Journal of Bionic Engineering (2024) 21:374–408
https://doi.org/10.1007/s42235-023-00437-8

RESEARCH ARTICLE

Geyser Inspired Algorithm: A New Geological-inspired Meta-heuristic for Real-parameter and Constrained Engineering Optimization

Mojtaba Ghasemi[1] · Mohsen Zare[2] · Amir Zahedi[3] · Mohammad-Amin Akbari[4] · Seyedali Mirjalili[5,6] · Laith Abualigah[7,8,9,10]

Received: 23 March 2023 / Revised: 14 August 2023 / Accepted: 24 August 2023 / Published online: 26 September 2023
© Jilin University 2023

Abstract

Over the past years, many efforts have been accomplished to achieve fast and accurate meta-heuristic algorithms to optimize a variety of real-world problems. This study presents a new optimization method based on an unusual geological phenomenon in nature, named Geyser inspired Algorithm (GEA). The mathematical modeling of this geological phenomenon is carried out to have a better understanding of the optimization process. The efficiency and accuracy of GEA are verified using statistical examination and convergence rate comparison on numerous CEC 2005, CEC 2014, CEC 2017, and real-parameter benchmark functions. Moreover, GEA has been applied to several real-parameter engineering optimization problems to evaluate its effectiveness. In addition, to demonstrate the applicability and robustness of GEA, a comprehensive investigation is performed for a fair comparison with other standard optimization methods. The results demonstrate that GEA is noticeably prosperous in reaching the optimal solutions with a high convergence rate in comparison with other well-known nature-inspired algorithms, including ABC, BBO, PSO, and RCGA. Note that the source code of the GEA is publicly available at https://www.optim-app.com/projects/gea.

Keywords Nature-inspired algorithms · Real-world and engineering optimization · Mathematical modeling · Geyser algorithm (GEA)

1 Introduction

Real-world problems, such as what happens in nature, are continually evolving. We struggle with new complex problems every day; as a result, the use of evolutionary algorithms and swarm intelligence inspired by nature is in the spotlight to solve various optimization problems [1, 2]. Optimization is a key design process that can eventually minimize or maximize an objective function [1, 3, 4].

1.1 Background

In recent years, several optimization methods inspired by various natural search phenomena have been introduced,

✉ Mohsen Zare
mzare@jahromu.ac.ir

✉ Laith Abualigah
aligah.2020@gmail.com; aligah@ammanu.edu.jo

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

2 Department of Electrical Engineering, Faculty of Engineering, Jahrom University, Jahrom, Fras, Iran

3 Department of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran

4 Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus

5 Centre for Artificial Intelligence Research and Optimisation, Torrens University Australia, Brisbane, QLD 4006, Australia

6 University Research and Innovation Center, Obuda University, 1034 Budapest, Hungary

7 Department of Electrical and Computer Engineering, Lebanese American University, Byblos 13-5053, Lebanon

8 Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman 19328, Jordan

9 MEU Research Unit, Middle East University, Amman 11831, Jordan

10 Applied Science Research Center, Applied Science Private University, Amman 11931, Jordan

which some of the most significant meta-heuristic algorithms are briefly mentioned as follows: A basic Genetic Algorithm (GA) [5], Simulated Annealing (SA) [6] is commonly used to estimate the global optimization. This algorithm is based on the metallurgical annealing process in 1983. During the annealing process, the physical and occasionally the chemical properties of the material change, during which the metal is first heated and then held at a particular temperature and then progressively cooled, Particle Swarm Optimization (PSO) [7, 8], Ant Colony Optimization (ACO) [8] that has been proposed in 1999 is based on the natural activity and lives of ant colonies, as well as the worker ants who labor in them. The foraging process in these colonies is extremely efficient, and the workers automatically discover a logical and ideal path between their dwelling quarters and various food sources, which serves as the foundation for optimization in this method.

The Artificial Bee Colony (ABC) [9] is a powerful algorithm based on honeybee behaviors which has been suggested in 2005. The Gravitational Search Algorithm (GSA) [10], which was based on Newton's principles of motion and gravitational attraction in 2009, drives individuals that are random masses in the issue space to search for the problem solution. Biogeography-Based Optimization (BBO) [11] is classified as a meta-heuristic because it has numerous variants and does not make presumptions about the situation, allowing it to be used for a wide range of challenges which was proposed in 2009. Based on the utilization of chemical activity categories and occurrences, Artificial Chemical Reaction Optimization Algorithm (ACROA) [12] is a robust algorithm with minimal control parameters which was created in 2011. The Heat Transfer Search (HTS) [13] is premised on heat exchange and the law of thermodynamics by authors in 2015. The

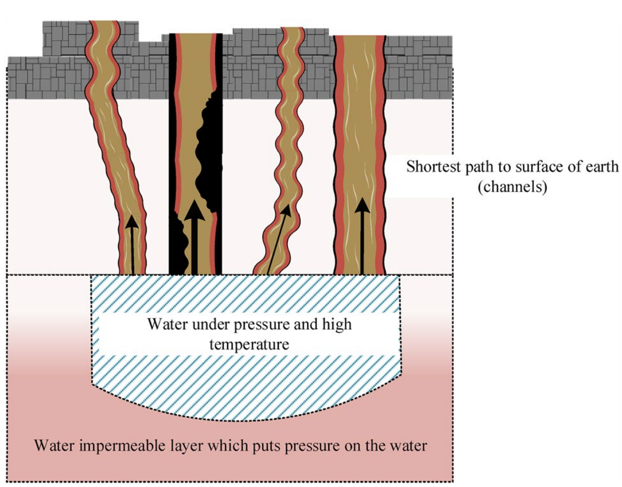

Fig. 1 Groundwater Pond and possible geyser to spout

Fig. 2 Geyser erupts, **a** graphically [25], **b** natural occurrences (pixabay.com)

Table 1 The correspondence of the geyser phenomena with mathematical modeling

(1) Phase in nature	(2) Equivalent phase in GEA	(3) Executive effect
Mass of water under pressure	Algorithm Population	The responsible for finding the optimum solution
Movement of water in channels to find a way to erupt	Mutation in population towards the best current neighbor solution	The movement toward an optimum position
The nearest neighbor channel to the mass of water	Finding the most similar member of the population to *i*th member	Leader position, which the *i*th particle tends to reach its position
The pressure and temperature forced on water particles	An equation in terms of algorithm iteration and fitness function value for each member	Random throwing of particles for escaping from local optimums
Channel	Members with better fitness function values	Member's leader for a better solution
More efforts to create Geyser	Number of iterations	More chances to find optimal solution

system's molecules act as search agents in the algorithm, interacting with each other and the environment to achieve a temperature range. Several heat transfer mechanisms are used to link molecules. Water Evaporation Optimization (WEO) [14] can be used to explore the evaporation of a minor number of water molecules as the algorithm's population on a hard surface with varying wettability. Water molecules comprise the algorithm's civilization in this meta-heuristic proposed in 2016.

Electromagnetic Field Optimization (EFO) [15] was proposed in 2016 as a new and effective meta-heuristic to engineering optimization. The usage of a nature-inspired ratio, termed the golden ratio is one of the major advantages of EFO. Yin-Yang-pair Optimization (YYO) [16] which is a novel lightweight optimizer, is centered on preserving a balance between exploitation and exploration in the problem space, and it uses a special process or physical occurrence. The user determines the three adjustable parameters of the algorithm based on the design model. This algorithm was suggested in 2016. Chemotherapy aims to eradicate cancer illness and malignant cells and replace them with suitable solutions, i.e., healthy cells, which has been the motivation in order to design Chemotherapy Science based Optimization (CSA) [17] in 2017. Rain-Fall Optimization (RFO) algorithm [18] was proposed in 2017 as a new meta-heuristic that uses the behavior of raindrops as a control parameter. Raindrops make up the algorithm's population, and the algorithm's functioning is based on the fact that when water particles fall on a rock, they tend to enter the pool by the deepest and shortest passage. Particles are continually aware of their surroundings and try to imitate their movements. In Thermal Exchange Optimization (TEO) algorithm [19], which was proposed via Newton's law of cooling in 2017, one group of people is treated as a cold item, while the rest perform the role of environmental factors or other populations, and the heat is moved between the two groups of people, which is the basis of the method's optimization. The Supernova Optimizer (SO) [20] is an artificial intelligence algorithm that was created in 2018. It is based on motivation drawn from the physical occurrence of a supernova, and represents a new form of optimization algorithm. The primary characteristic of this method is to increase exploration, exploitation, and local minima avoidance effectively compared to other algorithms.

The Coulomb's and Franklin's laws Algorithm (CFA) [21, 22] was proposed in 2018 separates the electrical particles into various categories. The electrical particles, as the algorithm's population, are then divided into weak and strong groups, corresponding to positive and negative charges, respectively. Finally, the absorption and repulsion actions and population conduction between members and groups ducted in each iteration of the CFA, which is the foundation of optimization in the CFA. The population members, which are the equivalent of atoms in the atomic motion model, absorb each other through Lennard–Jones forces, and this is the basis of optimization in Atom Search Optimization (ASO) [23] has been suggested in 2018. The activity controlled by Henry's law has been imitated by Henry Gas Solubility Optimization (HGSO) [24] in 2019. This is an integral gas law that defines how much of a particular gas melt into a

```
Generate initial population in size Npop X_i (i=1, 2, 3, …, Npop);
While (Stop Criterion)
    for i=1:Npop
        Calculate the channel probabilities using Eq. (1)
        Determine the target channel corresponding to population, i,
        using roulette wheel mechanism
        Determine the neighbor of population based on the shortest
        distance to X_i (Eqs. (3)-(4)), i.e. X_n,i
        Update the position of X_i using Eq. (4) i.e. X_i^new,1 , select the
        better solution
        Calculate the pressure value corresponding to X_i, using
        Eq. (6), i.e. P_i
        Update the channel probability using Eq. (7)
        Update the position of X_i, using Eq. (8), i.e. X_i^new,2 , select the
        better solution
    endfor
endwhile
```

Fig. 3 Pseudo code of GEA

certain shape and volume of liquid at a specific temperature. To balance exploitation and discovery in the search region while avoiding local optima, the HGSO algorithm simulates the huddling behavior of gas. The Cheetah optimizer (CO) [25] is designed with a combination of sit-and-wait, search, and attack strategies, mirroring the behavior of its real-life counterparts (2022). With its impressive ability to maintain population diversity, the CO algorithm is highly effective in tackling complex optimization problems, particularly those involving high dimensions. Its use of the initial population also offers a viable strategy that can be adapted to other evolutionary algorithms.

1.2 Motivation and Contribution

Evolutionary algorithms have garnered widespread adoption in optimizing various fields, ranging from engineering and computer science to biology and finance [26, 27]. Despite their success, these algorithms continue encountering challenges such as premature convergence, slow convergence speed, and difficulty balancing local and global searches [28]. Researchers in this field need to understand how to model natural processes as optimization methods. In most optimization algorithms, moving towards the best solutions accelerates the convergence speed and yet heightens the likelihood of encountering local optima. Conversely, not utilizing the best solution to enhance population in one step of evolution can also decelerate convergence speed. Some evolutionary algorithms can aptly identify optimal answers with zero values, but their ability to determine the optimal answer is significantly reduced when the optimal variables deviate from zero [29].

This paper introduces a highly efficient optimization algorithm that draws inspiration from the natural phenomenon of geyser formation and its subsequent transfer to the Earth's surface. This algorithm has been meticulously crafted to tackle the aforementioned challenges in a precise and effective manner.

Adapting various elements of this natural process through an evolutionary algorithm allows for easier expansion of this method to other natural phenomena. Logical clustering improves feasible solutions in a manner that creates a balance between the exploitation and exploration processes. The balance of this algorithm in local and global searches ensures the identification of optimal solutions due to its searchability feature. This reduces the risk of getting stuck in local optima while enabling improvement through modification. Applying these concepts results in a potent optimization algorithm inspired by an unusual geological phenomenon called Geyser Algorithm (GEA).

To show the accuracy and effectiveness of GEA, we have used the IEEE CEC 2005 tests. GEA is compared with four modern standard algorithms, including ABC, BBO, PSO, and RCGA, to verify the simulation results. Furthermore, a comprehensive analysis has been conducted on additional benchmark functions utilizing CEC 2014 and CEC 2017, and subsequently compared with various well-known evolutionary algorithms.

A geyser is an opening on the surface of the Earth that evacuates a pillar of hot water and steam on a regular basis. Even a little geyser is a spectacular sight; nevertheless, some geysers have massive eruptions that send hundreds of tons of boiling water several of feet into the air. Geysers are exceedingly uncommon. They only happen when a set of exceptional circumstances happens at the same time. There

Table 2 The selected tests with $f_{\min} = 0$ [37] and [38]

Real-parameter unimodal test functions	
Shifted Sphere	$[-100, 100]^D$
Shifted Schwefel's Problem 1.2	$[-100, 100]^D$
Shifted Rotated High Conditioned Elliptic	$[-100, 100]^D$
Shifted Schwefel's Problem 1.2 with Noise in Fitness	$[-100, 100]^D$
Schwefel's Problem 2.6 with Global Optimum on Bounds	$[-100, 100]^D$
Real-parameter multimodal test functions	
Shifted Rosenbrock's	$[-100, 100]^D$
Shifted Rotated Griewank's without Bounds	$[-600, 600]^D$
Shifted Rotated Ackley's with Global Optimum on Bounds	$[-32, 32]^D$
Shifted Rastrigin's	$[-5, 5]^D$
Shifted Rotated Rastrigin's	$[-5, 5]^D$
Shifted Rotated Weierstrass	$[-0.5, 0.5]^D$
Schwefel's Problem 2.1	$[-\pi,\pi]^D$
Real-parameter expanded multimodal test functions	
Shifted Expanded Griewank's plus Rosenbrock's Function	$[-5, 5]^D$
Shifted Rotated Expanded Scaffer's F6	$[-100, 100]^D$

Table 3 The results of the different values Nc

Test functions (type/number)		Nc = 5 Mean ± Std Rank	Nc = 10 Mean ± Std Rank	Nc = 20 Mean ± Std Rank	Nc = 40 Mean ± Std Rank	Nc = 60 Mean ± Std Rank
Unimodal functions	f_1	3.10e−28 ± 1.85e−28 3	2.0e−28 ± 1.96e−28 2	**3.15e−29 ± 2.67e−29** **1**	2.23e−18 ± 2.23e−18 4	1.05e−14 ± 9.23e−15 5
	f_2	0.0613 ± 0.0309 3	0.0529 ± 0.0241 2	0.0613 ± 0.0463 3	**0.041 ± 0.0204** **1**	0.069 ± 0.037 4
	f_3	**517,226 ± 242,789** **1**	653,435 ± 218,261 5	538,732 ± 252,030 2	552,326 ± 277,745 3	644,491 ± 223,597 4
	f_4	6.54 ± 5.017 5	3.42 ± 3.10 3	**2.0 ± 1.01** **1**	2.81 ± 2.29 2	3.56 ± 1.71 4
	f_5	2741 ± 756 5	2590 ± 931 4	2007 ± 1323 3	**1560 ± 662** **1**	1832 ± 1021 2
Basic multimodal functions	f_6	29.28 ± 16.83 2	51.14 ± 40.39 5	29.6 ± 18.8 3	**23.1 ± 2.0** **1**	39.52 ± 39.95 4
	f_7	0.0106 ± 0.0101 2	0.012 ± 0.0078 3	0.012 ± 0.0141 3	0.0126 ± 0.0052 4	**0.00942 ± 0.0113** **1**
	f_8	20.45 ± 0.128 2	20.5 ± 0.087 4	20.45 ± 0.1386 2	20.47 ± 0.108 3	**20.43 ± 0.0789** **1**
	f_9	126 ± 33.46 3	**114 ± 28.9** **1**	123 ± 23.34 2	129 ± 36.75 4	135 ± 36.81 5
	f_{10}	195 ± 65.66 2	213 ± 57.85 3	**183 ± 60.58** **1**	219 ± 87.66 5	215 ± 63.56 4
	f_{11}	25.8 ± 5.06 4	26.6 ± 5.54 5	24.8 ± 3.57 3	23 ± 4.20 2	**22.13 ± 4.74** **1**
	f_{12}	18,468 ± 21,246 2	21,383 ± 37,155 3	**13,763 ± 13,778** **1**	25,538 ± 28,047 4	71,070 ± 106,606 5
Expanded multimodal functions	f_{13}	9.0 ± 2.28 5	7.12 ± 2.44 4	7.0 ± 2.83 3	**6.22 ± 1.82** **1**	6.63 ± 1.65 2
	f_{14}	13.08 ± 0.216 3	13.19 ± 0.502 5	12.8 ± 0.422 2	**12.72 ± 0.43** **1**	13.11 ± 0.274 4
Nb/Nw/Mr		1/3/3.0	1/5/3.5	4/0/2.14	5/2/2.57	3/4/3.5

The bolded numbers are the best solutions for each function

are only approximately 1000 geysers in the world, with the majority of them in Yellowstone National Park (USA).

Geysers require specific conditions to erupt, such as (1) hot rocks below, (2) a plentiful supply of groundwater, (3) water storage under the surface, and (4) water will be sent to the surface via cracks.

To comprehend the operation of a geyser, you must first understand the connection between water and steam. Water in the form of steam is in a gaseous state. When the water reaches its boiling temperature, steam is created. Because steam fills 1600 times the volume of water, when it changes to steam at the surface, it expands dramatically. A geyser's eruption is fueled by a "steam explosion," which occurs when boiling water expands rapidly into considerably more voluminous steam. To conclude, a geyser occurs when overheated groundwater that has been contained at depth gets hot enough to explode at the surface [30].

Overall, the advantages of GEA are summarized as follows:

1. A novel optimizer based on an unusual geological phenomenon in nature.
2. It shows effectiveness and robustness in solving benchmark functions.
3. It has the ability to find global optimum in comparison to nature-inspired algorithms such as ABC, BBO, PSO, and RCGA.
4. It owns superior convergence capability for engineering optimization.

The remainder of this paper is as follows: the mathematical formulation of GEA based on its fundamental concepts is presented in Sect. 2. Section 3 concisely delineates the distinguishing characteristics between GEA and PSO-based algorithms. Section 4 is devoted to the numerical analysis of GEA. In this regard, the effects of different parameters of GEA are investigated. Then a comparison study on CEC 2005 test functions between the GEA and four other well-known evolutionary methods is evaluated. This section presents a comprehensive analysis of the performance of GEA in comparison to several powerful, improved algorithms across CEC 2005 functions. Additionally, a thorough investigation of GEA alongside well-known original algorithms is conducted on CEC 2017 functions, while the comparison

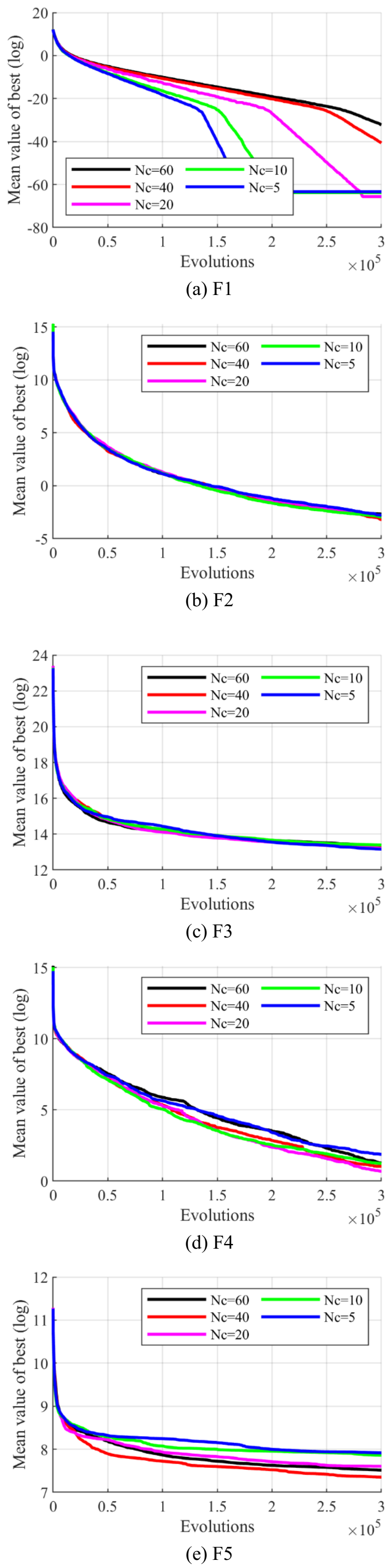

Fig. 4 Convergence rate comparison for GEA with different Nc for the unimodal benchmark functions such as **a** F1, **b** F2, **c** F3, **d** F4 and **e** F5

on CEC 2014 functions is carried out using improved algorithms. The section also delves into statistical tests and explores the exploitation and exploration process of GEA. Afterward, several real-parameter engineering problems are used to examine the effectiveness of GEA. Some relevant conclusions are presented in Sect. 5.

2 Geyser Algorithm (GEA)

In general, the concept of optimization is to search for optimum values of decision variables $\boldsymbol{X}$ to minimize or maximize a fitness function $f(\boldsymbol{X})$. In this part, the modeling of the recommended **GEA** as an optimization algorithm is described.

2.1 Basic Concept of Geyser Phenomena

Geysers are unusual geological phenomena from which boiling water and steam erupt [30, 31]. In other words, they are astonishing natural fountains that form strong eruptions of hot water into the sky. They are viable to erupt in the areas in which geothermal activities exist; consequently, their locations are mostly found in the vicinity of volcanic regions. These geological phenomena include numerous channels that extend deep into the ground. The available water in the channels is heated up to 200 ℃ by the molten rock named magma, which is located under the channels. The water progressively heats up until it boils. As the pressure builds, a portion of the water is obliged to move towards the Earth's surface, and the geyser spouts. The duration of eruptions varies between a few minutes to many days, and their height may fluctuate from a few meters to more than a hundred meters. The eruption will last until all of the water in the geyser has been pushed out or the temperature within the geyser has dropped. The reservoir progressively fills with water again over time, and the cycle is repeated [32, 33].

