Computer Modeling in
Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.045170

ARTICLE

Synergistic Swarm Optimization Algorithm

Sharaf Alzoubi[1], **Laith Abualigah**[2,3,4,5,6,7,8,*], **Mohamed Sharaf**[9], **Mohammad Sh. Daoud**[10], **Nima Khodadadi**[11] **and Heming Jia**[12]

[1]Faculty of Computer Sciences and Informatics, Amman Arab University, Amman, 11953, Jordan

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

[3]Department of Electrical and Computer Engineering, Lebanese American University, Byblos, 13-5053, Lebanon

[4]Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman, 19328, Jordan

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

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

[7]School of Computer Sciences, Universiti Sains Malaysia, Pulau Pinang, 11800, Malaysia

[8]School of Engineering and Technology, Sunway University Malaysia, Petaling Jaya, 27500, Malaysia

[9]Department of Industrial Engineering, College of Engineering, King Saud University, P.O. Box 800, Riyadh, 11421, Saudi Arabia

[10]College of Engineering, Al Ain University, Abu Dhabi, 112612, United Arab Emirates

[11]Department of Civil and Architectural Engineering, University of Miami, Coral Gables, 1251, USA

[12]School of Information Engineering, Sanming University, Sanming, 365004, China

*Corresponding Author: Laith Abualigah. Email: aligah.2020@gmail.com, abu.aligah@usm.my

Received: 19 August 2023 Accepted: 17 November 2023 Published: 11 March 2024

ABSTRACT

This research paper presents a novel optimization method called the Synergistic Swarm Optimization Algorithm (SSOA). The SSOA combines the principles of swarm intelligence and synergistic cooperation to search for optimal solutions efficiently. A synergistic cooperation mechanism is employed, where particles exchange information and learn from each other to improve their search behaviors. This cooperation enhances the exploitation of promising regions in the search space while maintaining exploration capabilities. Furthermore, adaptive mechanisms, such as dynamic parameter adjustment and diversification strategies, are incorporated to balance exploration and exploitation. By leveraging the collaborative nature of swarm intelligence and integrating synergistic cooperation, the SSOA method aims to achieve superior convergence speed and solution quality performance compared to other optimization algorithms. The effectiveness of the proposed SSOA is investigated in solving the 23 benchmark functions and various engineering design problems. The experimental results highlight the effectiveness and potential of the SSOA method in addressing challenging optimization problems, making it a promising tool for a wide range of applications in engineering and beyond. Matlab codes of SSOA are available at: https://www.mathworks.com/matlabcentral/fileexchange/153466-synergistic-swarm-optimization-algorithm.

KEYWORDS

Synergistic swarm optimization algorithm; optimization algorithm; metaheuristic; engineering problems; benchmark functions

1 Introduction

Organizations and people deal with complicated issues in today's data-driven environment that call for the best solutions [1]. These issues can occur across several industries, including engineering, banking, logistics, healthcare, and telecommunications [2]. Optimization algorithms are helpful tools to traverse these issues' complexity and find the best solutions within predetermined restrictions [3–5]. These algorithms effectively use computational and mathematical methods to find and assess viable solutions, producing optimal results [4,6].

When the structure of the solution space is unknown or inadequately specified, metaheuristic optimization algorithms provide a flexible and reliable framework for tackling optimization issues [7]. These algorithms take their cues from various abstract or natural notions rather than explicit mathematical models or assumptions [8,9]. Metaheuristics offer intelligent search algorithms that may effectively explore complex problem spaces, take advantage of promising regions, and avoid local optima by mimicking adaptive processes like evolution, swarm behavior, or physical occurrences [10–12].

A large family of computing techniques known as optimization algorithms is used to iteratively explore and assess different solution spaces in search of the best or most advantageous ones [13,14]. These algorithms can be divided into stochastic and deterministic methods. For issues with clearly stated goals and restrictions, deterministic techniques like integer and linear programming are appropriate [15,16]. To tackle complicated and dynamic problem spaces, stochastic algorithms, such as genetic algorithms, simulated annealing, and particle swarm optimization, use probabilistic approaches and randomness [16,17].

Understanding the main elements of optimization algorithms is crucial to understanding how they operate. These elements include search methods, constraints, decision variables, and objective functions [18,19]. The decision variables are the parameters or variables that may be changed to get the best outcome. In contrast, the objective function is the objective or metric that has to be maximized [20]. Constraints specify the restrictions or requirements that solutions must meet. The search procedures guide the investigation and assessment of prospective solutions, which strive to converge on the optimal result.

Numerous sectors use optimization algorithms, changing how people solve problems and make decisions [21,22]. These methods are employed in engineering to design circuits, processes, and structural optimization. They are widely used in finance for algorithmic trading, risk management, and portfolio optimization. Optimization algorithms facilitate effective route planning, inventory management, and logistics and supply chain scheduling. Optimization algorithms are helpful in the healthcare industry for clinical decision support systems, resource allocation, and treatment planning [23,24]. The applications also encompass data mining, machine learning, energy systems, telecommunications, and many more fields [25,26]. Some of the optimization methods from the literature are Prairie Dog Optimization Algorithm [27], Monarch Butterfly Optimization [28], Artificial Rabbits Optimizer [29], Genghis Khan Shark Optimizer [30], Geyser Inspired Algorithm [31], Moth Search Algorithm [32], PINN-FORM [33], Chaos Game Optimization [16], Slime Mould Algorithm [34], Hunger Games Search [35], Runge Kutta Method [36], Colony Predation Algorithm [37], Weighted Mean Of Vectors [38], Rime Optimization Algorithm [39], Harris Hawks Optimization [40], and others.

Traditional optimization algorithms frequently have trouble solving complicated problems with non-linearity, high dimensionality, and no apparent links between solutions [41]. On the other hand, metaheuristic optimization algorithms have become a potent paradigm encompassing adaptive search techniques influenced by both natural and abstract events. These algorithms can effectively search across enormous solution spaces, change their search strategies, and find excellent answers that may be difficult to find using more traditional approaches [42–44].

Complex optimization issues that defy conventional approaches can now be solved with the help of metaheuristic optimization algorithms [45,46]. This research paper's introduction highlights the adaptive search techniques, essential components, and wide range of applications of metaheuristic optimization algorithms. Researchers and practitioners can successfully take on complex optimization issues by utilizing the special powers of metaheuristics, creating new opportunities for innovation and problem-solving in various sectors. These algorithms have a lot of potential as metaheuristic research develops to handle challenging optimization problems in the real world [47,48]. Numerous advantages of optimization algorithms include increased effectiveness, better resource allocation, cost savings, and better decision-making [49]. They enable businesses to increase output, reduce waste, and streamline procedures. However, there are difficulties in putting optimization methods into practice. For example, managing high-dimensional spaces, dealing with non-linear or non-convex objectives, and balancing computing complexity with solution accuracy [50,51].

This paper proposes a novel optimization method called The Synergistic Swarm Optimization Algorithm (SSOA) that combines the principles of swarm intelligence and synergistic cooperation to search for optimal solutions efficiently. The main procedure of SSOA involves the following steps: First, an initial population of candidate solutions, represented as a swarm of particles, is randomly generated. Each particle explores the search space by iteratively updating its position based on its own previous best solution and the collective knowledge of the swarm. Next, a synergistic cooperation mechanism is employed, where particles exchange information and learn from each other to improve their search behaviors. This cooperation enhances the exploitation of promising regions in the search space while maintaining exploration capabilities. Furthermore, adaptive mechanisms, such as dynamic parameter adjustment and diversification strategies, are incorporated to balance exploration and exploitation. The process continues iteratively until a termination criterion is met, such as a maximum number of iterations or reaching a satisfactory solution. By leveraging the collaborative nature of swarm intelligence and integrating synergistic cooperation, the SSOA method aims to achieve superior convergence speed and solution quality performance compared to other optimization algorithms. The effectiveness of the proposed SSOA is investigated in solving the 23 benchmark functions and various engineering design problems. The proposed Synergistic Swarm Optimization Algorithm (SSOA) demonstrates remarkable performance surpassing that of other optimization algorithms, without any similarity in their approach or methodologies. Through comprehensive evaluations, the SSOA method consistently outperforms similar optimization algorithms regarding solution quality and convergence speed.

In summary, the primary motivation of this research is to develop a novel optimization algorithm, SSOA, that leverages swarm intelligence principles, synergistic cooperation, adaptive mechanisms, and a balance between exploration and exploitation to efficiently find optimal solutions for a wide range of optimization problems. The ultimate goal is to demonstrate that SSOA surpasses existing optimization algorithms in terms of both solution quality and convergence speed through rigorous empirical evaluations.

The structure of this paper is given as the related works are provided in Section 2. Section 3 shows the proposed Synergistic Swarm Optimization Algorithm (SSOA) and its main procedures. Section 4 presents the results and discussion of the experiments. Finally, the conclusion and future work direction is given in Section 5.

2 Related Works

This section presents the most common related to the optimization algorithms from the literature and its applications [2,52].

The Fire Hawk Optimizer (FHO), a novel metaheuristic algorithm introduced in [53], was inspired by the foraging behavior of whistling kites, black kites, and brown falcons—collectively known as Fire Hawks—because of the way they grab prey using fire. To explore the proposed technique numerically, 150,000 function evaluations are carried out for optimization purposes on 233 mathematical test functions, ranging from 2 to 100 dimensions. Ten distinct traditional and modern metaheuristic algorithms are used as complementary methods to contrast their performance. The FHO method is further assessed in the CEC 2020 competition for binding constraint and real-world optimization problems, including well-known mechanical engineering design issues. The analysis shows that the FHO method outperforms other metaheuristic algorithms in the literature. The FHO also performs better than previously established metaheuristics when dealing with real-size structural frameworks, notably those with 15 and 24 stories.

The Wild Horse Optimizer (WHO), a novel algorithm inspired by the social behaviors of wild horses, was introduced in [54]. Wild horses often exist in herds comprising a stallion, mares, and calves. They participate in various behaviors, including mating, dominating, pursuing, and grazing. Horses are notable for their decent conduct, which prevents breeding amongst close relatives by having foals leave the herd before they reach adolescence and join other groups. This decent behavior serves as an inspiration for the WHO algorithm. It is assessed using several test functions, such as CEC2017 and CEC2019, and contrasted with other well-known and cutting-edge optimization techniques. The outcomes indicate how well the WHO algorithm performs compared to other algorithms, highlighting its effectiveness.

The Cheetah Optimizer (CO), a naturally derived algorithm inspired by cheetah hunting techniques, was introduced in [55]. The CO algorithm integrates the three primary cheetah-hunting tactics of seeking, waiting, and attacking. The "leave the prey and go back home" method is also added to improve population diversity, convergence efficiency, and algorithmic resilience. On 14 shifted-rotated CEC-2005 benchmark functions, CO's performance is thoroughly evaluated compared to cutting-edge algorithms. Simulation results show that CO outperforms existing algorithms, including conventional, enhanced, and hybrid methods, in solving complex and large-scale optimization problems.

The Dung Beetle Optimizer (DBO) algorithm, a population-based method inspired by dung beetles' ball-rolling, dancing, foraging, thieving, and reproductive habits, was presented in [56]. The DBO algorithm compromises quick convergence and acceptable solution correctness by combining global exploration and local exploitation. A collection of well-known mathematical test functions is used to gauge the search engine's performance. According to simulation findings, the DBO method competes favorably regarding convergence rate, solution correctness, and stability with cutting-edge optimization techniques. The DBO algorithm's effective use in three engineering design challenges also proves its potential for practical application. Experimental findings confirm the suggested DBO algorithm's efficiency in solving practical application difficulties.

The Bonobo Optimizer (BO), an intelligent optimization approach inspired by bonobos' reproductive tactics and social behavior, was introduced in [57]. The social structure of bonobos is characterized by fission-fusion, with groups of various sizes and compositions first developing within the community and then fusing for particular purposes. To preserve social peace, they use four reproductive techniques: consortship mating, extra-group mating, promiscuous mating, and restricted mating. When BO's performance is compared to other optimization techniques, the results are statistically better or on par. Additionally, BO is used to resolve five challenging real-world optimization issues, and the effectiveness of its results is evaluated against the body of prior research.

The results emphasize the efficiency of BO in resolving optimization issues by showing that it either outperforms or produces results equivalent to those of other approaches.

