	2.32E+02	1.56E+02	1.01E+02	1.01E+02
	5.84E+01	1.83E+02	2.52E+01	5.11E+01	6.78E+01	2.40E+01	6.87E+01	4.82E−02	1.39E−01
	6/-	9/-	3/-	4/-	7/-	8/-	5/-	1.5/=	1.5
F27	1.92E+03	2.66E+03	1.50E+03	6.99E+02	3.75E+03	5.78E+03	6.99E+02	1.55E+03	1.69E+03
	8.67E+01	2.56E+02	2.42E+02	1.80E+02	1.09E+02	9.35E+03	1.80E+02	9.16E+01	1.44E+02
	6/-	7/-	3.5/+	1.5/+	8/-	9/-	1.5/+	3.5/+	5
F28	6.34E+03	1.05E+04	2.61E+03	2.88E+03	5.82E+03	4.26E+03	9.25E+03	**2.07E+03**	3.22E+03
	1.07E+03	2.35E+03	5.55E+02	2.18E+02	5.45E+02	9.47E+02	3.65E+02	**7.69E+01**	8.65E+02
	7/-	9/-	2/+	3/+	6/-	5/-	8/-	**1/+**	4
F29	4.55E+06	4.80E+07	9.36E+07	**2.62E+04**	9.26E+06	5.86E+07	3.72E+08	6.04E+05	1.62E+08
	7.86E+05	1.65E+08	1.03E+08	**1.31E+04**	8.64E+05	3.34E+07	2.83E+07	2.04E+05	1.22E+08
	3/+	5/+	7/+	**1/+**	4/+	6/+	9/-	2/+	8
F30	2.85E+04	6.09E+06	1.46E+04	5.20E+04	9.53E+04	8.62E+04	6.98E+04	6.79E+04	**1.43E+04**
	1.26E+04	5.98E+06	4.82E+03	1.10E+04	5.41E+03	1.53E+04	2.36E+04	1.10E+04	**4.51E+03**
	3/-	9/-	2/-	4/-	8/-	7/-	6/-	5/-	**1**
±/=	4/25/1	2/28/0	10/17/3	12/18/0	6/24/0	6/24/0	6/23/1	6/22/2	——
Mean *Fr*	5.4833	7.7500	3.8000	3.7333	5.0833	5.2667	6.0167	4.9833	**2.8833**
Final rank	7	9	3	2	5	6	8	4	**1**

The bold numbers indicate the best solutions for each function

Table 24 Summary of the results for FLA with $D = 100$ and NFEs = 1.00E+06 on CEC2014

Function		*Npop* = 30 Mean *Fr*	*Npop* = 60	*Npop* = 90	*Npop* = 30 Std	*Npop* = 60	*Npop* = 90	*Npop* = 30 Best	*Npop* = 60	*Npop* = 90
F1	Unimodal	4.65E+05 1	5.52E+05 2	6.20E+05 3	1.61E+05	1.74E+05	2.21E+05	**3.19E+05**	3.54E+05	4.31E+05
F2		2.91E+04 2	3.97E+04 3	1.78E+04 1	4.39E+04	3.32E+04	2.90E+04	2.32E+03	1.22E+03	**9.06E+01**
F3		5.23E−08 1	2.92E−03 2	8.69E−01 3	1.10E−07	3.09E−03	9.99E−01	**5.25E−10**	3.50E−05	1.72E−01
F4	Simple Multimodal	1.30E+02 2	1.20E+02 1	1.43E+02 3	3.24E+01	3.91E+01	1.37E+01	7.94E+01	**7.65E+01**	1.27E+02
F5		2.03E+01 1	2.05E+01 2	2.06E+01 3	1.05E−01	8.08E−02	1.73E−01	**2.02E+01**	2.04E+01	2.04E+01
F6		1.30E+02 3	1.16E+02 2	1.03E+02 1	1.03E+01	1.55E+00	5.11E+00	1.15E+02	1.15E+02	**9.63E+01**
F7		1.97E−03 1	2.46E−03 2	3.94E−03 3	4.41E−03	5.51E−03	8.80E−03	1.14E−13	**0.00E+00**	1.14E−13
F8		7.16E+02 3	5.12E+02 1	5.49E+02 2	2.53E+01	7.24E+01	4.55E+01	6.90E+02	**4.33E+02**	5.01E+02
F9		7.54E+02 3	6.86E+02 2	6.80E+02 1	6.27E+01	6.59E+01	3.49E+01	6.70E+02	**5.77E+02**	6.41E+02
F10		1.45E+04 2.5	1.44E+04 2.5	1.38E+04 1	1.51E+03	1.97E+03	1.30E+03	1.26E+04	**1.17E+04**	1.25E+04
F11		1.66E+04 2.5	1.54E+04 1	1.63E+04 2.5	2.74E+03	2.60E+03	2.04E+03	1.47E+04	**1.25E+04**	1.39E+04
F12		2.12E+00 2	2.05E+00 1	2.23E+00 3	6.70E−01	7.69E−01	5.92E−01	9.76E−01	**7.51E−01**	1.39E+00

Table 24 (continued)

Function		*Npop* = 30 Mean *Fr*	*Npop* = 60	*Npop* = 90	*Npop* = 30 Std	*Npop* = 60	*Npop* = 90	*Npop* = 30 Best	*Npop* = 60	*Npop* = 90
F13		6.37E-01 3	5.87E−01 1.5	5.99E−01 1.5	1.07E−01	7.80E−02	8.95E−02	5.08E−01	5.35E−01	**4.80E−01**
F14		5.32E−01 2	5.56E−01 3	4.39E−01 1	2.80E−01	3.21E−01	2.26E−01	2.98E−01	**2.82E−01**	3.22E−01
F15		3.50E+02 3	1.76E+02 1.5	1.73E+02 1.5	1.02E+02	3.61E+01	5.90E+01	2.26E+02	1.40E+02	**1.13E+02**
F16		4.36E+01 2	4.39E+01 2	4.41E+01 2	1.38E+00	2.11E+00	4.71E−01	**4.19E+01**	**4.18E+01**	4.35E+01
F17	Hybrid	6.12E+04 1	1.37E+05 2	1.52E+05 3	1.98E+04	5.73E+04	4.12E+04	**3.23E+04**	6.43E+04	9.57E+04
F18		3.29E+03 1	5.51E+03 2	7.00E+03 3	2.79E+03	3.54E+03	6.57E+03	**9.04E+02**	1.44E+03	9.28E+02
F19		8.80E+01 1	9.78E+01 3	9.61E+01 2	3.36E+01	3.24E+01	2.74E+01	**3.98E+01**	**4.00E+01**	6.88E+01
F20		1.34E+03 2.5	1.37E+03 2.5	1.25E+03 1	1.77E+02	2.69E+02	1.61E+02	1.15E+03	**1.00E+03**	1.06E+03
F21		4.30E+04 1	5.05E+04 3	4.57E+04 2	2.47E+04	3.37E+04	2.14E+04	2.41E+04	1.70E+04	**9.60E+03**
F22		3.82E+03 1.5	2.89E+03 1.5	2.92E+03 3	8.78E+02	3.61E+02	3.06E+02	3.20E+03	**2.32E+03**	2.66E+03

Table 24 (continued)

Function		*Npop* = 30 Mean *Fr*	*Npop* = 60	*Npop* = 90	*Npop* = 30 Std	*Npop* = 60	*Npop* = 90	*Npop* = 30 Best	*Npop* = 60	*Npop* = 90
F23	Composition	3.48E+02 2	3.48E+02 2	3.48E+02 2	2.49E−13	2.49E−13	0.00E+00	**3.48E+02**	**3.48E+02**	**3.48E+02**
F24		4.67E+02 2.5	4.66E+02 2.5	4.43E+02 1	1.56E+01	1.79E+01	1.24E+01	4.48E+02	4.46E+02	**4.29E+02**
F25		3.32E+02 3	2.35E+02 1.5	2.33E+02 1.5	5.79E+01	1.02E+01	7.96E+00	2.55E+02	**2.21E+02**	**2.23E+02**
F26		1.01E+02 2	1.01E+02 2	1.01E+02 2	1.00E−01	4.50E−02	9.55E−02	**1.01E+02**	**1.01E+02**	**1.01E+02**
F27		4.01E+03 3	3.82E+03 2	3.50E+03 1	3.08E+02	3.73E+02	1.55E+02	3.77E+03	3.39E+03	**3.33E+03**
F28		9.02E+03 3	7.53E+03 2	6.32E+03 1	2.80E+03	1.50E+03	1.13E+03	5.14E+03	5.75E+03	**4.83E+03**
F29		1.23E+09 3	8.83E+08 2	4.47E+08 1	1.38E+09	3.08E+08	1.79E+08	**4.39E+07**	6.65E+08	2.68E+08
F30		1.43E+04 3	1.16E+04 1	1.30E+04 2	1.36E+03	1.76E+03	3.72E+03	1.24E+04	9.38E+03	**8.64E+03**
Mean *Fr*/Rnak, *Nb*/Rank		2.117/3	1.95/1.5	1.933/1.5	–-	–-	–-	10/3	15/1	12/2

The bold numbers indicate the best solutions for each function

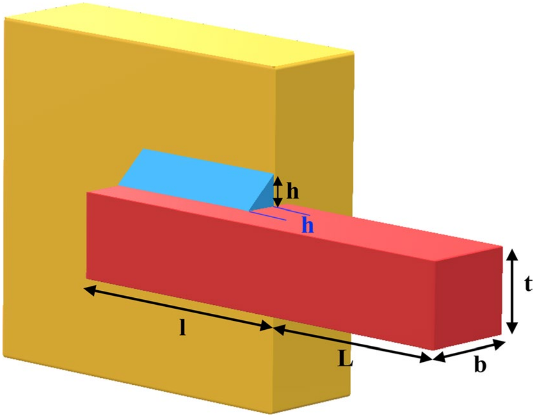


Fig. 16 Schematic of the welded beam problem

mean coil diameter [141, 166]. The mathematical expression of the issue is given in appendix A [142]:

The optimal solutions in Table 33 reveal FLA outcomes to optimize this issue alongside those in recent studies. FLA succeeded in creating a solution that was better than the more extensively researched methods. FLA won the more optimizers, with effective and acceptable solutions in the mean, worst, best cost, and Std. The best solutions obtained by FLA have been given in Table 34.

5.6 Pressure vessel optimization

The vessels have been shown in Fig. 21; the vessel's fabrication cost had to be cut so that shaping, material, and welding costs could be eschewed. Control parameters $\{s_1(T_s), s_2(T_h), s_3(R), s_4(L)\}$ denote the shell thickness, the spherical head thickness, the inner radius, and the length of the headless cylindrical section, respectively [183]. The mathematical expression of this problem is provided in appendix A [174].

There have been some studies to optimize this issue. Additional methods are found in Table 35. FLA obtained the best optimal solutions, which is clear. The obtained results seem to reveal that FLA did excellent in each statistical criterion. Table 36 reveals the best-obtained results for FLA, which will generate the optimum design for this problem.

5.7 Speed reducer optimization

As this optimization design is guided by seven control variables [176], as shown in Fig. 22, it is assumed to be a difficult one to address and solve. The formulation of this problem is presented in appendix A [176].