When the subsurface reservoir runs out of water or steam, the geyser ceases to erupt. Underground reservoirs store the water. Geyser steam and water jets can soar up to over 300 feet in the air (100 m).

In GEA, an initial population is generated randomly. This population expresses the water particles that are exposed to pressure and temperature in underground. The position of water particles indicates the possible solutions to the problem. These water particles in deep underground reservoirs, which are under pressure and heat, tend to find the shortest path to erupt. This is the main inspiration in the GEA, which is equivalent to finding the optimum solution in each iteration. The particles depending on their current positions, are directed to a better position in the channels or are randomly thrown to a corner of a channel. Water flow is the main factor to influence the particle positions. In Fig. 1, a sample of a

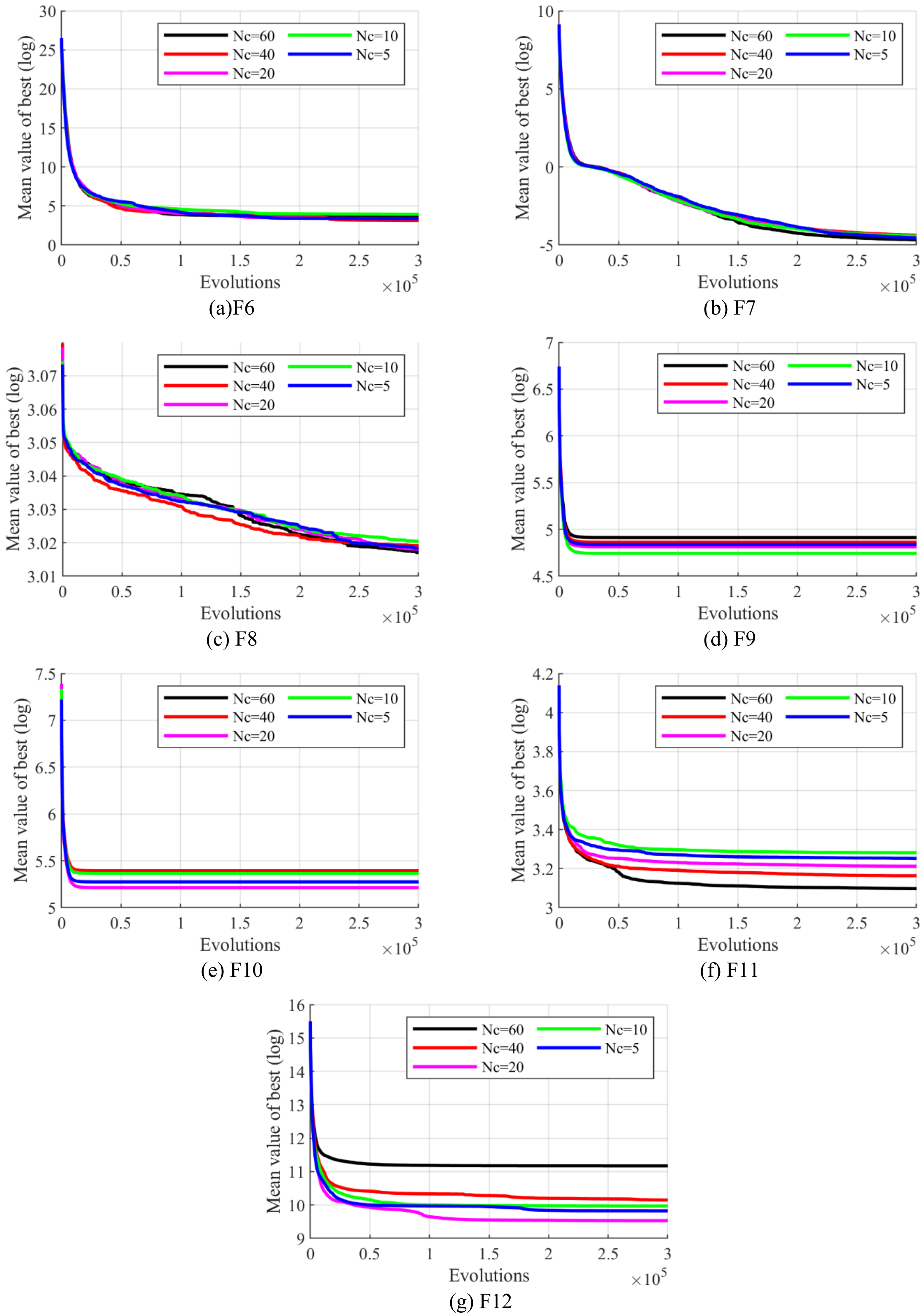

Fig. 5 Convergence rate comparison for GEA with different Nc for the basic multimodal benchmark functions such as **a** F6, **b** F7, **c** F8, **d** F9, **e** F10, **f** F11 and **g** F12

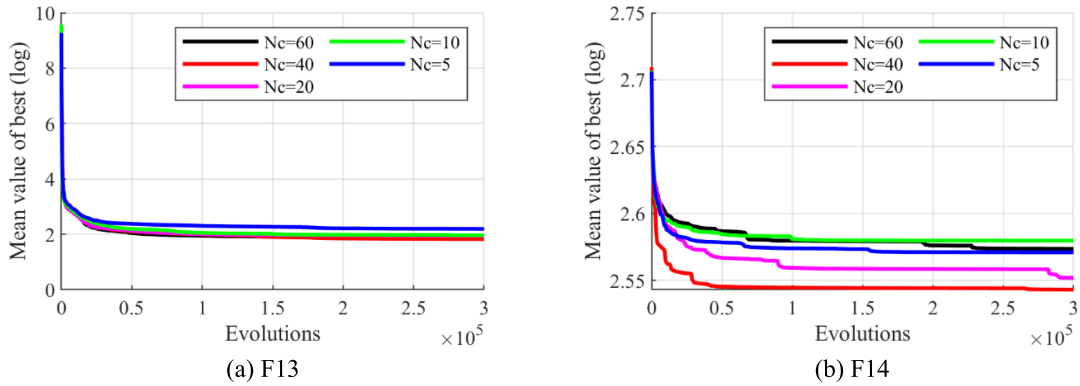

Fig. 6 Convergence rate comparison for GEA with different Nc for the expanded multimodal benchmark functions such as **a** F13 and **b** F14

groundwater pond and potential geyser is shown. In Fig. 2a, the geyser is depicted graphically, while Fig. 2b showcases its natural occurrences as examples.

2.2 Mathematical Modeling of GEA

The similarities of a geyser to an optimization algorithm can be understood by closely paying attention to its formation process. Water on the surface can be transported below ground, where it has the potential to merge with other subterranean waters and create a unified water source [33]. These subterranean reservoirs experience a rise in temperature upon encountering the heated rocks of the Earth's strata, yet owing to the immense underground pressure, they cannot convert into vapor. If there is a pathway with less pressure in the direction of movement, the superheated water will be pushed out of it, creating a geyser. Based on the explanation provided, the formation of geysers is primarily influenced by the movement of subterranean water, their interconnectivity, the rise in temperature and pressure, and the potential for a pathway to the Earth's surface. The various factors involved in the geyser phenomenon can be directly linked to the requirements of an optimization process. The primary objective of geyser phenomenon is to bring water to the Earth's surface, while the goal of optimization problems is to reach an optimal point through the population. Thus, the mass of water can be regarded as analogous to the initial population. The subsurface water is in motion such that the trajectories nearby amalgamate. Therefore, groundwater transport can be simulated through the population's mutation. The merging of nearby water paths and their movement towards an exit path is modeled through neighboring members and channels in a mutation process. This process mimics the movement of members toward a local optimum point. The selection of population members for adjacency will depend on their physical location and be determined by the precise model of nearby waterways' flow. However, it should be noted that any path can be selected as a channel, and this issue is derived from the fact that the geysers are spread across a plain in some areas. The selection of channels is performed twice to achieve optimal algorithm performance and prevent becoming trapped in local optima. During the first selection process, preference is given to members with higher objective function values. The second selection process grants greater chances of being chosen as channels to those members who have demonstrated better performance. The final considerations in creating the geyser are the temperature and pressure of the groundwater. These variables increase over time due to heightened heat transfer with water. Therefore, it can be inferred that they can be modeled as a functional variable of iteration and the value of the objective function.

Table 1 summarizes different phases of geyser phenomenon and their related mathematical modeling phases. Based on these explanations, this section is devoted to the mathematical formulation of GEA.

Based on concepts presented in Table 1, the mathematical modeling of GEA can be summarized in the following steps:

– Search for channels

This algorithm defines the members (particles) with more optimal fitness functions as the channels. The *i*th particle ($\boldsymbol{X_i}$) selects a possible channel based on the Roulette Wheel selection and moves towards it with the help of its neighbor.

– Roulette wheel selection

This is a stochastic method in which the probability of possible selections is relative to their fitness value. The wheel is divided into many portions, and each proportion is dedicated to one individual probability.

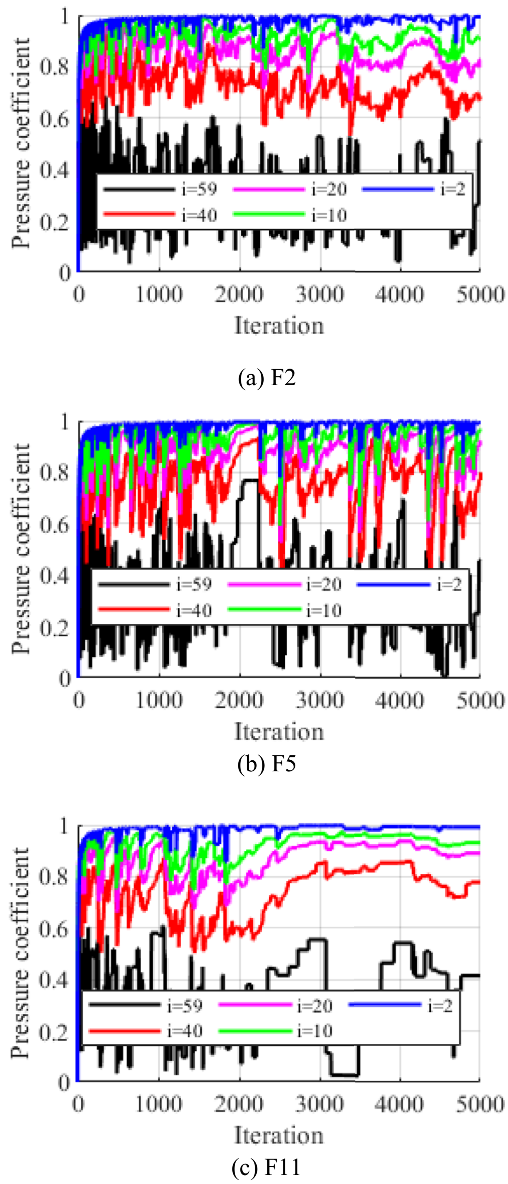

Fig. 7 The pressure coefficient variations for the 2-, 10-, 20-, 40-, and 59-*th* particle, for **a** F2, **b** F5, and **c** F11

In this method, the probability of selecting a member of the population as a channel is proportional to that member's fitness value, i.e., $f(\boldsymbol{X}_i)$. Since the number of channels is constant in this algorithm, the sum of all probabilities for selecting members is equal to unity. The probability of selecting the *i*th channel of the population on this wheel is equal to:

$$p_i = \frac{f(\boldsymbol{X}_i)}{\sum_{j=1}^{\mathrm{Nc}} f(\boldsymbol{X}_j)} \forall i \epsilon \{1, 2, \ldots, \mathrm{Nc}\} \tag{1}$$

In this equation, Nc is the number of channels. One effective method for selecting channels is to imagine a Roulette Wheel with segments representing the probability of each member of the population being chosen as a channel. Each time the wheel is rotated, each sample shown by the wheel marker is selected for the target. Here we are going to use this wheel to select a channel among Nc number of channels.

The sum of all probabilities for channel selection is equal to unity. To perform the Roulette Wheel selection, first, the sum of the cumulative probabilities of all cases must be obtained.

$$\sum_{i=1}^{\mathrm{Nc}} p_i = 1 \tag{2}$$

Based on deploying the Roulette Wheel mechanism, a possible channel, named $\boldsymbol{X}_{c,i}$, is selected for the *i*th particle ($\boldsymbol{X}_i$) to spout. Each channel is likely to be selected based on its fitness function value.

In this algorithm, we define the neighbor criterion as the distance similarity criterion. It means, any member that has the least value of the following equation is considered as the neighbor of the *i*th member ($\boldsymbol{X}_i$). The distance $d(\boldsymbol{X}, \boldsymbol{Y})$ between $\boldsymbol{X}$ and $\boldsymbol{Y}$ members is calculated as:

$$d\left(\boldsymbol{X}_m, \boldsymbol{Y}_l\right) = \frac{\sum_{j=1}^{\mathrm{D}} x_{m,j} y_{l,j}}{\left[\sum_{j=1}^{\mathrm{D}} x_{m,j}^2 \sum_{j=1}^{\mathrm{D}} y_{l,j}^2\right]^{\frac{1}{2}}} \tag{3}$$

$$\begin{aligned} \boldsymbol{X}_m &= \left[x_{m,1}, x_{m,2}, \ldots, x_{m,\mathrm{D}}\right], \boldsymbol{Y}_m \\ &= \left[y_{m,1}, y_{m,2}, \ldots, y_{m,\mathrm{D}}\right], \mathrm{m}, \mathrm{l} \in \{1, 2, \ldots, \mathrm{Npop}\} \end{aligned} \tag{4}$$

D is the number of decision variables. This distance is calculated for all particles (in size Npop) with respect to $\boldsymbol{X}_i$, and its neighbor ($\boldsymbol{X}_{n,i}$) is nominated based on the shortest distance to $\boldsymbol{X}_i$ (Eq. (3)). Now, we can obtain the new position of *i*th particle ($\boldsymbol{X}_i^{new,1}$) in the upper channel ($\boldsymbol{X}_{c,i}$) with the guide of its neighbor ($\boldsymbol{X}_{n,i}$) to find a way to erupt, which is determined by the following equation:

$$\boldsymbol{X}_i^{new,1} = \boldsymbol{X}_{n,i} + rand \times \left(\boldsymbol{X}_{c,i} - \boldsymbol{X}_i\right) + rand \times \left(\boldsymbol{X}_{c,i} - \boldsymbol{X}_{n,i}\right) \tag{5}$$

In the above equation, the target channel for $\boldsymbol{X}_i$, i.e. $\boldsymbol{X}_{c,i}$ is determined by the Roulette Wheel mechanism, *rand* generates a vector with D dimension filled with random numbers between 0 and 1. If $\boldsymbol{X}_i^{new,1}$ can find a better position, $\boldsymbol{X}_i$ replaces with $\boldsymbol{X}_i^{new,1}$, otherwise, $\boldsymbol{X}_i$ keeps its current position.

Equation (5) can be classified as a PSO-based equation. However, it has been transformed into a distinctive equation in GEA by incorporating the adjacency principle and employing the Roulette Wheel mechanism to select channels.

Table 4 The results for the real-parameter unimodal and multimodal test functions with Nc = Npop/3

Test functions (type/number)		Npop = 30 Mean ± Std Rank	Npop = 45 Mean ± Std Rank	Npop = 60 Mean ± Std Rank	Npop = 75 Mean ± Std Rank	Npop = 90 Mean ± Std Rank
Unimodal functions	f_1	9.13e−28 ± 7.87e−28 3	1.37e−28 ± 1.08e−28 2	**3.15e−29 ± 2.67e−29** **1**	1.43e−16 ± 1.55e−16 4	5.59e−10 ± 2.34e−10 5
	f_2	**0.0023 ± 0.0011** **1**	0.0136 ± 0.00522 2	0.0613 ± 0.0463 3	0.11 ± 0.0543 4	0.149 ± 0.075 5
	f_3	**487,585 ± 256,618** **1**	496,142 ± 162,167 2	538,732 ± 252,030 3	798,139 ± 422,876 4	894,657 ± 341,891 5
	f_4	1.92 ± 1.55 2	**1.66 ± 1.11** **1**	2.0 ± 1.01 3	4.76 ± 3.61 4	6.41 ± 3.22 5
	f_5	**1805 ± 595** **1**	2258 ± 552 5	2007 ± 1323 2	2249 ± 711 4	2228 ± 635 3
Basic multimodal functions	f_6	**27.43 ± 27.55** **1**	32.77 ± 21.94 4	29.6 ± 18.8 2	34.89 ± 33.094 5	30.02 ± 17.0 3
	f_7	**0.0091 ± 0.00845** **1**	0.0155 ± 0.0162 3	0.012 ± 0.0141 2	0.0215 ± 0.0105 4	0.0342 ± 0.0167 5
	f_8	**20.38 ± 0.1068** **1**	20.49 ± 0.1017 4	20.45 ± 0.1386 2	20.48 ± 0.0844 3	20.52 ± 0.126 5
	f_9	139 ± 33.37 4	140 ± 32 5	**123 ± 23.34** **1**	135 ± 27.08 3	132 ± 35.88 2
	f_{10}	182 ± 72.55 2	**165 ± 67.5** **1**	183 ± 60.58 3	191 ± 46.52 4	191 ± 50.05 4
	f_{11}	24.68 ± 5.75 2	**20.44 ± 3.83** **1**	24.8 ± 3.57 3	27.5 ± 5.07 5	25.36 ± 2.76 4
	f_{12}	49,405 ± 44,057 3	59,400 ± 131,372 4	**13,763 ± 13,778** **1**	20,463 ± 15,081 2	186,596 ± 18,306 5
Expanded multimodal functions	f_{13}	8.64 ± 2.28 5	8.19 ± 2.94 4	**7.0 ± 2.83** **1**	7.20 ± 1.42 2	7.60 ± 2.32 3
	f_{14}	12.8 ± 0.494 2	13.12 ± 0.303 4	12.8 ± 0.422 2	**12.76 ± 0.458** **1**	13.09 ± 0.369 3
Nb/Nw/Mr		**6**/1/**2.07**	3/3/3	4/**0/2.07**	1/3/3.5	0/8/4.07

The bolded numbers are the best solutions for each function

As mentioned above, the pressure has great effects on spout from the Earth. This pressure can be mathematically modeled by the probability given in Eq. (6).

$$P_i = \sqrt{\frac{\mathrm{Iter}}{\mathrm{Iter}-1}}\sqrt{\left(\frac{\mathrm{f}(\mathrm{X_i})-\mathrm{f_{min}}}{\mathrm{f_{max}}-\mathrm{f_{min}}}\right)^{\frac{2}{\mathrm{Iter}}}-\left(\frac{\mathrm{f}(\mathrm{X_i})-\mathrm{f_{min}}}{\mathrm{f_{max}}-\mathrm{f_{min}}}\right)^{\frac{\mathrm{Iter}+1}{\mathrm{Iter}}}} \quad \forall i \in \{1,2,\ldots,\mathrm{N_{pop}}\} \tag{6}$$

In the above equation, P_i is the pressure probability for the ith particle, $Iter$ is the current iteration of the algorithm, and f_{min} and f_{max} are the best and the worst values of the objective function, respectively, by the current iteration. It should be noted that $Iter$ starts from number 2 because number 1 is for initial population generation.

As mentioned earlier, the pressure on the ith particle can direct it toward the channel $\boldsymbol{X}_{c,i}^{new}$ which is selected from the candidate position of channels and can be determined by Roulette Wheel selection. The new probability for channel selection is given as follows:

$$p_i^{new} = 1 - p_i \tag{7}$$

Thus, the ith particle's new position is given as:

$$\boldsymbol{X}_i^{new,2} = \boldsymbol{X}_{c,i}^{new} + rand \times (P_i - rand) \times unifrand(\boldsymbol{X}_{max} - \boldsymbol{X}_{min}) \tag{8}$$

In the above equation, P_i is the pressure coefficient for the ith particle (Eq. (6)), and $unifrnd(\boldsymbol{X}_{max} - \boldsymbol{X}_{min})$ generates a random number from the continuous uniform distribution between the lower endpoint $\boldsymbol{X}_{min}$ and upper endpoint $\boldsymbol{X}_{max}$, which are the search area limitations of the problem. Update the particle positions with possible better solutions; otherwise, keep the current positions.

Equation (8) performs a local search around higher quality solutions. This equation is unique to the GEA, utilizing the pressure equations and the Roulette Wheel mechanism from Eqs. (6–7). Its effectiveness in preventing premature convergence makes it a valuable improvement that could be implemented in other algorithms.