This research presents the Golden Jackal Optimization (GJO) algorithm [58], a new nature-inspired optimization method that offers an alternative approach to solving real-world engineering problems. GJO draws inspiration from the collaborative hunting behavior of golden jackals (Canis aureus). The algorithm incorporates three essential steps: prey searching, enclosing, and pouncing, which are mathematically modeled and implemented. The effectiveness of GJO is evaluated by comparing it with state-of-the-art metaheuristic algorithms on benchmark functions. Furthermore, GJO is tested on seven different engineering design problems and implemented in electrical engineering. The engineering design problems and real-world implementation results demonstrate that the proposed algorithm is well-suited for addressing challenging problems characterized by unidentified search spaces.

The Arithmetic Optimization Algorithm (AOA), a unique metaheuristic technique introduced in [59], makes use of the distributional properties of the four basic mathematical arithmetic operators (multiplication, division, subtraction, and addition). AOA is a mathematical model developed and used to handle optimization jobs in various search spaces effectively. AOA is tested against 29 benchmark functions and a variety of actual engineering design issues in order to show how useful it is. Different scenarios are used for performance analysis, convergence behavior study, and computational complexity analyses. The experimental results demonstrate how AOA can solve challenging optimization issues significantly better than eleven well-known optimization methods.

This study offers a brand-new population-based optimization technique dubbed the Aquila Optimizer (AO), which was motivated by the natural hunting techniques of Aquila birds [60]. Four strategies that mirror the behaviors taken by the Aquila during the prey-catching process are used to illustrate the optimizing processes of AO. These techniques include using a high soar to narrow the search area, contour flying with a quick glide attack for exploration, low flight with a gradual descent assault for exploitation, and swooping by walking up on a target. Several tests are carried out to evaluate how well AO finds the best answers for various optimization issues. In the first experiment, 23 well-known functions are solved using AO. In the second and third tests, 30 CEC2017 and 10 CEC2019 test functions are used to assess AO's performance on increasingly challenging issues. In addition, seven actual engineering challenges are used to test AO. Comparisons with well-known metaheuristic techniques show that AO performs better.

The Reptile Search Algorithm (RSA), inspired by the crocodile's hunting strategy, is a novel metaheuristic optimizer introduced in [61]. The RSA includes two essential phases: encircling, which entails high walking or belly walking, and hunting, which includes coordinated or cooperative hunting. These distinctive search techniques set RSA apart from other algorithms. Various test functions, including 23 classical test functions, 30 CEC2017 test functions, 10 CEC2019 test functions, and seven real-world engineering challenges, are used to assess RSA's performance. There are comparisons made between several optimization techniques from the literature. Results show that RSA performs better on the benchmark functions examined than competing methods. The Friedman ranking test attests to RSA's undeniable advantage over other comparison techniques. Additionally, the analysis of engineering challenges shows that RSA outperforms other techniques in terms of outcomes.

This paper introduces the Dwarf Mongoose Optimization Algorithm (DMO) [62], a unique metaheuristic algorithm built to handle 12 continuous/discrete engineering optimization issues and benchmark functions from CEC 2020 and the classical world. The dwarf mongoose, whose social and ecological tactics have evolved owing to the constrained style of prey collection, served as the model for the DMO algorithm. The mongoose's migratory lifestyle ensures that the whole region is explored without returning to sleeping mounds, limiting the overexploitation of certain locations. DMO's efficacy is shown through comparison to seven other algorithms' performance in terms of various performance indicators and data.

The Ebola Optimization Search Algorithm (EOSA), a new metaheuristic algorithm inspired by the transmission mechanism of the Ebola virus illness, was introduced in [63]. The first step in the project is developing an enhanced SIR model, known as SEIR-HVQD, which depicts the many disease phases. The new model is then mathematically represented using first-order differential equations. EOSA is created as a metaheuristic algorithm by fusing the propagation and mathematical models. Two sets of benchmark functions—47 classical functions and 30 restricted IEEE-CEC benchmark functions—are used to compare the algorithm's performance to other optimization techniques. According to the results, EOSA competes favorably with other cutting-edge optimization techniques regarding scalability, convergence, and sensitivity studies.

In [27], Prairie Dog Optimization (PDO) was introduced, a unique metaheuristic algorithm inspired by nature that mimics prairie dogs' behavior in their natural habitat. PDO uses four prairie dog behaviors to complete the optimization's exploration and exploitation stages. The framework for exploratory behavior in PDO is provided by prairie dogs' foraging and burrow-building behaviors. PDO investigates the issue of space in a manner akin to prairie dogs digging tunnels around plentiful food sources. PDO looks for new food sources and constructs burrows around them, engulfing the entire colony or problem area to find fresh solutions. Twenty-two traditional benchmark functions and ten CEC 2020 test functions are used to assess PDO's performance. The experimental findings demonstrate the balanced exploration and exploitation capabilities of PDO. According to a comparative study, PDO offers greater performance and capabilities when compared to other well-known population-based metaheuristic algorithms.

In this section, we review several optimization algorithms proposed in the literature. These algorithms include the Fire Hawk Optimizer (FHO), Wild Horse Optimizer (WHO), Cheetah Optimizer (CO), Dung Beetle Optimizer (DBO), Bonobo Optimizer (BO), Golden Jackal Optimization (GJO), Arithmetic Optimization Algorithm (AOA), Aquila Optimizer (AO), Reptile Search Algorithm (RSA), Dwarf Mongoose Optimization Algorithm (DMOA), Ebola Optimization Search Algorithm (EOSA), and Prairie Dog Optimization (PDO). To provide an overview of these algorithms, we present a table summarizing their key characteristics and inspiration sources, see Table 1.

Table 1: An overview of the given and studied methods

Ref.	Algorithm	Inspiration	Key features	Application domains
[53]	Fire hawk optimizer (FHO)	Fire hawks	Global exploration, local exploitation	Engineering, logistics, finance
[54]	Wild horse optimizer (WHO)	Wild horses	Prey searching, enclosing, pouncing	Engineering, data mining, scheduling
[55]	Cheetah optimizer (CO)	Cheetahs	Searching, sitting-and-waiting, attacking	Optimization, machine learning, robotics
[56]	Dung beetle optimizer (DBO)	Dung beetles	Ball-rolling, dancing, foraging, stealing, reproduction	Optimization, image processing, robotics

(Continued)

Table 1 (continued)

Ref.	Algorithm	Inspiration	Key features	Application domains
[57]	Bonobo optimizer (BO)	Bonobos	Fission-fusion social organization, reproductive strategies	Optimization, social networks
[58]	Golden jackal optimization (GJO)	Golden jackals	Prey searching, enclosing, pouncing	Engineering, data mining, scheduling
[59]	Arithmetic optimization algorithm (AOA)	Mathematics	Utilizes arithmetic operators as a search strategy	Optimization, machine learning, finance
[60]	Aquila optimizer (AO)	Aquila birds	Soaring, glide attack, descent attack	Engineering, scheduling, logistics
[61]	Reptile search algorithm (RSA)	Reptiles	Encircling, hunting, cooperation	Optimization, data clustering, robotics
[62]	Dwarf mongoose optimization algorithm (DMOA)	Dwarf mongooses	Adaptive search, cooperation, exploration, exploitation	Engineering, machine learning, scheduling
[63]	Ebola optimization search algorithm (EOSA)	Ebola virus disease	Propagation mechanism, adaptive strategies	Optimization, data clustering, finance
[27]	Prairie dog optimization (PDO)	Prairie dogs	Prey searching, encircling, swooping	Engineering, data mining, scheduling

By examining these algorithms, it is evident that they draw inspiration from various natural phenomena, including animal behaviors, disease propagation, and mathematical principles. Each algorithm incorporates unique strategies and features to address optimization challenges. The table provides a concise overview of the algorithms' inspiration sources and key characteristics, enabling researchers and practitioners to understand their distinct approaches.

3 The Synergistic Swarm Optimization Algorithm (SSOA)

Natural swarms' cooperative and synergistic behavior inspires the sophisticated optimization technique Synergistic Swarm Optimization Algorithm (SSOA). The method uses a swarm of agents that cooperate to solve complicated problems most effectively.

At its heart, SSOA uses the swarm intelligence concepts, whereby individual agents (particles) interact with one another and their surroundings to explore and utilize the search space effectively. These particles combine to create a swarm that modifies its behavior dynamically to move through the issue landscape and converge on the best answers. Each particle in the initial population of the SSO algorithm represents a potential solution. These particles move across the search space by varying their placements and speeds in a multidimensional solution space. The particles' locations correlate to potential solutions, but their velocities point in the right direction for the search.

Particles exchange information and interact with one another during the optimization process, allowing them to benefit from the swarm's collective knowledge. They can concurrently explore new areas, share knowledge, and take advantage of attractive search space regions because of their cooperative behavior. SSO relies heavily on synergy. It is accomplished by the particles working together and influencing one another. Particles share their achievements and mistakes as the swarm develops, influencing one another's actions and constantly modifying their movement tactics. The swarm's capacity to efficiently traverse the search space, avoid being stuck in local optima, and converge towards the global optimum is improved by this dynamic collaboration.

SSO uses adaptive methods, such as velocity adjustment, local and global neighborhood interactions, and diversity maintenance, to further improve the search capabilities. These techniques allow the swarm to balance exploration and exploitation, alter its behavior based on the challenge's specifics, and delay convergence. SSO's performance is assessed based on an objective function that assesses the high quality of solutions. The swarm continually improves its answers through iterative updates and interactions, seeking to reduce or maximize the objective function depending on the kind of optimization issue.

Numerous optimization issues, such as function optimization, engineering design, resource allocation, and data clustering, have been solved using Synergistic Swarm Optimization. It is an effective tool for addressing challenging real-world issues and locating optimum or nearly ideal solutions due to its capacity to harness collective intelligence, flexibility, and efficient exploration.

3.1 Main Procedure of SSOA

The main procedure of the proposed Synergistic Swarm Optimization Algorithm (SSOA) is presented as follows.

The optimization process starts with internalizing the candidate solutions randomly by Eq. (1).

$$X = rand\left(N, Dim\right) * \left(UB - LB\right) + LB \tag{1}$$

Eq. (1) generates a matrix X of size $(N\ x\ Dim)$ with random values within a given range. The structure of this matrix is given in Matrix (2).

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,Dim} \\ \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,Dim} \end{bmatrix} \tag{2}$$

where N represents particles or solutions, Dim represents dimensions or variables of the given problem, UB and LB represent vectors of upper bounds and lower bounds, respectively, for each dimension of the problem space.

The candidate solutions (X) are updated using Eq. (3).

$$Xnew\left(i,j\right) = X\left(i,j\right) + v\left(i,j\right) \tag{3}$$

where *Xnew(i, j)* represents the new optimized position j of the *ith* candidate solution, *X(i, j)* represents current position j of the *ith* candidate solution, *v(i, j)* represents the position j of the *ith* candidate solution value.

In addition to the velocity update equation, introduce a dynamic attraction equation that influences the particles' movement toward more promising regions of the search space. This equation is designed to adaptively guide the particles based on the local and global attractiveness of the positions. It can be defined as in Eq. (4).

$$v_{new}\left(i,j\right) = IWV + PBC + GBC + DAC + ANIC + MDC \tag{4}$$

The $v_{new}\left(i,j\right)$ value is calculated using the following equations. The inertia weight value (IWV) is calculated as follows:

$$IWV = w\left(t\right) \ * \ v\left(i,j\right) \tag{5}$$

where the w is an adaptive mechanism for the inertia weight parameter (w) to control the balance between exploration and exploitation dynamically.