The comparison of the best achieved optimal results with variously studied optimizers can be found in Table 37. The optimum result of FLA is $S^*=(3.5, 0.7, 17, 7.3, 7.71531992, 3.35021467, 5.28665447)$, and its fitness value is f

Table 25 Optimal results of studied optimizers for optimal design of the welded beam problem

Methods	Best	Mean	Worst	Std
CPO [125]	1.72487	1.724866	N.A	1.44E−16
IAS [126]	1.7249	N.A	N.A	3.16E−19
SCHO [127]	1.72516	N.A	N.A	N.A
LSO [128]	1.7248658	N.A	N.A	N.A
KOA [129]	1.7248658492	1.724866	N.A	1.82E−15
SWO [130]	1.72486585	1.72486585	1.72486585	2.26E−16
GSO [131]	1.724856	1.726427	1.744697	4.29E−03
DSA [132]	1.725555	N.A	N.A	N.A
VCO [133]	1.724852	1.725133	1.725773	1.92E−03
SAO [134]	N.A	**1.72E+00**	N.A	9.54E−02
OA [135]	1.8014	1.80250	N.A	3.2E−06
MMLA [136]	1.7248544	1.7249990	1.7282878	6.25E−4
AD-IFA [137]	1.81	2.40	N.A	0.50
LS-LF-FA [137]	1.85	2.81	N.A	0.63
LF-FA [137]	1.93	3.13	N.A	0.80
FA [137]	1.88	3.41	N.A	0.92
WCA [32]	1.724856	1.726427	1.744697	4.29E−03
SFO [52]	1.73231	N.A	N.A	N.A
EPSO [138]	1.7248530	1.7282190	1.7472200	5.62E−03
FSA [139]	2.3811	2.4041	2.4889	N.A
CPSO [140]	1.728024	1.748831	1.782143	1.2926E−02
TEO [141]	1.725284	1.768040	1.931161	5.8166E−02
CDE [142]	1.73346	1.768158	1.824105	2.2194E−02
UPSO [143]	1.92199	2.83721	N.A	6.83E−01
PFA [144]	1.7248530	N.A	N.A	N.A
HGSO [49]	1.7260	1.7265	1.7325	7.66E−03
EO [145]	1.724853	1.726482	1.736725	3.257E−03
GWO [33]	1.72624	N.A	N.A	N.A
IPSO [146]	2.3810	2.3819	N.A	5.23E−03
HMS [147]	1.7255	N.A	N.A	N.A
POA [66]	1.7250306573	1.7327229652	N.A	5.146E−03
CPO [125]	1.724968	1.726504	1.728593	0.004328
FLA	**1.7248523**	**1.7248527**	**1.7248536**	3.08E−06

The bold numbers indicate the best solutions for each function

$(S^{*}) = 2994.47107$ as shown in Table 38. As is observed, FLA outperformed other comparative algorithms or provided the best results.

Table 26 Optimum variables for the welded beam problem

Variables	Value
s_1	0.20573
s_2	3.47049
s_3	9.03662
s_4	0.20573
$g_1(S)$	− 2.26533E−07
$g_2(S)$	− 3.19327E−07
$g_3(S)$	0.0
$g_4(S)$	− 3.432984
$g_5(S)$	− 0.080730
$g_6(S)$	− 0.235540
$g_7(S)$	− 1.105493E−06
Best	1.7248523

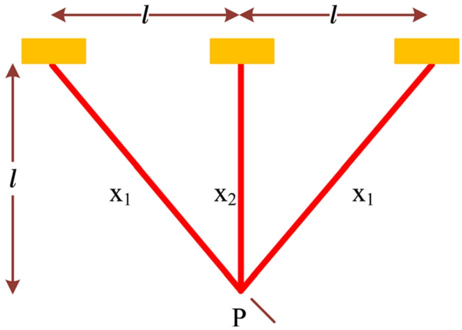


Fig. 17 Schematic of the three-bar truss problem

Table 27 Optimal results of studied optimizers for optimal design of the three-bar truss problem

Methods	Best	Mean	Worst	Std
SCHO [127]	**263.8958476**	N.A	N.A	N.A
PSA [151]	263.8958824	63.8984017	N.A	2.215E−03
AMO [152]	2.6390E+02	2.6390E+02	N.A	2.605E−04
DSA [132]	263.8958434	263.8959	263.8959	1.13E−05
ESOA [153]	**263.896**	263.909	263.948	0.0146
iLSHADEε [154, 155]	2.64E+02	2.64E+02	2.65E+02	4.47E − 01
RL-BA [156]	**263.89584**	263.9003	263.924700	6.06E − 03
AD-IFA [137]	282.84	282.84	N.A	0.00
LS-LF-FA [137]	282.84	282.84	N.A	0.00
LF-FA [137]	282.84	283.20	N.A	1.10
FA [137]	282.84	287.84	N.A	4.94
SFO [52]	263.89592128	N.A	N.A	N.A
mGWO [157]	263.8961	N.A	N.A	N.A
PSO-HBF [158]	**263.89584**	263.89617	263.89893	6.9120e − 04
FLA	**263.89584**	263.89586	263.89665	7.10E−05

The bold numbers indicate the best solutions for each function

Table 28 Optimum variables for the three-bar truss problem

Variables	Value
s_1	0.7887089
s_2	0.4081528
$g_1(S)$	− 3.14339E−09
$g_2(S)$	− 1.46421
$g_3(S)$	− 0.53579
Best	263.89584

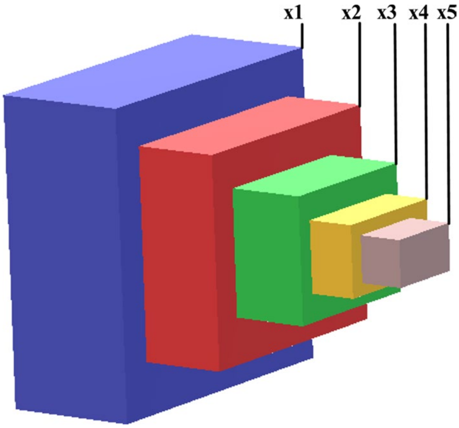


Fig. 18 Schematic of the cantilever beam problem

Table 29 Optimal results of studied optimizers for optimal design of the cantilever beam problem

Methods	Best	Mean	Worst	Std
BLPSO [159]	1.339957	1.339959	1.339965	2.11E−06
MBWO [160]	**1.3399565**	N.A	N.A	N.A
CCEO [161]	**1.33996**	N.A	N.A	N.A
IAS [126]	**1.34000**	N.A	N.A	5.27E−14
MPDO [162]	1.3400522	N.A	N.A	N.A
PSA [151]	1.339957	1.339959	1.339963	1.945E−06
EEFO [163]	1.34	1.54	N.A	0.44
AD-IFA [137]	1.34	1.95	N.A	0.91
LS-LF-FA [137]	1.45	4.91	N.A	1.63
LF-FA [137]	3.59	8.72	N.A	1.93
FA [137]	1.339956541	N.A	N.A	N.A
GCHHO [164]	1.33996	N.A	N.A	N.A
GOA [41]	1.33998	N.A	N.A	N.A
MFO [34]	1.347944	1.436524	1.664954	0.076076
WOA [159]	1.339957	N.A	N.A	N.A
SMA [55]	1.33999	N.A	N.A	N.A
m-SCA [114]	1.33996	1.33997	**N.A**	1.1E−5
FLA	**1.339956**	**1.339958**	**1.339963**	**6.48E−07**

The bold numbers indicate the best solutions for each function

Table 30 Optimum variables for the cantilever beam problem

Variables	Value
s_1	6.0166040039
s_2	5.3087497655
s_3	4.4941815962
s_4	3.5012893106
s_5	2.1528352736
$g(S)$	− 1.199011E−08
Best	1.339956

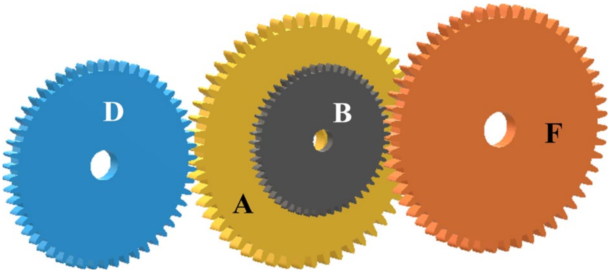


Fig. 19 Schematic of the gear train problem

Table 31 Optimal results of studied optimizers for optimal design of the gear train problem

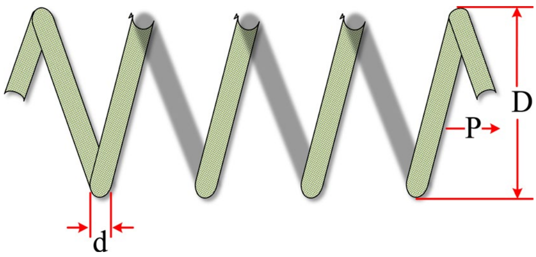
Methods	Best	Mean	Worst	Std
mGWO [157]	**2.7009E−12**	N.A	N.A	N.A
BES [165]	2.308E−11	2.096E−08	1.110E−07	2.596E−08
GOA [165]	1.166E−10	1.590E−03	4.563E−02	8.327E−03
EBS [165]	**2.701E−12**	1.705E−09	8.701E−09	2.079E−09
UPSO [143]	**2.700857E−12**	3.80562E−8	N.A	1.09E−7
FLA	**2.700857E−12**	**8.7526E−10**	**1.4069E−9**	**2.76E−9**

The bold numbers indicate the best solutions for each function

Table 32 Optimum variables for the gear train problem

Variables	Value
s_1	43
s_2	19
s_3	16
s_4	49
Best	2.700857E−12

Fig. 20 Schematic of the tension/compression spring problem



5.8 The vertical deflection optimal design of an I-beam

The problem's parameters include the height, length, and two thicknesses [198]. The problem model can be observed in Fig. 23. Also, the problem is mathematically presented in Appendix [198]:

Table 39 reports comparing the FLA and CS, SNS, AOS, SOS, and CPA when applied to solving the speed reducer design problem. As one can easily observe, the FLA provides more relevant results than its counterparts. In addition, the solutions obtained by FLA for the I-beam vertical deflection optimal design are listed in Table 40.

5.9 Tubular column optimization

An example of a design for this optimal design to take away a pressing load $P=2500$ kgf at optimum solution [137] is shown in Fig. 24. The L (length) of it is 250 cm [137]. The column is made of a ρ (density) of 0.0025 kgf cm^{-3}, a modulus of E (elasticity) of 0.85×106 kgf cm^{-2}, and a material with σ_y (yield stress) of 500 kgf cm^{-2} [130]. The stress in the column should be less than the buckling and yield stresses (constraints g_1 and g_2, respectively). The average diameter of the column is between 2 and 14 cm (constraints g_3 and g_4, respectively), and columns thicker than 0.2–0.8 cm are unavailable in the market (constraints g_5 and g_6). The objective function consists of construction and body costs and is modeled as $9.82dt+2d$, where d is the average diameter of the column in cm and t represents the thickness of the tube. The objective function of this problem is formulated as presented in Appendix A [137].

The results of FLA for this issue have been depicted in Table 41. The obtained results from applying FLA are compared with those of its counterparts. As is seen, the FLA is superior to popular techniques. In addition, the best solutions via FLA for the tubular column optimal design have been depicted in Table 42.

5.10 Piston lever optimization

Piston components, that is, $s_1(H)$, $s_2(B)$, $s_3(X)$, and $s_4(D)$, are located by solving the piston level optimal design problem. To this end, the oil volume needs to be minimized when the piston lever is lifted to 45° (refer to Fig. 25) [202]. The expression in Appendix A models the optimization problem. Several inequality limits, such as

Table 33 Optimal results of studied optimizers for optimal design of the tension/compression spring problem