As can be seen, the entire population is formed as a pressured mass of water. Each particle attempts to erupt by

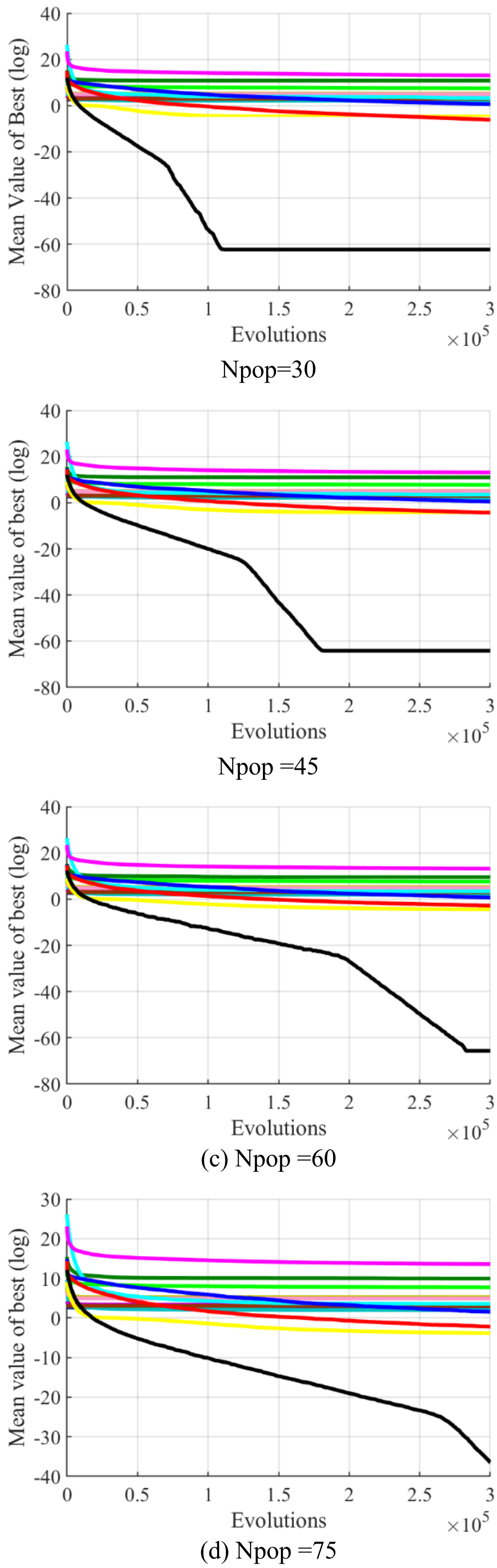

Fig. 8 Convergence rate comparison for GEA with different Npop such as **a** 30, **b** 45, **c** 60, **d** 75 and **e** 90 for all the selected benchmark functions of the CEC 2005

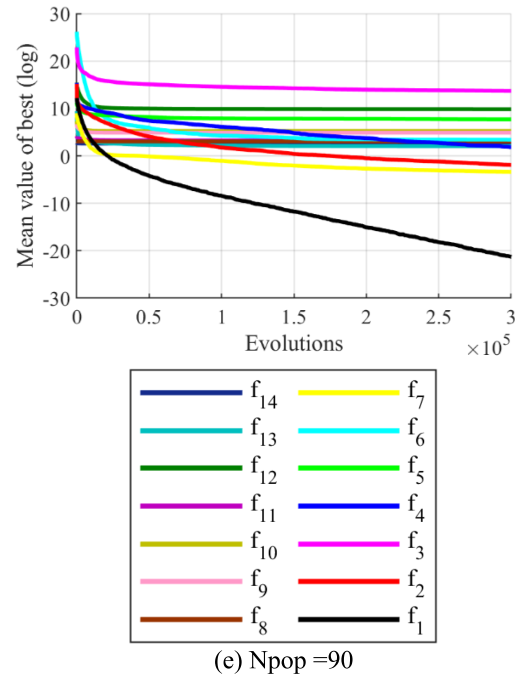

Fig. 8 (continued)

utilizing its neighbor (Eq. (5)), which is chosen depending on the distances between the particles (Eqs. (3–4)).

Indeed, the underground water flows are connected to one another, and each line of flow is connected to its neighbor, resulting in a massive mass of water. According to the definition, channels are members who do better in fitness, but to better adapt to reality, it is also possible to select a solution from the weaker options. As a result, the Roulette Wheel mechanism is used to define these channels, while the selection probability is generated using the fitness function of each particle (Eqs. (1–2)). This approach is entirely consistent with real-world conditions in which water can erupt into the Earth through the channel randomly. Until now, the path flows have fused and are traveling in the direction of their probable channels. The pressure and temperature of a mass of water (i.e., the algorithm's population) are determined here using Eq. (6). A new channel is chosen for each population, and the probability selection is updated in accordance with Eq. (7). Now, the eruptive process is analyzed using Eq. (8), with the channel, pressure, and temperature all having a significant effect on it. In general, water mass moves along the quickest path to water, whereas the eruptive process is governed by pressure and temperature.

To have a better overview of implementing the GEA, the above-mentioned procedure is illustrated in a pseudo code as shown in Fig. 3.

Table 5 Summary of the results for the real-parameter unimodal and multimodal test functions with D = 30

Test functions (type/number)		ABC Mean ± Std Rank, Winner	BBO Mean ± Std Rank, Winner	PSO Mean ± Std Rank, Winner	RCGA Mean ± Std Rank	GEA Mean ± Std Rank
Unimodal functions	f_1	5.79e−16 ± 1.4e−16 2, –	0.0231 ± 0.0064 4, –	605.4440 ± 224.4084 5, –	3.85e−9 ± 7.29e−10 3,–	**9.13e−28 ± 7.87e−28** **1**
	f_2	3.70e+03 ± 9.11e+02 5, –	11.03 ± 2.17 2, –	2.31e+03 ± 4.40e+02 4, –	1.07e+02 ± 1.15e+02 3,–	**0.0023 ± 0.0011** **1**
	f_3	1.18e+07 ± 2.29e+07 4, –	2.55e+06 ± 8.51e+05 3, –	1.61e+07 ± 1.04e+07 5, –	1.94e+06 ± 1.03e+06 2,–	**4.88e+05 ± 2.57e+05** **1**
	f_4	4.53e+04 ± 4.57e+03 5, –	45.34 ± 7.57 2, –	1.11e+04 ± 4.01e+03 4, –	4.08e+03 ± 8.75e+02 3,–	**1.92 ± 1.55** **1**
	f_5	9.35e+03 ± 1.25e+03 5, –	4.66e+03 ± 5.90e+02 3, –	6.62e+03 ± 2.25e+03 4, –	3.53e+03 ± 1.90e+03 2,–	**1.81e+03 ± 5.95e+02** **1**
Basic multimodal functions	f_6	4.45 ± 2.48 2, –	169.0 ± 31.68 4, –	2.14e+07 ± 2.38e+07 5, –	4.00e+01 ± 3.54e+01 3,–	**27.43 ± 27.55** **1**
	f_7	0.0214 ± 0.012 2, –	0.9050 ± 0.1058 4, –	137.0 ± 79.15 5, –	0.127 ± 0.091 3,–	**0.0091 ± 0.00845** **1**
	f_8	20.92 ± 0.038 4, –	20.61 ± 0.125 2, –	20.67 ± 0.1027 3, –	20.61 ± 0.542 2,–	**20.38 ± 0.107** **1**
	f_9	**4.44e−−15 ± 4.59e−−15** **1, +**	34.95 ± 11.06 3, +	164.66 ± 37.25 5, –	3.56e−−02 ± 3.13e−−02 2, +	139.0 ± 33.37 4
	f_{10}	268.0 ± 53.75 5, –	**75.17 ± 10.82** **1, +**	206.83 ± 68.23 4, –	194.0 ± 80.50 3,–	182.0 ± 72.55 2
	f_{11}	27.78 ± 1.57 5, –	24.74 ± 4.12 2, –	26.75 ± 4.034 4, –	2.63e+01 ± 7.42e+00 3,–	**24.68 ± 5.75** **1**
	f_{12}	1.30e+04 ± 4.07e+03 2, +	8.96e+03 ± 6.98e+03 **1, +**	8.92e+04 ± 3.90e+03 5, –	5.17e+04 ± 9.16e+03 4,–	4.94e+04 ± 4.41e+04 3
Expanded multimodal functions	f_{13}	**0.444 ± 0.059** **1, +**	2.37 ± 0.246 2, +	10.79 ± 1.64 5, –	10.00 ± 7.02 4,–	8.64 ± 2.28 3
	f_{14}	13.01 ± 0.191 3, –	12.82 ± 0.997 2, –	13.16 ± 0.105 4, –	13.20 ± 0.489 5,–	**12.8 ± 0.494** **1, +**
Nb/Nw/Mr		2/6/3.2857	2/0/2.5000	0/7/4.4286	0/1/3.0000	**10/0/1.5714**
+/–/=		3/11/0	4/10/0	0/14/0	1/13/0	**–**

The bolded numbers are the best solutions for each function

Based on the above explanations, each particle in GEA is updated twice throughout each step of evolution. The first step is to apply Eq. (5). This equation requires a channel that is chosen according to the Roulette Wheel mechanism (Eqs. (1–2)) and a neighbor for particle *i*th that is chosen according to the shortest distance to the target particle (Eqs. (3–4). Equation (8) is then used to update the location of each member. This equation requires a pressure value, which is derived using the iteration number and some objective function values (Eq. (6)), and a channel for each particle, which is chosen based on the particle's new probability (Eq. (7)). Between these two updated particles, i.e., Eqs. (5) and (8), the first one aims to maintain population variety throughout the search process, whereas the second tries to converge the particle toward solutions with better objective function values. These two concepts can create a powerful optimization methodology called Geyser Algorithm (GEA).

3 Differences Between GEA and PSO-based Algorithms

In classical algorithms such as PSO, a local search is performed based on the particle's position, while a global search is performed based on the position of the best solution. However, particles that are far from the global best solution may become trapped in local optima when they move towards the best solution. The other PSO-based algorithms have attempted to eliminate these defects. For example, the population is divided into two groups in Cat Swarm

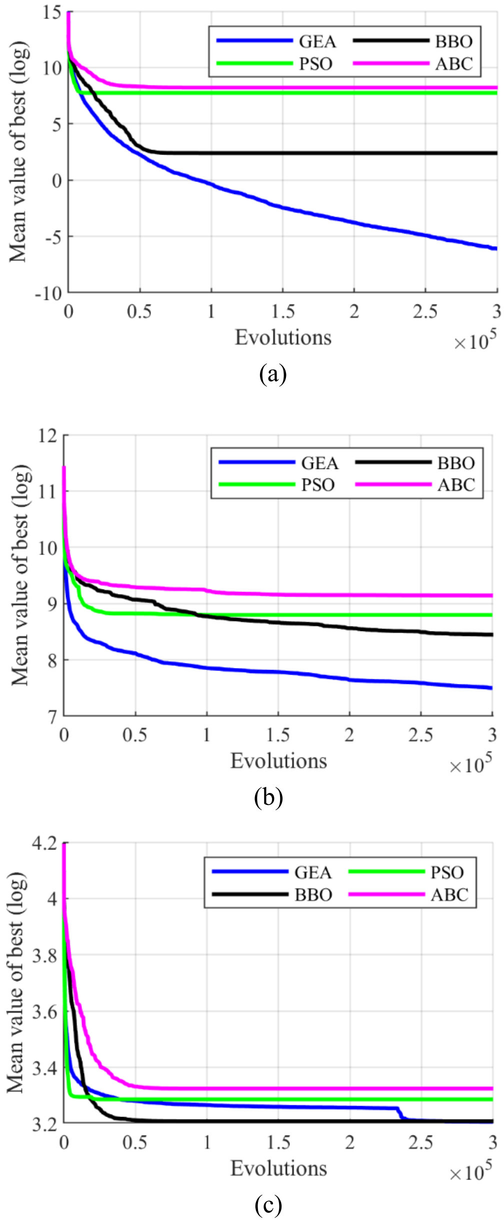

Fig. 9 Convergence rate comparison for benchmark functions **a** F2, **b** F5, and **c** F11

Optimization (CSA) [34]. One group follows the global search with the best solution, known as the Tracing mode. The other group uses a mutation method to preserve population diversity and perform a local search called the Seeking mode. An interval-based probability with the worst solution is used in this mode to select the next population.

It should be noted that maintaining the population dispersion through the mutation that occurred in the seeking mode of the CSA actually happens through the displacement of some variables of the selected members, which can be considered a discrete step process. On the other hand, the selection probability is determined such that the better solutions experience a higher mutation probability, which can confuse the final steps of the evolution process. In the GEA, not all members move towards the best solution but towards the channels instead. This movement towards the channels occurs through nearby members, which reduces the probability of being trapped in local optima. Therefore, moving towards channels in local search happens completely continuously. The global search in the GEA can also be considered a clustered search. In this context, the movement of each cluster's members will be towards the respective channel. The greater the distance a member has from the channel, the more drastic changes will be felt. Therefore, it can be confident that the best solutions will not experience confusion, in addition to maintaining population diversity.

The idea of dividing the population to move toward channels in GEA can be similar to the memeplex concept in Shuffled Frog Leaping Algorithm (SFLA) [35]. However, dividing the population between channels and being placed in memeplexes is entirely different. In GEA, the population moves towards the channel through the adjacent members, distinct from the movement inside the memeplexes in the SFLA algorithm [36]. Therefore, the global and local search processes in GEA and SFLA are different.

4 Experimental Study

We have considered the most investigated test functions used in IEEE CEC 2005 [37]. In this experimental study, fourteen CEC 2005 test functions [38] are selected for global real-parameter optimization, which the summary of the selected tests has been provided in Table 2.

4.1 Experimental Analysis of GEA

In this section, the effects of two parameters of GEA, Nc and Npop, on algorithm performance are investigated. Then the performance of GEA method on some constrained engineering optimization problems is investigated.

4.1.1 The Impact of Nc Selection on the GEA

As mentioned earlier, the proposed GEA has a control parameter named Nc which signifies the number of possible channels for water particle eruption. Different values of this parameter may lead to dissimilar results for different problems. Therefore, in this section, we have selected different values for ***Nc*** and run the algorithm for all benchmark functions.

Table 6 The results of the modern algorithms on CEC 2005 benchmark functions (D = 30 and NFEs = 3.00E+05)

Test functions	GL-25 [40, 41]	CLPSO [40, 42]	EPSDE [43, 44]	jDE [45, 46]	AuDE2 [46]	FIPS [47, 48]	OLPSO [47, 49]	sHPSO [47, 50]	DMS-PSO [51]	GEA
	Mean(Rank) Std	Mean(Rank) Std	Mean(Rank) Std	Mean(Rank) Std	Mean(Rank) Std	Mean(Rank) Std	Mean(Rank) Std	Mean(Rank) Std	Mean(Rank) Std	Mean(Rank) Std
f_1	5.60e−27(3)	**0.00e+00(1)**	**0.00e+00(1)**	5.38e−12(5)	3.29e+01(6)	**0.00e+00(1)**	**0.00e+00(1)**	**0.00e+00(1)**	1.89e−15(4)	9.13e−28(2)
	1.76e−26	0.00e+00	0.00e+00	2.12e−12	5.41e+01	0.00e+00	0.00e+00	0.00e+00	1.03e−14	7.87e−28
f_2	4.04e+01(6)	8.40e+02(9)	**4.25e − 10(1)**	1.23e+02(8)	1.27e+04(10)	77.94(7)	13.79(4)	1.44e–02(3)	21.44(5)	2.30e−03(2)
	6.28e+01	1.90e+02	2.12e − 09	2.00e+01	3.24e+03	27.05	8.33	7.10e–02	13.80	1.10e−03
f_3	2.19e+06(4)	1.42e+07(7)	1.37e+06(3)	4.21e+06(5)	4.31e+07(10)	2.45e+07(9)	1.60e+07(8)	8.75e+05(2)	5.89e+06(6)	**4.88e+05(1)**
	1.08e+06	4.19e+06	4.97e+06	9.02e+05	1.81e+07	6.29e+06	7.04e+06	5.34e+05	2.59e+06	2.57e+07
f_4	9.07e+02(2)	6.99e+03(7)	2.30e+05(10)	5.99e+03(6)	1.81e+04(8)	1.15e+03(3)	2.18e+03(5)	2.02e+04(9)	1.35e+03(4)	**1.92(1)**
	4.25e+02	1.73e+03	6.26e+05	1.36e+03	4.17e+03	3.73e+02	1.09e+03	9.94e+03	5.39e+02	1.55
f_5	2.51e+03(5)	3.86e+03(7)	**9.15e+02(1)**	5.14e+03(8)	1.28e+04(10)	2.22e+03(3)	3.30e+03(6)	6.94e+03(9)	2.41e+03(4)	1.81e+03(2)
	1.96e+02	4.35e+02	5.45e+02	7.36e+02	1.92e+03	5.14e+02	3.75e+02	1.43e+03	4.09e+02	5.95e+02
f_6	21.50(4)	4.16(2)	**3.18e − 01(1)**	35.70(6)	7.94e+06(10)	37.70(7)	20.68(3)	115.27(9)	52.25(8)	27.43(5)
	1.17	3.48	1.10e+00	3.46	1.27e+07	35.03	24.97	228.56	38.22	27.55
f_7	2.78e−02(5)	4.51e−01(9)	1.22e−02(3)	6.22e−02(8)	9.94e+01(10)	0.03(6)	0.01(2)	0.04(7)	1.75e−02(4)	**9.10e−03(1)**
	3.62e−02	8.47e−02	1.34e−02	1.97e−02	4.79e+01	0.02	0.01	0.04	1.06e−02	8.45e−03
f_8	20.90(4)	20.90(4)	20.90(4)	20.90(4)	20.90(4)	20.94(5)	20.96(6)	**20.18(1)**	20.53(3)	20.38(2)
	5.94e−02	4.41e−02	6.24e−02	4.59e−02	5.72e−02	0.06	0.08	0.19	4.65e−02	0.1068
f_9	2.45e+01(3)	**0.00e+00(1)**	**0.00e+00(1)**	4.34e+01(4)	9.14e+01(7)	57.11(5)	**0.00e+00(1)**	82.54(6)	13.73(2)	1.39e+02(8)
	7.35e+00	0.00e+00	0.00e+00	6.04e+00	2.08e+01	14.55	0.00e+00	24.35	10.19	3.34e+01
f_{10}	1.42e+02(5)	1.04e+02(3)	6.14e+01(2)	1.92e+02(9)	1.66e+02(6)	1.78e+02(7)	1.09e+02(4)	2.43e+02(10)	**54.05(1)**	1.82e+02(8)
	6.45e+01	1.53e+01	9.02e+00	1.35e+01	3.00e+01	9.25	18.94	89.17	6.59	7.26e+01
f_{11}	32.70(8)	26.00(3)	35.70(9)	27.90(4)	29.80(6)	38.36(10)	25.05(2)	32.29(7)	28.28(5)	**24.68(1)**
	7.79e+00	1.63e+00	2.87e+00	1.18e+00	2.95e+00	1.52	3.14	3.84	1.49	5.75
f_{12}	6.53e+04(9)	1.79e+04(3)	5.75e+04(8)	3.78e+04(5)	9.49e+04(10)	5.62e+04(7)	1.22e+04(2)	2.43e+04(4)	**6.78e+03(1)**	4.94e+04(6)
	4.69e+04	5.24e+03	1.05e+04	5.63e+03	4.07e+04	2.00e+04	5.42e+03	2.66e+04	5.49e+03	4.41e+04
f_{13}	6.23(7)	2.06(2)	2.52(3)	6.04(5)	6.14(6)	12.51(9)	**1.86(1)**	6.14(6)	3.95(4)	8.64(8)
	4.88e+00	2.15e-01	1.77e−01	5.82e−01	1.72e+00	0.96	0.28	2.31	0.51	2.28
f_{14}	13.10(6)	12.80(3)	13.40(9)	12.90(4)	**11.90(1)**	13.12(7)	13.13(8)	13.07(5)	12.44(2)	12.80(3)
	1.84E−01	2.48e−01	1.98e−01	1.56e−01	3.99e−01	2.10e−01	2.00e−01	3.90e−01	2.40e−01	4.94e−01
Mr	5.0714	4.3571	3.8571	5.7857	7.4286	6.1429	3.7857	5.9286	3.7857	**3.7143**
Fr	5	4	3	6	9	8	2	7	2	**1**

The bolded numbers are the best solutions for each function

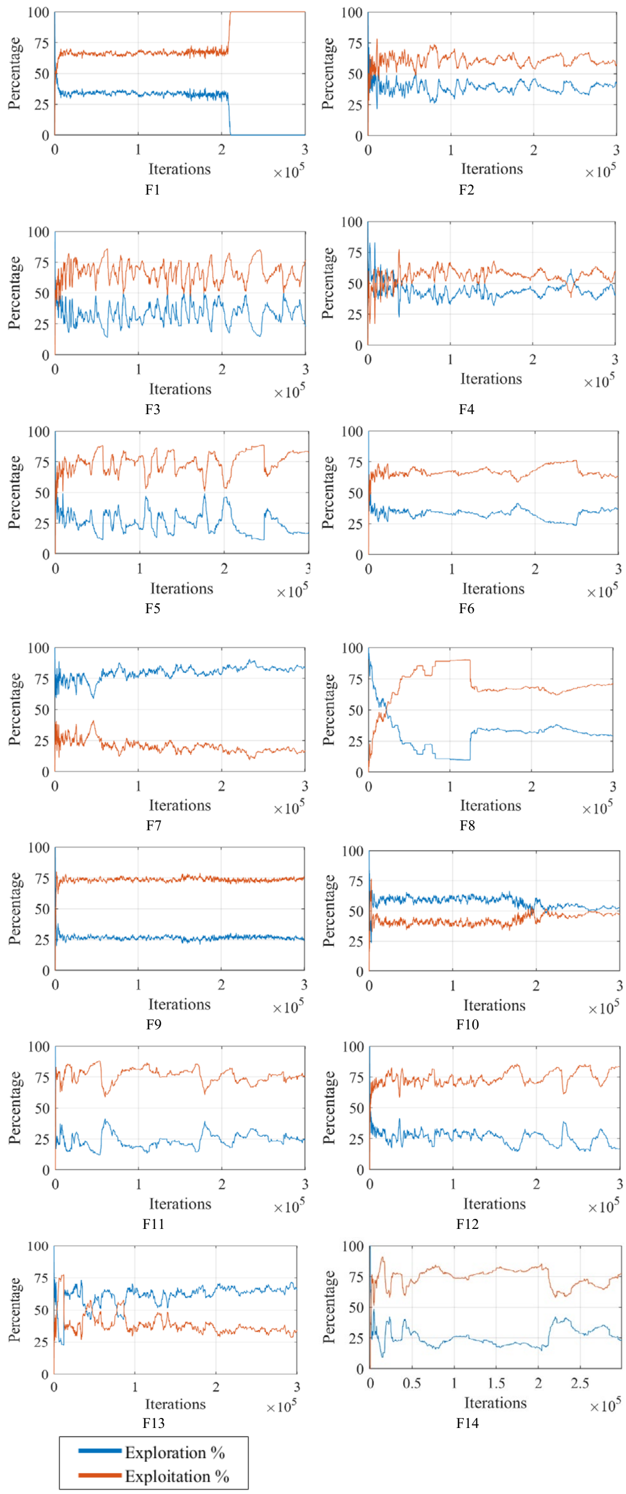

Fig. 10 Exploration-exploration of GEA

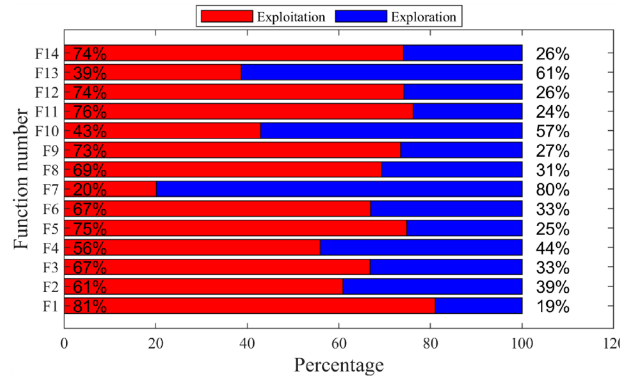

Fig. 11 Mean exploration–exploitation of test functions by GEA

The effectiveness of GEA has been compared for different values of Nc in the same condition. For each case, 30 independent runs with 300,000 number of function evaluation (NFEs) [37], and 30 dimensions are considered. The population size Npop is considered 60 for all the tests.