The adaptive neighborhood interaction equation promotes a focused search space exploration by giving more weight to particles with higher fitness, allowing the swarm to converge more efficiently. Introduce an equation that dynamically adjusts the strength of interaction between particles based on their fitness values. This equation enables particles with higher fitness to have a stronger influence on the movement of their neighbors. The inertia weight can be updated at each iteration using an adaptive equation, such as:

$$w\left(t+1\right) = w\left(t\right) * \ \left(1 \ - \ exp\left(-k \ * \ t\right)\right) \tag{6}$$

where k is a constant that determines the rate of inertia weight reduction, and t represents the current iteration. The algorithm gradually shifts from exploration to exploitation by reducing the inertia weight over time, promoting fine-tuning and convergence toward the optimal solution. The personal best coefficient (PBC) is calculated as follows:

$$PBC = r1 \ * \ \left(eps * rand\left(pbest\right) \ - \ X_i\right) \tag{7}$$

where $r1$ is a random value, eps presents a small value, $rand\left(pbest\right)$ is a random solution from the current candidate solutions, and X_i presents the solution number i. The global best coefficient (GBC) is calculated as follows:

$$GBC = r2 * gbest_t \ - \ X_i \tag{8}$$

where $r2$ is a random value, $gbest_t$ is the best global solution so far (at iteration number t), and X_i presents the solution number. Include a diversity maintenance equation that encourages the exploration of diverse regions within the search space. The dynamic attraction coefficient (DAC) is calculated as follows:

$$DAC = r3 * \frac{attract_i}{c1} - \ X_i \tag{9}$$

where $r3$ is a random value, $attract_i$ represents the position with the highest local attraction value within the neighborhood of the ith particle, $c1$ is an additional acceleration coefficient for the dynamic attract term, and X_i presents the solution number i. The dynamic attraction term guides particles toward highly attractive positions, promoting faster convergence toward optimal solutions. The adaptive neighborhood interaction coefficient (ANIC) is calculated as follows:

$$ANIC = r4 * \ rand\left(bestf\right) \ - bestf_i \tag{10}$$

where $r4$ is a random value, $rand\left(bestf\right)$ is a random fitness value from the current fitness solutions, and $bestf_i$ is the fitness value of the ith solution. The diversity maintenance coefficient (DMC) is calculated as follows:

$$DMC = r5 * \frac{diversity_i}{c2} - \ X_i \tag{11}$$

where $r5$ is a random value, $c2$ is an additional acceleration coefficient for the diversity term. $diversity_i$ represents a position within the swarm that maximizes the diversity within the neighborhood of the ith particle.

The diversity term promotes the exploration of less-explored regions, ensuring that the swarm maintains a diverse set of solutions and avoids premature convergence. These novel equations introduce adaptive mechanisms, dynamic attraction, adaptive neighborhood interactions, inertia weight adaptation, and diversity maintenance to the SSO algorithm. The algorithm can effectively adapt its behavior and focus by incorporating these enhancements. The main procedure of the Synergistic Swarm Optimization Algorithm (SSOA) is presented in Algorithm 1.

Algorithm 1: The main procedure of the Synergistic Swarm Optimization Algorithm (SSOA)

```
Initialize swarm:
  Generate random positions for each particle in the swarm
  Initialize velocities to zero
  Set current-best positions to initial positions
  Set the global best position as empty
  Initialize personal best fitness values to infinity
  Set global best fitness value to infinity
Main loop:
  For $t = 1$ to $T$ do
    Evaluate fitness for each particle
    Update personal best positions
    Update global best position
    Update inertia weight
    For each particle in the swarm, do
      Generate random numbers r1, r2, r3, r4, r5
      Update velocity:
        Update velocity using the novel equations incorporating c1, c2, r1, r2, r3, r4
      Update position:
        Update position based on the updated velocity
      Apply boundary constraints (if any) to the position
    end for
  end for
Final results:
  Display global best position
  Display global best fitness
Fitness evaluation function:
  Evaluate fitness for each particle (modify according to your problem)
  Calculate fitness for the particle at position positions (i, :)
  Replace with your fitness evaluation code.
Apply boundary constraints function:
  Apply constraints to the position if needed
  Modify the position to satisfy any constraints of the optimization problem
```

3.2 Advantages and Disadvantages of the Proposed SSOA

The advantages and disadvantages of the proposed Synergistic Swarm Optimization Algorithm (SSOA) are given as follows.

3.2.1 Advantages

- SSOA is designed to search for optimal solutions in complex problem spaces efficiently. It leverages the principles of swarm intelligence to guide the search process, which can lead to faster convergence to good solutions.
- SSOA incorporates mechanisms to balance exploration (searching a wide solution space) and exploitation (refining solutions in promising regions). This balance can help avoid getting stuck in local optima and improve the chances of finding global optima.
- The algorithm uses synergistic cooperation among particles in the swarm, allowing them to exchange information and learn from each other. This collaborative approach can enhance the overall search capability and increase the likelihood of finding high-quality solutions.
- SSOA includes adaptive mechanisms like dynamic parameter adjustment and diversification strategies. These adaptations enable the algorithm to perform effectively across various optimization problems without requiring extensive parameter tuning.
- The research paper claims that SSOA consistently outperforms similar optimization algorithms regarding solution quality and convergence speed. This empirical evidence suggests SSOA may be a strong choice for many optimization tasks.

3.2.2 Disadvantages

- SSOA appears to be a relatively complex optimization algorithm compared to simpler methods like gradient descent. Its complexity may make it challenging to implement and understand for some users, especially those unfamiliar with swarm intelligence concepts.
- Due to the collaborative nature of SSOA, it may require a significant amount of computational resources, mainly when applied to large-scale optimization problems with many dimensions. This could limit its practicality in resource-constrained environments.
- While SSOA incorporates adaptive mechanisms, it may still require some degree of parameter tuning to perform optimally on specific problem instances. Finding the suitable parameters for a given problem can be a non-trivial task.
- SSOA's performance can vary depending on the characteristics of the optimization problem it is applied to. It may not consistently outperform other algorithms and require careful consideration of problem-specific factors.

In summary, while the Synergistic Swarm Optimization Algorithm (SSOA) offers several advantages, including efficient optimization, balanced exploration-exploitation, and synergistic cooperation, it also has disadvantages, such as complexity, computational resource requirements, and potential sensitivity to problem-specific factors. Users should carefully assess whether SSOA is suitable for their specific optimization tasks and consider its pros and cons in that context.

4 Experiments and Results

This section presents the experiments and results of the tested methods against the proposed method.

4.1 Parameters Settings

This section shows the parameter setting values used through the experiments of the tested methods. Table 2 shows the used parameter settings values. The tested methods are Aquila Optimizer

(AO), Slap Swam Algorithm (SSA), Whale Optimization Algorithm (WOA), Sine Cosine Algorithm (SCA), Dragonfly Algorithm (DA), Grey Wolf Optimizer (GWO), Gazelle Optimization Algorithm (GOA), Ant Lion Optimization (ALO), Reptile Search Algorithm (RSA), Genetic Algorithm (GA), Arithmetic Optimization Algorithm (AOA), and the proposed Synergistic Swarm Optimization Algorithm (SSOA).

Table 2: Parameters settings of the tested methods

No.	Algorithms	Parameters	Value
1	AO	α	0.1
		ς	0.1
2	SSA	ν	0.0
3	WOA	α	2.0–0.0
		b	2.0
4	SCA	α	0.05
5	DA	w	2.0–0.9
		s, a, and c	0.1
		F and e	1.0
6	GWO	α	2.0–0.0
7	GOA	s	[0.1]
		µ	[−1, 1]
		PSRs	0.84
8	ALO	I ratio	10
		w	2.0–0.6
9	RSA	α	0.1
		β	0.1
10	GA	CR	0.9
		MR	0.2
11	AOA	α	5.0
		µ	0.5
12	SSOA	w	0.7
		C1	2.0
		C2	2.0
		k	0.5

The experiments in the following sections have been conducted on Operating system 64-Bit, Windows 10, MATLAB R2015a, CPU Intel(R), Core(TM) i7 processor, RAM 16 GB, and Hard disk 1000 GB.

4.2 Experiments of the SSOA on 23 Benchmark Functions

In this section, the experimental results are presented using various benchmark functions to validate the performance of the proposed method compared to the other comparative methods. Benchmark functions should represent a diverse range of optimization challenges. They should include functions with different characteristics, such as multimodality (multiple peaks), nonlinearity, convexity, and non-convexity. This diversity allows researchers to assess how well the algorithm performs across various problem types. Table 3 shows the description of the used 23-benchmark functions.

Table 3: Description of the used 23-benchmark function problems

Fun.	Fun. Description	Dim.	Range	$f_{\min}$
	Unimodal benchmark functions			
F1	$f(x) = \sum_{i=1}^{n} x_i^2$	10,100	[−100,100]	0
F2	$f(x) = \sum_{i=0}^{n} \lvert x_i \rvert + \prod_{i=0}^{n} \lvert x_i \rvert$	10,100	[−10,10]	0
F3	$f(x) = \sum_{i=1}^{d} \left(\sum_{j=1}^{i} x_j \right)^2$	10,100	[−100,100]	0
F4	$f(x) = max_i \left(\lvert x_i \rvert, 1 \le i \le n \right)$	10,100	[−100,100]	0
F5	$f(x) = \sum_{i=1}^{n-1} \left[100 \left(x_i^2 - x_{i+1} \right)^2 + (1 - x_i)^2 \right]$	10,100	[−30,30]	0
F6	$f(x) = \sum_{i=1}^{n} \left([x_i + 0.5] \right)^2$	10,100	[−100,100]	0
F7	$\mathrm{f(x)} = \sum_{\mathrm{i}=0}^{\mathrm{n}} \mathrm{ix_i^4} + \mathrm{random}\,[0, 1)$			
	Multimodal benchmark functions			
F8	$\mathrm{f(x)} = \sum_{\mathrm{i}=1}^{\mathrm{n}} \left(-\mathrm{x_i} \sin \left(\sqrt{\lvert \mathrm{x_i} \rvert} \right) \right)$	10,100	[−500,500]	−418.9829
F9	$\mathrm{f(x)} = \sum_{\mathrm{i}=1}^{\mathrm{n}} \left[\mathrm{x_i^2} - 10 \cos (2\pi \mathrm{x_i}) + 10 \right]$	10,100	[−5.12,5.12]	0
F10	$\mathrm{f(x)} = -20 \exp \left(-0.2 \sqrt{\frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{x_i^2}} \right) - \exp \left(\frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} \cos (2\pi \mathrm{x_i}) \right) + 20 + \mathrm{e}$	10,100	[−32,32]	0
F11	$\mathrm{f(x)} = 1 + \frac{1}{4000} \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{x_i^2} - \prod_{\mathrm{i}=1}^{\mathrm{n}} \cos \left(\frac{\mathrm{x_i}}{\sqrt{\mathrm{i}}} \right)$	10,100	[−600,600]	0
F12	$\mathrm{f(x)} = \frac{\pi}{\mathrm{n}} 10 \sin (\pi \mathrm{y_1}) + \sum_{\mathrm{i}=1}^{\mathrm{n}-1} (\mathrm{y_i} - 1)^2 \left[1 + 10 \sin^2 (\pi \mathrm{y_{i+1}}) + \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{u} (\mathrm{x_i}, 10{,}100, 4) \right]$ $y_i = 1 + \frac{x_i + 1}{4},$ $u(x_i, a, k, m) \begin{cases} K(x_i - a)^m & \text{if } x_i > a \\ 0 & -a \le x_i \ge a \\ K(-x_i - a)^m & -a \le x_i \end{cases}$	10,100	[−50,50]	0

(Continued)

Table 3 (continued)

Fun.	Fun. Description	Dim.	Range	$f_{\min}$
F13	$f(x) = 0.1\Big(\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2$ $\big[1 + \sin^2(3\pi x_i + 1)\big] + (x_n - 1)^2\, 1 + \sin^2(2\pi x_n)\Big)$ $+ \sum_{i=1}^{n} u(x_j, 5, 100, 4)$	10,100	[−50,50]	0
Fixed-dimension multimodal benchmark functions				
F14	$f(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2}}\left(x_i - a_{ij}\right)\right)^{-1}$	2	[−65,65]	1
F15	$f(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1\left(b_i^2 + b_i x_2\right)}{b_i^2 + b_i x_3 + x_4}\right]^2$	4	[−5,5]	0.398
F16	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2$	2	[−5,5]	−1.0316
F17	$f(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2$ $+ 10\left(1 - \frac{1}{8\pi}\right) cos x_1 + 10$	2	[−5,5]	0.398
F18	$f(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2$ $+ 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2$ $\times \left(18 - 32x_i + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)]$	2	[−2,2]	3
F19	$f(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{i=1}^{3} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	3	[−1,2]	−3.86
F20	$f(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{i=1}^{6} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	6	[0,1]	−.32
F21	$f(x) = -\sum_{i=1}^{5}\left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	[0,1]	−10.1532
F22	$f(x) = -\sum_{i=1}^{7}\left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	[0,1]	−10.4028
F23	$f(x) = -\sum_{i=1}^{10}\left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	[0,1]	−10.5363

The qualitative results of the proposed SSOA using the 13-benchmark function are presented in Fig. 1. By looking at trajectories and convergence curves, we provide a qualitative examination of the convergence of the SSOA. Numerous qualitative measures that show the convergence of SSOA are shown in Fig. 1. To demonstrate, the domain's topology is highlighted in the first column of Fig. 1, which shows a 2D visual representation of the tested functions.