Methods	Best	Mean	Worst	Std
SCHO [127]	0.0126656	N.A	N.A	N.A
PSA [151]	0.0127226	0.0132851	N.A	0.0006535
KOA [129]	0.0126652328	0.0126652328	N.A	6.00E−08
DSA [132]	0.012668	N.A	N.A	N.A
EEFO [163]	0.012666021	0.012667867	0.012682986	3.562E−06
VCO [133]	0.012665397	0.012722	0.012889	6.78E−05
GGO [167]	0.01267	0.01267	0.01267	0.0
ESOA [153]	0.0127434	0.0127839	0.0128516	3.09E−05
WO [168]	**0.012665**	0.013363	0.014771	7.07E−04
DE–QL [169]	**1.266523E − 02**	1.266581E − 02	1.267668E − 02	2.27594E − 06
VMCH [170]	**1.27E − 02**	**1.27E − 02**	**1.27E − 02**	1.58E − 08
EnMODE [171]	**1.2665E − 02**	1.2710E − 02	1.2719E − 02	2.0138E − 05
QS [166]	**0.012665**	**0.012666**	0.012669	N.A
GCHHO [164]	**0.012665264**	N.A	N.A	N.A
SMA-AGDE [172]	**0.0126652**	0.0128764	0.0138429	0.0002361
COOT [173]	**0.012665293034089**	N.A	N.A	N.A
SDO [150]	0.0126663	0.0126724	0.0126828	6.1899E−06
CPSO [140]	0.0126747	0.012730	0.012924	5.19E−05
mGWO [157]	0.012668	N.A	N.A	N.A
PFA [144]	**0.01266528**	N.A	N.A	N.A
G-QPSO [174]	**0.012665**	0.013524	0.017759	1.268E−03
WCA [32]	**0.012665**	0.012746	0.012952	8.06E−05
DDAO [175]	0.0129065	0.0151829	0.0173199	1.26E−03
CDE [142]	0.012670	0.012703	0.012790	2.07E−05
(1+λ)-ES [176]	0.012689	0.013165	N.A	3.9E−04
HPSO [124]	**0.0126652**	0.0127072	0.0127190	1.58E−05
EO [145]	0.012666	0.013017	0.013997	3.91E−04
INFO [177]	0.012666	N.A	N.A	N.A
NRBO [178]	0.0127	0.0127	N.A	2.108E−06
IMSCSO [179]	**0.012665**	N.A	N.A	N.A
LSO [128]	0.012665233	N.A	N.A	N.A
EBS [165]	**0.01266523**	0.0128560	N.A	0.00035
HGA [180]	0.012668	0.013481	0.016155	N.A
TDO [181]	0.012671024	0.012681410	0.012701561	0.00002042
UPSO [143]	0.01312	0.02294	N.A	7.2E−03
CSA [182]	0.012667428	0.012669345	0.012672344	1.23211E−03
SCA [182]	0.012709667	0.012839637	0.012998448	0.000078
MVO [182]	0.012816930	0.014464372	0.017839737	0.001622
MFO [182]	0.012753902	0.014023657	0.017236590	0.001390
FLA	**0.0126652**	**0.012666**	**0.012667**	6.29E−07

The bold numbers indicate the best solutions for each function

Table 34 Optimum variables for the tension/compression spring problem

Variables	Value
s_1	0.051734882
s_2	0.357820340
s_3	11.224654530
$g_1(S)$	– 1.14952E–06
$g_2(S)$	– 1.64586E–06
$g_3(S)$	– 4.05594
$g_4(S)$	– 0.72696
Best	0.0126652

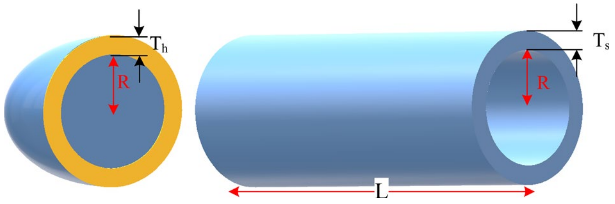


Fig. 21 Schematic of the pressure vessel problem

Table 35 Optimal results of studied optimizers for optimal design of the pressure vessel problem

Methods	Best	Mean	Worst	Std
YDSE [184]	6.0597E+03	6.0597E+03	6.0597E+03	9.25E−13
VCO [133]	6.61E+03	6.77E+03	6.79E+03	6.48E+01
BP-εMAg-ES [185]	**6.0597E+03**	6.0671E+03	6.0905E+03	1.3431E+01
COLSHADE [186]	**6059.7143**	6062.179252	6090.5262	8.359059
DE−QL [169]	8.077438E+03	1.085138E+04	1.355810E+04	1.459232E+03
VMCH [170]	6.07E+03	6.06E+03	6.37E+03	6.23E+01
UPSO [143]	6154.70	8016.37	9387.77	745.869
G-QPSO [174]	6059.7208	6440.3786	7544.4925	448.4711
CPSO [140]	6061.0777	6147.1332	6363.8041	86.4545
mGWO [157]	6059.7140	N.A	N.A	N.A
RFO [60]	6113.3195	N.A	N.A	N.A
EO [145]	**6059.7143**	6668.114	7544.4925	566.24
CDE [142]	6059.7340	6085.2303	6371.0455	43.013
DHOA [187]	6103.842	N.A	N.A	N.A
FLA	**6059.714335**	**6060.210562**	**6090.5261**	**3.86**

The bold numbers indicate the best solutions for each function

geometrical conditions, minimum piston stroke, maximum bending moment of the lever, and imposed-force equilibrium, are also considered [198].

Table 36 Optimum variables for the pressure vessel problem

Variables	Value
s_1	0.8125
s_2	0.4375
s_3	42.098445590
s_4	176.63659592
$g_1(S)$	– 1.13E–10
$g_2(S)$	– 0.03588
$g_3(S)$	– 2.78875E–05
$g_4(S)$	– 63.36340
Best	6059.714335

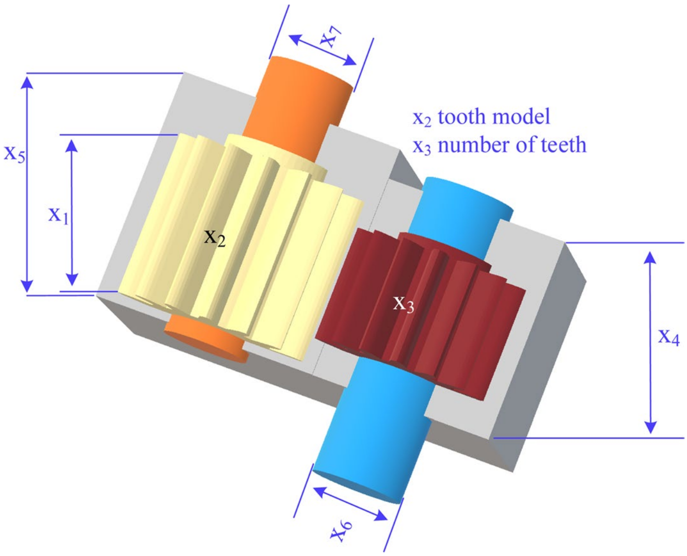


Fig. 22 Schematic of the speed reducer problem

Table 43 compares the results obtained via FLA and some other methods when applied to solving the piston lever design problem. It is seen that the FLA provides more suitable results than its counterparts. In addition, the best solutions via FLA for the piston lever optimal design have been shown in Table 44.

5.11 Corrugated bulkhead optimization

Some of the corrugated bulkhead design applications are in chemical tankers, bulk carriers, and product oil carriers. This problem attempts to minimize the weight of corrugated bulkheads for a tanker (refer to Fig. 26) [137]. The variables include

Table 37 Optimal results of studied optimizers for optimal design of the speed reducer problem

Methods	Best	Mean	Worst	Std
WOA [188]	2996.604340	3003.042915	3233.598124	4.0888E+01
SSA [188]	2996.021720	3005.574377	3015.662612	4.63871
MBA [188]	2994.471371	2944.744438	2994.484789	2.4195E−03
GWO [188]	2995.704435	3001.556162	3009.944297	4.1218
ER-WCA [188]	**2994.471066**	2996.744541	3007.436552	4.3876
ALO [188]	2996.521745	3005.644280	3014.379001	4.7422
LFD [189]	3007.7820	N.A	N.A	N.A
ACVO [190]	2994.4718	N.A	N.A	N.A
EChOA [189]	**2994.4711**	N.A	N.A	N.A
I-GWO [189]	2996.3482	N.A	N.A	N.A
HFPSO [189]	3003.7076	N.A	N.A	N.A
HEAA [148]	2994.499107	2994.613368	2994.752311	7.0E−02
SHO [191]	2998.5507	2999.64	3003.889	1.93E+00
SETO [192]	2994.4991	N.A	N.A	N.A
LFD [192]	2994.5173	N.A	N.A	N.A
SELO [192]	2999.2274	N.A	N.A	N.A
AHA [193]	2994.471158	2994.471652	2994.473229	4.2512E−04
AO [62]	3007.7328	N.A	N.A	N.A
MBWO [160]	2995.438303	N.A	N.A	N.A
CCEO [161]	2995.4374	N.A	N.A	N.A
MPDO [162]	2995.2477	N.A	N.A	N.A
SCHO [127]	2996.3482	N.A	N.A	N.A
GAO [194]	2.9944E+03	2.9944E+03	2.9944E+03	7.778E−10
YDSE [184]	N.A	2994.7831	N.A	1.0037
LEA [195]	**2994.471**	2995.1210	2998.0923	2.9012E−02
CSA [182]	3030.5633	3065.9172	3104.7791	18.0742
SCA [182]	3002.9281	3028.8411	3060.9582	13.0186
MVO [182]	3009.5717	3021.2565	3054.5248	11.0235
MFO [182]	2996.5157	N.A	N.A	N.A
RSA [196]	3029.873076	N.A	N.A	N.A
hHHO-SCA [197]	2997.9157	N.A	N.A	N.A
AOA [103]	2996.3482	2999.88	3001.491	1.782335
FLA	**2994.47107**	2994.471589	2994.473074	2.09E−04

width (s_1), depth (s_2), length (s_3), and thickness of the plate (x_4). The formulation in Appendix A models the problem in a mathematical $\left[\left(s_1, s_2, s_3, s_4\right) = (b, h, l, t)\right]$ [137].

Heuristic approaches were applied to this issue, including AEFA-C, FPSA, AD-IFA, LS-LF-FA, LF-FA, and FA. FLA is also applied to look for the optimum solution to this issue with 200 iterations. Table 45 lists a summary of the results presented by mentioned methods. In comparison, FLA finds the most suitable solution with the same computational burden. So, this is inferred that the FLA provides more

Table 38 Optimum variables for the speed reducer problem

Variables	Value
s_1 (b)	3.5
s_2 (m)	0.70
sx_3 (z)	17.0
sx_4 (l_1)	7.30
s_5 (l_2)	7.71531992
s_6 (d_1)	3.35021467
s_7 (d_2)	5.28665447
$g_1(S)$	− 0.0739153
$g_2(S)$	− 0.19799853
$g_3(S)$	− 0.49917225
$g_4(S)$	− 0.9046439
$g_5(S)$	− 3.4954948E−09
$g_6(S)$	− 2.84692292E−09
$g_7(S)$	− 0.70250
$g_8(\mathrm{S})$	0.0
$g_9(S)$	− 0.58333333
$g_{10}(\mathrm{S})$	− 0.051325753
$g_{11}(S)$	− 3.8883663E−10
Best	2994.47107

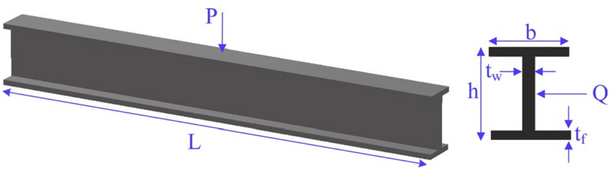


Fig. 23 Schematic of the I-beam vertical deflection problem

Table 39 Optimal results of studied optimizers for optimal design of the I-beam problem

Methods	Best	Mean	Worst	Std
YDSE [184]	**1.3074E−02**	**1.3074E−02**	**1.3074E−02**	8.8219E−18
SRS [199]	**0.0130741**	0.0130893	0.01322180	3.371E− 05
CPA [63]	0.0132240	N.A	N.A	N.A
SOS [113]	**0.0130741**	0.0130884	N.A	4.0E−05
FLA	**0.013074**	**0.01307445**	0.01307579	6.91E−06

The bold numbers indicate the best solutions for each function

desirable results. The performance of other algorithms is also convincing to some degree. The best optimal results of FLA have been shown in Table 46.