The results for the real-parameter unimodal and multimodal test functions for different values of Nc have been summarized in Table 3. In the last row, there are three parameters in which Nb and Nw represent how many times the algorithm has gotten better and worst results than expected, respectively, and Mr is the mean rank of the corresponding case. It can be observed that the value of Nc = Npop/3 can be a proper choice for all test functions, which we have used this value in the comparative study section. In addition, the convergence rate comparison for the test functions is shown in Figs. 4, 5, and 6 in order to show the effect of the different values Nc on the performance of the proposed GEA for the various test functions.

In addition, in Fig. 7 the pressure coefficient variations (P) for the 2-, 10-, 20-, 40-, and 59-*th* particle are shown for the F2, F5, and F11 of test functions that will be given in Table 3. For these curves, the number of 5000 iterations as well as the population size of 60 are considered. The particles are sorted from the best to worst.

4.1.2 The Impact of Npop on the GEA Performance

Another control parameter in proposed GEA is Npop, which should be determined by user. In order to evaluate its impact on the algorithm, five different values for Npop (30, 45, 60,

Table 7 Configuring parameters in compared optimization algorithms

Algorithm	Parameter settings with NFEs = 3.00E+05
HHO	C = [0, 2], E = [-2,2], C = [0, 2]
SCA	a = 2, r_1 = a – Iter × (a / Itermax), r_2 = 2π × rand(), r_3 = 2 × rand(), r_4 = rand()
GWO	Convergence constant a = [2 0] (linear reduction), Npop = 100
BA	The factors updating α and γ are set to 0.9, the pulse rate (r), Npop = 50
EHO	β = 0.1, α = 0.5, clan = 5
BOA	c = 0.01, α = 0.1, p = 0.8
GSA	α = 20, G_0 = 100, Npop = 100

75, and 90) are selected and the algorithm runs for all benchmark functions.

The results for the real-parameter unimodal and multimodal test functions for different values of Npop have been given in Table 4, the value of Npop = 60 is the best choice for all test functions. Other parameters such as Nb, Nw, and Mr are defined as before. Also, the convergence rate comparison for the test functions is shown in Fig. 8 to show the effect of the value Npop in GEA.

4.2 A Comparative Analysis for GEA on CEC Standard Benchmark Functions

4.2.1 Results and Analysis of Category 1 Experiments: The Basic Optimization Algorithms

In this category, we have used the same tests of CEC 2005 [37] mentioned in Table 2 to assess the robustness of GEA. The effectiveness of GEA has been compared with four nature-inspired optimization algorithms, including PSO [7], ABC [9], BBO [11], and real-coded genetic algorithm (RCGA) [39] in a fair manner, and also, the parameter settings of the four algorithms have been chosen as those indicated in its original paper. For each method, 30 independent runs with 300,000 Number of Function Evaluations (NFEs) are considered. The number of initial population in each algorithm is set to 30. In Table 5, the mean and the standard deviation (Std.) of the results obtained by each algorithm are given. It is obvious from the recorded results that GEA succeeds in finding the best solution on most of the cases compared to all other methods. N_b signifies how often does this algorithm produce superior results as expected, N_w represents how often does this algorithm produce inferior results when compared to other methods, and M_r is the mean rank of the corresponding method.

In addition, in the last row of this table, comparative indexes for each algorithm are given. The plus sign denotes the cases which the corresponding algorithm outperforms the proposed GEA, and the minus sign indicates the cases which the GEA underperforms the corresponding algorithm.

The convergence rate comparison for three test functions (TF2, TF5, and TF11) has been given in Fig. 9 in order to show the convergence performance of GEA in comparison to well-known ABC, BBO, PSO, and RCGA algorithms.

The convergence behavior curves indicate that GEA has a better performance with adequate accuracy compared to other optimization algorithms (ABC, BBO, PSO, and RCGA) for highly complex benchmark functions.

These results showed the proposed GEA optimizer have a competitive and better quality solution with the suitable ability to converge for most of the CEC 2005 real-parameter benchmark functions in compare with ABC, BBO, PSO, and RCGA popular algorithms.

The convergence characteristics depicted in Fig. 9 indicate that preserving the population diversity during the optimization process is of significant importance. Specifically, Fig. 9a illustrates that the BBO algorithm attains convergence at approximately 17% of the optimization process, whereas the GEA effectively maintains population diversity. Moreover, Fig. 9b demonstrates the superior performance of the GEA in comparison to the BBO algorithm in identifying more optimal solutions.

4.2.2 Results and Analysis of Category 2 Experiments: The Modern Algorithms

In this category of experiments, Table 6 has compared the efficiency of GEA with the performance of some other modern optimizers (including GL-25: Global and Local real-coded genetic algorithms [40, 41]; CLPSO: Comprehensive Learning PSO [40, 42]; EPSDE: an Ensemble of control Parameters and mutation Strategies with the DE [43, 44]; jDE: a self-adaptive DE [45, 46]; AuDE2: an Adaptive Unified DE [46]; FIPS: a Fully-Informed PSO [47, 48]; OLPSO: Orthogonal Learning PSO [47, 49]; sHPSO: static Heterogeneous PSO [47, 50]; and DMS-PSO: Dynamic Multi-Swarm PSO with two mutation operators [51])].

EPSDE and GEA achieved the highest accuracy on unimodal functions in the CEC 2005 test set. The proposed GEA attained the greatest result on F7 and F11, and the second most successful performance on F8 for the fundamental multimodal functions. EPSDE outperformed DMS-PSO on F6 and F9, whereas DMS-PSO outperformed EPSDE on

Table 8 Evaluation of different algorithms using CEC2017 benchmarks (D = 30 and NFEs = 3.00E+05)

Function	GEA Mean Std	BOA Mean Std	EHO Mean Std	HHO Mean Std	BA Mean Std	GWO Mean Std	GSA Mean Std	SCA Mean Std
F1	1.92E+3 4.15E+2	4.47E+10 6.73E+9	2.68E+10 5.87E+9	1.04E+7 1.87E+6	3.37E+11 3.48E+9	**1.72E+3** 8.91E+8	1.90E+3 1.03E+3	1.25E+10 2.94E+9
F3	**6.86E-03** 5.77E-03	6.48E+4 9.30E+3	7.23E+4 9.76E+3	4.85E+3 1.70E+3	2.85E+9 4.62E+8	2.82E+4 9.70E+3	8.27E+4 4.33E+3	3.47E+4 7.52E+3
F4	**1.69E+0** 1.92E+0	1.94E+4 3.77E+3	4.60E+3 3.91E+2	1.25E+2 2.36E+1	3.54E+4 1.36E+3	1.70E+2 4.98E+1	1.42E+2 1.59E+1	1.05E+3 3.75E+2
F5	1.71E+2 4.18E+1	3.71E+2 2.33E+1	3.34E+2 1.80E+1	2.30E+2 2.98E+1	5.26E+3 2.13E+1	**9.20E+1** 2.63E+1	2.26E+2 2.01E+1	2.73E+2 2.24E+1
F6	3.06E+1 1.68E+1	7.00E+1 9.54E+0	7.10E+1 6.00E+0	6.34E+1 5.40E+0	6.85E+2 6.55E+0	**4.00E+0** 2.33E+0	5.00E+1 2.75E+0	4.90E+1 5.20E+0
F7	2.42E+2 5.78E+1	6.10E+2 5.58E+1	7.30E+2 4.43E+1	5.30E+2 8.35E+1	1.80E+3 1.63E+2	1.35E+2 4.95E+1	**8.70E+1** 1.19E+1	4.40E+2 4.15E+1
F8	2.47E+2 4.72E+1	3.00E+2 1.85E+1	3.00E+2 2.15E+1	1.69E+2 2.33E+1	1.65E+4 3.98E+1	**8.10E+1** 1.25E+1	1.51E+2 1.31E+1	2.50E+2 1.62E+1
F9	2.70E+3 7.21E+2	8.45E+3 1.03E+3	8.48E+3 1.20E+3	5.86E+3 5.95E+2	1.01E+4 1.69E+3	**2.80E+2** 1.39E+2	2.03E+3 3.92E+2	4.81E+3 1.43E+3
F10	4.44E+3 1.02E+3	7.49E+3 2.75E+2	7.15E+3 3.14E+2	4.39E+3 6.57E+2	9.88E+4 7.65E+3	**2.73E+3** 5.49E+2	3.87E+3 4.34E+2	7.04E+3 3.42E+2
F11	2.19E+2 1.01E+2	4.51E+3 1.83E+3	1.03E+3 1.46E+2	**1.52E+2** 3.99E+1	3.46E+4 3.37E+3	4.10E+2 4.42E+2	3.50E+2 8.92E+1	1.05E+3 2.65E+2
F12	**3.82E+4** 3.97E+4	1.05E+10 3.41E+9	3.17E+9 5.20E+8	9.24E+6 5.95E+6	3.95E+9 9.45E+7	3.31E+7 3.81E+7	1.03E+7 1.93E+7	1.13E+9 2.52E+8
F13	**1.43E+3** 6.44E+2	6.52E+9 3.65E+9	1.06E+9 2.71E+8	3.14E+5 1.17E+5	4.17E+9 6.17E+8	6.63E+6 2.33E+7	2.97E+4 6.45E+3	3.87E+8 1.24E+8
F14	**4.96E+2** 1.89E+2	9.60E+5 1.84E+6	1.79E+5 7.54E+4	5.19E+4 1.03E+5	1.35E+7 1.65E+6	7.96E+4 1.76E+5	4.73E+5 1.31E+5	1.42E+5 8.76E+4
F15	**7.98E+3** 1.01E+4	1.35E+8 1.80E+8	2.60E+7 1.04E+7	6.89E+4 6.59E+4	1.73E+8 6.11E+7	2.43E+5 5.82E+5	1.02E+4 1.93E+3	1.19E+7 1.07E+7
F16	1.10E+3 2.92E+2	4.58E+3 1.38E+3	2.57E+3 1.94E+2	1.60E+3 3.65E+2	4.46E+4 2.96E+2	**7.20E+2** 2.39E+2	1.58E+3 2.84E+2	2.01E+3 3.47E+2
F17	6.22E+2 2.11E+2	6.17E+3 5.38E+3	8.90E+2 1.65E+2	9.54E+2 2.84E+2	3.07E+4 1.82E+2	**2.30E+2** 1.16E+2	1.20E+3 1.70E+2	7.10E+2 1.88E+2
F18	**2.85E+4** 9.82E+3	1.08E+7 1.50E+7	1.38E+6 6.64E+5	8.29E+5 7.16E+5	2.26E+9 1.98E+7	7.73E+5 1.40E+6	3.18E+5 1.76E+5	3.92E+6 2.66E+6
F19	**4.40E+3** 4.64E+3	2.64E+8 2.69E+8	6.23E+7 2.53E+7	2.41E+5 1.77E+5	1.66E+9 6.24E+7	2.04E+5 3.88E+5	1.23E+4 5.13E+3	2.13E+7 1.10E+7
F20	6.46E+2 3.42E+2	8.40E+2 1.04E+2	5.80E+2 7.75E+1	7.37E+2 2.12E+2	3.96E+3 1.36E+2	**3.30E+2** 1.66E+2	1.03E+3 2.36E+2	6.30E+2 1.01E+2
F21	**3.90E+1** 4.22E+1	3.90E+2 1.51E+2	5.00E+2 1.60E+1	4.60E+2 5.25E+1	2.87E+3 2.55E+1	2.70E+2 1.85E+1	4.60E+2 1.95E+1	4.50E+2 1.50E+1
F22	**1.91E+2** 4.94E+1	2.20E+3 7.00E+2	2.99E+3 3.14E+2	4.23E+3 1.77E+3	5.33E+4 1.70E+3	2.27E+3 1.45E+3	4.19E+3 1.69E+3	5.98E+3 2.36E+3
F23	7.74E+2 1.27E+2	9.00E+2 1.23E+2	9.90E+2 5.15E+1	8.07E+2 1.05E+2	3.37E+4 4.95E+2	**4.30E+2** 3.04E+1	1.26E+3 1.23E+2	6.80E+2 2.64E+1
F24	9.13E+2 3.57E+2	1.37E+3 1.67E+2	1.08E+3 6.37E+1	1.05E+3 1.47E+2	5.77E+4 5.19E+5	**5.00E+2** 4.70E+1	8.90E+2 5.57E+1	7.60E+2 2.46E+1
F25	4.36E+2 3.24E+1	3.15E+3 4.64E+2	2.22E+3 5.33E+2	**4.10E+2** 2.02E+1	4.95E+5 4.86E+7	4.60E+2 2.69E+1	4.30E+2 1.22E+1	6.70E+2 7.39E+1
F26	3.84E+3 8.38E+2	7.80E+3 1.17E+3	6.00E+3 5.87E+2	**3.75E+3** 2.12E+3	7.85E+4 6.76E+2	1.83E+3 2.45E+2	4.26E+3 8.95E+2	4.33E+3 3.06E+2
F27	1.18E+3 2.65E+2	1.06E+3 1.35E+2	1.23E+3 9.38E+1	6.31E+2 6.78E+1	3.65E+4 6.75E+1	**5.30E+2** 1.78E+1	1.97E+3 3.21E+2	6.90E+2 2.60E+1
F28	4.90E+2 2.68E+1	5.05E+3 5.08E+2	2.29E+3 3.00E+2	**4.52E+2** 2.65E+1	5.64E+4 4.33E+2	5.30E+2 4.83E+1	5.10E+2 4.94E+1	1.03E+3 1.63E+2

Table 8 (continued)

Function	GEA Mean Std	BOA Mean Std	EHO Mean Std	HHO Mean Std	BA Mean Std	GWO Mean Std	GSA Mean Std	SCA Mean Std
F29	9.87E+2 2.81E+2	6.47E+3 2.74E+3	2.30E+3 2.14E+2	1.37E+3 2.96E+2	5.46E+4 3.57E+2	**8.10E+2** 1.26E+2	1.81E+3 2.10E+2	1.69E+3 2.52E+2
F30	**1.01E+4** 7.62E+3	5.74E+8 3.63E+8	8.97E+7 3.04E+7	1.65E+6 8.27E+5	1.65E+9 6.31E+7	3.90E+6 3.10E+6	1.67E+5 1.24E+5	7.06E+7 3.47E+7

The bolded numbers are the best solutions for each function

Table 9 Wilcoxon's test results between GEA and some other algorithms on CEC2017

i	j	MoNR	MoPR	SNR	SPR	F(i) < F(j)	F(j) < F(i)	*p*-value	0.95 Confidence interval	
GEA	BOA	15.42857	3	432	3	28	1	1.86E−08	− 1.3E+08	− 2883.43
	EHO	15.39286	4	431	4	28	1	2.61E−08	− 3.1E+07	− 1390.36
	HHO	17.36364	7.571429	382	53	22	7	0.000156	− 156,314	− 197.614
	BA	15	–	435	0	29	0	3.73E−09	− 1.1E+09	− 42,993.9
	GWO	18.21429	12	255	180	14	15	0.429367	− 117,423	176.1159
	GSA	17.09091	8.428571	376	59	22	7	0.000299	− 41,359.8	− 236.198
	SCA	16.36	6.5	409	26	25	4	3.98E−06	− 1.1E+07	− 556.776

i,j algorithm index, *MoNR* mean of negative rank, *MoPR* mean of positive rank, *SNR* sum of negative rank, *SPR* sum of positive rank, *F(.)*set of objective functions

Table 10 Friedman's test results and ranking original algorithms on CEC2017

GEA	BOA	EHO	HHO	BA	GWO	GSA	SCA
2.21	6.36	5.78	3.40	7.93	2.24	3.64	4.45
1	7	6	3	8	2	4	5

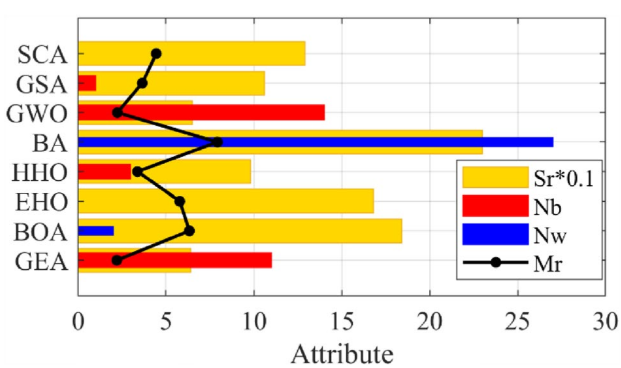

Fig. 12 The statistical analysis of GEA and other original algorithms on CEC 2017 functions

F10 and F12. CLPSO, EPSDE, and OLPSO all produce the same best mean values for the F9 benchmark function, as shown in Table 6.

The proposed new algorithm, GEA (winner in chosen functions of the CEC 2005 Special Session), has attained the best performance and the top rank according to the average (Mr) and (Fr) final ranks. Second place went to the OLPSO and DMS-PSO algorithms. Third and fourth place went to EPSDE and CLPSO, respectively. Although OLPSO and DMS-PSO are marginally superior in several test functions, there is little difference between the suggested new algorithms, GEA. This phenomenon has a multifaceted cause that merits in-depth examination.

4.2.3 Exploration and Exploitation Analysis

To achieve optimal performance, an optimization algorithm must strike a delicate balance between exploration and exploitation, which can be achieved by promoting individual population diversity. Such an approach effectively prevents the algorithm from becoming trapped in local optima and expedites identifying more effective optimization solutions. However, relying solely on suitable exploration and exploitation cannot determine algorithmic superiority when solving optimization problems. Figure 10 provides a visual representation of the changes in exploration and exploitation

Table 11 Evaluation of different algorithms using CEC2014 benchmarks (D = 30 and NFEs = 3.00E+05)