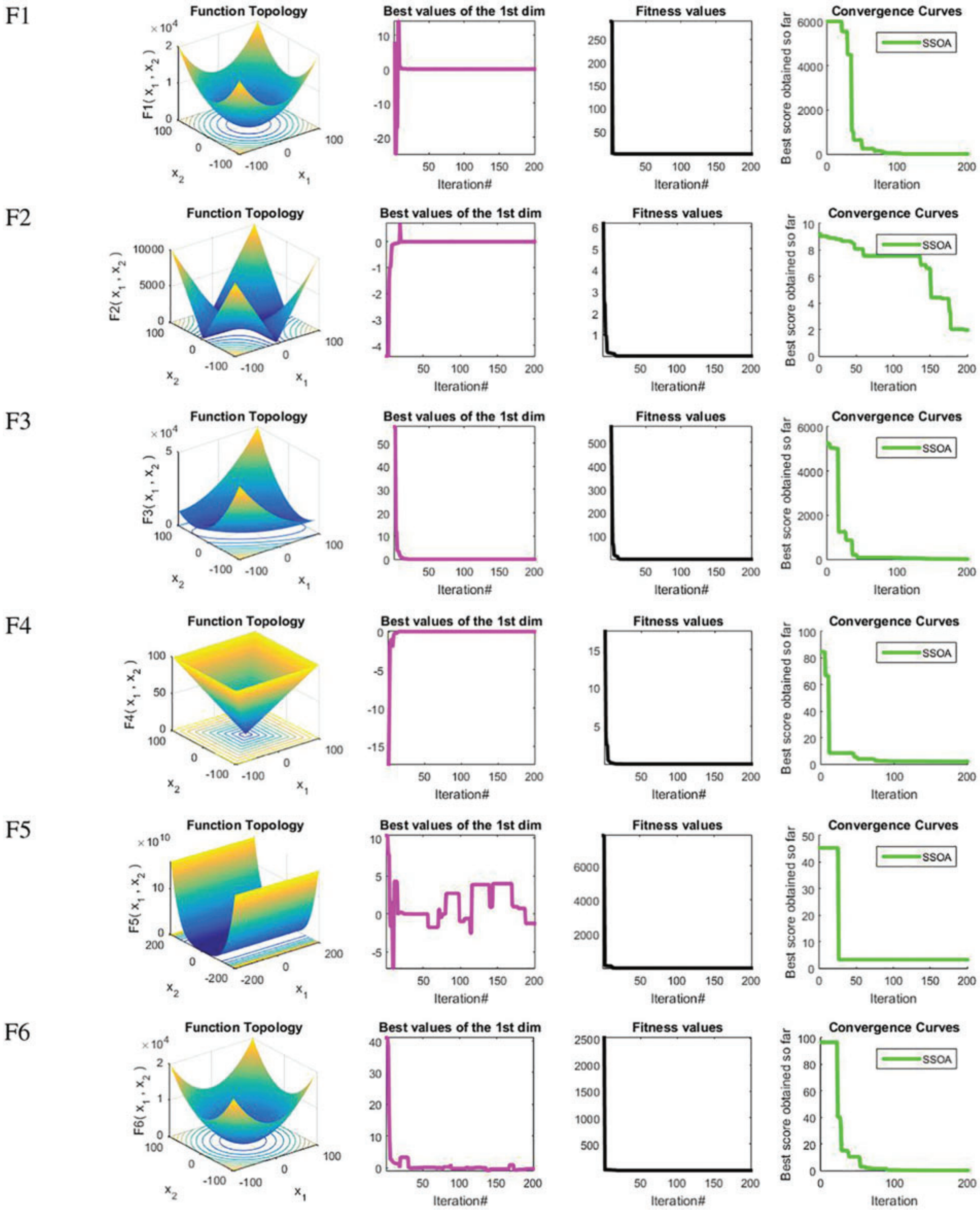

Figure 1: (Continued)

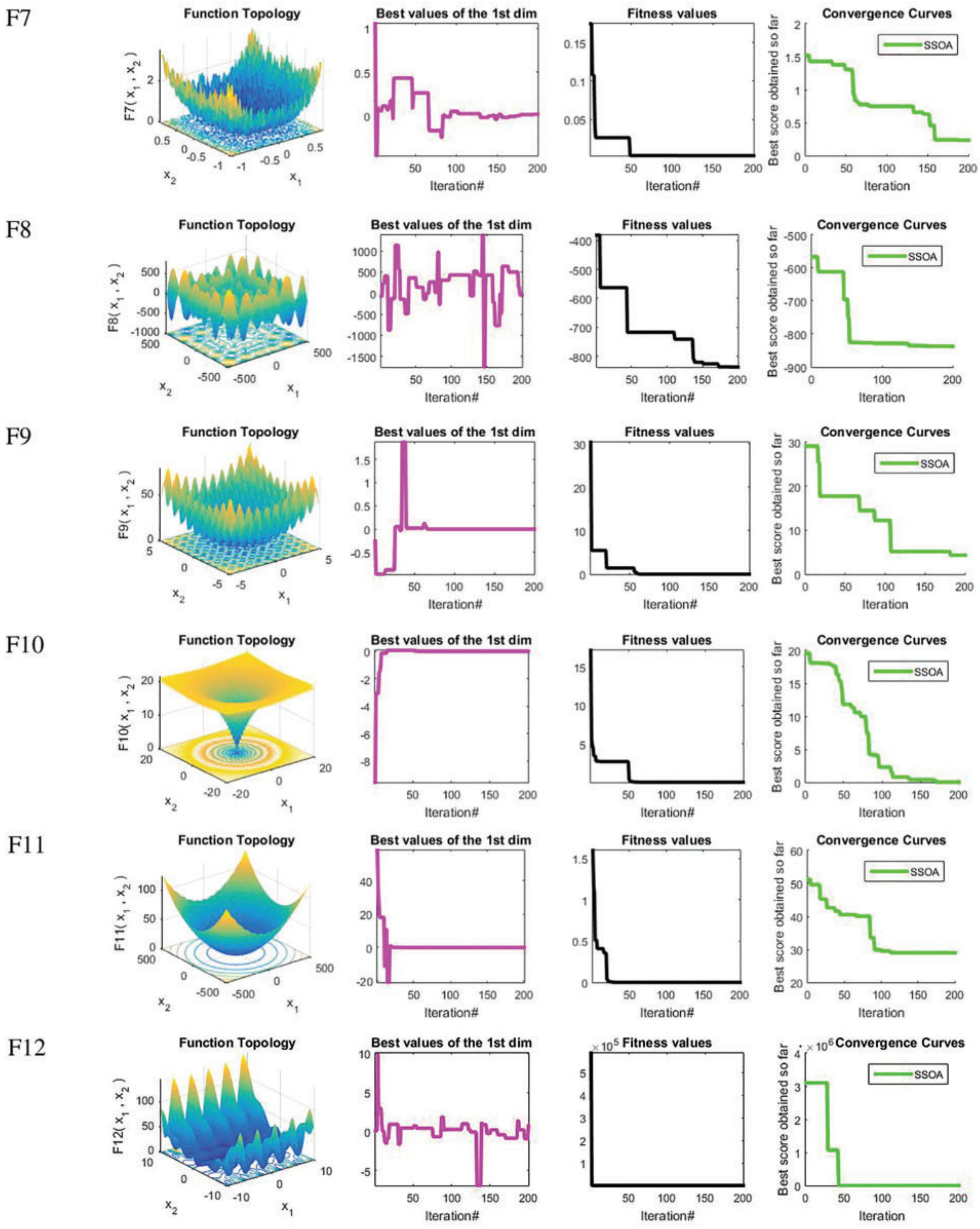

Figure 1: (Continued)

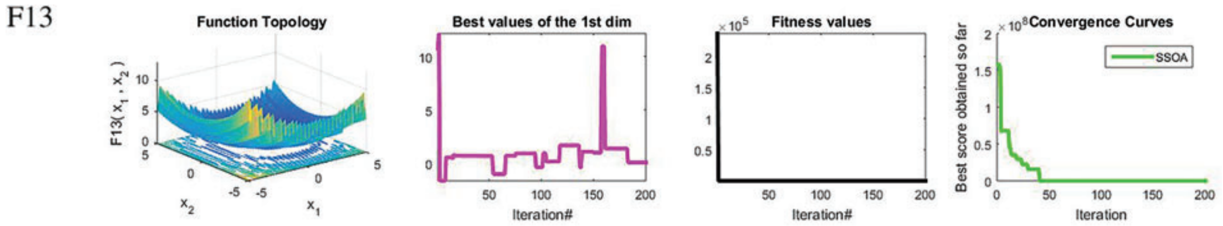

Figure 1: Qualitative results for the given 13 problems

Furthermore, it is clear that compared to unimodal functions, multimodal and composition functions display topological fluctuations more frequently, more significantly, and for longer. This finding emphasizes how well SSOA performs with various optimization functions due to its extraordinary robustness and versatility. The algorithm has a notable aptitude for striking a balance between exploration and exploitation, enabling it to negotiate challenging terrain successfully.

Additionally, the average fitness value of the agents throughout the iterations is depicted in the third column of Fig. 1. These curves illustrate that the agents initially exhibit diverse fitness values, and in most cases, this diversity persists throughout the entire range of iterations. This observation confirms the SSOA's exceptional capability to preserve diversity throughout the iterations.

The convergence curve representing the best agent identified so far can be observed in the last column of Fig. 1. Each function type exhibits a distinct pattern in its convergence curve. For instance, the convergence curve for unimodal functions displays a smooth progression and achieves optimal results with a small number of iterations. In contrast, the convergence curve for multimodal functions exhibits a stepwise behavior, reflecting the complexity of these functions. Additionally, it is noticeable that in the initial iterations of SSOA, the agents tend to encircle the optimal agent while striving to improve their positions as the iterations progress. The agents explore the search space throughout the entire iterations to determine optimal positions and enhance their overall performance.

To compare the proposed SSOA method with the other tested methods using 13 benchmark functions with ten dimensions. To show that the proposed method (SSOA) is the best, we can analyze the results in Table 4. Here is a discussion of the results for each benchmark function.

For the F1, the proposed SSOA method achieved the 2nd best result in the "Best" measure. It obtained an average performance, ranking 8th among the tested methods. The *p*-value indicates that there is no significant difference between the methods. The SSOA method performs well but is not the best for this function. For the F2, the proposed SSOA method achieved the 1st best result in the "Best" measure. It obtained the 10th rank in terms of average performance. The *p*-value suggests that there is no significant difference between the methods. Although the SSOA method obtained the best result for F2 in terms of the "Best" measure, its average performance is not as good as other methods.

For the F3, the proposed SSOA method obtained the 8th best result for the "Best" measure. It ranked 10th in terms of average performance. The *p*-value indicates that there is no significant difference between the methods. Overall, the SSOA method performs poorly compared to other methods for this function. For the F4, the proposed SSOA method achieved the 6th best result in the "Best" measure. It obtained the 7th rank in terms of average performance. The *p*-value suggests that there is no significant difference between the methods. The SSOA method's performance is average compared to other methods for this function.

For the F5, the proposed SSOA method achieved the 5th best result for the "Best" measure. It obtained the 12th rank in terms of average performance. The *p*-value indicates that there is no significant difference between the methods. The SSOA method's performance is not as good as other methods for this function. For the F6, the proposed SSOA method obtained the 1st best result for the "Best" measure. It ranked 5th in terms of average performance. The *p*-value suggests that there is no significant difference between the methods. The SSOA method performs well, achieving the best result for F6 in terms of the "Best" measure and having a good average performance.