Table 40 Optimum variables for the I-beam problem

Variables	Value
s_1	80.0
s_2	50.0
s_3	0.90
s_4	2.321792260
$g_1(S)$	− 6.80E−08
$g_2(S)$	− 1.5702285
Best	0.013074

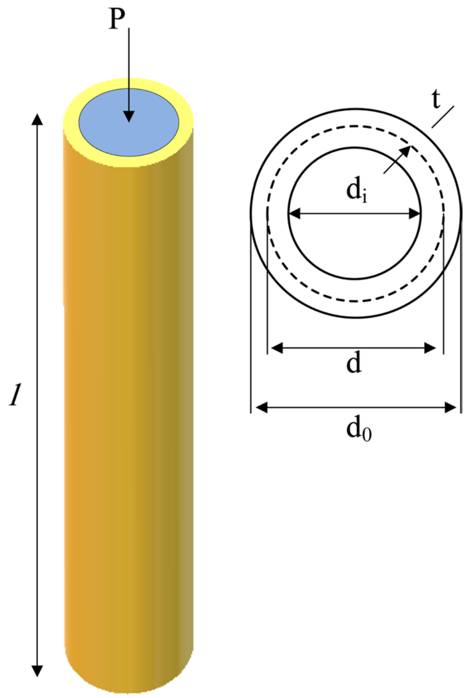


Fig. 24 Schematic of the tubular column problem

Table 41 Optimal results of studied optimizers for optimal design of the tubular column problem

Methods	Best	Mean	Worst	Std
AD-IFA [137]	26.50	26.54	N.A	0.07
LS-LF-FA [137]	26.50	26.58	N.A	0.19
LF-FA [137]	26.50	27.46	N.A	1.36
FA [137]	26.52	28.74	N.A	2.08
DO [200]	26.499497	26.499499	N.A	1.41E−06
KOA [129]	26.499497	26.499497	N.A	1.81E−14
DBB-BC [201]	26.49954	26.50117	N.A	2.0E−03
FLA	**26.4994969**	26.4995014	26.5100331	1.41E−04

The bold number indicates the best solutions for each function

Table 42 Optimum variables for the tubular column problem

Optimal variables	Value
s_1	5.45115622
s_2	0.29196548
$g_1(S)$	− 7.168066E−09
$g_2(S)$	− 2.00896E−09
$g_3(S)$	− 0.6331054
$g_4(S)$	− 0.6106317
$g_5(S)$	− 0.3149875
$g_6(S)$	− 0.63504315
Best	26.4994969

5.12 *Car* side impact optimization

Gu et al. [205] discussed this problem for the first time. This design problem attempts to discover the best weight of the car with the help of eleven mixed control parameters. The eighth and the nine control parameters are discrete, while the rest are continuous control parameters. The problem formulation and its constraints are described in Appendix A [188].

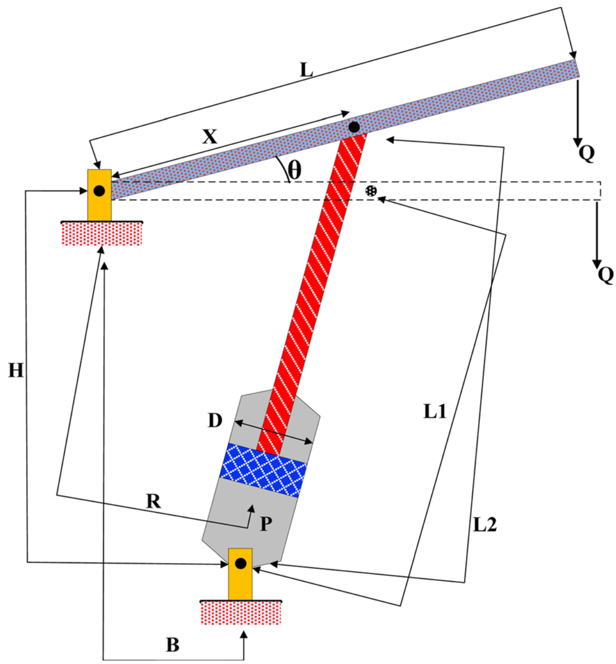


Fig. 25 Schematic of the piston lever problem

Table 43 Optimal results of studied optimizers for optimal design of the piston lever problem

Methods	Best	Mean	Worst	Std
PSO [202]	122	166	294	51.7
DE [202]	159	187	199	14.2
GA [202]	161	185	216	18.2
HPSO [202]	162	187	197	13.4
HPSO-Q [202]	129	151	168	13.4
SNS [198]	8.41269835	24.318974	167.472775	47.71792646
FLA	**8.41269832410**	23.82125	167.232196	47.2

The bold number indicates the best solutions for each function

Table 44 Optimum variables for the piston lever problem

Optimal variables	Value
s_1	0.05
s_2	2.04151359
s_3	120
s_4	4.08302718
$g_1(S)$	− 1.7449124E+03
$g_2(S)$	− 600,000
$g_3(S)$	− 1.17185459E+02
$g_4(S)$	− 4.50E−04
Best	8.41269832410

The statistical solutions of FLA can be found in Table 47. As per Table 47, FLA gives the best solution compared to its counterpart. In terms of robustness, FLA obtained the best value of the standard deviation. The best optimal results of FLA have been shown in Table 48.

6 Conclusion and future works

Due to the variety of complex and nonlinear problems in the real world, including single or multi-objective, constrained or non-constraint with continuous or discrete variables, existing methods need to be improved, and new techniques developed so that researchers can achieve optimal solutions to any problem. In this study, we suggested an effective meta-heuristic algorithm called FLA, inspired by a destructive and invasive natural phenomenon, namely floods in watersheds. The flood is moving very fast and has turbulence in the water mass, so we were inspired to use this algorithm. FLA prevents the algorithm's population from getting stuck in the local optimal solution by creating extraordinary turbulence in the particles. This algorithm can search globally and locally well, so it can be seen as valuable and important. The power and performance of FLA on the modern and standard functions CEC2005, 2014 with dimension 30, which covers the optimization functions

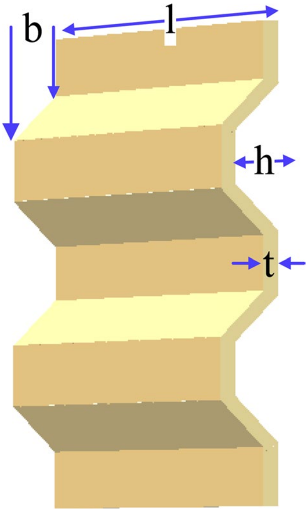


Fig. 26 Schematic of the corrugated bulkhead problem

Table 45 Optimal results of studied optimizers for optimal design of the corrugated bulkhead problem

Methods	Best	Mean	Worst	Std
AEFA-C [203]	6.845841	6.846779	6.846779	0.0
FPSA [204]	7.008391	N.A	N.A	N.A
AD-IFA [137]	6.84	7.21	N.A	0.58
LS-LF-FA [137]	6.86	7.44	N.A	0.67
LF-FA [137]	6.95	8.83	N.A	1.26
FA [137]	7.21	10.23	N.A	1.95
FLA	**6.842958010**	6.8429676	6.8432916	1.25E−05

The bold number indicates the best solutions for each function

Table 46 Optimum variables for the corrugated bulkhead problem

Optimal variables	Value
s_1	57.69230769
s_2	34.14762035
s_3	57.69230769
s_4	1.05
$g_1(S)$	− 2.406946E+02
$g_2(S)$	− 2.326884E−06
$g_3(S)$	− 3.60E−11
$g_4(S)$	− 3.60E−11
$g_5(S)$	0.0
$g_6(S)$	− 23.54468
Best	6.842958010

Table 47 Optimal results of studied optimizers for optimal design of the car side impact problem

Methods	Best	Mean	Worst	Std
MPDO [162]	23.19869131	N.A	N.A	N.A
MGO [206]	22.84324	23.14917	23.3004	5.59E−02
RAO-3 [206]	23.23209	24.37533	28.4253	2.09E−01
PSA [207]	26.48636	26.48636	N.A	3.16E−15
WOA [188]	23.0421622	24.81448617	27.36081368	9.6570E−01
SSA [188]	22.846514	23.253716	23.829531	3.0557E−01
MBA [188]	22.843596	22.936421	23.488942	1.5258E−01
WCA [188]	22.84303648	22.97516442	23.37093376	1.9772E−01
ER-WCA [188]	22.84326461	23.06992534	24.4553128	3.5021E−01
ALO [188]	22.842981	23.108403	23.824366	2.9093E−01
MFO [188]	22.8429708736	22.972835	23.687547	2.0794E−01
T-CSS [208]	22.847848	22.903653	23.800904	0.078565
CSS [208]	23.007336	23.523265	24.863563	0.562345
FACSS [209]	22.84907401	22.91212354	23.05362562	4.726E−02
FLA	**22.8429706205**	22.8891475	23.17638342	7.38E−03

The bold number indicates the best solutions for each function

Table 48 Optimum variables for the car side impact problem

Optimal variables	Value
s_1	0.5
s_2	1.11623305
s_3	0.50000008
s_4	1.30241786
s_5	0.5
s_6	1.5
s_7	0.5
s_8	0.3450
s_9	0.3450
s_{10}	− 19.58523995
s_{11}	0.00058232
$g_1(S)$	− 0.6548805
$g_2(S)$	− 0.07425069
$g_3(S)$	− 0.0649099
$g_4(S)$	− 0.0468826
$g_5(S)$	− 3.7172318
$g_6(S)$	− 6.125633
$g_7(S)$	− 3.996794E−07
$g_8(S)$	− 1.552761E−09
$g_9(S)$	− 0.9657752
$g_{10}(S)$	− 0.296410
Best	22.8429706205

in the real world, are compared with the standard and widely used meta-heuristic algorithms. The findings indicate that a larger population size increases the algorithm's efficacy, and it exhibits commendable proficiency in addressing high dimensions within the optimization problem. In addition, the FLA has been studied on 12 well-known engineering problems, and the results of which show the robustness and superiority of FLA compared to the outcomes of a large number of recent articles.

As inferred from applying FLA to some well-known optimization problems, FLA provides more desirable solutions based on the quality of the results and computational time and speed. Moreover, this algorithm shows a satisfactory performance in terms of avoiding the local optima. For problems that include several different search spaces with different constraints and limits, further study is required to propose a binary FLA method to solve real problems—noting that FLA's application can be widened to include multi-objective optimization problems. Eventually, the FLA can be considered a reliable approach to solving complicated problems in the engineering realm.

Appendix

A: Formulation of engineering optimization problems

A 1. Formulation of the welded beam optimal design

Minimize:

$$f(S) = f\left(S\left(\left[s_1, s_2, s_3, s_4\right]\right)\right) = 1.10471 s_2 s_1^2 + 0.04811 s_3 s_4 \left(14 + s_2\right) \tag{15}$$

By considering:

$$g_1(X) = g_1(S) = \tau(S) - \tau_{max} \leq 0 \tag{16}$$

$$g_2(X) = g_2(S) = \sigma(S) - \sigma_{max} \leq 0 \tag{17}$$

$$g_3(X) = g_3(S) = s_1 - s_4 \leq 0 \tag{18}$$

$$g_4(X) = g_4(S) = 0.10471 s_1^2 + 0.04811 s_3 s_4 \left(14 + s_2\right) - 5 \leq 0 \tag{19}$$

$$g_5(X) = g_5(S) = 0.125 - s_1 \leq 0 \tag{20}$$

$$g_6(X) = g_6(S) = \delta(S) - \delta_{max} \leq 0 \tag{21}$$

$$g_7(X) = g_7(S) = P - P_c(S) \leq 0 \tag{22}$$

where

$$\tau(S) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{s_2}{2R} + (\tau'')^2} \tag{23}$$

$$\tau' = \frac{P}{\sqrt{2}s_1 s_2}, \tau'' = \frac{MR}{J} \tag{24-25}$$

$$M = P\left(L + \frac{s_2}{2}\right), R = \sqrt{\frac{s_2^2}{4} + \left(\frac{s_1 + s_3}{2}\right)^2}, \delta(S) = \frac{4PL^3}{Es_3^3 s_4} \tag{26-28}$$

$$J = 2\left[\sqrt{2}s_1 s_2\left\{\frac{s_2^2}{12} + \left(\frac{s_1 + s_3}{2}\right)^2\right\}\right], \sigma(S) = \frac{6PL}{s_4 s_3^2} \tag{29-30}$$

$$P_c(S) = \frac{4.013E\sqrt{\frac{s_4^6 s_3^2}{36}}}{L^2}\left(1 - \frac{s_3}{2L}\sqrt{\frac{E}{4G}}\right) \tag{31}$$

E = 30e6 psi, σ_{max} = 30,000 L = 14 in, psi G = 12e6 psi, P = 6,000 lb, τ_{max} = 13,000 psi,

δ_{max} = 0.25 in, $0.1 \le s_1, s_4 \le 2$, $0.1 \le s_2, s_3 \le 10$.