	GEA	MG-SCA	TOGPEAe	ne-cLFA	mTLBO	RCBBOG	NIWTLBO	pe-cLFA	LJA	m-SCA	PPE
F1	**2.94E+5**	2.92E+7	6.54E+6	1.61E+7	6.46E+7	3.77E+6	4.95E+5	1.32E+7	6.31E+7	2.26E+7	1.11E+7
	2.09E+5	2.07E+7	3.49E+6	3.04E+7	4.03E+7	2.16E+6	3.70E+5	3.29E+7	1.87E+7	6.35E+6	–
F2	2.48E+3	9.00E+0	1.79E+7	**2.09E−2**	1.18E+9	9.36E+3	1.78E+2	2.45E−2	4.77E+9	8.11E+7	2.41E+5
	1.27E+3	2.26E+9	2.26E+7	3.54E−2	1.52E+9	1.08E+4	2.49E+2	4.65E−2	6.03E+8	5.59E+7	–
F3	3.09E+1	1.69E+9	5.47E+3	1.23E+1	**4.68E+0**	1.53E+4	2.52E+1	2.19E+1	6.91E+4	2.59E+4	2.82E+3
	1.05E+1	1.00E+1	4.07E+3	1.86E+1	1.73E+1	1.19E+4	5.73E+1	2.56E+1	1.07E+4	6.43E+3	–
F4	**1.27E+1**	1.77E+4	1.44E+2	1.22E+2	3.85E+2	7.70E+1	9.32E+1	1.28E+2	4.08E+2	1.83E+2	1.38E+2
	2.83E+1	6.63E+3	4.37E+1	4.89E+1	1.17E+2	3.37E+1	3.17E+1	5.63E+1	5.38E+1	2.58E+1	–
F5	**2.00E+1**	9.00E+0	2.05E+1	2.00E+1	2.09E+1	2.00E+1	2.09E+1	2.00E+1	2.09E+1	2.09E+1	2.00E+1
	3.70E−2	2.76E+2	3.56E−1	5.99E−4	5.60E−2	4.44E−3	4.68E−2	4.61E−4	4.97E−2	7.07E−2	–
F6	2.07E+1	6.55E+1	2.16E+1	1.85E+1	2.42E+1	2.09E+1	2.88E+1	2.13E+1	3.39E+1	**1.46E+1**	2.50E+1
	2.54E+0	9.00E+0	3.46E+0	3.76E+0	2.27E+0	2.94E+0	3.01E+0	3.27E+0	1.29E+0	3.35E+0	–
F7	4.52E−3	2.04E+1	1.40E+0	6.00E−3	6.00E+1	1.98E−2	3.15E−1	1.04E−2	1.58E+1	2.01E+0	**0.00E+0**
	7.38E−3	1.44E−1	3.12E−1	8.10E−3	2.36E+1	2.75E−2	1.42E+0	9.20E−3	2.80E+0	4.62E−1	–
F8	1.39E+2	1.00E+0	5.89E+1	1.43E+2	1.16E+2	**2.83E+1**	1.42E+2	1.43E+2	2.24E+2	1.10E+2	1.02E+2
	4.09E+1	1.94E+1	1.80E+1	1.61E+1	2.46E+1	7.33E+0	1.96E+1	1.81E+1	9.93E+0	1.24E+1	–
F9	1.63E+2	2.89E+0	7.22E+1	1.67E+2	1.25E+2	**5.54E+1**	1.72E+2	1.70E+2	2.61E+2	1.31E+2	1.60E+2
	3.87E+1	7.00E+0	2.17E+1	2.26E+1	2.89E+1	1.51E+1	2.15E+1	1.99E+1	1.47E+1	9.81E+0	–
F10	3.96E+3	1.99E+1	3.79E+3	2.57E+4	2.87E+3	**1.61E+2**	3.15E+3	2.59E+3	5.68E+3	3.58E+3	2.01E+3
	1.18E+3	1.18E+1	1.91E+3	3.99E+2	6.57E+2	1.22E+2	4.38E+2	4.53E+2	3.95E+2	4.72E+2	–
F11	3.91E+3	1.00E+1	4.53E+3	3.39E+3	3.28E+3	3.22E+3	**3.05E+3**	3.56E+3	6.88E+3	4.93E+3	3.81E+3
	4.27E+2	1.07E+2	1.45E+3	6.61E+2	5.48E+2	5.42E+2	7.21E+2	5.40E+2	3.12E+2	4.71E+2	–
F12	1.65E−1	2.14E+1	7.74E−1	1.36E−1	2.50E+0	2.29E−1	1.93E+0	2.12E−1	2.49E+0	1.76E+0	**0.00E+0**
	1.02E−1	7.00E+0	1.10E+0	8.92E−2	2.66E−1	1.17E−1	6.19E−1	1.40E−1	2.73E−1	2.89E−1	–
F13	3.91E−1	1.39E+2	4.93E−1	4.80E−1	1.77E+0	3.85E−1	5.93E−1	5.16E−1	1.08E+0	3.86E−1	**0.00E+0**
	4.94E−2	2.56E+1	1.15E−1	1.10E−1	9.83E−1	9.87E−2	1.44E−1	1.06E−1	1.19E−1	5.92E−2	–
F14	3.59E−1	8.00E+0	2.63E−1	2.82E−1	2.04E+1	4.77E−1	2.91E−1	2.85E−1	4.33E+0	2.65E−1	**0.00E+0**
	1.39E−1	2.82E+3	5.02E−2	4.10E−2	8.51E+0	2.24E−1	1.67E−1	1.12E−1	1.70E+0	2.94E−2	–
F15	8.64E+0	6.83E+2	2.43E+1	**4.70E+0**	1.20E+3	4.37E+1	1.74E+2	8.38E+0	5.05E+1	1.51E+1	2.00E+1
	4.49E+0	1.00E+0	7.54E+0	1.15E+0	1.46E+3	1.49E+1	2.96E+2	3.90E+0	9.36E+0	1.47E+0	–
F16	1.24E+1	3.30E+3	1.23E+1	1.21E+1	1.11E+1	1.18E+1	1.16E+1	1.25E+1	1.28E+1	1.20E+1	**1.00E+1**
	6.06E−1	6.26E+2	6.54E−1	4.93E−1	7.48E−1	7.54E−1	4.92E−1	3.64E−1	1.78E−1	2.89E−1	–
F17	2.63E+4	3.00E+0	7.12E+4	1.24E+6	6.56E+5	1.13E+6	**2.52E+4**	1.00E+6	2.63E+6	5.99E+5	4.60E+5
	2.13E+4	6.33E−1	5.59E+4	1.94E+6	1.24E+6	7.61E+5	3.08E+4	2.03E+6	9.76E+5	3.59E+5	–
F18	2.74E+3	3.36E−1	3.07E+3	**1.04E+3**	1.92E+4	3.67E+3	2.06E+3	1.42E+3	1.26E+7	1.61E+5	1.24E+3
	4.46E+3	1.00E+0	3.05E+3	6.57E+2	8.10E+4	4.95E+3	2.34E+3	6.56E+2	1.06E+7	8.25E+4	–
F19	**9.92E+0**	5.51E−1	1.60E+1	2.52E+1	8.02E+1	1.49E+1	2.30E+1	2.60E+1	3.78E+1	1.93E+1	2.00E+1
	1.54E+0	8.94E−2	1.80E+1	5.59E+0	3.50E+1	1.11E+1	1.40E+1	3.51E+0	3.46E+1	5.95E+0	–
F20	3.20E+2	8.00E+0	3.31E+3	4.24E+3	**2.72E+2**	3.83E+4	3.75E+2	5.46E+3	9.92E+3	1.22E+4	5.69E+3
	1.15E+2	2.34E+0	3.67E+3	2.73E+3	1.21E+2	1.81E+4	1.56E+2	2.87E+3	3.69E+3	3.70E+3	–
F21	2.45E+4	3.31E+0	1.73E+4	4.72E+4	3.18E+4	4.35E+5	**1.44E+4**	3.10E+4	6.94E+5	1.10E+5	2.11E+5
	1.13E+4	1.00E+1	1.74E+4	5.45E+4	2.86E+4	3.50E+5	8.91E+3	3.82E+4	2.03E+5	5.66E+4	–
F22	5.75E+2	8.72E+1	6.87E+2	7.42E+2	5.32E+2	4.90E+2	6.41E+2	7.88E+2	5.47E+2	**2.57E+2**	6.60E+2
	1.70E+2	1.01E+2	1.65E+2	2.06E+2	2.09E+2	1.97E+2	2.93E+2	2.32E+2	1.05E+2	5.71E+1	–
F23	3.15E+2	9.00E+0	**2.16E+1**	2.68E+2	3.53E+2	3.15E+2	2.00E+2	2.80E+2	3.43E+2	3.21E+2	3.20E+2
	2.88E−13	1.16E+1	3.46E+0	6.56E+1	1.98E+1	1.68E−2	0.00E+0	6.60E+1	3.41E+0	1.58E+0	–
F24	2.40E+2	6.91E−1	3.17E+2	2.02E+2	2.00E+2	2.47E+2	2.00E+2	2.10E+2	2.57E+2	2.00E+2	2.30E+2
	1.01E+1	2.00E+0	1.93E+0	3.51E+0	7.93E−4	4.37E+0	1.99E−5	6.70E+0	4.04E+0	4.29E−2	–
F25	2.06E+2	9.56E+5	2.09E+2	**2.00E+2**	2.04E+2	2.15E+2	2.00E+2	2.00E+2	2.16E+2	2.01E+2	2.10E+2
	4.02E+0	7.62E+5	7.21E+0	1.18E+0	7.51E+0	6.88E+0	0.00E+0	5.27E−13	2.58E+0	3.19E+0	

Table 11 (continued)

	GEA	MG-SCA	TOGPEAe	ne-cLFA	mTLBO	RCBBOG	NIWTLBO	pe-cLFA	LJA	m-SCA	PPE
F26	1.17E+2	1.00E+1	1.04E+2	2.00E+2	1.41E+2	1.01E+2	1.90E+2	2.00E+2	1.01E+2	**1.00E+2**	1.80E+2
	4.07E+1	1.48E+5	1.83E+1	5.90E−3	4.93E+1	1.48E−1	3.03E+1	4.80E−3	1.02E−1	5.40E−2	–
F27	8.53E+2	9.00E+5	9.26E+2	1.90E+3	9.40E+2	6.89E+2	5.24E+2	2.03E+3	9.86E+2	**4.33E+2**	5.80E+2
	1.10E+2	9.00E+0	8.92E+1	9.67E+2	2.48E+2	2.20E+2	2.76E+2	1.06E+3	2.48E+2	2.01E+1	–
F28	1.27E+3	2.28E+1	1.58E+3	2.22E+3	2.36E+3	1.14E+3	1.88E+3	2.56E+3	1.13E+3	1.05E+3	3.99E+3
	1.72E+2	1.43E+1	2.22E+2	9.32E+2	5.34E+2	2.44E+2	3.98E+2	1.24E+3	6.63E+1	2.74E+2	–
F29	1.53E+6	7.00E+0	2.09E+3	1.96E+4	5.45E+6	**1.50E+3**	9.18E+5	6.26E+6	9.82E+5	4.44E+4	7.40E+3
	3.73E+6	4.24E+3	5.16E+2	8.31E+4	1.21E+7	3.29E+2	3.58E+6	3.10E+6	2.07E+6	1.77E+4	–
F30	**2.15E+3**	3.82E+3	1.09E+4	1.08E+4	8.61E+4	4.20E+3	2.93E+3	1.04E+4	1.09E+4	3.48E+4	7.50E+3
	1.06E+3	9.00E+0	6.80E+3	1.47E+4	6.81E+4	4.67E+3	7.84E+2	1.61E+4	4.24E+3	1.05E+4	–

The bolded numbers are the best solutions for each function

Table 12 Wilcoxon's test results between GEA and some other algorithms on CEC2014

i	j	MoNR	MoPR	SNR	SPR	F(i) < F(j)	F(j) < F(i)	*p*-value	0.95 Confidence interval	
GEA	MG-SCA	16	14.75	288	177	18	12	0.262122	− 72,060.1	15.07213
	TOGPEAe	15.85714	14.66667	333	132	21	9	0.038418	− 1662.21	− 0.25632
	ne-cLFA	17	12.53846	272	163	16	13	0.242947	− 4307.15	7.355083
	mTLBO	17.25	12	345	120	20	10	0.019661	− 41,951.9	− 5.8897
	RCBBOG	14.55556	15.72727	262	173	18	11	0.341391	− 7579.15	50.94184
	NIWTLBO	13.125	18.21429	210	255	16	14	0.655438	− 31.4261	407.2985
	pe-cLFA	15.52632	14	295	140	19	10	0.095915	− 3237.66	2.961857
	LJA	15.46154	15.75	402	63	26	4	0.000232	− 334,724	− 22.9169
	m-SCA	16.94118	13.61538	288	177	17	13	0.262122	− 32,651.9	13.76537
	PPE	18.125	11.15385	290	145	16	13	0.119501	− 2687.03	2.390157

Table 13 Friedman's test results and ranking algorithms on CEC2014

GEA	MG-SCA	TOGPEAe	ne-cLFA	mTLBO	RCBBOG	NIWTLBO	pe-cLFA	LJA	m-SCA	PPE
4.70	6.45	5.65	5.45	7.37	4.83	5.18	6.20	9.40	5.80	4.97
1	9	6	5	10	2	4	8	11	7	3

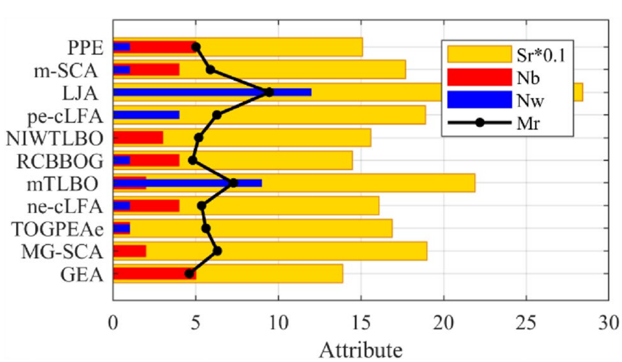

Fig. 13 The statistical analysis of GEA and other algorithms on CEC 2014 functions

throughout iterations of the GEA. With a well-designed exploration and exploitation process, algorithms like GEA can effectively improve using modification methods. The key is maintaining a diverse population that can be optimized through enhanced search methods. These calculations are carried out using the methodology outlined in [52].

The GEA operates under the influence of channel count, which has a marked effect on exploitation and exploration rates. The experiments show that the exploitation percentage increases from 20% when utilizing five channels to 33% when using twenty. This emphasizes the critical role of channel count as a key parameter in the GEA. The methodology for determining the share of exploitation and exploration during the optimization process can be found in Appendix A.

Table 14 Comparison between GEA and RSA on CEC2014 benchmarks (D = 30 and NFEs = 3.00E+05)

Function	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
GEA	**2.94E+05**	**2.48E+03**	**3.09E+01**	**1.27E+01**	**2.00E+01**	2.07E+01	**4.52E-03**	1.39E+02	1.63E+02	3.96E+03
RSA	1.53E+08	7.35E+09	9.67E+03	1.28E+03	2.05E+01	**1.02E+01**	1.05E+02	**7.88E+01**	**6.28E+01**	**1.14E+03**
Function	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20
GEA	3.91E+03	**1.65E-01**	**3.81E-01**	**3.59E-01**	**8.64E+00**	1.24E+01	**2.63E+04**	**2.74E+03**	**9.92E+00**	**3.20E+02**
RSA	**1.47E+03**	1.28E+00	3.92E+00	1.16E+01	4.09E+03	**3.85E+00**	4.99E+05	1.79E+05	1.93E+01	2.49E+04
Function	F21	F22	F23	F24	F25	F26	F27	F28	F29	F30
GEA	**2.45E+04**	5.75E+02	3.15E+02	2.40E+02	2.06E+02	1.17E+02	8.53E+02	1.27E+03	1.53E+06	2.15E+03
RSA	1.83E+06	**1.95E+02**	**2.00E+02**	**2.00E+02**	**2.00E+02**	**1.04E+02**	**2.00E+02**	**2.00E+02**	**2.00E+02**	**2.00E+02**

The bolded numbers are the best solutions for each function

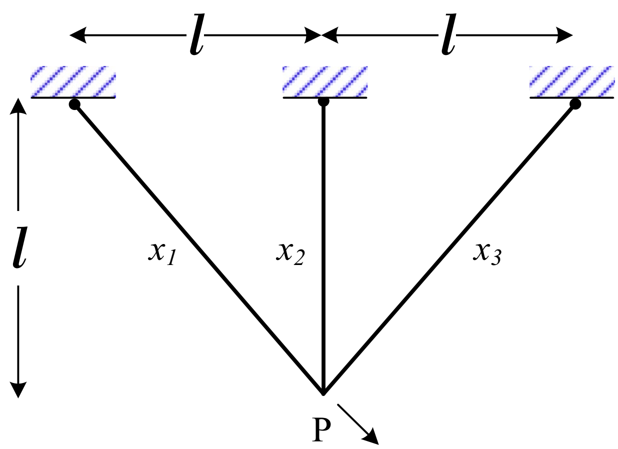

Fig. 14 Schematic of the three-bar truss [72]

Figure 11 displays the average results of the exploration and exploitation process utilized by GEA on the fourteen functions outlined in Table 6. It is essential to note that maintaining a balance between these two stages throughout the optimization process is a critical feature of the GEA. This ensures optimal performance and effectiveness in achieving desired results.

4.2.4 Results and Analysis of Category 3 Experiments: CEC2017 Benchmark Functions

The GEA was tested against other evolutionary methods on various functions, including CEC2017 [53, 54], in a

Table 15 Best statistical results of various algorithms on the three-bar truss problem

Methods	Best	Mean	Worst	Std
RL-BA (an adaptive reinforcement Learning-based Bat Algorithm) [73]	**263.89584**	263.9003	263.924700	6.06e−03
MBA (Mine Blast Algorithm) [2]	263.895852	263.897996	263.915983	3.93e−03
DSS-MDE (DE with Dynamic Stochastic Selection) [74]	**263.8958434**	263.8981518	263.95226	9.19e−03
Society and civilization algorithm [75]	263.8958466	263.90335672	263.96975638	1.3e−02
PSO-DE (Hybridizing PSO with DE) [76]	**263.89584338**	**263.89584338**	**263.89584338**	1.2e−10
CSA (Crow Search Algorithm) [72]	**263.8958433765**	**263.8958433765**	**263.8958433770**	**1.0123e−10**
SFO (Sailfish Optimizer) [77]	263.89592128	N.A	N.A	N.A
SDO (Supply–Demand-based Optimization) [78]	**263.895843**	263.895845	263.895847	1.12e−06
CS (Cuckoo Search) [79]	263.97156	264.0669	N.A	9.0e−05
HEA-ACT (Hybrid Evolutionary Algorithm and Adaptive Constraint-handling Technique) [80]	**263.895843**	263.895865	263.896099	4.9e−05
GOA (Gannet Optimization Algorithm) [81]	**263.895843**	N.A	N.A	N.A
AVOA (African Vultures Optimization Algorithm) [82]	**263.895843**	N.A	N.A	N.A
ABC (this study)	263.895921	263.896174	263.898325	7.7e−05
BBO (this study)	263.896069	264.008712	264.145629	8.6e−02
PSO (this study)	263.896272	263.898854	263.920047	5.1e−03
RCGA (this study)	263.89605	264.293025	264.421497	3.5e−01
GEA (the proposed algorithm)	**263.895843**	263.9261	264.0399	4.74e−02

The bolded numbers are the best solutions for each function

Table 16 The obtained best solutions on the three-bar truss problem

Design variables	ABC (this study)	BBO (this study)	PSO (this study)	RCGA (this study)	GEA (the proposed algorithm)
x_1	0.78883270	0.7882220	0.78857560	0.78909940	0.7886740
x_2	0.40780340	0.40953220	0.40853410	0.40705030	0.40825150
$g_1(X)$	− 4.47047e−07	− 5.597988e−07	− 3.189517e−06	− 5.25825e−07	− 3.091607e−10
$g_2(X)$	− 1.4646077	− 1.4626432	− 1.463778	− 1.465465	− 1.464098
$g_3(X)$	− 0.535393	− 0.537357	− 0.5362249	− 0.534536	− 0.5359020
Best	**263.895921**	**263.896069**	**263.896272**	**263.89605**	**263.895843**

The bolded numbers are the best solutions for each function

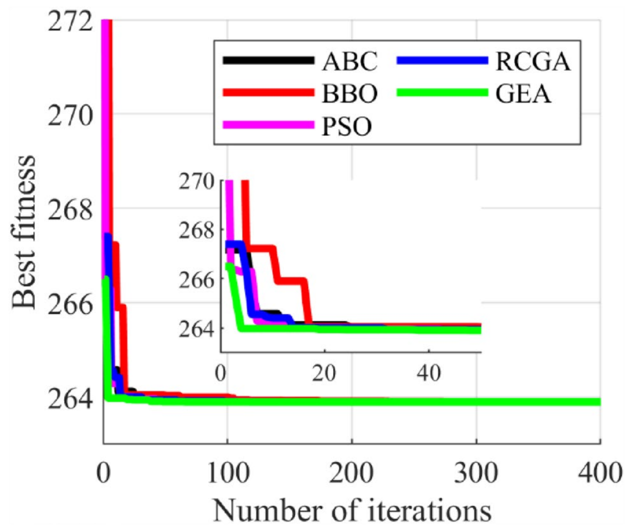

Fig. 15 Convergence characteristic to the best solution (three-bar truss problem)

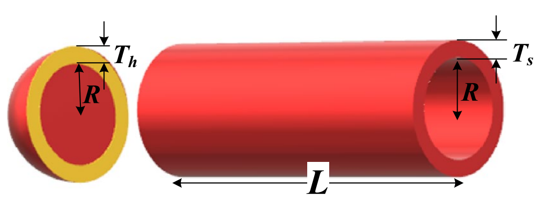

Fig. 16 Schematic of the pressure vessel [72]

30-dimensional space with 30 runs per function. Parameter settings for different algorithms are listed in Table 7, and simulation results for D = 30 are compared in Table 8. The results clearly demonstrate that GEA outperforms other modern algorithms like Grey Wolf Optimizer (GWO) [55], Gravitational Search Algorithm (GSA) [55], Elephant Herding Optimization (EHO) [56], Butterfly Optimization Algorithm (BOA), Sine Cosine Algorithm (SCA) [56], Bat Algorithm (BA) [57], and Harris Hawks Optimization (HHO) [58] on several functions, including F1, F3, F4, …, and F30. This proves that GEA is dependable and robust.

In accordance with the Wilcoxon signed-rank test [27, 59] with a significance level of 0.05, the results of a comparison study between the GEA and various other original algorithms are presented in Table 9. One algorithm is considered significantly better if the p-value is less than the significance level and the 0.95 confidence interval does not include zero. Based on these criteria, GEA outperforms BOA, EHO, HHO, BA, GSA, and SCA. Although the p-value is greater than 0.05 and the confidence interval includes the zero value when comparing GEA with GWO, further analysis of the confidence interval reveals that the preceding interval to zero is wider than the following one. This indicates that GEA provides better solutions than GWO. Table 10 presents the average ranking of all algorithms as per the results of the Friedman test, with a p-value of zero (= 0.000).

The graphical representation of statistical analysis of the solutions obtained in Table 8 is presented in Fig. 12. Sr, which represents the sum of the rank of each algorithm in 29 functions, and Mr, the average of the obtained ranks, indicate that the GEA and GWO algorithms are closely ranked with each other. However, the GEA shows slightly better results in these metrics. Additionally, Nb and Nw, representing the number of times each method achieves the best and worst solutions, demonstrate that the GEA has never produced the worst solution.