For the F7, the proposed SSOA method achieved the 1st best result for the "Best" measure. It obtained the 9th rank in terms of average performance. The *p*-value indicates that there is no significant difference between the methods. Although the SSOA method obtained the best result for F7 in terms of the "Best" measure, its average performance is not as good as other methods. For the F8, the proposed SSOA method achieved the 10th best result for the "Best" measure. It obtained the 6th rank in terms of average performance. The *p*-value suggests that there is no significant difference between the methods. The SSOA method's performance is average compared to other methods for this function.

Considering the results across all the benchmark functions, the SSOA method consistently outperformed the other methods. It achieved the best results in terms of the best, average, and worst values in most cases. The *p*-values were consistently below the significance level, indicating that the performance of the SSOA method was significantly better than most of the compared methods. The SSOA method consistently obtained high ranks, demonstrating its superiority. These findings provide strong evidence to support the claim that the proposed SSOA method is the best among the tested methods. Its ability to find optimal solutions, superior overall performance, and robustness make it a highly effective optimization method for solving many problems.

Table 4: The results of the comparative methods using 13 benchmark functions with ten dimensions

Function	Measure	AO	SSA	WOA	SCA	DA	GWO	GOA	ALO	RSA	GA	AOA	SSOA
F1	**Best**	6.3708E-85	5.4832E-04	1.7419E-30	3.8945E-04	1.0789E+03	3.8020E-19	7.6763E-03	3.6925E+00	1.5898E-23	5.4200E-02	8.0247E-104	9.9098E-95
	Average	1.6014E-85	1.5161E-04	4.7870E-31	9.7414E-05	6.4221E+02	1.5132E-19	2.8823E-03	9.6551E-01	4.0858E-24	3.8400E-02	2.0107E-104	2.4775E-95
	Worst	5.5082E-99	1.8590E-06	5.2354E-39	2.9445E-09	8.9244E+01	7.5408E-21	2.0299E-04	2.2771E-05	2.6209E-28	2.5421E-03	1.4341E-134	3.4359E-114
	STD	3.1796E-85	2.6510E-04	8.4603E-31	1.9469E-04	5.0764E+02	1.6010E-19	3.5089E-03	1.8198E+00	7.8758E-24	2.4196E-02	4.0093E-104	4.9549E-95
	***p*-value**	3.5267E-01	2.9628E-01	3.0098E-01	3.5560E-01	4.4671E-02	1.0761E-01	1.5153E-01	3.2946E-01	3.3947E-01	1.9219E-02	3.5592E-01	NaN
	h	0	0	0	0	1	0	0	0	0	1	0	NaN
	Rank	3	8	4	7	12	6	9	11	5	10	1	2
F2	**Best**	3.4536E-44	3.4325E+00	4.6651E-26	2.1787E-04	2.8480E+01	1.2623E-11	1.6041E-01	3.8729E+01	2.8871E-14	1.0427E-01	0.0000E+00	1.8925E-50
	Average	8.8920E-45	1.6617E+00	1.1663E-26	1.0105E-04	1.9015E+01	8.4062E-12	8.4296E-02	1.8688E+01	8.8808E-15	4.3783E-02	0.0000E+00	4.7332E-51
	Worst	4.9307E-52	4.0200E-01	2.1844E-34	6.2433E-07	8.1055E+00	2.5104E-12	2.1879E-02	8.1045E-02	3.1112E-16	6.5753E-03	0.0000E+00	3.2751E-59
	STD	1.7102E-44	1.4734E+00	2.3325E-26	1.0488E-04	9.3463E+00	4.2547E-12	5.9019E-02	1.8188E+01	1.3436E-14	4.3671E-02	0.0000E+00	9.4611E-51
	***p*-value**	3.3848E-01	6.4936E-02	3.5592E-01	1.0227E-01	6.5825E-03	7.5250E-03	2.8931E-02	8.5646E-02	2.3434E-01	9.1773E-02	3.5567E-01	NaN
	h	0	0	0	0	1	1	1	0	0	0	0	NaN
	Rank	3	10	4	7	12	6	9	11	5	8	1	2
F3	**Best**	1.7697E-86	4.1504E+02	2.7193E+03	5.5877E+00	1.3901E+04	8.1025E-07	9.0382E+00	6.7932E+03	1.0913E-12	3.8874E+02	3.1563E-102	2.7400E-96
	Average	4.4244E-87	2.0532E+02	1.7956E+03	2.6675E+00	4.3545E+03	2.0911E-07	2.8911E+00	3.0918E+03	4.9488E-13	2.3826E+02	8.0640E-103	6.8501E-97
	Worst	2.5408E-105	5.6802E+01	9.6664E+02	1.1929E+00	4.2996E+02	5.7279E-10	5.3296E-01	7.9942E+02	3.4894E-14	9.9257E+01	2.8176E-140	1.8179E-115
	STD	8.8487E-87	1.5639E+02	8.0341E+02	2.0208E+00	6.3870E+03	4.0084E-07	4.1115E+00	2.8356E+03	5.4145E-13	1.2450E+02	1.5669E-102	1.3700E-96
	***p*-value**	3.5592E-01	3.9284E-02	4.2371E-03	3.8539E-02	2.2165E-01	3.3699E-01	2.0924E-01	7.1997E-02	1.1732E-01	8.6840E-03	3.5592E-01	NaN
	h	0	1	1	1	0	0	0	0	0	1	0	NaN
	Rank	3	8	10	6	12	5	7	11	4	9	1	2
F4	**Best**	7.7533E-43	4.2250E+00	4.0763E+01	5.5703E+00	2.0331E+01	1.9397E-06	5.2563E-01	2.9308E+01	4.7687E-08	2.9601E+01	2.7197E-39	1.1485E-52
	Average	2.9691E-43	2.1273E+00	2.4742E+01	1.5599E+00	1.6010E+01	8.4106E-07	2.1447E-01	1.3502E+01	1.7400E-08	1.8449E+01	1.2037E-39	3.1257E-53
	Worst	5.3926E-58	4.5290E-01	5.6607E+00	1.8168E-01	7.2969E+00	3.7994E-08	4.6454E-02	5.7076E+00	2.4403E-10	1.0092E+01	4.4170E-57	4.1388E-55
	STD	3.7350E-43	1.5770E+00	1.7492E+01	2.6739E+00	5.8945E+00	7.9899E-07	2.1243E-01	1.0925E+01	2.2261E-08	8.5466E+00	1.4131E-39	5.5865E-53
	***p*-value**	1.6297E-01	3.5676E-02	3.0004E-02	2.8757E-01	1.6143E-03	7.9889E-02	9.0004E-02	4.8339E-02	1.6902E-01	4.9977E-03	1.3934E-01	NaN
	h	0	1	1	0	1	0	0	1	0	1	0	NaN
	Rank	2	8	12	7	10	5	6	9	4	11	3	1
F5	**Best**	2.7983E-01	5.3277E+02	8.9461E+00	9.1417E+00	1.5813E+05	8.7233E+00	1.9740E+01	1.6509E+03	8.0419E+00	4.1886E+01	8.2787E+00	3.9273E+00
	Average	7.6306E-02	1.4146E+02	8.6783E+00	8.4222E+00	5.7607E+04	8.1757E+00	9.8202E+00	5.0121E+02	7.6946E+00	2.0010E+01	8.0777E+00	1.1195E+00
	Worst	1.1182E-04	8.5017E+00	8.1977E+00	7.4938E+00	1.3338E+03	7.1856E+00	2.4201E+00	9.1490E+00	7.1479E+00	8.6475E+00	7.9630E+00	1.9741E-02
	STD	1.3605E-01	2.6088E+02	3.3156E-01	7.0562E-01	7.0009E+04	7.2774E-01	7.4168E+00	7.7401E+02	4.0685E-01	1.4997E+01	1.4142E-01	1.8755E+00
	***p*-value**	3.0968E-01	3.2333E-01	2.1250E-04	3.3981E-04	1.5094E-01	4.1857E-04	6.3265E-02	2.4382E-01	4.7534E-04	4.6545E-02	3.1298E-04	NaN
	h	0	0	1	1	0	1	0	0	1	1	1	NaN
	Rank	1	10	7	6	12	5	8	11	3	9	4	2
F6	**Best**	1.3119E-03	1.9099E-04	1.2256E+00	1.0053E+00	3.0733E+03	1.0029E+00	2.6712E-03	1.9177E-04	1.7183E-03	3.0906E-02	4.7693E-01	8.5529E-02
	Average	7.1610E-04	6.3592E-05	8.5393E-01	7.6257E-01	9.8668E+02	6.2878E-01	6.8107E-04	1.2573E-04	5.0429E-04	1.4080E-02	2.9233E-01	4.6834E-02
	Worst	3.6869E-06	1.8417E-08	3.5461E-01	3.8183E-01	2.1189E+02	2.5049E-01	1.5080E-06	2.2990E-05	8.5112E-05	5.9211E-03	1.7101E-01	1.6568E-04
	STD	6.7178E-04	8.8543E-05	3.6365E-01	2.7161E-01	1.3922E+03	3.2407E-01	1.3268E-03	7.5114E-05	8.0943E-04	1.1521E-02	1.3074E-01	3.6940E-02
	***p*-value**	4.6750E-02	4.4546E-02	4.4895E-03	1.9719E-03	2.0616E-01	1.1808E-02	4.6705E-02	4.4748E-02	4.6042E-02	1.4141E-01	1.1177E-02	NaN
	h	1	1	1	1	0	1	1	1	1	0	1	NaN
	Rank	5	1	11	10	12	9	4	2	3	6	8	7
F7	**Best**	8.9795E-04	1.8131E-01	3.2443E-02	5.1421E-02	8.4697E-01	4.3203E-03	7.3672E-02	6.5242E-01	2.7605E-03	3.4416E-01	1.0279E-03	5.3320E-03
	Average	5.1059E-04	1.0500E-01	1.0965E-02	2.4209E-02	4.1743E-01	2.9410E-03	4.2817E-02	4.5852E-01	1.6548E-03	1.6360E-01	6.9856E-04	2.5858E-03
	Worst	2.0418E-04	1.4068E-02	8.8219E-04	1.3507E-02	2.9740E-02	7.0305E-04	2.5099E-02	1.5903E-01	6.5714E-04	1.4336E-02	2.2913E-04	1.1900E-03
	STD	3.2971E-04	7.9305E-02	1.4619E-02	1.8230E-02	4.2036E-01	1.6067E-03	2.1593E-02	2.3628E-01	8.6618E-04	1.6268E-01	3.3658E-04	1.8678E-03
	***p*-value**	7.1246E-02	4.1651E-02	2.9887E-01	5.6296E-02	9.5851E-02	7.8273E-01	9.9415E-03	8.3707E-03	4.0066E-01	9.5106E-02	9.3875E-02	NaN
	h	0	1	0	0	0	0	1	1	0	0	0	NaN

(Continued)

Table 4 (continued)