A 2. Formulation of the Three-Bar truss optimization

Minimize:

$$f(S) = l \times \left(2\sqrt{2}s_1 + s_2\right) \tag{32}$$

By considering:

$$g_1(S) = P \times \frac{\sqrt{2}s_1 + s_2}{\sqrt{2}s_1^2 + 2s_1 s_2} - \sigma \le 0 \tag{33}$$

$$g_2(S) = P \times \frac{s_2}{\sqrt{2}s_1^2 + 2s_1 s_2} - \sigma \le 0 \tag{34}$$

$$g_3(S) = P \times \frac{1}{\sqrt{2}s_2 + s_1} - \sigma \le 0 \tag{35}$$

σ = 2kN/cm^2, L = 100 cm, P = 2kN/cm.^{2}

A 3. Formulation of the cantilever beam optimization

Minimize:

$$f(S) = 0.0624 \times \left(s_1 + s_2 + s_3 + s_4 + s_5\right) \tag{36}$$

By considering:

$$g(S) = \frac{61}{s_1^3} + \frac{37}{s_2^3} + \frac{19}{s_3^3} + \frac{7}{s_4^3} + \frac{1}{s_5^3} - 1 \leq 0 \tag{37}$$

where $0.01 \leq s_1, s_2, s_3, s_4, s_5 \leq 100$.

A 4. Formulation of the cantilever beam optimization

Minimize:

$$f(S) = \left(\left(\frac{1}{6.931}\right) - \left(\frac{s_2 s_3}{s_1 s_4}\right)\right)^2 \tag{38}$$

$12 \leq s_i \leq 60, i = 1, 2, 3, 4.$

A 5. Formulation of the tension/compression spring optimization

Minimize:

$$f(S) = \left(s_3 + 2\right) s_2 s_1^2 \tag{39}$$

By considering:

$$g_1(S) = 1 - \frac{s_2^3 s_3}{71785 s_1^4} \leq 0 \tag{40}$$

$$g_2(S) = \frac{4s_2^2 - s_1 s_2}{12566\left(s_1^3 s_2 - s_1^4\right)} + \frac{1}{5108 s_1^2} - 1 \leq 0 \tag{41}$$

$$g_3(S) = 1 - \frac{140.45 s_1}{s_2^2 s_3} \leq 0 \tag{42}$$

$$g_4(S) = \frac{s_1 + s_2}{1.5} - 1 \leq 0 \tag{43}$$

$0.05 \leq s_1 \leq 2, 0.25 \leq s_2 \leq 1.3, 2 \leq s_3 \leq 15.$

A 6. Formulation of the tension pressure vessel optimal design

Minimize:

$$f(S) = 0.6224s_1s_3s_4 + 1.7781s_2s_3^2 + 3.1661s_1^2s_4 + 19.84s_1^2s_3 \tag{44}$$

By considering:

$$g_1(S) = -s_1 + 0.0193s_3 \le 0 \tag{45}$$

$$g_2(S) = -s_2 + 0.00954s_3 \le 0 \tag{46}$$

$$g_3(S) = -\pi s_3^2s_4 - \frac{4}{3}\pi s_3^3 + 1296000 \le 0 \tag{47}$$

$$g_4(S) = s_4 - 240 \le 0 \tag{48}$$

$0 \le s_i \le 100$, i=1, 2.
$10 \le s_i \le 200$, i=3, 4.

A 7. Formulation of the speed reducer optimization

Minimize:

$$f\left(X : \left[x_1(b), x_2(m), x_3(z), x_4\left(l_1\right), x_5\left(l_2\right), x_6\left(d_1\right), x_7\left(d_2\right)\right]\right)$$

$$\begin{aligned} &= f\left(S:\left[s_1, s_2, s_3, s_4, s_5, s_6, s_7\right]\right) \\ &= 0.7854bm^2\left(3.3333z^2 + 14.9334z - 43.0934\right) \\ &\quad - 1.508b\left(d_1^2 + d_2^2\right) + 7.4777\left(d_1^3 + d_2^3\right) \\ &\quad + 0.7854\left(l_1d_1^2 + l_2d_2^2\right) \end{aligned} \tag{49}$$

By considering:

$$g_1(S) = \frac{27}{bm^2z} - 1 \le 0 \tag{50}$$

$$g_2(S) = \frac{397.5}{bm^2z^2} - 1 \le 0 \tag{51}$$

$$g_3(S) = \frac{1.93 \times l_1^3}{mzd_1^4} - 1 \le 0 \tag{52}$$

$$g_4(S) = \frac{1.93 \times l_2^3}{mzd_2^4} - 1 \leq 0 \tag{53}$$

$$g_5(S) = \frac{\sqrt{\left(\frac{745l_1}{mz}\right)^2 + 1.69e6}}{110d_1^3} - 1 \leq 0 \tag{54}$$

$$g_6(S) = \frac{\sqrt{\left(\frac{745l_2}{mz}\right)^2 + 157.5e6}}{85d_2^3} - 1 \leq 0 \tag{55}$$

$$g_7(S) = \frac{mz}{40} - 1 \leq 0 \tag{56}$$

$$g_8(S) = \frac{5m}{b} - 1 \leq 0 \tag{57}$$

$$g_9(S) = \frac{b}{12m} - 1 \leq 0 \tag{58}$$

$$g_{10}(S) = \frac{1.5 \times d_1 + 1.9}{l_1} - 1 \leq 0 \tag{59}$$

$$g_{11}(S) = \frac{1.1 \times d_2 + 1.9}{l_2} - 1 \leq 0 \tag{60}$$

A 8. Formulation of the vertical deflection optimal design of an I-Beam

Minimize:

$$f\left(s_1, s_2, s_3, s_4\right) = f\left(h, b, t_w, t_f\right) = \frac{5000}{\frac{t_w(h-2t_f)^3}{12} + \frac{bt_f^3}{6} + 2bt_f\left(\frac{h-t_f}{2}\right)^2} \tag{61}$$

By considering:

$$g_1 = 2bt_f + t_w\left(h - 2t_f\right) \leq 300 \tag{62}$$

$$g_2 = \frac{18h \times 10^4}{t_w(h - 2t_f)^3 + 2bt_f\left(4t_f^2 + 3h\left(h - 2t_f\right)\right)} + \frac{15b \times 10^3}{\left(h - 2t_f\right)t_w^3 + 2t_f b^3} \leq 6 \tag{63}$$

$$0.9 \leq t_f \leq 50.9 \leq t_w \leq 510 \leq b \leq 5010 \leq h \leq 80$$

A 9. Formulation of the tubular column optimal design

Minimize:

$$f(S) = f(d, t) = 9.8dt + 2d \tag{64}$$

By considering:

$$g_1 = \frac{P}{\pi dt\sigma_y} - 1 \leq 0 \tag{65}$$

$$g_2 = \frac{8P}{\pi^3 Edt\left(d^2 + t^2\right)} - 1 \leq 0 \tag{66}$$

$$g_3 = \frac{2}{d} - 1 \leq 0 \tag{67}$$

$$g_4 = \frac{d}{14} - 1 \leq 0 \tag{68}$$

$$g_5 = \frac{0.2}{t} - 1 \leq 0 \tag{69}$$

$$g_6 = \frac{t}{0.8} - 1 \leq 0 \tag{70}$$

A 10. Formulation of the piston lever optimal design

Minimize:

$$f(S) = f(H, B, X, D) = \frac{1}{4}\pi D^2\left(L_2 - L_1\right) \tag{71}$$

By considering:

$$g_1 = QL\cos\theta - RF \leq 0 at \theta = 45^\circ \tag{72}$$

$$g_2 = -M_{max} + Q(L - X) \leq 0 \tag{73}$$

$$g_3 = 1.2\left(L_2 - L_1\right) - L_1 \leq 0 \tag{74}$$

$$g_4 = \frac{D}{2} - B \leq 0 \tag{75}$$

$$R = \frac{|-X(X\sin\theta + H) + H(B - X\cos\theta)|}{\sqrt{(X-B)^2 + H^2}} \tag{76}$$

$$F = \frac{\pi P D^2}{4} \tag{77}$$

$$L_1 = \sqrt{(X-B)^2 + H^2} \tag{78}$$

$$L2 = \sqrt{(X\sin 45 + H)^2 + (B - X\cos 45)^2} \tag{79}$$

$0 \leq s_1(H), s_2(B), s_4\,(D) \leq 500, 005 \leq s_3(X) \leq 120.$

In this equation, the payload equals Q=10,000 *lbs*, L (lever) =240 *in*, M_{max} (maximum bending moment) =1.8×10[6] *lbs.in*, and P (oil pressure) =1,500 *psi*.

A 11. Formulation of the corrugated bulkhead optimal design

Minimize:

$$f(S) = f(b,h,l,t) = \frac{5.885t(b+l)}{b + \sqrt{l^2 - h^2}} \tag{80}$$

By considering:

$$g_1 = th\left(0.4b + \frac{l}{6}\right) - 8.94\left(b + \sqrt{l^2 - h^2}\right) \geq 0 \tag{81}$$

$$g_2 = th^2\left(0.2b + \frac{l}{12}\right) - 2.2(8.94\left(b + \sqrt{l^2 - h^2}\right))^{\frac{4}{3}} \geq 0 \tag{82}$$

$$g_3 = t - 0.0156b - 0.15 \geq 0 \tag{83}$$

$$g_4 = t - 0.0156l - 0.15 \geq 0 \tag{84}$$

$$g_5 = t - 1.05 \geq 0 \tag{85}$$

$$g_6 = l - h \geq 0 \tag{86}$$

$$0 < b, h, l < 1000 < t < 5$$

A 12. Formulation of the car side impact optimization

Minimize:

$$f(S) = 1.98 + 4.90s_1 + 6.67s_2 + 6.98s_3 + 4.01s_4 + 1.78s_5 + 2.73s_7 \tag{87}$$

By considering:

$$g_1(S) = 1.16 - 0.3717s_2s_4 - 0.00931s_2s_{10} - 0.484s_3s_9 + 0.01343s_6s_{10} \leq 1 \tag{88}$$

$$\begin{aligned} g_2(S) = 0.261 &- 0.0159s_1s_2 - 0.188s_1s_8 - 0.019s_2s_7 + 0.0144s_3s_5 + 0.0008757s_5s_{10} \\ &+ 0.08045s_6s_9 + 0.00139s_8s_{11} + 0.00001575s_{10}s_{11} \leq 0.32 \end{aligned} \tag{89}$$

$$\begin{aligned} g_3(S) = 0.214 &+ 0.00817s_5 - 0.131s_1s_8 - 0.0704s_1s_9 + 0.03099s_2s_6 \\ &- 0.018s_2s_7 + 0.0208s_3s_8 + 0.121s_3s_9 - 0.00364s_5s_6 \\ &+ 0.0007715s_5s_{10} - 0.0005354s_6s_{10} + 0.00121s_8s_{11} \leq 0.32 \end{aligned} \tag{90}$$

$$g_4(S) = 0.74 - 0.61s_2 - 0.163s_3s_8 + 0.001232s_3s_{10} - 0.166s_7s_9 + 0.227s_2^2 \leq 0.32 \tag{91}$$

$$g_5(S) = 28.98 + 3.818s_3 - 4.2s_1s_2 + 0.0207s_5s_{10} + 6.63s_6s_9 - 7.7s_7s_8 + 0.32s_9s_{10} \leq 32 \tag{92}$$

$$\begin{aligned} g_6(S) = 33.86 &+ 2.95s_3 + 0.1792s_{10} - 5.057s_1s_2 - 11s_2s_8 \\ &- 0.0215s_5s_{10} - 9.98s_7s_8 + 22s_8s_9 \leq 32 \end{aligned} \tag{93}$$

$$g_7(S) = 46.36 - 9.9s_2 - 12.9s_1s_8 + 0.1107s_3s_{10} \leq 32 \tag{94}$$

$$g_8(S) = 4.72 - 0.5s_4 - 0.19s_2s_3 - 0.0122s_4s_{10} + 0.009325s_6s_{10} + 0.000191s_{11}^2 \leq 4 \tag{95}$$

$$g_9(S) = 10.58 - 0.674s_1s_2 - 1.95s_2s_8 + 0.02054s_3s_{10} - 0.0198s_4s_{10} + 0.028s_6s_{10} \leq 9.9 \tag{96}$$

$$g_{10}(S) = 16.45 - 0.489s_3s_7 - 0.843s_5s_6 + 0.0432s_9s_{10} - 0.0556s_9s_{11} - 0.000786s_{11}^2 \leq 15.7 \tag{97}$$

$$\begin{aligned} &0.5 \leq s_1, s_3, s_4 \leq 1.5, 0.45 \leq s_2 \leq 1.35, 0.875 \leq s_5 \leq 2.625 \\ &0.4 \leq s_6, s_7 \leq 1.2, s_8, s_9 \in \{0.192, 0.345\}, 0.5 \leq s_{10}, s_{11} \leq 1.5 \end{aligned}$$

Author contributions CRediT author statementMojtaba Ghasemi was involved in the supervision, conceptualization, methodology, software, investigation, validation, and writing—original draft preparation. Keyvan Golalipour assisted in the writing—original draft preparation, visualization, and investigation. Mohsen Zare was involved in the writing—original draft preparation, visualization, and investigation. Seyedali Mirjalili contributed to the writing—original draft preparation, visualization, and investigation. Pavel Trojovský contributed to the writing—original draft preparation, visualization, and investigation.