4.2.5 Results and Analysis of Category 4 Experiments: CEC2014 Benchmark Functions

Table 11 compares the GEA and ten other algorithms, including nine modified algorithms and one original algorithm, on a total of CEC2014 benchmark functions [54]. The reviewed algorithms in Table 11 are: LJA: Lévy flight Jaya Algorithm [60], NIWTLBO: Nonlinear Inertia Weighted Teaching–Learning-Based Optimization [61, 62], TOGPEAe: enriched Grey Prediction Evolutionary algorithm based on the Topological Opposition-Based Learning [63],

Table 17 Optimal results of the various algorithms on the pressure vessel problem

Methods	Best	Mean	Worst	Std
QPSO (Quantum-behaved PSO) [83]	6059.7209	6440.3786	8017.2816	479.2671
ABC (Artificial Bee Colony algorithm) [84]	6059.714339	6245.308144	N.A	2.05e+02
GA4 (GA with a dominance-based tournament selection) [85]	6059.9463	6177.2533	6469.3220	130.9297
CPSO (Co-evolutionary PSO) [86]	6061.0777	6147.1332	6363.8041	86.4545
CDE (Co-evolutionary DE) [87]	6059.7340	6085.2303	6371.0455	43.013
G-QPSO (Gaussian Quantum-behaved PSO [83]	6059.7208	6440.3786	7544.4925	448.4711
UPSO (Unified PSO) [88]	6154.70	8016.37	9387.77	745.869
CB-ABC (Crossover-Based artificial ABC) [89]	**6059.714335**	6126.623676	N.A	1.14e+02
CSA (Crow Search Algorithm) [72]	6059.71436343	6342.49910551	7332.8416211	384.94541634
EO (Equilibrium Optimizer) [90]	**6059.7143**	6668.114	7544.4925	566.24
HAIS-GA (Hybridizing a GA with an Artificial Immune System) [91]	6832.584	7187.314	8012.615	276
NHAIS-GA (a New Hybrid AIS-GA) [92]	6061.1229	6743.0848	7368.0602	457.99
BFOA (Bacterial Foraging Optimization Algorithm) [93]	6060.460	6074.625	N.A	156
ES (Evolution Strategies) [94]	6059.746	6850.00	7332.87	426
T-Cell (a modified version of a T-Cell algorithm) [95]	6390.554	6737.065	7694.066	357
GA3 (GA with a self-adaptive penalty approach) [96]	6288.7445	6293.8432	6308.4970	7.4133
BA (Bat Algorithm) [97]	**6059.7143348**	6179.13	6318.95	137.223
CVI-PSO (PSOwith a new constraint-handling mechanism) [98]	**6059.7143**	6292.1231	6820.4101	288.4550
PVS (Passing Vehicle Search) [99]	**6059.714**	6065.877	6090.526	N.A
QS (Queuing Search) [100]	**6059.714**	6060.947	6090.526	N.A
BIANCA (the automatic dynamic penalisation method (ADP) with GA) [101]	6059.9384	6182.0022	6447.3251	122.3256
Sinusodial AFA (Adaptive firefly algorithm via sinusodial chaos) [102]	6059.71441707	6061.25542415	6090.52625202	6.88963799
DEC-PSO (Diversity-Enhanced PSO) [103]	**6059.714335**	**6060.33057699**	6090.52620169	4.35745530
DHOA (Deer Hunting Optimization Algorithm) [104]	6103.8420	N.A	N.A	N.A
ABC (this study)	6060.299390	6060.845730	**6061.405879**	**5.2**
BBO (this study)	6059.951290	6182.687452	6311.940267	147.0
PSO (this study)	6059.928702	6063.475504	6070.109476	10.8
RCGA (this study)	6059.8616218	6268.070258	6402.930865	217.6
GEA (the proposed algorithm)	**6059.714335**	6084.727025	6371.338743	86.1

The bolded numbers are the best solutions for each function

Table 18 The obtained best solutions on the pressure vessel problem

Design variables	ABC (This study)	BBO (This study)	PSO (This study)	RCGA (This study)	GEA (The proposed algorithm)
x_1	0.8125	0.8125	0.8125	0.8125	0.8125
x_2	0.4375	0.4375	0.4375	0.4375	0.4375
x_3	42.09758455	42.09741762	42.096868510	42.09821837	42.098445590
x_4	176.66780532	176.65411485	176.65709292	176.6445278	176.63659592
$g_1(X)$	− 1.6618185e−05	− 1.9839934e−05	− 3.0437757e−05	− 4.385459e−06	− 1.13000054e−10
$g_2(X)$	− 0.0358890	− 0.03589063	− 0.035895874	− 0.035883	− 0.03588083
$g_3(X)$	− 1.1435445e+02	− 26.61451377	− 5.308396627	− 28.48582	− 2.78875232e−05
$g_4(X)$	− 63.332195	− 63.34588515	− 63.34290708	− 63.3554722	− 63.36340408
Best	6060.299390	6059.951290	6059.928702	6059.8616218	**6059.714335**

The bolded number is the best solution for each function

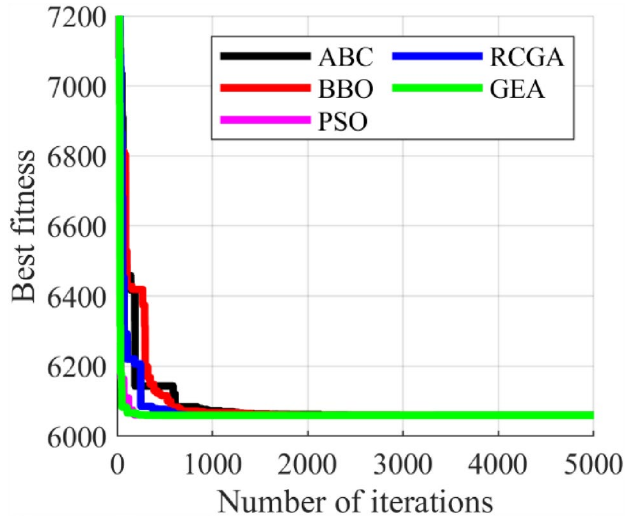

Fig. 17 Convergence characteristic to the best solution (pressure vessel problem)

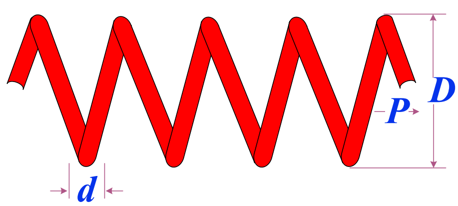

Fig. 18 Schematic of the tension/compression spring

MG-SCA: Memory Guided Sine Cosine Algorithm [64], mTLBO: modified TLBO [62], m-SCA: modified Sine Cosine Algorithm [65], ne-cLFA: non-permanent Elitism-based Compact Lévy Firefly Algorithm [66], RCBBOG: Real Code Biogeography-Based Optimization with Gaussian mutation [67], pe-cLFA: permanent elitism-based Compact LFA [66], PPE: Phasmatodea Population Evolution algorithm [68, 69].

The results of the Wilcoxon signed-rank test in Table 12 indicate that the GEA outperforms algorithms TOGPEAe, mTLBO, and LJA, while its performance is comparable to the other seven algorithms. Moreover, the 0.95 confidence interval preceding zero confirms the superiority of GEA solutions over other algorithms. The Friedman tests' results in Table 13 also confirm the acceptable performance of the proposed algorithm, where GEA has the lowest average rating value. Finally, Fig. 13 provides a graphical representation of the statistical information on the performance of the algorithms on CEC2014 functions.

Table 14 presents the mean values achieved by the GEA and the Reptile Search Algorithm (RSA) [70, 71] on the benchmark functions of CEC2014. A line below its mean value shows the best response for each function. Both algorithms have been successful in 15 functions. However, it is essential to note that the RSA algorithm tends to outperform the GEA in functions where a zero value for variables leads to a suitable local optimum. In contrast, the GEA's results are solely based on searching within the feasible space of the problems.

4.3 GEA for Solving Constrained Engineering Optimization Problems

In this section, various real-world complex engineering problems are deployed to test the GEA and its effectiveness will be confirmed. The population size has been chosen to be 60.

4.3.1 Three-Bar Truss Optimal Design

The primary objective of this task is to decrease the volume of a three-bar truss that is statistically loaded. Figure 14 illustrates this problem, which can be expressed as follows [72–75]:

Minimize:

$$f(\boldsymbol{X}) = 1 \times \left(2\sqrt{2}x_1 + x_2\right). \tag{9}$$

by considering:

$$g_1(\boldsymbol{X}) = \mathrm{P} \times \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} - \sigma \le 0, \tag{10}$$

$$g_2(\boldsymbol{X}) = \mathrm{P} \times \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} - \sigma \le 0, \tag{11}$$

$$g_3(\boldsymbol{X}) = \mathrm{P} \times \frac{1}{\sqrt{2}x_2 + x_1} - \sigma \le 0, \tag{12}$$

$0 \le \boldsymbol{x_i} \le 1, \boldsymbol{i} = 1,2$

$\mathrm{L} = 100$ cm, $\mathrm{P} = 2\mathrm{kN/cm}^2$, $\sigma = 2\mathrm{kN/cm}^2$.

It can be noticed from Table 15 that the proposed GEA outperforms all other methods such as HEAA, CSA, PSO-DE, DSS-MDE, and RL-BA, and by achieving Std. of 3.6e-014, offers the most robust and reliable solution for solving this problem. Besides, Table 16 has a list of the decision variables and restrictions associated with the optimal solutions for this problem. The convergence characteristics to the best solution by GEA has been shown in Fig. 15.

Table 19 Optimal results of the various algorithms on the tension/compression spring problem

Methods	Best	Mean	Worst	Std
EO (Equilibrium Optimizer) [90]	0.012666	0.013017	0.013997	3.91e-04
BFOA (Bacterial Foraging Optimization Algorithm) [93]	0.012671	0.012759	N.A	1.36e-04
T-Cell (a modified version of a T-Cell algorithm) [95]	**0.012665**	0.012732	0.013309	9.4e-05
CDE (Co-evolutionary DE) [87]	0.012670	0.012703	0.012790	2.07e-05
CPSO (Co-evolutionary PSO) [86]	0.0126747	0.012730	0.012924	5.19e-05
SI (swarm with an intelligent information sharing among individuals) [105]	0.013060	0.015526	0.018992	N.A
CA (Cultural Algorithm) [106]	0.012721	0.013568	0.0151156	8.4e-04
HGA (hybrid GA) [107]	0.012668	0.013481	0.016155	N.A
GA4 (GA with a dominance-based tournament selection) [85]	0.012681	0.012742	0.012973	9.5e-05
GA3 (GA with a self-adaptive penalty approach) [96]	0.0127048	0.012769	0.012822	3.93e-05
CSA (Crow Search Algorithm) [72]	**0.0126652328**	**0.0126659984**	0.012670182	**1.35708e-06**
HPSO (A Hybrid PSO with a feasibility-based rule) [108]	**0.0126652**	0.0127072	0.0127190	1.58e-05
TEO (Thermal Exchange Optimization) [19]	**0.012665**	0.012685	0.012715	4.4079e-06
G-QPSO (Gaussian Quantum-behaved PSO) [83]	**0.012665**	0.013524	0.017759	1.268e-03
Society and civilization algorithm [75]	0.012669249	0.012922669	0.016717272	5.92e-04
DSS-MDE (DE with Dynamic Stochastic Selection) [73]	**0.012665233**	0.012669366	0.012738262	1.25e-05
(l+λ)-ES (Evolutionary algorithms) [109]	0.012689	0.013165	N.A	3.9e-04
UPSO (Unified PSO) [88]	0.01312	0.02294	N.A	7.2e-03
GWO (Grey Wolf Optimizer) [110]	0.0126660	N.A	N.A	N.A
DDAO (Dynamic Differential Annealed Optimization) [111]	0.0129065	0.0151829	0.0173199	1.26e-03
PFA (Pathfinder Algorithm) [112]	**0.01266528**	N.A	N.A	N.A
SDO (Supply–Demand-based Optimization) [78]	0.0126663	0.0126724	0.0126828	6.1899e-06
QS (Queuing Search) [100]	**0.012665**	**0.012666**	**0.012669**	N.A
BA (Bat Algorithm) [97]	**0.01266522**	0.01350052	0.0168954	1.42027e-03
CVI-PSO (PSO with a new constraint-handling mechanism) [98]	0.0126655	0.012731	0.0128426	5.58e-05
WCA (Water Cycle Algorithm) [113]	**0.012665**	0.012746	0.012952	8.06e-05
BIANCA (the automatic dynamic penalisation method (ADP) with genetic algorithms) [101]	0.012671	0.012681	0.012913	5.1232e-05
Sinusodial AFA (Adaptive FireFly Algorithm via Sinusodial chaos) [102]	0.0126660517	0.0127072216	0.0127680353	2.57941e-05
AVOA (African Vultures Optimization Algorithm) [82]	0.01266524	N.A	N.A	N.A
MRFO (Manta Ray Foraging Optimization) [114]	0.0126813	0.0127007	0.0131811	2.1378E−04
COA (Coati Optimization Algorithm) [115]	0.012666	0.012688	0.012697	1e-3
ABC (this study)	**0.012665**	0.012681	0.012736	3.1e-03
BBO (this study)	0.012692	0.012779	0.013324	1.9e-02
PSO (this study)	0.012666	0.012674	0.012805	5.8e-03
RCGA (this study)	0.01270524	0.013617	0.0169325	1.15e-03
GEA (the proposed algorithm)	**0.012665**	0.012677	0.012685	6.41E-06

The bolded numbers are the best solutions for each function

4.3.2 Pressure Vessel Optimal Design

The main aim of this problem is to minimize the entire cost of a pressure vessel, as shown in Fig. 16, in which x_1 or T_s is the thickness of the shell, x_2 or T_h is the thickness of the head, x_3 or R is the inner radius and x_4 or L is the length of the cylindrical section of the vessel. This problem can be expressed as follows [72–75]:

Minimize:

$$f(\boldsymbol{X}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \tag{13}$$

by considering:

$$g_1(\boldsymbol{X}) = -x_1 + 0.0193x_3 \le 0 \tag{14}$$

$$g_2(\boldsymbol{X}) = -x_2 + 0.00954x_3 \le 0 \tag{15}$$

Table 20 The obtained best solutions on the tension/compression spring problem

Design variables	ABC (This study)	BBO (This study)	PSO (This study)	RCGA (This study)	GEA (The proposed algorithm)
x_1	0.05168300	0.05098910830	0.05167699740	0.050948890	0.05170431680
x_2	0.356570500	0.3400442159	0.35641991890	0.338689336	0.357084692800
x_3	11.29767450	12.35650429870	11.30709115480	12.451465788	11.26749115160
$g_1(X)$	− 1.667e−06	− 0.00129	− 3.22425e−05	− 1.2075e−04	− 1.43239e−08
$g_2(X)$	− 0.8393	− 1.56573e−04	− 0.8394	− 0.001132	− 3.8394e−07
$g_3(X)$	− 4.05347	− 4.0123	− 4.052941	− 4.010	− 4.05451
$g_4(X)$	− 0.72783	− 0.73931	− 0.72794	− 0.74024	− 0.727474
Best	**0.012665**	0.012692	0.012666	0.01270524	**0.012665**

The bolded numbers are the best solutions for each function

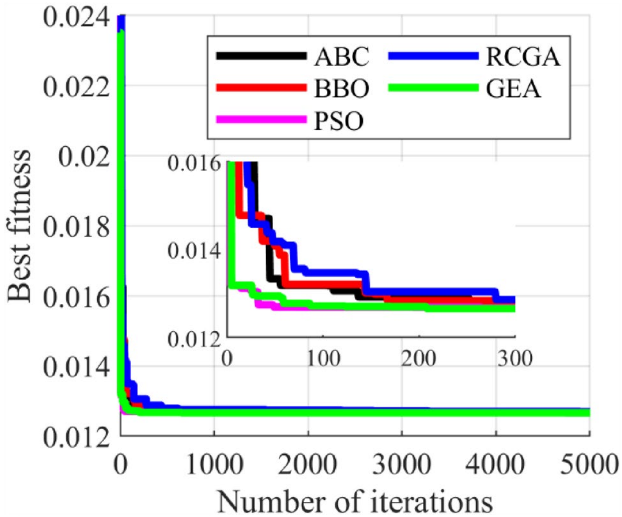

Fig. 19 Convergence characteristic to the best solution (tension/compression spring problem)

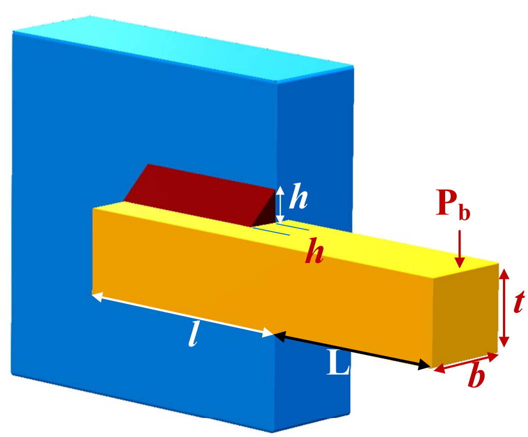

Fig. 20 Schematic of the welded beam [72]

$$g_3(\boldsymbol{X}) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0 \quad (16)$$

$$g_4(\boldsymbol{X}) = x_4 - 240 \leq 0 \quad (17)$$

$0 \leq x_i \leq 100, i = 1, 2$
$10 \leq x_i \leq 200, i = 3, 4$

It is clear from Table 17 that the fitness function value of 6059.714335 has been achieved by the proposed GEA. Moreover, among all algorithms, the GEA suggests the most reliable and robust method for solving this optimization problem. Moreover, Table 18 has a list of the decision variables and restrictions associated with the optimal solutions for this problem. Figure 17 shows the convergence characteristics to the best solution of the problem.

4.3.3 Tension/Compression Spring Optimal Design

The main goal of this design is to optimize the weight of a tension/compression spring by considering the material, shaping and welding costs, as shown in Fig. 18, in which d or x_1 is the wire diameter, D or x_2 is the mean coil diameter, and P or x_3 is the number of active coils. This problem can be expressed as follows [72–75]:

Minimize:

$$f(X) = \left(x_3 + 2\right)x_2 x_1^2 \quad (18)$$

by considering:

$$g_1(\boldsymbol{X}) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0, \quad (19)$$

$$g_2(\boldsymbol{X}) = \frac{4x_2^2 - x_1 x_2}{12566\left(x_1^3 x_2 - x_1^4\right)} + \frac{1}{5108 x_1^2} - 1 \leq 0, \quad (20)$$

Table 21 Optimal results of the various algorithms on the welded beam problem

Methods	Best	Mean	Worst	Std
MRFO (Manta Ray Foraging Optimization) [114]	**1.7248523**	**1.7248547**	**1.7248648**	**3.832e-06**
EPSO (cooperative PSO with stochastic movements) [116]	1.7248530	1.7282190	1.7472200	5.62e-03
EO (Equilibrium Optimizer) [90]	1.724853	1.726482	1.736725	3.257e-03
BFOA (Bacterial Foraging Optimization Algorithm) [93]	2.3868	2.4040	N.A	1.6e-02
T-Cell (new version of a T-Cell algorithm) [95]	2.3811	2.4398	2.7104	9.314e-02
CDE (Co-evolutionary DE) [87]	1.73346	1.768158	1.824105	2.2194e-02
CPSO (Co-evolutionary PSO) [85]	1.728024	1.748831	1.782143	1.2926e-02
SBM (Socio-Behavioural simulation Model) [117]	2.4426	2.5215	2.6315	N.A
HSA-GA (hybrid real-parameter GA) [118]	2.2500	2.26	2.28	7.8e-03
FSA (Derivative-free Filter Simulated Annealing) [119]	2.3811	2.4041	2.4889	N.A
RAER (multiagent evolutionary algorithm) [120]	2.3816	N.A	2.38297	3.4e-04
TEO (Thermal Exchange Optimization) [19]	1.725284	1.768040	1.931161	5.81661e-02
IPSO (new Improved PSO) [121]	2.3810	2.3819	N.A	5.23e-03
HPSO (Hybrid PSO with a feasibility-based rule) [108]	**1.724852**	1.749040	1.814295	4.01e-02
Society and civilization algorithm [75]	2.3854347	3.0025883	6.3996785	9.59e-01
GA4 (GA with a dominance-based tournament selection) [85]	1.728226	1.792654	1.993408	7.47e-02
(1+λ)-ES (Evolutionary algorithms) [109]	**1.724852**	1.777692	N.A	8.8e-02
UPSO (Unified PSO) [86]	1.92199	2.83721	N.A	6.83e-01
GWO (Grey Wolf Optimizer) [110]	1.72624	N.A	N.A	N.A
SFO (Sailfish Optimizer) [76]	1.73231	N.A	N.A	N.A
HGSO (Henry Gas Solubility Optimization) [24]	1.7260	1.7265	1.7325	7.66e-03
PFA (Pathfinder Algorithm) [112]	1.7248530	N.A	N.A	N.A
BA (Bat Algorithm) [97]	1.7312065	1.8786560	2.3455793	2.677989e-01
CVI-PSO (PSO with a new constraint handling mechanism) [98]	**1.724852**	1.725124	1.727665	6.12e-04
WCA (Water Cycle Algorithm) [113]	1.724856	1.726427	1.744697	4.29e-03
BIANCA (the automatic dynamic penalisation method (ADP) with GA) [101]	1.725436	1.752201	1.793233	2.3001e-02
Sinusodial AFA (Adaptive firefly algorithm via Sinusodial chaos) [102]	1.724868	1.724953	1.725022	4.6e-05
GOA (Gannet Optimization Algorithm) [81]	1.7249	N.A	N.A	N.A
AVOA (African Vultures Optimization Algorithm) [82]	1.724852	N.A	N.A	N.A
NGO (Northern goshawk optimization) [122]	1.725202	1.725312	1.725496	1.06e-5
ABC (this study)	1.726256	1.741024	1.750081	8.4e-04
BBO (this study)	1.733720	1.756059	1.912450	4.6e-02
PSO (this study)	1.727792	1.746645	1.801599	2.7e-03
RCGA (this study)	1.7324625	1.774069	1.935528	8.5e-02
GEA (the proposed algorithm)	**1.724852**	1.728164	1.733048	3e-3

The bolded numbers are the best solutions for each function

$$g_3(\boldsymbol{X}) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0, \tag{21}$$

$$g_4(\boldsymbol{X}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0. \tag{22}$$

$$0.05 \leq x_1 \leq 2, 0.25 \leq x_2 \leq 1.3, 2 \leq x_3 \leq 15.$$

It is evident from Table 19 that the fitness function value of 0.012665 has been achieved by the proposed GEA. Moreover, among all algorithms, the GEA recommends the most robust and reliable way for solving this optimal design. Additionally, Table 20 has a list of the decision variables and constraints associated with the optimal solutions for this problem; also the convergence curve is depicted in Fig. 19.