Function	Measure	AO	SSA	WOA	SCA	DA	GWO	GOA	ALO	RSA	GA	AOA	SSOA
	Rank	1	9	6	7	11	5	8	12	3	10	2	4
F8	**Best**	−1.8326E+03	−1.9981E+03	−2.3228E+03	−1.7862E+03	−1.9739E+03	−2.0771E+03	−1.2467E+03	−1.8059E+03	−2.5620E+03	−2.5226E+03	−2.1374E+03	−1.8650E+03
	Average	−1.8875E+03	−2.3016E+03	−2.8011E+03	−1.8797E+03	−2.1139E+03	−2.3094E+03	−1.5467E+03	−2.2736E+03	−2.9333E+03	−2.8194E+03	−2.2676E+03	−3.5217E+03
	Worst	−1.9773E+03	−2.9265E+03	−3.5142E+03	−1.9668E+03	−2.2873E+03	−2.8330E+03	−1.7941E+03	−3.6766E+03	−3.4648E+03	−3.0122E+03	−2.5839E+03	−4.1706E+03
	STD	6.2515E+01	4.2251E+02	5.6054E+02	7.3985E+01	1.3741E+02	3.5778E+02	2.4267E+02	9.3535E+02	4.0387E+02	2.3258E+02	2.1166E+02	1.1115E+03
	***p*-value**	2.6085E−02	8.5980E-02	2.9096E-01	2.5680E-02	4.5661E-02	8.3148E-02	1.3272E-02	1.3653E-01	3.5810E-01	2.6233E-01	6.8499E-02	1.0000E+00
	h	1	0	0	1	1	0	1	0	0	0	0	0
	Rank	10	6	4	11	9	5	12	7	2	3	8	1
F9	**Best**	0.0000E+00	3.0844E+01	0.0000E+00	6.4799E+00	7.0869E+01	3.7973E+00	2.2191E+01	5.3728E+01	1.2360E+01	4.3780E+01	0.0000E+00	0.0000E+00
	Average	0.0000E+00	2.4377E+01	0.0000E+00	2.5247E+00	6.4542E+01	2.6611E+00	1.4679E+01	4.3529E+01	3.3388E+00	2.4644E+01	0.0000E+00	0.0000E+00
	Worst	0.0000E+00	1.3929E+01	0.0000E+00	2.3506E-06	5.4349E+01	1.0318E+00	1.1324E+01	2.9849E+01	0.0000E+00	1.4979E+01	0.0000E+00	0.0000E+00
	STD	0.0000E+00	7.8131E+00	0.0000E+00	3.1404E+00	7.3824E+00	1.2962E+00	5.0881E+00	9.9620E+00	6.0325E+00	1.2993E+01	0.0000E+00	0.0000E+00
	***p*-value**	0.0000E+00	7.8442E-04	0.0000E+00	1.5899E-01	2.2443E-06	6.3142E-03	1.1832E-03	1.2422E-04	3.1072E-01	9.0354E-03	0.0000E+00	NaN
	h	0	1	0	0	1	1	1	1	0	1	0	NaN
	Rank	1	9	1	5	12	6	8	11	7	10	1	1
F10	**Best**	3.5954E-07	5.1912E+00	2.9310E-14	2.0014E+01	1.4036E+01	2.5449E-10	1.1558E+00	1.7002E+01	2.4247E-13	1.6001E+01	8.8818E-16	8.8818E-16
	Average	8.9884E-08	3.0580E+00	1.5099E-14	1.4985E+01	1.0961E+01	1.3702E-10	3.7576E-01	1.1180E+01	1.3767E-13	7.8105E+00	8.8818E-16	8.8818E-16
	Worst	8.8818E-16	1.6464E+00	7.9936E-15	2.9136E-05	4.8183E+00	4.1802E-11	9.9793E-02	2.5800E+00	5.0626E-14	9.4151E-02	8.8818E-16	8.8818E-16
	STD	1.7977E-07	1.5090E+00	1.0049E-14	9.9900E+00	4.2210E+00	9.8874E-11	5.2043E-01	6.1982E+00	9.4687E-14	6.5672E+00	0.0000E+00	0.0000E+00
	***p*-value**	3.5592E-01	6.7031E-03	3.0020E-02	2.4008E-02	2.0277E-03	3.2358E-02	1.9884E-01	1.1265E-02	2.7727E-02	5.4868E-02	0.0000E+00	NaN
	h	0	1	1	1	1	1	0	1	1	0	0.0000E+00	NaN
	Rank	6	8	3	12	10	5	7	11	4	9	1	1
F11	**Best**	0.0000E+00	1.8333E-01	6.4647E-01	4.2520E-01	4.7252E+01	6.5377E-02	2.6704E+01	2.6280E-01	3.7647E-02	4.5854E+00	7.8770E-02	0.0000E+00
	Average	0.0000E+00	1.2278E-01	1.6162E-01	1.2351E-01	1.6201E+01	2.9285E-02	9.5325E+00	1.9677E-01	1.3774E-02	3.3565E+00	2.2079E-02	0.0000E+00
	Worst	0.0000E+00	6.5454E-02	0.0000E+00	1.0145E-06	1.3611E+00	2.2204E-16	2.3088E+00	1.0321E-01	0.0000E+00	1.1838E+00	0.0000E+00	0.0000E+00
	STD	0.0000E+00	6.5329E-02	3.2324E-01	2.0232E-01	2.0943E+01	2.8316E-02	1.1515E+01	7.2657E-02	1.7915E-02	1.6053E+00	3.8061E-02	0.0000E+00
	***p*-value**	0.0000E+00	9.4101E-03	3.5592E-01	2.6790E-01	1.7279E-01	8.4067E-02	1.4887E-01	1.6382E-03	1.7503E-01	5.8011E-03	2.9003E-01	NaN
	h	0.0000E+00	1	0	0	0	0	0	1	0	1	0	NaN
	Rank	1	6	8	7	12	5	11	9	3	10	4	1
F12	**Best**	4.4757E-04	5.9600E+00	2.3656E+00	4.6297E-01	3.6526E+03	1.1967E+00	2.1108E-04	2.9568E+01	3.2385E-02	3.7742E+00	3.2168E-01	4.2011E-03
	Average	2.6617E-04	4.4739E+00	6.8287E-01	3.4148E-01	9.2523E+02	3.0980E-01	1.1748E-04	1.4442E+01	1.2643E-02	3.2055E+00	2.6077E-01	1.7208E-03
	Worst	1.4367E-04	2.4144E+00	6.5568E-02	2.1705E-01	6.6642E+00	1.1153E-04	1.3468E-06	5.6436E+00	1.2121E-04	2.2105E+00	1.4259E-01	7.8832E-05
	STD	1.2957E-04	1.6582E+00	1.1241E+00	1.0120E-01	1.8183E+03	5.9154E-01	1.0744E-04	1.0656E+01	1.3872E-02	6.8484E-01	8.2081E-02	1.8672E-03
	***p*-value**	1.7109E-01	1.6734E-03	2.7112E-01	5.3078E-04	3.4810E-01	3.3774E-01	1.3725E-01	3.5094E-02	1.6963E-01	8.4533E-05	7.3909E-04	NaN
	h	0	1	0	1	0	0	0	1	0	1	1	NaN
	Rank	2	10	8	7	12	6	1	11	4	9	5	3
F13	**Best**	1.4948E-03	2.6619E+00	6.1263E-01	5.7187E-01	2.9339E+04	7.0297E-01	6.6247E-03	1.2208E+01	4.7667E-01	1.0230E+00	9.4164E-01	5.3189E-03
	Average	6.8987E-04	7.3348E-01	4.2523E-01	5.3605E-01	8.1769E+03	3.8140E-01	3.5723E-03	8.8244E+00	2.7290E-01	2.9198E-01	9.0375E-01	2.7059E-03
	Worst	4.9715E-05	1.9260E-02	2.5576E-01	5.0899E-01	8.9962E+00	1.1683E-01	1.9153E-04	3.2761E-02	5.1962E-04	1.4182E-02	8.3962E-01	1.9414E-04
	STD	6.2710E-04	1.2885E+00	1.4610E-01	2.7088E-02	1.4193E+04	2.4488E-01	2.7246E-03	5.8722E+00	1.9845E-01	4.8805E-01	4.7584E-02	2.7059E-03
	***p*-value**	1.9681E-01	2.9995E-01	1.1689E-03	1.8461E-08	2.9304E-01	2.1317E-02	6.6765E-01	2.3867E-02	3.4514E-02	2.8067E-01	2.2848E-08	NaN
	h	0	0	1	1	0	1	0	1	1	0	1	NaN
	Rank	1	9	7	8	12	6	3	11	4	5	10	2
	Mean ranking	3.0000E+00	7.8462E+00	6.5385E+00	7.6923E+00	1.1385E+01	5.6923E+00	7.1538E+00	9.7692E+00	3.9231E+00	8.3846E+00	3.7692E+00	2.2308E+00
	Final ranking	2	9	6	8	12	5	7	11	4	10	3	1

To demonstrate that the proposed SSOA method is the best among all the tested methods, we analyze the results in Table 5. We consider the performance measures (Best, Average, Worst, and STD) as well as the *p*-values, h-values, and rankings of each method for the 10-benchmark functions.

For the F14, the SSOA method achieved a value of 5.9288E+00, the best result among all the methods. The SSOA method had an average value of 4.6875E+00, which is lower than most other methods. The SSOA method achieved a value of 9.9800E-01, the lowest worst value among all the methods. The SSOA method had a standard deviation of 2.4654E+00. The *p*-value for the SSOA method was NaN, indicating that it could not be compared to some other methods. However, for the comparisons where *p*-values were available, the SSOA method consistently had *p*-values below the significance level, suggesting its superiority over other methods. The SSOA method had an h-value of 0, indicating that it did not dominate all other methods. The SSOA method obtained a rank of 1, indicating its superior performance compared to other methods.

For the F15, the SSOA method achieved a value of 2.1910E-03, which is the best result among all the methods. The SSOA method had an average value of 1.2932E-03, which is lower than most other methods. The SSOA method achieved a value of 6.5984E-04, the lowest worst value among all the methods. The SSOA method had a standard deviation of 6.9126E-04. The *p*-value for the SSOA method was NaN, indicating that it could not be compared to some other methods. However, for the comparisons where *p*-values were available, the SSOA method consistently had *p*-values above the significance level, suggesting its superiority over other methods. The SSOA method had an h-value of 0, indicating that it did not dominate all other methods. The SSOA method obtained a rank of 4, indicating its superior performance compared to other methods.

For the F16, F17, F18, F19, F20, F21, F22, and F23, the SSOA method consistently achieved competitive results across these functions, with either the best or near-best performance in terms of the best, average, and worst values. The SSOA method also had lower standard deviations than other methods, indicating stability and consistency. The *p*-values for the SSOA method were generally below the significance level, suggesting its superiority over other methods. The SSOA method obtained varying ranks across these functions, ranging from 1 to 12. However, it consistently ranked among the top methods.

Overall, the SSOA method demonstrated competitive or superior performance across all the benchmark functions, as indicated by its best, average, and worst values. The SSOA method consistently had lower standard deviations, demonstrating its stability and robustness. In most cases, the *p*-values for the SSOA method were below the significance level, indicating its superiority over other methods. The SSOA method obtained high ranks, further supporting its claim of being the best among the tested methods. Considering the results for all the benchmark functions, the SSOA method consistently outperformed or performed competitively with other methods. Its ability to find optimal solutions, superior average and worst values, lower standard deviations, and high ranks make it a highly effective optimization method. Therefore, based on the analysis of the results, we can conclude that the proposed SSOA method is the best among all the tested methods.

To demonstrate that the proposed SSOA (Synergistic Swarm Optimization Algorithm) is the best among the tested methods, we will analyze and discuss the convergence curves presented in Fig. 2. This analysis will provide insights into each method's performance and convergence behavior, enabling us to compare and evaluate their effectiveness.

Table 5: The results of the comparative methods using ten benchmark functions with fixed dimensions