Laith Abualigah was involved in the writing—original draft preparation, visualization, and investigation. Rasul Hemmati assisted in the writing—reviewing and editing

Data availability Data are available from the authors upon reasonable request.

Declarations

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

1. Zhang H, Zhang Y, Niu Y et al (2023) T cell immune algorithm: a novel nature-inspired algorithm for engineering applications. IEEE Access 11:95545
2. Houssein EH, Oliva D, Samee NA et al (2023) Liver cancer algorithm: a novel bio-inspired optimizer. Comput Biol Med 165:107389
3. Rao RV, Savsani VJ, Rao RV, Savsani VJ (2012) Mechanical design optimization using advanced optimization techniques. Springer, London
4. Toklu YC, Bekdas G, Nigdeli SM (2021) Metaheuristics for structural design and analysis. Wiley, London
5. Qi H, Liu X, Gani A, Gong C (2024) Quantum particle swarm optimized extreme learning machine for intrusion detection. J Supercomput. https://doi.org/10.1007/s11227-024-06022-y
6. Furui K, Ohue M (2024) Fastlomap: faster lead optimization mapper algorithm for large-scale relative free energy perturbation. J Supercomput. https://doi.org/10.1007/s11227-024-06006-y
7. Zhao L, Bai Y (2024) Unlocking the ocean 6G: a review of path-planning techniques for maritime data harvesting assisted by autonomous marine vehicles. J Mar Sci Eng 12:126. https://doi.org/10.3390/jmse12010126
8. Dalklint A, Wallin M, Tortorelli D (2024) Simultaneous shape and topology optimization of inflatable soft robots. Comput Method Appl Mech Eng 420:116751. https://doi.org/10.1016/j.cma.2024.116751
9. Wang Y, Ye C, Wang Y (2024) Hierarchical multi-objective optimization of proton exchange membrane fuel cell with parameter uncertainty. Swarm Evol Comput 86:101464. https://doi.org/10.1016/j.swevo.2023.101464
10. Yang X-S (2010) Engineering optimization: an introduction with metaheuristic applications. Wiley, London
11. Salehnia T, Fathi A (2021) Fault tolerance in LWT-SVD based image watermarking systems using three module redundancy technique. Expert Syst Appl 179:115058
12. Drigo M (1996) The ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybernet-Part B 26:1–13
13. Tarasewich P, McMullen PR (2002) Swarm intelligence: power in numbers. Commun ACM 45:62–67
14. Ali AM, Abdelhafeez A (2022) DeepHAR-Net: a novel machine intelligence approach for human activity recognition from inertial sensors. Sustain Mach Intell J 2(1):1
15. Abdel-Monem A, Abouhawwash M (2022) A machine learning solution for securing the internet of things infrastructures. Sustain Mach Intell J 1(1):4–1
16. Abdelhafeez A, Aziz A, Khalil N (2022) Building a sustainable social feedback loop: a machine intelligence approach for twitter opinion mining. Sustain Mach Intell J 5(1):6–1
17. Givi H, Dehghani M, Montazeri Z et al (2021) GBUO: "the good, the bad, and the ugly" optimizer. Appl Sci 11:2042

18. Kirkpatrick S, Gelatt CD Jr, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680
19. Rechenberg I (1989) Evolution strategy: nature's way of optimization. In: optimization: methods and applications, possibilities and limitations. In: Proceedings of an international seminar organized by Deutsche Forschungsanstalt für Luft-und Raumfahrt (DLR), Bonn, Berlin, Heidelberg. Springer, Berlin, Heidelberg, pp 106–126
20. Holland JH (1992) Genetic algorithms. Sci Am 267:66–73
21. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks. pp 1942–1948
22. Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11:341–359
23. Mehrabian AR, Lucas C (2006) A novel numerical optimization algorithm inspired from weed colonization. Ecol Inform 1:355–366
24. Atashpaz-Gargari E, Lucas C (2007) Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: 2007 IEEE congress on evolutionary computation. pp 4661–4667
25. Yang X-S (2010) Nature-inspired metaheuristic algorithms. Luniver press, Luniver
26. Simon D (2008) Biogeography-based optimization. IEEE Trans Evol Comput 12:702–713
27. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci (N Y) 179:2232–2248
28. Yang X-S, Deb S (2009) Cuckoo search via Lévy flights. In: 2009 World congress on nature \& biologically inspired computing (NaBIC). pp 210–214
29. Yang X-S (2010) A new metaheuristic bat-inspired algorithm. In: Nature inspired cooperative strategies for optimization (NICSO 2010). Springer, pp 65–74
30. Rao RV, Savsani VJ, Vakharia DP (2011) Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des 43:303–315
31. Yang X-S (2012) Flower pollination algorithm for global optimization. In: International conference on unconventional computing and natural computation. pp 240–249
32. Eskandar H, Sadollah A, Bahreininejad A, Hamdi M (2012) Water cycle algorithm—a novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput Struct 110:151–166
33. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
34. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl Based Syst 89:228–249
35. Cheng R, Jin Y (2014) A competitive swarm optimizer for large scale optimization. IEEE Trans Cybern 45:191–204
36. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
37. Huang G (2016) Artificial infectious disease optimization: a SEIQR epidemic dynamic model-based function optimization algorithm. Swarm Evol Comput 27:31–67
38. Yazdani M, Jolai F (2016) Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm. J Comput Des Eng 3:24–36
39. Kaveh A, Bakhshpoori T (2016) Water evaporation optimization: a novel physically inspired optimization algorithm. Comput Struct 167:69–85
40. Labbi Y, Ben AD, Gabbar HA et al (2016) A new rooted tree optimization algorithm for economic dispatch with valve-point effect. Int J Electric Power Energy Syst 79:298–311
41. Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. Adv Eng Softw 105:30–47
42. Baykasoğlu A, Akpinar Ş (2015) Weighted superposition attraction (wsa): a swarm intelligence algorithm for optimization problems–part 2: constrained optimization. Appl Soft Comput 1(37):396–415
43. Mirjalili S, Gandomi AH, Mirjalili SZ et al (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw 114:163–191
44. Zhang Q, Wang R, Yang J et al (2017) Collective decision optimization algorithm: a new heuristic optimization method. Neurocomputing 221:123–137
45. Dhiman G, Kumar V (2018) Emperor penguin optimizer: a bio-inspired algorithm for engineering problems. Knowl Based Syst 159:20–50
46. Kallioras NA, Lagaros ND, Avtzis DN (2018) Pity beetle algorithm—a new metaheuristic inspired by the behavior of bark beetles. Adv Eng Softw 121:147–166

47. Ghasemi M, Ghavidel S, Aghaei J et al (2018) CFA optimizer: a new and powerful algorithm inspired by Franklin's and Coulomb's laws theory for solving the economic load dispatch problems. Int Trans Electric Energy Syst. https://doi.org/10.1002/etep.2536
48. Cheraghalipour A, Hajiaghaei-Keshteli M, Paydar MM (2018) Tree Growth Algorithm (TGA): a novel approach for solving optimization problems. Eng Appl Artif Intell 72:393–414
49. Hashim FA, Houssein EH, Mabrouk MS et al (2019) Henry gas solubility optimization: a novel physics-based algorithm. Futur Gener Comput Syst 101:646–667
50. Dehghani M, Montazeri Z, Malik OP (2019) DGO: dice game optimizer. Gazi Univ J Sci 32:871–882
51. Heidari AA, Mirjalili S, Faris H et al (2019) Harris hawks optimization: algorithm and applications. Futur Gener Comput Syst 97:849–872
52. 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
53. Faramarzi A, Heidarinejad M, Mirjalili S, Gandomi AH (2020) Marine predators algorithm: a nature-inspired metaheuristic. Expert Syst Appl 152:113377
54. Dehghani M, Montazeri Z, Malik OP et al (2020) Shell game optimization: a novel game-based algorithm. Int J Intell Eng Syst 13:246–255
55. Li S, Chen H, Wang M et al (2020) Slime mould algorithm: a new method for stochastic optimization. Futur Gener Comput Syst 111:300–323
56. Ahmadianfar I, Bozorg-Haddad O, Chu X (2020) Gradient-based optimizer: a new metaheuristic optimization algorithm. Inf Sci 540:131–159
57. Ghasemi M, Davoudkhani IF, Akbari E et al (2020) A novel and effective optimization algorithm for global optimization and its engineering applications: turbulent flow of water-based optimization (TFWO). Eng Appl Artif Intell 92:103666
58. Zervoudakis K, Tsafarakis S (2020) A mayfly optimization algorithm. Comput Ind Eng 145:106559. https://doi.org/10.1016/j.cie.2020.106559
59. Yang Y, Chen H, Heidari AA, Gandomi AH (2021) Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. Expert Syst Appl 177:114864
60. Połap D, Woźniak M (2021) Red fox optimization algorithm. Expert Syst Appl 166:114107
61. Ghasemi M, Rahimnejad A, Hemmati R et al (2021) Wild geese algorithm: a novel algorithm for large scale optimization based on the natural life and death of wild geese. Array 11:100074
62. Abualigah L, Yousri D, Abd Elaziz M et al (2021) Aquila optimizer: a novel meta-heuristic optimization algorithm. Comput Ind Eng 157:107250
63. Tu J, Chen H, Wang M, Gandomi AH (2021) The colony predation algorithm. J Bionic Eng 18:674–710
64. Ayyarao TSLV, RamaKrishna NSS, Elavarasan RM et al (2022) War strategy optimization algorithm: a new effective metaheuristic algorithm for global optimization. IEEE Access 10:25073–25105
65. Hashim FA, Hussien AG (2022) Snake optimizer: a novel meta-heuristic optimization algorithm. Knowl Based Syst 242:108320
66. Trojovsky P, Dehghani M (2022) Pelican optimization algorithm: a novel nature-inspired algorithm for engineering applications. Sensors 22:855
67. Kusuma PD, Nugrahaeni RA, Dinimaharawati A (2022) Cone search: a simple metaheuristic optimization algorithm. IAENG Int J Appl Math 52:1–8
68. Akbari MA, Zare M, Azizipanah-abarghooee R et al (2022) The cheetah optimizer: a nature-inspired metaheuristic algorithm for large-scale optimization problems. Sci Rep 12:10953. https://doi.org/10.1038/s41598-022-14338-z
69. Zervoudakis K, Tsafarakis S (2023) A global optimizer inspired from the survival strategies of flying foxes. Eng Comput 39:1–34
70. Zhang Q, Gao H, Zhan Z-H et al (2023) Growth optimizer: a powerful metaheuristic algorithm for solving continuous and discrete global optimization problems. Knowl Based Syst 261:110206
71. Yuan Y, Shen Q, Wang S et al (2023) Coronavirus mask protection algorithm: a new bio-inspired optimization algorithm and its applications. J Bionic Eng 20:1–19
72. Rezvani K, Gaffari A, Dishabi MRE (2023) The bedbug meta-heuristic algorithm to solve optimization problems. J Bionic Eng 20:1–21