4.3.4 Welded Beam Optimal Design

The main aim of this problem is to optimize the cost of a welded beam. The welded beam optimal design has four continuous decision variables named x_1 or h, x_2 or l, x_3

Table 22 The obtained best solutions on the welded beam problem

Design variables	ABC (This study)	BBO (This study)	PSO (This study)	RCGA (This study)	GEA (The proposed algorithm)
x_1	0.20572123540	0.203201423200	0.20540621750	0.2046984619	0.20572963980
x_2	3.47311219040	3.52724440950	3.48030403980	3.4938096593	3.47048866550
x_3	9.03134815750	9.04248086640	9.0320571350	9.0704911556	9.0366239101
x_4	0.20598932760	0.20627399270	0.20611194220	0.2057562799	0.20572963980
$g_1(X)$	− 1.2814562	− 11.6577605	− 3.35797069	− 43.69436	− 2.2653330e−07
$g_2(X)$	− 2.804932	− 1.17917222e+02	− 25.3563746	− 2.27464e+02	− 3.1932723e−07
$g_3(X)$	− 2.680922e−04	− 0.0030726	− 7.057247e−04	− 0.001058	0.0
$g_4(X)$	− 3.4316903	− 3.4228490	− 3.430006	− 3.4249	− 3.432984
$g_5(X)$	− 0.0807212	− 0.0782014	− 0.080406	− 0.079698	− 0.080730
$g_6(X)$	− 0.2355332	− 0.2356065	− 0.2355452	− 0.235704	− 0.23554
$g_7(X)$	− 20.437234	− 50.329657	− 31.505976	− 17.09198	− 1.105493e−06
Best	1.726256	1.733720	1.727792	1.7324625	**1.724852**

The bolded number is the best solution for each function

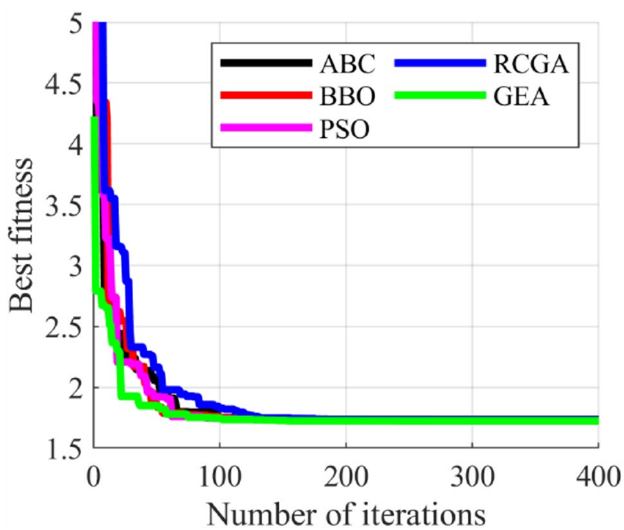

Fig. 21 Convergence characteristic to the best solution (welded beam problem)

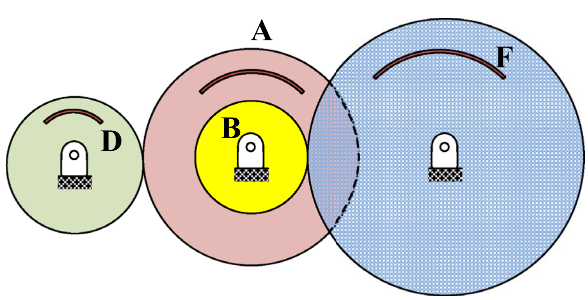

Fig. 22 Schematic of the gear train [72]

or t and x_4 or b, as shown in Fig. 20, the problem can be expressed as follows [72–75]:

Minimize:

$$f(\boldsymbol{X}) = 1.10471x_2x_1^2 + 0.04811x_3x_4\left(14 + x_2\right) \tag{23}$$

by considering:

$$g_1(\boldsymbol{X}) = \tau(x) - \tau_{\max} \tag{24}$$

$$g_2(\boldsymbol{X}) = \sigma(x) - \sigma_{\max} \tag{25}$$

$$g_3(\boldsymbol{X}) = x_1 - x_4 \leq 0 \tag{26}$$

$$g_4(\boldsymbol{X}) = 0.10471x_1^2 + 0.04811x_3x_4\left(14 + x_2\right) - 5 \leq 0 \tag{27}$$

$$g_5(\boldsymbol{X}) = 0.125 - x_1 \leq 0, \tag{28}$$

$$g_6(\boldsymbol{X}) = \delta(x) - \delta_{\max} \tag{29}$$

$$g_7(\boldsymbol{X}) = \mathrm{P} - P_c(x) \leq 0, \tag{30}$$

where

$$\tau(x) = \sqrt{(\tau\prime)^2 + 2\tau\prime\tau^{\varepsilon}\frac{x_2}{2R} + (\tau^{\varepsilon})^2} \tag{31}$$

$$\tau\prime = \frac{\mathrm{P}}{\sqrt{2}x_1x_2}, \tag{32}$$

$$\tau^{\varepsilon} = \frac{MR}{J}, \tag{33}$$

Table 23 Optimal results of the various algorithms on the gear train problem

Methods	Best	Mean	Worst	Std
CS (Cuckoo Search) [79]	**2.7009e−12**	1.9841e−9	2.3576e−9	3.5546e−9
TOGPEAe [63]	**2.700e−12**	6.300e−10	2.400e−9	8.600e−10
UPSO (Unified PSO) [88]	**2.700857e−12**	3.80562e−8	N.A	1.09e−07
MBA (Mine Blast Algorithm) [2]	**2.700857e−12**	2.471635e−9	2.06290e−8	3.94e−09
ABC (this study)	**2.700857e−12**	**5.313982e−11**	**2.828166e−10**	1.4e−09
BBO (this study)	2.307816e−11	7.099246e−8	1.990372e−7	8.2e−07
PSO (this study)	**2.700857e−12**	2.114953e−8	4.725086e−8	5.7e−08
RCGA (this study)	2.768925e−12	1.2453314e−6	5.802142e−6	4.3e−06
GEA (the proposed algorithm)	**2.700857e−12**	3.392666e−10	9.92E−10	**4.90E−10**

The bolded numbers are the best solutions for each function

Table 24 The obtained best solutions on the gear train problem

Design variables	ABC (This study)	BBO (This study)	PSO (This study)	RCGA (This study)	GEA (The proposed algorithm)
x_1	43	51	43	43	43
x_2	19	15	19	19	16
x_3	16	26	16	16	19
x_4	49	53	49	49	49
Best	**2.700857e−12**	**2.307816e−11**	**2.700857e−12**	**2.700857e−12**	**2.700857e−12**

The bolded numbers are the best solutions for each function

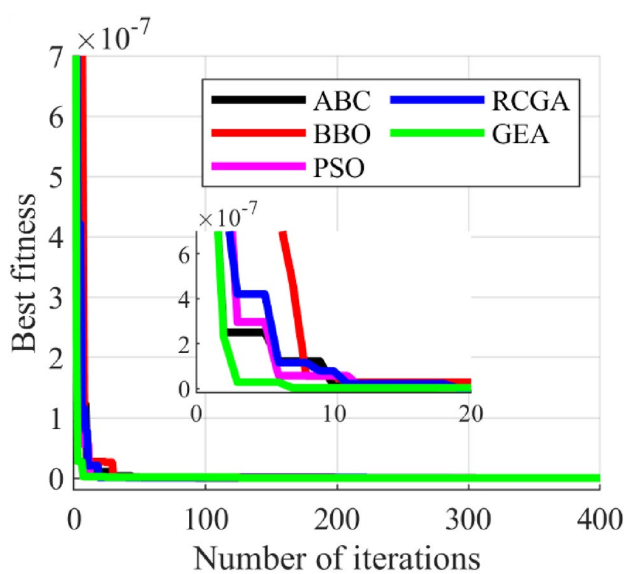

Fig. 23 Convergence characteristic by algorithms (gear train problem)

$$M = \mathrm{P}\left(\mathrm{L} + \frac{x_2}{2}\right), \tag{34}$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}, \tag{35}$$

$$\delta(x) = \frac{4PL^3}{\mathrm{E}x_3^3 x_4} \tag{36}$$

$$J = 2\left[\sqrt{2}x_1x_2\left\{\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right\}\right], \tag{37}$$

$$\sigma(x) = \frac{6\mathrm{PL}}{x_4 x_3^2} \tag{38}$$

$$P_c(x) = \frac{4.013\mathrm{E}\sqrt{\frac{x_4^6 x_3^2}{36}}}{\mathrm{L}^2}\left(1 - \frac{x_3}{2\mathrm{L}}\sqrt{\frac{\mathrm{E}}{4\mathrm{G}}}\right) \tag{39}$$

P = 6000 lb; L = 14 in; E = 30e6 psi G = 12e6 psi, τ_{max} =13,000 psi, σ_{max}=30,000 psi.

δ_{max}=0.25 in, $0.1 \le x_1 \le 2, 0.1 \le x_2 \le 10, 0.1 \le x_3 \le 10$,$0.1 \le x_4 \le 2$.

It is obvious from Table 21 that the fitness function value of 1.724852 has been attained by the proposed GEA. Besides, among all algorithms, the GEA proposes the most robust and reliable process for solving the optimization problem. In addition, the decision variables associated with the optimal solutions have been shown in Table 22, also the convergence characteristic is depicted in Fig. 21.

4.3.5 Gear Train Optimal Design

The schematic of this problem is shown in Fig. 22. This problem's major goal is to reduce the gear train's cost per gear ratio. Only the parameters' boundaries are constrained when solving for the optimal design of a gear train. The decision variables, $\boldsymbol{n_A}$ $(\boldsymbol{x_1})$, $\boldsymbol{n_B}$ $(\boldsymbol{x_2})$, $\boldsymbol{n_D}$ $(\boldsymbol{x_3})$ and $\boldsymbol{n_F}$ $(\boldsymbol{x_4})$, are in discrete. This problem has been defined as follows [72–75]:

Minimize:

$$f(\boldsymbol{X}) = \left(\left(\frac{1}{6.931} \right) - \left(\frac{x_2 x_3}{x_1 x_4} \right) \right)^2 \tag{40}$$

$12 \le x_i \le 60,\ i = 1, 2, 3, 4.$

It can be observed from Table 23 that the fitness function value of 2.700857e-12 has been achieved by the proposed GEA. In addition, among all algorithms, the GEA offers the most robust and reliable method for solving the optimization problem. The best solutions for this problem are listed in Table 24. Figure 23 shows the convergence characteristics of algorithms for the problem.

The result section compared the GEA's performance to that of other well-known evolutionary algorithms using three distinct categories of test functions. To begin, the GEA was compared to several novel evolutionary algorithms that were shown to be capable of dealing with these methodologies, namely ABC, BBO, PSO, and RCGA. The efficacy of GEA was then compared to that of certain modified evolutionary algorithms. This section demonstrated the advantage of GEA in terms of exploration and exploitation, as determined by two techniques of population update. Finally, the performance of GEA was evaluated and compared to that of other techniques on constrained optimization problems. Additionally, the final section validated the usefulness of GEA's convergence speed and ability to cover the entire search field.

5 Conclusion

In this paper, a new meta-heuristic algorithm inspired by an unusual geological phenomenon in nature, called Geyser Algorithm (GEA), is proposed. The modeling process of the algorithm is designed to consider the distinctive traits of geyser formation to inspire and guide researchers in exploring the modeling of various other natural occurrences. To assess the performance of the GEA compared to different original and improved algorithms, standard benchmark functions, namely CEC2005, CEC2014, and CEC2017, were evaluated. Upon analyzing the graphical representations of the exploitation and exploration procedures within the GEA, it is observed that a harmonious balance between these two processes is maintained during the optimization iterations. GEA has been rigorously tested on real-parameter engineering optimization problems and has proven to be effective and practical for complex optimization challenges. Four algorithms, namely ABC, BBO, PSO, and RCGA, were evaluated under the same conditions as GEA for a comprehensive analysis.

The GEA has demonstrated its efficacy in situations where its implementation is straightforward, requiring only the adjustment of two parameters: the initial population size and the number of channels. The latter parameter governs the algorithm's local search power, and its manipulation can sustain population diversity throughout the optimization process. The GEA's inherent simplicity and efficient search capabilities render it susceptible to commonly employed modification techniques, thereby enhancing its capacity to tackle intricate problems. In considering future research directions, exploring the adaptation of the GEA to address multi-objective optimization problems is beneficial [26, 123]. Additionally, presenting a binary version of the algorithm may be worthwhile.

Appendix A: Portion of Exploration and Exploitation

The following equations should be implemented to determine the proportion of exploitation and exploration processes within evolutionary algorithms and the level of population diversity [52].

Suppose the size of the initial population is Npop and the number of decision variables is D:

$$\boldsymbol{X_j} = [x_{1,j}, x_{2,j}, \ldots, x_{Npop,j}] \forall \mathrm{j} = 1, 2, \ldots, \mathrm{D} \tag{41}$$

$$K_j = median(\boldsymbol{X_j}) \forall \mathrm{j} = 1, 2, \ldots, \mathrm{D} \tag{42}$$

$\boldsymbol{X_j}$ is a vector that contains all the jth variables of all population. Now the following two parameters can be calculated.

$$dv_j = \sum\nolimits_{i=1}^{Npop} K_j - x_{i,j} \forall \mathrm{j} = 1, 2, \ldots, \mathrm{D} \tag{43}$$

$$DV = \frac{1}{D} \sum\nolimits_{j=1}^{D} dv_j \tag{44}$$

$x_{i,j}$ is the jth variable of ith population. DV is the average of diversity.

$$\%Pl = \frac{DV}{DV\_max} \times 100 \tag{45}$$

$$\%Pt = \frac{|DV - DV\_max|}{DV\_max} \times 100 \tag{46}$$

DV is the population diversity in each iteration and DV_max is the maximum diversity between all iterations. %Pl and %Pt are the percentage of exploration and exploitation corresponding to each iteration.

Data availability Data is 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. Ertenlice, O., & Kalayci, C. B. (2018). A survey of swarm intelligence for portfolio optimization: Algorithms and applications. *Swarm and Evolutionary Computation, 39*, 36–52. https://doi.org/10.1016/j.swevo.2018.01.009
2. Sadollah, A., Bahreininejad, A., Eskandar, H., & Hamdi, M. (2013). Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Applied Soft Computing, 13*(5), 2592–2612. https://doi.org/10.1016/j.asoc.2012.11.026
3. Yu, K., Zhang, D., Liang, J., Chen, K., Yue, C., Qiao, K., & Wang, L. (2022). A correlation-guided layered prediction approach for evolutionary dynamic multiobjective optimization. *IEEE Transactions on Evolutionary Computation*. https://doi.org/10.1109/TEVC.2022.3193287
4. Yu, K., Liang, J., Qu, B., Luo, Y., & Yue, C. (2022). Dynamic selection preference-assisted constrained multiobjective differential evolution. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 52*(5), 2954–2965. https://doi.org/10.1109/TSMC.2021.3061698
5. Holland, J. H. (1992). Genetic algorithms. *Scientific American, 267*(1), 66–73.
6. Kirkpatrick, S., Gelatt, C. D., Jr., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671–680. https://doi.org/10.1126/science.220.4598.671
7. 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*. IEEE, Nagoya, Japan, pp 39–43.
8. Dorigo, M., & Di Caro, G. (1999). Ant colony optimization: a new meta-heuristic. In: *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)*. IEEE, Washington, DC, USA, pp 1470–1477.
9. Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization, 39*(3), 459–471. https://doi.org/10.1007/s10898-007-9149-x
10. Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences, 179*(13), 2232–2248. https://doi.org/10.1016/j.ins.2009.03.004
11. Simon, D. (2008). Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation, 12*(6), 702–713. https://doi.org/10.1109/TEVC.2008.919004
12. Alatas, B. (2011). ACROA: Artificial chemical reaction optimization algorithm for global optimization. *Expert Systems with Applications, 38*(10), 13170–13180. https://doi.org/10.1016/j.eswa.2011.04.126
13. Patel, V. K., & Savsani, V. J. (2015). Heat transfer search (HTS): A novel optimization algorithm. *Information Sciences, 324*, 217–246. https://doi.org/10.1016/j.ins.2015.06.044
14. Kaveh, A., & Bakhshpoori, T. (2016). Water evaporation optimization: A novel physically inspired optimization algorithm. *Computers & Structures, 167*, 69–85. https://doi.org/10.1016/j.compstruc.2016.01.008
15. Abedinpourshotorban, H., Shamsuddin, S. M., Beheshti, Z., & Jawawi, D. N. A. (2016). Electromagnetic field optimization: A physics-inspired metaheuristic optimization algorithm. *Swarm and Evolutionary Computation, 26*, 8–22. https://doi.org/10.1016/j.swevo.2015.07.002
16. Punnathanam, V., & Kotecha, P. (2016). Yin-Yang-pair optimization: A novel lightweight optimization algorithm. *Engineering Applications of Artificial Intelligence, 54*, 62–79. https://doi.org/10.1016/j.engappai.2016.04.004
17. Salmani, M. H., & Eshghi, K. (2017). A metaheuristic algorithm based on chemotherapy science: CSA. *Journal of Optimization*. https://doi.org/10.1155/2017/3082024
18. Kaboli, S. H. A., Selvaraj, J., & Rahim, N. A. (2017). Rainfall optimization algorithm: A population based algorithm for solving constrained optimization problems. *Journal of Computational Science, 19*, 31–42. https://doi.org/10.1016/j.jocs.2016.12.010
19. Kaveh, A., & Dadras, A. (2017). A novel meta-heuristic optimization algorithm: Thermal exchange optimization. *Advances in Engineering Software, 110*, 69–84. https://doi.org/10.1016/j.advengsoft.2017.03.014
20. Hudaib, A. A., & Fakhouri, H. N. (2018). Supernova optimizer: A novel natural inspired meta-heuristic. *Modern Applied Science, 12*(1), 32–50. https://doi.org/10.5539/mas.v12n1p32
21. Ghasemi, M., Ghavidel, S., Aghaei, J., Akbari, E., & Li, L. (2018). CFA optimizer: a new and powerful algorithm inspired by Franklin's and Coulomb's laws theory for solving the economic load dispatch problems. *International Transactions on Electrical Energy Systems, 28*(5), e2536. https://doi.org/10.1002/etep.2536
22. Ghasemi, M., Zare, M., Zahedi, A., Hemmati, R., Abualigah, L., & Forestiero, A., A comparative study of the coulomb's and franklin's laws inspired algorithm (cfa) with modern evolutionary algorithms for numerical optimization. In: Comito, C., Talia, D. (eds) *Pervasive Knowledge and Collective Intelligence on Web and Social Media. PerSOM 2022. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol 494. Springer, Cham. Messina, Italy, 2022, pp 111–124. https://doi.org/10.1007/978-3-031-31469-8_8
23. Zhao, W., Wang, L., & Zhang, Z. (2019). A novel atom search optimization for dispersion coefficient estimation in groundwater. *Future Generation Computer Systems, 91*, 601–610. https://doi.org/10.1016/j.future.2018.05.037