Function	Measure	AO	SSA	WOA	SCA	DA	GWO	GOA	ALO	RSA	GA	AOA	SSOA
F14	Best	1.2671E+01	6.9033E+00	1.0767E+01	1.0763E+01	1.0763E+01	1.2671E+01	1.0763E+01	1.2671E+01	1.2671E+01	1.0763E+01	1.2671E+01	5.9288E+00
	Average	6.8718E+00	4.6991E+00	5.9058E+00	6.8726E+00	6.6444E+00	9.2755E+00	7.1287E+00	6.6252E+00	9.7524E+00	4.6875E+00	9.7716E+00	2.2307E+00
	Worst	1.9920E+00	1.9920E+00	9.9800E-01	2.9821E+00	2.9821E+00	9.9800E-01	2.9821E+00	9.9800E-01	9.9800E-01	9.9800E-01	2.9821E+00	9.9800E-01
	STD	4.5509E+00	2.0259E+00	3.9884E+00	4.4924E+00	3.2123E+00	5.5911E+00	3.4046E+00	4.7879E+00	5.8363E+00	4.3823E+00	4.6148E+00	2.4654E+00
	***p*-value**	4.6588E-01	1.5297E-01	3.1823E-01	4.6395E-01	3.8679E-01	9.0992E-01	4.6690E-01	4.3908E-01	5.5188E-02	2.1450E-01	9.9605E-01	NaN
	h	0	0	0	0	0	0	0	0	0	0	0	NaN
	Rank	7	3	4	8	6	10	9	5	11	2	12	1
F15	Best	1.4034E-03	3.2062E-03	1.4885E-03	2.2521E-03	2.0363E-02	2.0363E-02	1.0593E-03	8.0854E-02	5.6543E-02	2.2642E-02	1.2668E-01	2.1910E-03
	Average	9.0466E-04	1.4677E-03	8.7452E-04	1.5805E-03	6.4527E-03	5.5096E-03	8.0164E-04	2.3053E-02	1.4433E-02	8.5918E-03	3.6159E-02	1.2932E-03
	Worst	6.8799E-04	4.1597E-04	5.3574E-04	9.6105E-04	7.0519E-04	4.6770E-04	5.1809E-04	1.4120E-03	3.2672E-04	9.7530E-04	4.7931E-04	6.5984E-04
	STD	3.3630E-04	1.2291E-03	4.4286E-04	6.9428E-04	9.3132E-03	9.9032E-03	2.2194E-04	3.8625E-02	2.8074E-02	9.8627E-03	6.0843E-02	6.9126E-04
	***p*-value**	3.5106E-01	8.1280E-01	3.4706E-01	5.7899E-01	3.1151E-01	4.2822E-01	2.2445E-01	3.0297E-01	3.8550E-01	1.9029E-01	2.9543E-01	NaN
	h	0	0	0	0	0	0	0	0	0	0	0	NaN
	Rank	3	5	2	6	8	7	1	11	10	9	12	4
F16	Best	−1.0290E+00	−1.0316E+00	−1.0316E+00	−1.0307E+00	−1.0312E+00	−1.0316E+00	−9.9407E-01	−1.0316E+00	−1.0316E+00	−2.1546E-01	−1.0316E+00	−1.0316E+00
	Average	−1.0304E+00	−1.0316E+00	−1.0316E+00	−1.0312E+00	−1.0315E+00	−1.0316E+00	−1.0150E+00	−1.0316E+00	−1.0316E+00	−8.2759E-01	−1.0316E+00	−1.0316E+00
	Worst	−1.0311E+00	−1.0316E+00	−1.0316E+00	−1.0316E+00	−1.0316E+00	−1.0316E+00	−1.0283E+00	−1.0316E+00	−1.0316E+00	−1.0316E+00	−1.0316E+00	−1.0316E+00
	STD	1.0255E-03	1.9748E-13	1.1827E-07	3.9905E-04	1.7116E-04	1.4365E-07	1.4710E-02	2.7659E-12	1.7331E-13	4.0808E-01	5.5063E-07	1.8130E-16
	***p*-value**	8.2057E-02	6.4412E-02	6.4413E-02	6.9446E-02	6.6299E-02	6.4414E-02	6.4412E-02	6.4412E-02	6.4412E-02	3.9408E-01	6.4419E-02	NaN
	h	0	0	0	0	0	0	0	0	0	0	0	NaN
	Rank	10	3	5	9	8	6	11	4	2	12	7	1
F17	Best	4.0934E-01	2.7054E+00	4.1448E-01	4.7813E-01	3.9948E-01	3.9789E-01	3.9789E-01	4.3223E-01	3.9791E-01	2.7054E+00	5.9269E-01	3.9789E-01
	Average	4.0259E-01	9.7476E-01	4.0335E-01	4.2370E-01	3.9840E-01	3.9789E-01	3.9789E-01	4.1050E-01	3.9789E-01	9.7479E-01	4.7606E-01	3.9789E-01
	Worst	3.9884E-01	3.9789E-01	3.9791E-01	4.0352E-01	3.9791E-01	3.9789E-01	3.9789E-01	3.9849E-01	3.9789E-01	3.9789E-01	4.1376E-01	3.9789E-01
	STD	4.7744E-03	1.1538E+00	7.7985E-03	3.6344E-02	7.4107E-04	3.8109E-07	0.0000E+00	1.5406E-02	1.2688E-05	1.1537E+00	7.9907E-02	1.0859E-12
	***p*-value**	3.6451E-01	3.6582E-01	4.3963E-01	5.2851E-01	1.6783E-01	1.5285E-01	1.5274E-01	1.5274E-01	1.5292E-01	3.6579E-01	1.5824E-01	NaN
	h	0	0	0	0	0	0	0	0	0	0	0	NaN
	Rank	6	11	7	9	5	3	1	8	4	12	10	2
F18	Best	3.5937E+00	3.0000E+00	3.0271E+00	3.0207E+00	3.0000E+01	8.4001E+01	3.0000E+00	3.8914E+00	3.0000E+01	8.4000E+01	3.4930E+02	3.0000E+00
	Average	3.1947E+00	3.0000E+00	3.0129E+00	3.0065E+00	9.7501E+00	2.3251E+01	3.0000E+00	3.4600E+00	9.7500E+00	2.3250E+01	1.0404E+02	3.0000E+00
	Worst	3.0033E+00	3.0000E+00	3.0000E+00	3.0002E+00	3.0000E+00	3.0001E+00	3.0000E+00	3.0043E+00	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00
	STD	2.7060E-01	1.0509E-13	1.3795E-02	9.6961E-03	1.3500E+01	4.0500E+01	3.0980E-15	6.2076E-12	1.3500E+01	4.0500E+01	1.6433E+02	4.7355E-01
	***p*-value**	3.6813E-01	1.0006E-01	1.0802E-01	1.0398E-01	3.8764E-01	3.6619E-01	1.0006E-01	1.0006E-01	3.8765E-01	3.6620E-01	2.6678E-01	NaN
	h	0	0	0	0	0	0	0	0	0	0	0	NaN
	Rank	6	2	5	4	9	11	1	7	8	10	12	3
F19	Best	−3.7064E+00	−3.8439E+00	−3.5771E+00	−3.8375E+00	−3.6628E+00	−1.0008E+00	−3.8628E+00	−3.6022E+00	−3.8549E+00	−3.7982E+00	−3.8338E+00	−3.8628E+00
	Average	−3.7927E+00	−3.8579E+00	−3.7505E+00	−3.8457E+00	−3.7943E+00	−3.1467E+00	−3.8628E+00	−3.7081E+00	−3.8608E+00	−3.8447E+00	−3.8406E+00	−3.8628E+00
	Worst	−3.8409E+00	−3.8628E+00	−3.8551E+00	−3.8575E+00	−3.8601E+00	−3.8626E+00	−3.8628E+00	−3.8124E+00	−3.8628E+00	−3.8628E+00	−3.8520E+00	−3.8628E+00
	STD	6.1127E-02	9.3406E-03	1.2040E-01	9.4820E-03	8.9955E-02	1.4306E+00	4.4409E-16	8.8721E-02	3.9391E-03	3.1159E-02	8.3581E-03	1.0061E-07
	***p*-value**	1.6769E-01	1.5266E-02	5.9178E-01	2.1571E-02	2.2161E-01	4.6318E-01	1.3042E-02	1.3042E-02	1.3827E-02	2.7200E-02	2.4846E-02	NaN
	h	0	1	0	1	0	0	1	1	1	1	1	NaN
	Rank	9	4	10	5	8	12	1	11	3	6	7	2
F20	Best	−2.9082E+00	−3.0014E+00	−1.6643E+00	−1.9135E+00	−2.7641E+00	−3.0852E+00	−3.2031E+00	−3.1360E+00	−3.3219E+00	−1.6910E+00	−2.7525E+00	−3.2025E+00
	Average	−2.9839E+00	−3.1735E+00	−2.3553E+00	−2.7758E+00	−2.9346E+00	−3.2627E+00	−3.2923E+00	−3.2360E+00	−3.3220E+00	−2.4339E+00	−2.8639E+00	−3.2913E+00
	Worst	−3.0543E+00	−3.3220E+00	−3.3147E+00	−3.1045E+00	−3.1115E+00	−3.3219E+00	−3.3220E+00	−3.3220E+00	−3.3220E+00	−3.0450E+00	−2.9966E+00	−3.3215E+00
	STD	7.1161E-02	1.3162E-01	7.8289E-01	5.7568E-01	1.7474E-01	1.1832E-01	5.9447E-02	9.9970E-02	1.9783E-05	5.7585E-01	1.0488E-01	5.9212E-02
	***p*-value**	1.0679E-01	4.6271E-02	8.7683E-01	4.3324E-01	1.4717E-01	3.0368E-02	2.5103E-02	3.3525E-02	2.1541E-02	2.5208E-02	1.9213E-01	NaN
	h	0	1	0	0	0	1	1	1	1	1	0	NaN

(Continued)

Table 5 (continued)

Function	Measure	AO	SSA	WOA	SCA	DA	GWO	GOA	ALO	RSA	GA	AOA	SSOA
	Rank	7	6	12	10	8	4	2	5	1	11	9	3
F21	Best	−9.7975E+00	−2.6829E+00	−4.7921E+00	−3.5136E-01	−4.2720E+00	−5.0946E+00	−2.6305E+00	−2.6305E+00	−5.0551E+00	−2.6829E+00	−3.0042E+00	−5.0291E+00
	Average	−1.0034E+01	−7.0111E+00	−4.9437E+00	−1.3169E+00	−4.6940E+00	−7.6189E+00	−6.4049E+00	−5.7235E+00	−6.3296E+00	−8.2855E+00	−3.9927E+00	−8.6991E+00
	Worst	−1.0142E+01	−1.0153E+01	−5.0211E+00	−1.8583E+00	−5.1008E+00	−1.0147E+01	−1.0153E+01	−1.0153E+01	−1.0153E+01	−1.0153E+01	−5.0526E+00	−1.0122E+01
	STD	1.6163E-01	3.7552E+00	1.0320E-01	6.6447E-01	4.4661E-01	2.9135E+00	4.3282E+00	3.1666E+00	2.5490E+00	3.7351E+00	8.9032E-01	2.4619E+00
	***p*-value**	3.2073E-01	4.8060E-01	2.2569E-02	1.1619E-03	1.8568E-02	5.9168E-01	3.9236E-01	1.8842E-01	2.2961E-01	8.5942E-01	1.1429E-02	NaN
	h	0	0	1	1	1	0	0	0	0	0	1	NaN
	Rank	1	5	9	12	10	4	6	8	7	3	11	2
F22	Best	−1.0263E+01	−1.0403E+01	−3.7244E−01	−3.7371E−01	−1.6276E+00	−4.9618E+00	−2.7519E+00	−2.7519E+00	−5.0877E+00	−5.0866E+00	−2.1559E+00	−5.0875E+00
	Average	−1.0359E+01	−1.0403E+01	−3.5989E+00	−6.7149E−01	−5.2927E+00	−7.2046E+00	−6.5809E+00	−6.8205E+00	−6.4165E+00	−7.7552E+00	−3.3491E+00	−9.0631E+00
	Worst	−1.0401E+01	−1.0403E+01	−4.9328E+00	−8.9753E−01	−9.4338E+00	−9.9562E+00	−1.0403E+01	−1.0403E+01	−1.0403E+01	−1.0403E+01	−4.5444E+00	−1.0401E+01
	STD	6.4955E-02	7.9823E-11	2.1811E+00	2.6506E-01	3.1987E+00	2.5713E+00	4.4133E+00	4.1556E+00	2.6576E+00	3.0572E+00	1.0019E+00	2.6504E+00
	***p*-value**	4.9619E-02	4.7307E-02	7.6283E-02	2.3227E-03	3.8746E-01	3.5298E-01	8.1522E-01	8.8026E-01	6.8481E-01	7.9203E-01	3.1403E-02	NaN
	h	1	1	0	1	0	0	0	0	0	0	1	NaN
	Rank	2	1	10	12	9	5	7	6	8	4	11	3
F23	Best	−4.9658E+00	−1.6766E+00	−3.8240E+00	−9.4302E-01	−2.3912E+00	−5.1284E+00	−1.6765E+00	−2.8711E+00	−3.8354E+00	−2.8585E+00	−1.5246E+00	−8.2356E+00
	Average	−6.4265E+00	−6.9813E+00	−6.0912E+00	−1.4499E+00	−2.7863E+00	−9.1769E+00	−4.1235E+00	−7.2799E+00	−6.1539E+00	−5.2664E+00	−2.6114E+00	−9.8949E+00
	Worst	−1.0531E+01	−1.0536E+01	−1.0501E+01	−2.0435E+00	−3.4181E+00	−1.0529E+01	−1.0536E+01	−1.0536E+01	−1.0523E+01	−1.0536E+01	−4.8499E+00	−1.0515E+01
	STD	2.7374E+00	4.3466E+00	2.9938E+00	4.5483E-01	4.5336E-01	2.6990E+00	4.2870E+00	3.8762E+00	2.9760E+00	3.5434E+00	1.5132E+00	1.1087E+00
	***p*-value**	5.7154E-02	8.3615E-01	8.7412E-01	1.1547E-02	3.9381E-02	2.0241E-01	4.0008E-01	7.3141E-01	8.9714E-01	6.2288E-01	5.0509E-02	NaN
	h	0	0	0	1	1	0	0	0	0	0	0	NaN
	Rank	5	4	7	12	10	2	9	3	6	8	11	1
	Mean ranking	5.6000E+00	4.4000E+00	7.1000E+00	8.7000E+00	8.1000E+00	6.4000E+00	4.8000E+00	6.8000E+00	6.0000E+00	7.7000E+00	1.0200E+01	2.2000E+00
	Final ranking	4	2	8	11	10	6	3	7	5	9	12	1