73. Ghasemi M, Zare M, Zahedi A et al (2023) Geyser inspired algorithm: a new geological-inspired meta-heuristic for real-parameter and constrained engineering optimization. J Bionic Eng 21:1–35
74. Salehnia T, Seyfollahi A, Raziani S et al (2023) An optimal task scheduling method in IoT-fog-cloud network using multi-objective moth-flame algorithm. Multimed Tools 83:1–22
75. Di Baldassarre G, Viglione A, Carr G et al (2013) Socio-hydrology: conceptualising human-flood interactions. Hydrol Earth Syst Sci 17:3295–3303
76. Fan FM, Collischonn W, Quiroz KJ et al (2016) Flood forecasting on the tocantins river using ensemble rainfall forecasts and real-time satellite rainfall estimates. J Flood Risk Manag 9:278–288
77. Adib A, Salarijazi M, Shooshtari MM, Akhondali AM (2011) Comparison between characteristics of geomorphoclimatic instantaneous unit hydrograph be produced by GcIUH based clark model and clark IUH model. J Mar Sci Technol 19:12
78. Sikder S, Chen X, Hossain F et al (2016) Are general circulation models ready for operational streamflow forecasting for water management in the Ganges and Brahmaputra river basins? J Hydrometeorol 17:195–210
79. Aksoy H, Kirca VSO, Burgan HI, Kellecioglu D (2016) Hydrological and hydraulic models for determination of flood-prone and flood inundation areas. Proc Int Assoc Hydrol Sci 373:137–141
80. Suganthan PN, Hansen N, Liang JJ et al (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. KanGAL Rep 2005005:2005
81. Ghasemi M, Akbari M-A, Jun C et al (2022) Circulatory system based optimization (CSBO): an expert multilevel biologically inspired meta-heuristic algorithm. Eng Appl Comput Fluid Mech 16:1483–1525
82. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Global Optim 39:459–471
83. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: MHS'95. Proceedings of the sixth international symposium on micro machine and human science. pp 39–43
84. Shi Y, Eberhart R (1998) A modified particle swarm optimizer. In: 1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360). pp 69–73
85. Yang X-S (2010) Firefly algorithm, stochastic test functions and design optimisation. Int J Bio-inspir Comput 2:78–84
86. Rao R (2020) Rao algorithms: three metaphor-less simple algorithms for solving optimization problems. Int J Ind Eng Comput 11:107–130
87. Çimen ME, Yalçin Y (2022) A novel hybrid firefly–whale optimization algorithm and its application to optimization of MPC parameters. Soft comput 26:1845–1872
88. Derrac J, Garcia S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1:3–18
89. Ghasemi M, Zare M, Zahedi A et al (2024) Optimization based on performance of lungs in body: lungs performance-based optimization (LPO). Comput Methods Appl Mech Eng 419:116582
90. Wang Y, Cai Z, Zhang Q (2011) Differential evolution with composite trial vector generation strategies and control parameters. IEEE Trans Evol Comput 15:55–66
91. Garcia-Martinez C, Lozano M, Herrera F et al (2008) Global and local real-coded genetic algorithms based on parent-centric crossover operators. Eur J Oper Res 185:1088–1113
92. Liang JJ, Qin AK, Suganthan PN, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Trans Evol Comput 10:281–295
93. Qiang J, Mitchell C, Qiang A (2016) Tuning of an adaptive unified differential evolution algorithm for global optimization. In: 2016 IEEE congress on evolutionary computation (CEC). pp 4061–4068
94. Brest J, Greiner S, Boskovic B et al (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans Evol Comput 10:646–657
95. Du Y, Fan Y, Liu X et al (2019) Multiscale cooperative differential evolution algorithm. Comput Intell Neurosci 2019:1
96. Mallipeddi R, Suganthan PN, Pan Q-K, Tasgetiren MF (2011) Differential evolution algorithm with ensemble of parameters and mutation strategies. Appl Soft Comput 11:1679–1696
97. Lynn N, Suganthan PN (2015) Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation. Swarm Evol Comput 24:11–24

98. Mendes R, Kennedy J, Neves J (2004) The fully informed particle swarm: simpler, maybe better. IEEE Trans Evol Comput 8:204–210
99. Zhan Z-H, Zhang J, Li Y, Shi Y-H (2010) Orthogonal learning particle swarm optimization. IEEE Trans Evol Comput 15:832–847
100. Engelbrecht AP (2010) Heterogeneous particle swarm optimization. In: International conference on swarm intelligence. pp 191–202
101. Gutjahr WJ (2009) Convergence analysis of metaheuristics. In: Matheuristics: hybridizing metaheuristics and mathematical programming. Springer, Boston, MA, US, pp 159–187
102. Liang J, Qu B, Suganthan P (2014) Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. Computational intelligence laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, p 635
103. Abualigah L, Diabat A, Mirjalili S et al (2021) The arithmetic optimization algorithm. Comput Methods Appl Mech Eng 376:113609
104. Houssein EH, Saad MR, Hashim FA et al (2020) Lévy flight distribution: a new metaheuristic algorithm for solving engineering optimization problems. Eng Appl Artif Intell 94:103731
105. Long W, Cai S, Jiao J, Tang M (2020) An efficient and robust grey wolf optimizer algorithm for large-scale numerical optimization. Soft comput 24:997–1026
106. Hansen N, Müller SD, Koumoutsakos P (2003) Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evol Comput 11:1–18
107. Iacca G, dos Santos Junior VC, de Melo VV (2021) An improved Jaya optimization algorithm with Lévy flight. Expert Syst Appl 165:113902
108. Gupta S, Deep K, Engelbrecht AP (2020) A memory guided sine cosine algorithm for global optimization. Eng Appl Artif Intell 93:103718
109. Ks SR, Murugan S (2017) Memory based hybrid dragonfly algorithm for numerical optimization problems. Expert Syst Appl 83:63–78
110. Mlakar U, Fister I Jr, Fister I (2016) Hybrid self-adaptive cuckoo search for global optimization. Swarm Evol Comput 29:47–72
111. Dai C, Hu Z, Li Z et al (2020) An improved grey prediction evolution algorithm based on topological opposition-based learning. IEEE Access 8:30745–30762
112. Li MD, Zhao H, Weng XW, Han T (2016) A novel nature-inspired algorithm for optimization: virus colony search. Adv Eng Softw 92:65–88
113. Cheng M-Y, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. Comput Struct 139:98–112
114. Gupta S, Deep K (2019) A hybrid self-adaptive sine cosine algorithm with opposition based learning. Expert Syst Appl 119:210–230
115. Chen X, Xu B, Yu K, Du W (2018) Teaching-learning-based optimization with learning enthusiasm mechanism and its application in chemical engineering. J Appl Math 2018:1
116. Satapathy SC, Naik A (2014) Modified teaching–learning-based optimization algorithm for global numerical optimization—a comparative study. Swarm Evol Comput 16:28–37
117. Zhang L, Oh SK, Pedrycz W, Yang B, Wang L (2021) A promotive particle swarm optimizer with double hierarchical structures. IEEE Trans Cybernet 52(12):13308–13322
118. Band SS, Ardabili S, Seyed Danesh A et al (2022) Colonial competitive evolutionary Rao algorithm for optimal engineering design. Alex Eng J 61:11537–11563. https://doi.org/10.1016/J.AEJ.2022.05.018
119. Agarwal P, Mehta S (2018) Empirical analysis of five nature-inspired algorithms on real parameter optimization problems. Artif Intell Rev 50:383–439
120. Deng L-B, Li C-L, Sun G-J (2020) An adaptive dimension level adjustment framework for differential evolution. Knowl Based Syst 206:106388
121. Hu G, Yang R, Qin X, Wei G (2023) MCSA: multi-strategy boosted chameleon-inspired optimization algorithm for engineering applications. Comput Methods Appl Mech Eng 403:115676
122. Ghasemi M, Zare M, Mohammadi SK, Mirjalili S (2024) Applications of whale migration algorithm in optimal power flow problems of power systems. In: handbook of whale optimization algorithm. Elsevier, pp 347–364
123. Abdel-Basset M, Mohamed R, Zidan M et al (2023) Mantis search algorithm: a novel bio-inspired algorithm for global optimization and engineering design problems. Comput Methods Appl Mech Eng 415:116200

124. He Q, Wang L (2007) A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. Appl Math Comput 186:1407–1422
125. Abdel-Basset M, Mohamed R, Abouhawwash M (2024) Crested porcupine optimizer: a new nature-inspired metaheuristic. Knowl-Based Syst 25(284):111257
126. Jahangiri M, Hadianfard MA, Najafgholipour MA et al (2020) Interactive autodidactic school: a new metaheuristic optimization algorithm for solving mathematical and structural design optimization problems. Comput Struct 235:106268
127. Bai J, Li Y, Zheng M et al (2023) A sinh cosh optimizer. Knowl Based Syst 282:111081
128. Abdel-Basset M, Mohamed R, Sallam KM, Chakrabortty RK (2022) Light spectrum optimizer: a novel physics-inspired metaheuristic optimization algorithm. Mathematics 10:3466
129. Abdel-Basset M, Mohamed R, Azeem SAA et al (2023) Kepler optimization algorithm: a new metaheuristic algorithm inspired by Kepler's laws of planetary motion. Knowl Based Syst 268:110454
130. Abdel-Basset M, Mohamed R, Jameel M, Abouhawwash M (2023) Spider wasp optimizer: a novel meta-heuristic optimization algorithm. Artif Intell Rev 56:1–64
131. Naghdiani M, Jahanshahi M, Matin RK (2023) A garter snake optimization algorithm for constrained optimization. https://doi.org/10.21203/rs.3.rs-2899298/v1
132. Zhang M, Wen G (2024) Duck swarm algorithm: theory, numerical optimization, and applications. Cluster Comput 1–29
133. He C, Huang W, Ouyang H et al (2023) Virus control optimizer based on the prevention and control mechanism of COVID-19 for engineering optimization problems. Appl Soft Comput 147:110770
134. Deng L, Liu S (2023) Snow ablation optimizer: a novel metaheuristic technique for numerical optimization and engineering design. Expert Syst Appl 225:120069
135. Kaveh M, Mesgari MS, Saeidian B (2023) Orchard algorithm (OA): a new meta-heuristic algorithm for solving discrete and continuous optimization problems. Math Comput Simul 208:95–135
136. He S-X, Cui Y-T (2023) Multiscale medalist learning algorithm and its application in engineering. Acta Mech 235:1–27
137. Wu J, Wang Y-G, Burrage K et al (2020) An improved firefly algorithm for global continuous optimization problems. Expert Syst Appl 149:113340
138. Ngo TT, Sadollah A, Kim JH (2016) A cooperative particle swarm optimizer with stochastic movements for computationally expensive numerical optimization problems. J Comput Sci 13:68–82
139. Hedar A-R, Fukushima M (2006) Derivative-free filter simulated annealing method for constrained continuous global optimization. J Glob Optim 35:521–549
140. He Q, Wang L (2007) An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng Appl Artif Intell 20:89–99
141. Kaveh A, Dadras A (2017) A novel meta-heuristic optimization algorithm: thermal exchange optimization. Adv Eng Softw 110:69–84
142. Huang F, Wang L, He Q (2007) An effective co-evolutionary differential evolution for constrained optimization. Appl Math Comput 186:340–356
143. Parsopoulos KE, Vrahatis MN (2005) Unified particle swarm optimization for solving constrained engineering optimization problems. In: International conference on natural computation, vol 3612. Springer, Berlin, Heidelberg, pp 582–591
144. Yapici H, Cetinkaya N (2019) A new meta-heuristic optimizer: pathfinder algorithm. Appl Soft Comput 78:545–568
145. Faramarzi A, Heidarinejad M, Stephens B, Mirjalili S (2020) Equilibrium optimizer: a novel optimization algorithm. Knowl Based Syst 191:105190
146. He S, Prempain E, Wu QH (2004) An improved particle swarm optimizer for mechanical design optimization problems. Eng Optim 36:585–605
147. Mousavirad SJ, Ebrahimpour-Komleh H (2017) Human mental search: a new population-based metaheuristic optimization algorithm. Appl Intell 47:850–887
148. Wang Y, Cai Z, Zhou Y, Fan Z (2009) Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. Struct Multidiscip Optim 37:395–413
149. Ray T, Liew K-M (2003) Society and civilization: an optimization algorithm based on the simulation of social behavior. IEEE Trans Evol Comput 7:386–396
150. Zhao W, Wang L, Zhang Z (2019) Supply-demand-based optimization: a novel economics-inspired algorithm for global optimization. IEEE Access 7:73182–73206