24. Hashim, F. A., Houssein, E. H., Mabrouk, M. S., Al-Atabany, W., & Mirjalili, S. (2019). Henry gas solubility optimization: A novel physics-based algorithm. *Future Generation Computer Systems, 101*, 646–667. https://doi.org/10.1016/j.future.2019.07.015
25. Akbari, M. A., Zare, M., Azizipanah-abarghooee, R., Mirjalili, S., & Deriche, M. (2022). The cheetah optimizer: A nature-inspired metaheuristic algorithm for large-scale optimization problems. *Scientific Reports, 12*(1), 10953. https://doi.org/10.1038/s41598-022-14338-z
26. Liang, J., Qiao, K., Yu, K., Qu, B., Yue, C., Guo, W., & Wang, L. (2022). Utilizing the relationship between unconstrained and constrained pareto fronts for constrained multiobjective optimization. *IEEE Transactions on Cybernetics, 53*(6), 3873–3886. https://doi.org/10.1109/TCYB.2022.3163759
27. Zare, M., Ghasemi, M., Zahedi, A., Golalipour, K., Mohammadi, S. K., Mirjalili, S., & Abualigah, L. (2023). A Global Best-guided Firefly Algorithm for Engineering Problems. *Journal of Bionic Engineering*. https://doi.org/10.1007/s42235-023-00386-2
28. Yuan, Y., Shen, Q., Wang, S., Ren, J., Yang, D., Yang, Q., Fan, J., & Mu, X. (2023). Coronavirus mask protection algorithm: A new bio-inspired optimization algorithm and its applications. *Journal of Bionic Engineering, 20*(4), 1747–1765. https://doi.org/10.1007/s42235-023-00359-5
29. Abualigah, L., Habash, M., Hanandeh, E. S., Hussein, A. M., Shinwan, M. A., Zitar, R. A., & Jia, H. (2023). Improved reptile search algorithm by salp swarm algorithm for medical image segmentation. *Journal of Bionic Engineering, 20*(4), 1766–1790. https://doi.org/10.1007/s42235-023-00332-2
30. Wang, C.-Y., & Manga, M. G. (2021). *Water and earthquakes* (pp. 289–299). Cham: Springer International Publishing.
31. Rinehart, J.S. (1980). The geologic, thermal, and hydrologic state of the earth. In: *Geysers and geothermal energy*. Springer-Verlag, New York, pp.24–48.
32. Rinehart, J.S. (1980). Fundamentals of geyser operation. In: *Geysers and geothermal energy*, Springer-Verlag, New York, pp.49–77.
33. Rinehart, J.S. (1980). Practical uses of geothermal fluids. In: *Geysers and geothermal energy*, Springer-Verlag, New York, pp.175–204.
34. Chu, S.-C., Tsai, P.-W., & Pan, J.-S. (2006). Cat swarm optimization. In: *PRICAI 2006: Trends in Artificial Intelligence: 9th Pacific Rim International Conference on Artificial Intelligence*, Guilin, China, August 7–11, Proceedings 9. Springer, Cham, pp 854–858.
35. Eusuff, M., Lansey, K., & Pasha, F. (2006). Shuffled frog-leaping algorithm: A memetic meta-heuristic for discrete optimization. *Engineering Optimization, 38*(2), 129–154.
36. Niknam, T., Zare, M., Aghaei, J., & Farsani, E. A. (2011). A new hybrid evolutionary optimization algorithm for distribution feeder reconfiguration. *Applied Artificial Intelligence, 25*(10), 951–971.
37. Suganthan, P. N., Hansen, N., Liang, J. J., Deb, K., Chen, Y.-P., Auger, A., & Tiwari, S. (2005). Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. *KanGAL Report, 2005005*(2005), 2005.
38. Chen, D., Zou, F., Lu, R., & Wang, P. (2017). Learning backtracking search optimisation algorithm and its application. *Information Sciences, 376*, 71–94. https://doi.org/10.1016/j.ins.2016.10.002
39. Ono, I., Kita, H., & Kobayashi, S. (2003). A real-coded genetic algorithm using the unimodal normal distribution crossover. In: Ghosh, A., Tsutsui, S. (eds) *Advances in evolutionary computing*. Natural Computing Series. Springer, Berlin, pp. 213–237. https://doi.org/10.1007/978-3-642-18965-4_8
40. Wang, Y., Cai, Z., & Zhang, Q. (2011). Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Transactions on Evolutionary Computation, 15*(1), 55–66.
41. García-Martínez, C., Lozano, M., Herrera, F., Molina, D., & Sánchez, A. M. (2008). Global and local real-coded genetic algorithms based on parent-centric crossover operators. *European Journal of Operational Research, 185*(3), 1088–1113.
42. Liang, J. J., Qin, A. K., Suganthan, P. N., & Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation, 10*(3), 281–295.
43. Du, Y., Fan, Y., Liu, X., Luo, Y., Tang, J., & Liu, P. (2019). Multiscale cooperative differential evolution algorithm. *Computational Intelligence and Neuroscience*. https://doi.org/10.1155/2019/5259129
44. Mallipeddi, R., Suganthan, P. N., Pan, Q.-K., & Tasgetiren, M. F. (2011). Differential evolution algorithm with ensemble of parameters and mutation strategies. *Applied Soft Computing, 11*(2), 1679–1696.
45. 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)*. IEEE, Vancouver, BC, pp.4061–4068. https://doi.org/10.1109/CEC.2016.7744305
46. Brest, J., Greiner, S., Boskovic, B., Mernik, M., & Zumer, V. (2006). Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation, 10*(6), 646–657.
47. Lynn, N., & Suganthan, P. N. (2015). Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation. *Swarm and Evolutionary Computation, 24*, 11–24.
48. Mendes, R., Kennedy, J., & Neves, J. (2004). The fully informed particle swarm: Simpler, maybe better. *IEEE Transactions on Evolutionary Computation, 8*(3), 204–210. https://doi.org/10.1109/TEVC.2004.826074
49. Zhan, Z.-H., Zhang, J., Li, Y., & Shi, Y.-H. (2011). Orthogonal learning particle swarm optimization. *IEEE Transactions on Evolutionary Computation, 15*(6), 832–847. https://doi.org/10.1109/TEVC.2010.2052054
50. Engelbrecht, A.P. (2010). Heterogeneous particle swarm optimization. In: *Swarm Intelligence: 7th International Conference, ANTS 2010*, Brussels, Belgium, September 8–10, 2010. Proceedings 7. Springer, Cham, pp. 191–202
51. Wang, S., Liu, G., Gao, M., Cao, S., Guo, A., & Wang, J. (2020). Heterogeneous comprehensive learning and dynamic multi-swarm particle swarm optimizer with two mutation operators. *Information Sciences, 540*, 175–201. https://doi.org/10.1016/j.ins.2020.06.027
52. Hussain, K., Salleh, M. N. M., Cheng, S., & Shi, Y. (2019). On the exploration and exploitation in popular swarm-based metaheuristic algorithms. *Neural Computing and Applications, 31*, 7665–7683.
53. Wu, G., Mallipeddi, R., & Suganthan, P. (2016). *Problem definitions and evaluation criteria for the CEC 2017 competition and special session on constrained single objective real-parameter optimization*. National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report. 2017 Sep.
54. 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. 2013 Dec;635(2).

55. Lei, Z., Gao, S., Gupta, S., Cheng, J., & Yang, G. (2020). An aggregative learning gravitational search algorithm with self-adaptive gravitational constants. *Expert Systems with Applications, 152*, 113396.
56. Li, W., & Wang, G.-G. (2021). Improved elephant herding optimization using opposition-based learning and K-means clustering to solve numerical optimization problems. *Journal of Ambient Intelligence and Humanized Computing, 14*(3), 1753–1784. https://doi.org/10.1007/s12652-021-03391-7
57. Alsalibi, B., Abualigah, L., & Khader, A. T. (2021). A novel bat algorithm with dynamic membrane structure for optimization problems. *Applied Intelligence, 51*(4), 1992–2017. https://doi.org/10.1007/s10489-020-01898-8
58. Hu, J., Gui, W., Heidari, A. A., Cai, Z., Liang, G., Chen, H., & Pan, Z. (2022). Dispersed foraging slime mould algorithm: continuous and binary variants for global optimization and wrapper-based feature selection. *Knowledge-Based Systems, 237*, 107761. https://doi.org/10.1016/j.knosys.2021.107761
59. Derrac, J., García, 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 and Evolutionary Computation, 1*(1), 3–18.
60. Iacca, G., dos Santos Junior, V. C., Veloso de Melo, V., de Melo, V. V., & Veloso de Melo, V. (2021). An improved Jaya optimization algorithm with Lévy flight. *Expert Systems with Applications, 165*, 113902. https://doi.org/10.1016/j.eswa.2020.113902
61. Wu, Z.-S., Fu, W.-P., & Xue, R. (2015). Nonlinear inertia weighted teaching-learning-based optimization for solving global optimization problem. *Computational Intelligence and Neuroscience, 2015*, 87. https://doi.org/10.1155/2015/292576
62. Chen, X., Xu, B., Yu, K., & Du, W. (2018). Teaching-learning-based optimization with learning enthusiasm mechanism and its application in chemical engineering. *Journal of Applied Mathematics*. https://doi.org/10.1155/2018/1806947
63. Dai, C., Hu, Z., Li, Z., Xiong, Z., & Su, Q. (2020). An improved grey prediction evolution algorithm based on topological opposition-based learning. *IEEE Access, 8*, 30745–30762. https://doi.org/10.1109/ACCESS.2020.2973197
64. Gupta, S., Deep, K., & Engelbrecht, A. P. (2020). A memory guided sine cosine algorithm for global optimization. *Engineering Applications of Artificial Intelligence, 93*, 103718. https://doi.org/10.1016/j.engappai.2020.103718
65. Gupta, S., & Deep, K. (2019). A hybrid self-adaptive sine cosine algorithm with opposition based learning. *Expert Systems with Applications, 119*, 210–230. https://doi.org/10.1016/j.eswa.2018.10.050
66. Tighzert, L., Fonlupt, C., & Mendil, B. (2018). A set of new compact firefly algorithms. *Swarm and Evolutionary Computation, 40*, 92–115. https://doi.org/10.1016/j.swevo.2017.12.006
67. Chen, X., Tianfield, H., Mei, C., Du, W., & Liu, G. (2017). Biogeography-based learning particle swarm optimization. *Soft Computing, 21*(24), 7519–7541. https://doi.org/10.1007/s00500-016-2307-7
68. Song, P.-C., Chu, S.-C., Pan, J.-S., & Yang, H. (2022). Simplified phasmatodea population evolution algorithm for optimization. *Complex & Intelligent Systems, 8*(4), 2749–2767. https://doi.org/10.1007/s40747-021-00402-0
69. Song, P.-C., Chu, S.-C., Pan, J.-S., & Yang, H. (2020). Phasmatodea population evolution algorithm and its application in length-changeable incremental extreme learning machine. In: *2020 2nd International Conference on Industrial Artificial Intelligence (IAI)*. IEEE, Shenyang, China, pp 1–5. https://doi.org/10.1109/IAI50351.2020.9262236
70. Chen, T., Chen, Y., He, Z., Li, E., Zhang, C., & Huang, Y. (2023). A novel marine predators algorithm with adaptive update strategy. *The Journal of Supercomputing, 79*(6), 6612–6645. https://doi.org/10.1007/s11227-022-04903-8
71. Abualigah, L., Elaziz, M. A., Sumari, P., Geem, Z. W., & Gandomi, A. H. (2022). Reptile search algorithm (RSA): a nature-inspired meta-heuristic optimizer. *Expert Systems with Applications, 191*, 116158. https://doi.org/10.1016/j.eswa.2021.116158
72. Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Computers & Structures, 169*, 1–12. https://doi.org/10.1016/j.compstruc.2016.03.001
73. Meng, X.-B., Li, H.-X., & Gao, X.-Z. (2019). An adaptive reinforcement learning-based bat algorithm for structural design problems. *International Journal of Bio-Inspired Computation, 14*(2), 114–124. https://doi.org/10.1504/IJBIC.2019.101639
74. Zhang, M., Luo, W., & Wang, X. (2008). Differential evolution with dynamic stochastic selection for constrained optimization. *Information Sciences, 178*(15), 3043–3074. https://doi.org/10.1016/j.ins.2008.02.014
75. Ray, T., & Liew, K. M. (2003). Society and civilization: An optimization algorithm based on the simulation of social behavior. *IEEE Transactions on Evolutionary Computation, 7*(4), 386–396. https://doi.org/10.1109/TEVC.2003.814902
76. Liu, H., Cai, Z., & Wang, Y. (2010). Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Applied Soft Computing, 10*(2), 629–640. https://doi.org/10.1016/j.asoc.2009.08.031
77. Shadravan, S., Naji, H. R., & Bardsiri, V. K. (2019). The sailfish optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. *Engineering Applications of Artificial Intelligence, 80*, 20–34. https://doi.org/10.1016/j.engappai.2019.01.001
78. Zhao, W., Wang, L., & Zhang, Z. (2019). Supply-demand-based optimization: A novel economics-inspired algorithm for global optimization. *IEEE Access, 7*, 73182–73206. https://doi.org/10.1109/ACCESS.2019.2918753
79. Gandomi, A. H., Yang, X.-S., & Alavi, A. H. (2013). Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems. *Engineering with Computers, 29*(1), 17–35. https://doi.org/10.1007/s00366-011-0241-y
80. Wang, Y., Cai, Z., Zhou, Y., & Fan, Z. (2009). Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. *Structural and Multidisciplinary Optimization, 37*(4), 395–413. https://doi.org/10.1007/s00158-008-0238-3
81. Pan, J.-S., Zhang, L.-G., Wang, R.-B., Snášel, V., & Chu, S.-C. (2022). Gannet optimization algorithm: A new metaheuristic algorithm for solving engineering optimization problems. *Mathematics and Computers in Simulation, 202*, 343–373. https://doi.org/10.1016/j.matcom.2022.06.007
82. Abdollahzadeh, B., Gharehchopogh, F. S., & Mirjalili, S. (2021). African vultures optimization algorithm: a new nature-inspired metaheuristic algorithm for global optimization problems. *Computers & Industrial Engineering, 158*, 107408. https://doi.org/10.1016/j.cie.2021.107408
83. dos Santos Coelho, L. (2010). Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. *Expert Systems with Applications, 37*(2), 1676–1683. https://doi.org/10.1016/j.eswa.2009.06.044
84. Akay, B., & Karaboga, D. (2012). Artificial bee colony algorithm for large-scale problems and engineering design optimization. *Journal of Intelligent Manufacturing, 23*(4), 1001–1014. https://doi.org/10.1007/s10845-010-0393-4
85. Coello, C. A. C., Montes, E. M., Coello Coello, C. A., Mezura Montes, E., Coello, C. A. C., & Montes, E. M. (2002). Constraint-handling in genetic algorithms through the use of

dominance-based tournament selection. *Advanced Engineering Informatics, 16*(3), 193–203. https://doi.org/10.1016/S1474-0346(02)00011-3
86. He, Q., & Wang, L. (2007). An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering Applications of Artificial Intelligence, 20*(1), 89–99. https://doi.org/10.1016/j.engappai.2006.03.003
87. Huang, F., Wang, L., & He, Q. (2007). An effective co-evolutionary differential evolution for constrained optimization. *Applied Mathematics and Computation, 186*(1), 340–356. https://doi.org/10.1016/j.amc.2006.07.105
88. Parsopoulos, K.E., & Vrahatis, M.N. (2005). Unified particle swarm optimization for solving constrained engineering optimization problems. In: Wang, L., Chen, K., Ong, Y.S. (eds) *Advances in Natural Computation. ICNC 2005*. Lecture Notes in Computer Science, vol 3612. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11539902_71
89. Brajevic, I. (2015). Crossover-based artificial bee colony algorithm for constrained optimization problems. *Neural Computing and Applications, 26*(7), 1587–1601. https://doi.org/10.1007/s00521-015-1826-y
90. Faramarzi, A., Heidarinejad, M., Stephens, B., & Mirjalili, S. (2020). Equilibrium optimizer: A novel optimization algorithm. *Knowledge-Based Systems, 191*, 105190. https://doi.org/10.1016/j.knosys.2019.105190
91. Coello, C. A. C., & Cortés, N. C. (2004). Hybridizing a genetic algorithm with an artificial immune system for global optimization. *Engineering Optimization, 36*(5), 607–634. https://doi.org/10.1080/03052150410001704845
92. Bernardino, H.S., Barbosa, H.J.C., Lemonge, A.C.C., & Fonseca, L.G. (2008). A new hybrid AIS-GA for constrained optimization problems in mechanical engineering. In: *2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence)*. IEEE, Hong Kong, pp 1455–1462. https://doi.org/10.1109/CEC.2008.4630985
93. Mezura-Montes, E., & Hernández-Ocana, B. (2008). *Bacterial foraging for engineering design problems: preliminary results*. Centro de Investigación en Matemáticas Guanajuato, México
94. Mezura-Montes, E., & Coello, C. A. C. (2008). An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. *International Journal of General Systems, 37*(4), 443–473. https://doi.org/10.1080/03081070701303470
95. Aragón, V. S., Esquivel, S. C., & Coello, C. A. C. (2010). A modified version of a T-Cell algorithm for constrained optimization problems. *International Journal for Numerical Methods in Engineering, 84*(3), 351–378. https://doi.org/10.1002/nme.2904
96. Coello, C. A. C. (2000). Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry, 41*(2), 113–127. https://doi.org/10.1016/S0166-3615(99)00046-9
97. Gandomi, A. H., Yang, X.-S., Alavi, A. H., & Talatahari, S. (2013). Bat algorithm for constrained optimization tasks. *Neural Computing and Applications, 22*(6), 1239–1255. https://doi.org/10.1007/s00521-012-1028-9
98. Mazhoud, I., Hadj-Hamou, K., Bigeon, J., & Joyeux, P. (2013). Particle swarm optimization for solving engineering problems: A new constraint-handling mechanism. *Engineering Applications of Artificial Intelligence, 26*(4), 1263–1273. https://doi.org/10.1016/j.engappai.2013.02.002
99. Savsani, P., & Savsani, V. (2016). Passing vehicle search (PVS): A novel metaheuristic algorithm. *Applied Mathematical Modelling, 40*(5–6), 3951–3978. https://doi.org/10.1016/j.apm.2015.10.040
100. Zhang, J., Xiao, M., Gao, L., & Pan, Q. (2018). Queuing search algorithm: A novel metaheuristic algorithm for solving engineering optimization problems. *Applied Mathematical Modelling, 63*, 464–490. https://doi.org/10.1016/j.apm.2018.06.036
101. Montemurro, M., Vincenti, A., & Vannucci, P. (2013). The automatic dynamic penalisation method (ADP) for handling constraints with genetic algorithms. *Computer Methods in Applied Mechanics and Engineering, 256*, 70–87. https://doi.org/10.1016/j.cma.2012.12.009
102. Baykasoğlu, A., & Ozsoydan, F. B. (2015). Adaptive firefly algorithm with chaos for mechanical design optimization problems. *Applied Soft Computing, 36*, 152–164. https://doi.org/10.1016/j.asoc.2015.06.056
103. Chun, S., Kim, Y.-T., & Kim, T.-H. (2013). A diversity-enhanced constrained particle swarm optimizer for mixed integer-discrete-continuous engineering design problems. *Advances in Mechanical Engineering, 5*, 130750. https://doi.org/10.1155/2013/130750
104. Brammya, G., Praveena, S., Ninu Preetha, N. S., Ramya, R., Rajakumar, B. R., & Binu, D. (2019). Deer hunting optimization algorithm: a new nature-inspired meta-heuristic paradigm. *The Computer Journal*. https://doi.org/10.1093/comjnl/bxy133
105. Ray, T., & Saini, P. (2001). Engineering design optimization using a swarm with an intelligent information sharing among individuals. *Engineering Optimization, 33*(6), 735–748. https://doi.org/10.1080/03052150108940941
106. Coello Coello, C. A., & Becerra, R. L. (2004). Efficient evolutionary optimization through the use of a cultural algorithm. *Engineering Optimization, 36*(2), 219–236. https://doi.org/10.1080/03052150410001647966
107. Bernardino, H.S., Barbosa, H.J.C., & Lemonge, A.C.C. (2007). A hybrid genetic algorithm for constrained optimization problems in mechanical engineering. In: *2007 IEEE congress on evolutionary computation*. IEEE, Singapore, pp. 646–653. https://doi.org/10.1109/CEC.2007.4424532
108. He, Q., & Wang, L. (2007). A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Applied Mathematics and Computation, 186*(2), 1407–1422. https://doi.org/10.1016/j.amc.2006.07.134
109. Mezura-Montes, E., & Coello, C.A.C. (2005). Useful infeasible solutions in engineering optimization with evolutionary algorithms. In: Gelbukh, A., de Albornoz, Á., Terashima-Marín, H. (eds) *MICAI 2005: Advances in Artificial Intelligence. MICAI 2005*. Lecture Notes in Computer Science(), vol 3789. Springer, Berlin, Heidelberg, pp. 652–662. https://doi.org/10.1007/11579427_66
110. Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
111. Ghafil, H. N., & Jármai, K. (2020). Dynamic differential annealed optimization: new metaheuristic optimization algorithm for engineering applications. *Applied Soft Computing, 93*, 106392. https://doi.org/10.1016/j.asoc.2020.106392
112. Yapici, H., & Cetinkaya, N. (2019). A new meta-heuristic optimizer: Pathfinder algorithm. *Applied Soft Computing, 78*, 545–568. https://doi.org/10.1016/j.asoc.2019.03.012
113. Eskandar, H., Sadollah, A., Bahreininejad, A., & Hamdi, M. (2012). Water cycle algorithm–a novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers & Structures, 110*, 151–166. https://doi.org/10.1016/j.compstruc.2012.07.010
114. Zhao, W., Zhang, Z., & Wang, L. (2020). Manta ray foraging optimization: an effective bio-inspired optimizer for engineering applications. *Engineering Applications of Artificial Intelligence, 87*, 103300. https://doi.org/10.1016/j.engappai.2019.103300
115. Dehghani, M., Montazeri, Z., Trojovská, E., & Trojovský, P. (2023). Coati optimization algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems.

Knowledge-Based Systems, 259, 110011. https://doi.org/10.1016/j.knosys.2022.110011

116. Ngo, T. T., Sadollah, A., & Kim, J. H. (2016). A cooperative particle swarm optimizer with stochastic movements for computationally expensive numerical optimization problems. *Journal of Computational Science, 13*, 68–82. https://doi.org/10.1016/j.jocs.2016.01.004
117. Akhtar, S., Tai, K., & Ray, T. (2002). A socio-behavioural simulation model for engineering design optimization. *Engineering Optimization, 34*(4), 341–354. https://doi.org/10.1080/03052150212723
118. Hwang, S.-F., & He, R.-S. (2006). A hybrid real-parameter genetic algorithm for function optimization. *Advanced Engineering Informatics, 20*(1), 7–21. https://doi.org/10.1016/j.aei.2005.09.001
119. Hedar, A.-R., & Fukushima, M. (2006). Derivative-free filter simulated annealing method for constrained continuous global optimization. *Journal of Global optimization, 35*(4), 521–549. https://doi.org/10.1007/s10898-005-3693-z
120. Zhang, J., Liang, C., Huang, Y., Wu, J., & Yang, S. (2009). An effective multiagent evolutionary algorithm integrating a novel roulette inversion operator for engineering optimization. *Applied Mathematics and Computation, 211*(2), 392–416. https://doi.org/10.1016/j.amc.2009.01.048
121. He, S., Prempain, E., & Wu, Q. H. (2004). An improved particle swarm optimizer for mechanical design optimization problems. *Engineering Optimization, 36*(5), 585–605. https://doi.org/10.1080/03052150410001704854
122. Dehghani, M., Hubalovsky, S., & Trojovsky, P. (2021). Northern goshawk optimization: A new swarm-based algorithm for solving optimization problems. *IEEE Access, 9*, 162059–162080. https://doi.org/10.1109/ACCESS.2021.3133286
123. Liang, J., Ban, X., Yu, K., Qu, B., Qiao, K., Yue, C., Chen, K., & Tan, K. C. (2022). A survey on evolutionary constrained multiobjective optimization. *IEEE Transactions on Evolutionary Computation, 27*(2), 201–221. https://doi.org/10.1109/TEVC.2022.3155533

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.