151. Qais MH, Hasanien HM, Alghuwainem S, Loo KH (2023) Propagation search algorithm: a physics-based optimizer for engineering applications. Mathematics 11:4224
152. Sharafaddini AM, Mohammad Hasani Zade B, Mansouri N Attrition mill optimization: a novel approach for solving engineering optimization problems. Najme, Attrition mill optimization: a novel approach for solving engineering optimization problems
153. Chen Z, Francis A, Li S et al (2022) Egret swarm optimization algorithm: an evolutionary computation approach for model free optimization. Biomimetics 7:144
154. Kumar A, Wu G, Ali MZ et al (2020) A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. Swarm Evol Comput 56:100693
155. Sugantthan PN (2020) Guidelines for real-world single-objective constrained optimisation competition. PN-Suganthan/2020-RW-constrained-optimisation
156. Meng X-B, Li H-X, Gao X-Z (2019) An adaptive reinforcement learning-based bat algorithm for structural design problems. Int J Bio-Inspir Comput 14:114–124
157. Gupta S, Deep K (2020) A memory-based grey wolf optimizer for global optimization tasks. Appl Soft Comput 93:106367
158. Zare M, Akbari M-A, Azizipanah-Abarghooee R et al (2023) A modified particle swarm optimization algorithm with enhanced search quality and population using hummingbird flight patterns. Decis Anal J 7:100251. https://doi.org/10.1016/j.dajour.2023.100251
159. Liu Z, Nishi T (2020) Multipopulation ensemble particle swarm optimizer for engineering design problems. Math Prob Eng 2020(1):1450985
160. Jia H, Wen Q, Wu D et al (2023) Modified beluga whale optimization with multi-strategies for solving engineering problems. J Comput Des Eng 10:2065–2093
161. Zhou Y, Zhang S, Luo Q, Abdel-Baset M (2019) CCEO: cultural cognitive evolution optimization algorithm. Soft comput 23:12561–12583
162. Yu H, Wang Y, Jia H, Abualigah L (2023) Modified prairie dog optimization algorithm for global optimization and constrained engineering problems. Math Biosci Eng 20:19086–19132
163. Zhao W, Wang L, Zhang Z et al (2023) Electric eel foraging optimization: a new bio-inspired optimizer for engineering applications. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2023.122200
164. Song S, Wang P, Heidari AA et al (2021) Dimension decided Harris hawks optimization with Gaussian mutation: balance analysis and diversity patterns. Knowl Based Syst 215:106425
165. Shahrouzi M, Kaveh A (2022) An efficient derivative-free optimization algorithm inspired by avian life-saving manoeuvres. J Comput Sci 57:101483
166. Zhang J, Xiao M, Gao L, Pan Q (2018) Queuing search algorithm: a novel metaheuristic algorithm for solving engineering optimization problems. Appl Math Model 63:464–490
167. El-kenawy E-SM, Khodadadi N, Mirjalili S et al (2023) Greylag goose optimization: nature-inspired optimization algorithm. Expert Syst Appl 238:122147
168. Han M, Du Z, Yuen KF et al (2024) Walrus optimizer: a novel nature-inspired metaheuristic algorithm. Expert Syst Appl 239:122413
169. Kizilay D, Tasgetiren MF, Oztop H, et al (2020) A differential evolution algorithm with q-learning for solving engineering design problems. In: 2020 IEEE congress on evolutionary computation (CEC). pp 1–8
170. Wen X, Wu G, Fan M, et al (2020) Voting-mechanism based ensemble constraint handling technique for real-world single-objective constrained optimization. In: 2020 IEEE congress on evolutionary computation (CEC). pp 1–8
171. Sallam KM, Elsayed SM, Chakrabortty RK, Ryan MJ (2020) Improved multi-operator differential evolution algorithm for solving unconstrained problems. In: 2020 IEEE congress on evolutionary computation (CEC). pp 1–8
172. Houssein EH, Mahdy MA, Blondin MJ et al (2021) Hybrid slime mould algorithm with adaptive guided differential evolution algorithm for combinatorial and global optimization problems. Expert Syst Appl 174:114689
173. Naruei I, Keynia F (2021) A new optimization method based on COOT bird natural life model. Expert Syst Appl 183:115352
174. dos Santos CL (2010) Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. Expert Syst Appl 37:1676–1683
175. Ghafil HN, Jármai K (2020) Dynamic differential annealed optimization: new metaheuristic optimization algorithm for engineering applications. Appl Soft Comput 93:106392

176. Mezura-Montes E, Coello CAC (2005) Useful infeasible solutions in engineering optimization with evolutionary algorithms. In: Mexican international conference on artificial intelligence. Springer, pp 652–662
177. Ahmadianfar I, Heidari AA, Noshadian S, Chen H, Gandomi AH (2022) INFO: an efficient optimization algorithm based on weighted mean of vectors. Exp Syst Appl 1(195):116516
178. Sowmya R, Premkumar M, Jangir P (2024) Newton–Raphson-based optimizer: a new population-based metaheuristic algorithm for continuous optimization problems. Eng Appl Artif Intell 128:107532
179. Li X, Qi Y, Xing Q, Hu Y (2023) IMSCSO: an intensified sand cat swarm optimization with multi-strategy for solving global and engineering optimization problems. IEEE Access 11:122315–122344
180. Bernardino HS, Barbosa HJC, Lemonge ACC (2007) A hybrid genetic algorithm for constrained optimization problems in mechanical engineering. In: 2007 IEEE congress on evolutionary computation. IEEE, pp 646–653
181. Dehghani M, Hubálovsky Š, Trojovsky P (2022) Tasmanian devil optimization: a new bio-inspired optimization algorithm for solving optimization algorithm. IEEE Access 10:19599–19620
182. Braik MS (2021) Chameleon swarm algorithm: a bio-inspired optimizer for solving engineering design problems. Expert Syst Appl 174:114685
183. Chun S, Kim Y-T, Kim T-H (2013) A diversity-enhanced constrained particle swarm optimizer for mixed integer-discrete-continuous engineering design problems. Adv Mech Eng 5:130750
184. Abdel-Basset M, El-Shahat D, Jameel M, Abouhawwash M (2023) Young's double-slit experiment optimizer: a novel metaheuristic optimization algorithm for global and constraint optimization problems. Comput Method Appl Mech Eng 403:115652
185. Hellwig M, Beyer H-G (2020) A modified matrix adaptation evolution strategy with restarts for constrained real-world problems. In: 2020 IEEE congress on evolutionary computation (CEC). pp 1–8
186. Gurrola-Ramos J, Hernàndez-Aguirre A, Dalmau-Cedeño O (2020) COLSHADE for real-world single-objective constrained optimization problems. In: 2020 IEEE congress on evolutionary computation (CEC). pp 1–8
187. Brammya G, Praveena S, Ninu Preetha NS et al (2019) Deer hunting optimization algorithm: a new nature-inspired meta-heuristic paradigm. Comput J. https://doi.org/10.1093/comjnl/bxy133
188. Yildiz AR, Abderazek H, Mirjalili S (2020) A comparative study of recent non-traditional methods for mechanical design optimization. Arch Comput Method Eng 27:1031–1048
189. Jia H, Sun K, Zhang W, Leng X (2022) An enhanced chimp optimization algorithm for continuous optimization domains. Complex Intell Syst 8:65–82
190. Emami H (2022) Anti-coronavirus optimization algorithm. Soft Comput 26:4991–5023
191. Dhiman G, Kumar V (2017) Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications. Adv Eng Softw 114:48–70
192. Emami H (2022) Stock exchange trading optimization algorithm: a human-inspired method for global optimization. J Supercomput 78:2125–2174
193. Zhao W, Wang L, Mirjalili S (2022) Artificial hummingbird algorithm: a new bio-inspired optimizer with its engineering applications. Comput Methods Appl Mech Eng 388:114194
194. Alsayyed O, Hamadneh T, Al-Tarawneh H et al (2023) Giant Armadillo optimization: a new bio-inspired metaheuristic algorithm for solving optimization problems. Biomimetics 8:619
195. Dalirinia E, Jalali M, Yaghoobi M, Tabatabaee H (2023) Lotus effect optimization algorithm (LEA): a lotus nature-inspired algorithm for engineering design optimization. J Supercomput 80:761–799
196. Abualigah L, Abd Elaziz M, Sumari P et al (2022) Reptile search algorithm (RSA): a nature-inspired meta-heuristic optimizer. Expert Syst Appl 191:116158
197. Kamboj VK, Nandi A, Bhadoria A, Sehgal S (2020) An intensify Harris Hawks optimizer for numerical and engineering optimization problems. Appl Soft Comput 89:106018
198. Bayzidi H, Talatahari S, Saraee M, Lamarche CP (2021) Social network search for solving engineering optimization problems. Comput Intell Neurosci 2021(1):8548639
199. Goodarzimehr V, Talatahari S, Shojaee S, Hamzehei-Javaran S (2023) Special relativity search for applied mechanics and engineering. Comput Methods Appl Mech Eng 403:115734
200. Khalil AE, Boghdady TA, Alham MH, Ibrahim DK (2023) A novel cascade-loop controller for load frequency control of isolated microgrid via dandelion optimizer. Ain Shams Eng J 15:102526

201. Prayogo D, Cheng M-Y, Wu Y-W et al (2018) Differential big bang-big crunch algorithm for construction-engineering design optimization. Autom Constr 85:290–304
202. Kim P, Lee J (2009) An integrated method of particle swarm optimization and differential evolution. J Mech Sci Technol 23:426–434
203. Yadav A, Kumar N et al (2020) Artificial electric field algorithm for engineering optimization problems. Expert Syst Appl 149:113308
204. Abdel-Baset M, Hezam I (2016) A hybrid flower pollination algorithm for engineering optimization problems. Int J Comput Appl 140:10–23
205. Gu L, Yang RJ, Tho C-H et al (2001) Optimisation and robustness for crashworthiness of side impact. Int J Veh Des 26:348–360
206. Sadeeq HT, Abdulazeez AM (2023) Car side impact design optimization problem using giant trevally optimizer. In: Structures. pp 39–45
207. Gao Y (2023) PID-based search algorithm: a novel metaheuristic algorithm based on PID algorithm. Expert Syst Appl 232:120886
208. Talatahari S, Azizi M (2021) Tribe-charged system search for global optimization. Appl Math Model 93:115–133
209. Talatahari S, Azizi M, Toloo M (2021) Fuzzy adaptive charged system search for global optimization. Appl Soft Comput 109:107518

Authors and Affiliations

Mojtaba Ghasemi[1] · Keyvan Golalipour[2] · Mohsen Zare[3] · Seyedali Mirjalili[4,5] · Pavel Trojovský[6] · Laith Abualigah[7,8,9,10] · Rasul Hemmati[11]

✉ Mohsen Zare
mzare@jahromu.ac.ir

✉ Laith Abualigah
aligah.2020@gmail.com

Mojtaba Ghasemi
mojtaba.ghasemi3@gmail.com

Keyvan Golalipour
Golalipour@yahoo.com

Seyedali Mirjalili
ali.mirjalili@torrens.edu.au

Pavel Trojovský
pavel.trojovsky@uhk.cz

Rasul Hemmati
rasul.hemmati@marquette.edu

1 Department of Electronics and Electrical Engineering, Shiraz University of Technology, Shiraz, Iran

[2] Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran

[3] Department of Electrical Engineering, Faculty of Engineering, Jahrom University, Jahrom, Iran

[4] Centre for Artificial Intelligence Research and Optimisation, Torrens University Australia, Brisbane, QLD 4006, Australia

[5] Faculty of Electrical Engineering and Computer Science, VŠB-TU Ostrava, Ostrava, Czech Republic

[6] Department of Mathematics, Faculty of Science, University of Hradec Králové, Hradec Králové, Czech Republic

[7] Computer Science Department, Al Al-Bayt University, Mafraq 25113, Jordan

[8] MEU Research Unit, Middle East University, Amman 11831, Jordan

[9] Applied Science Research Center, Applied Science Private University, Amman 11931, Jordan

[10] Jadara Research Center, Jadara University, Irbid 21110, Jordan

[11] Department of Electrical and Computer Engineering, Marquette University, Milwaukee, USA