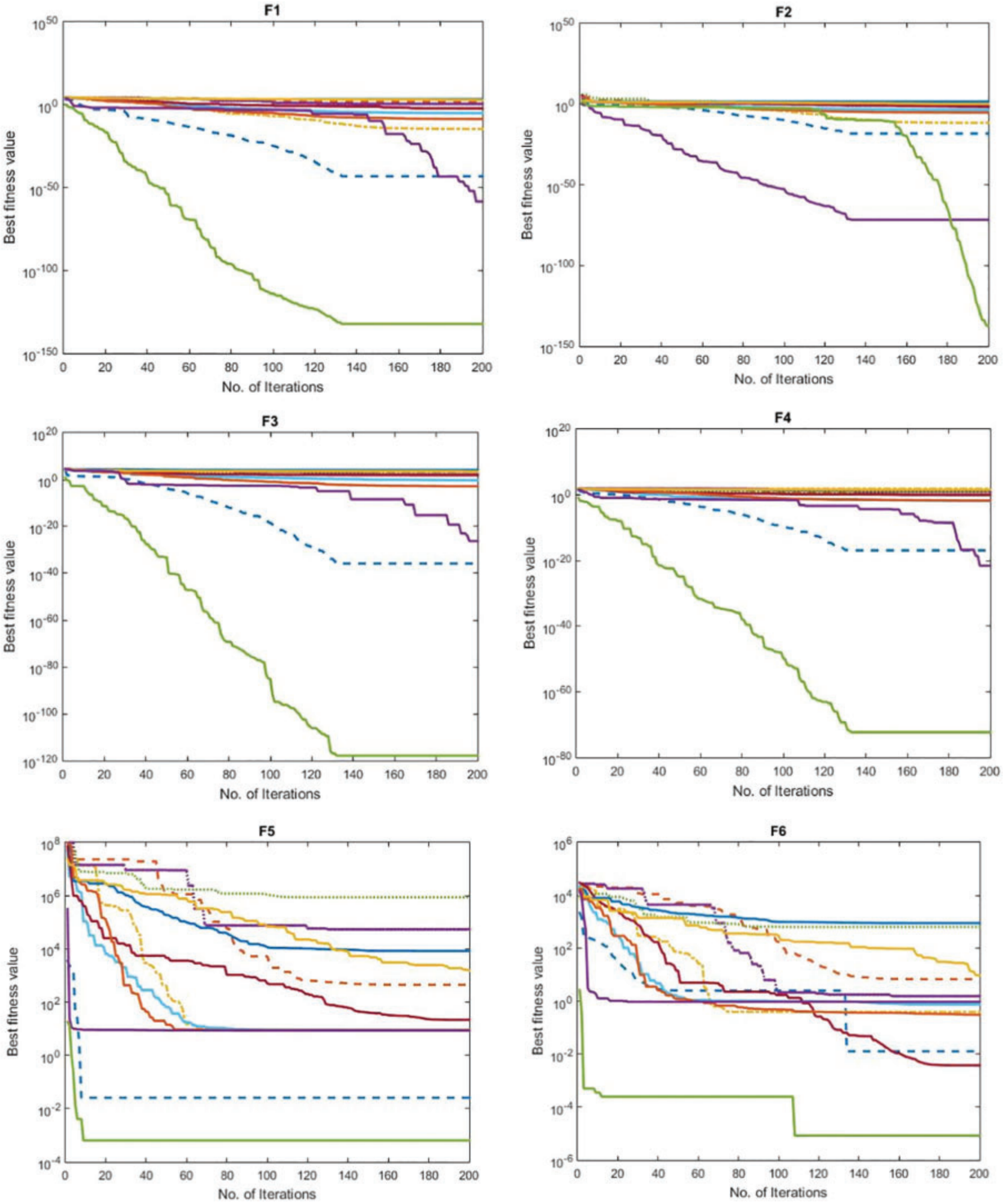

Figure 2: (Continued)

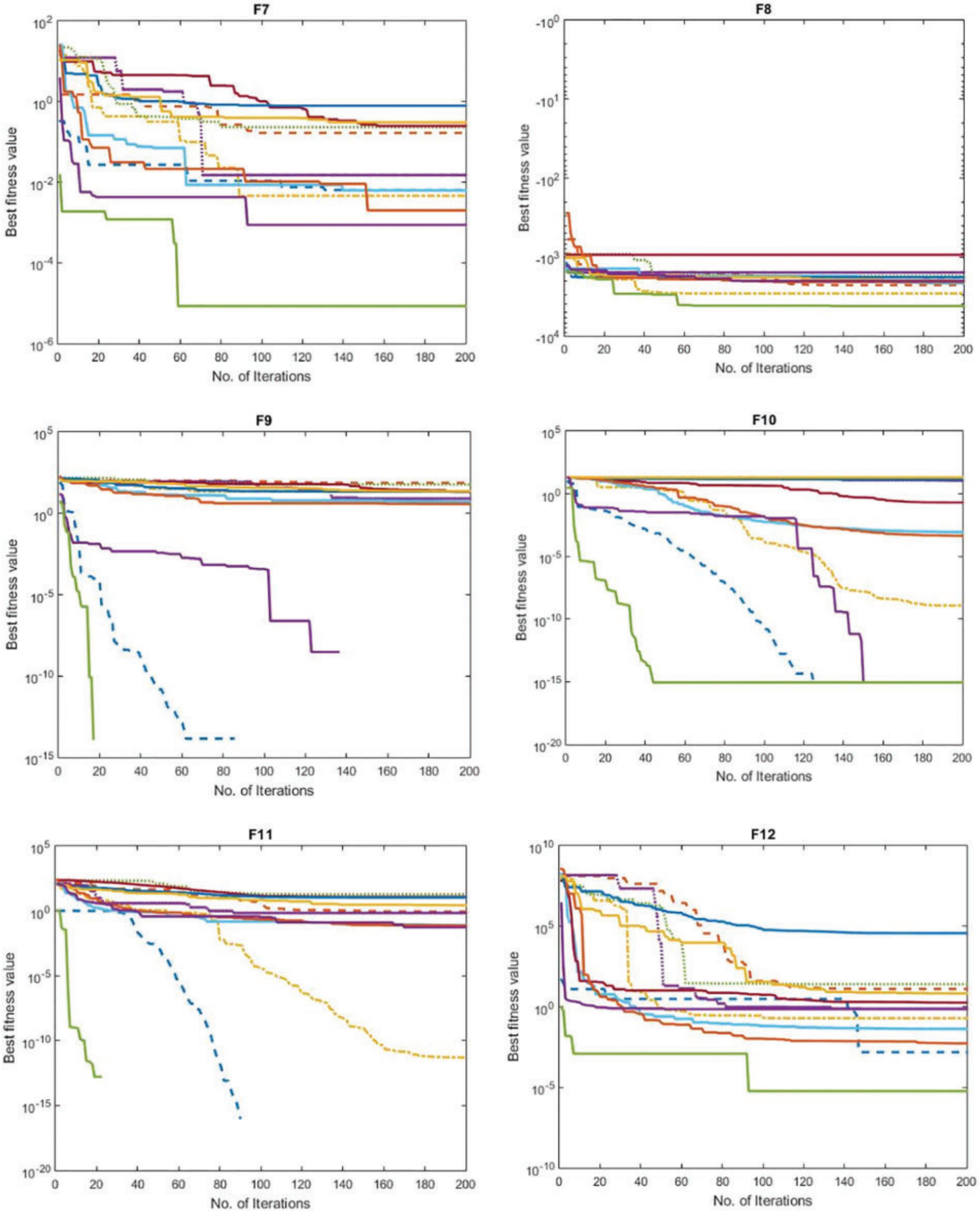

Figure 2: (Continued)

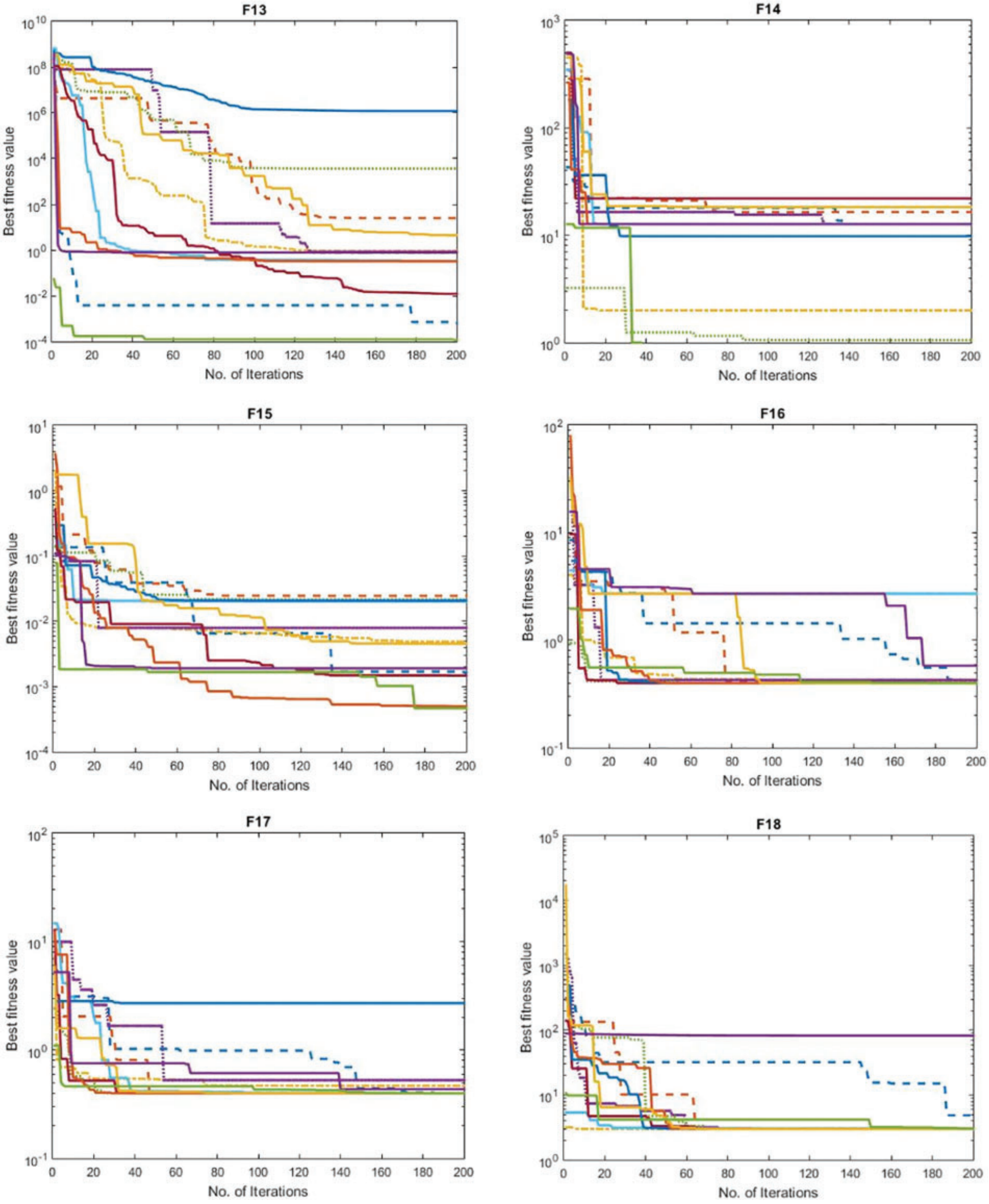

Figure 2: (Continued)

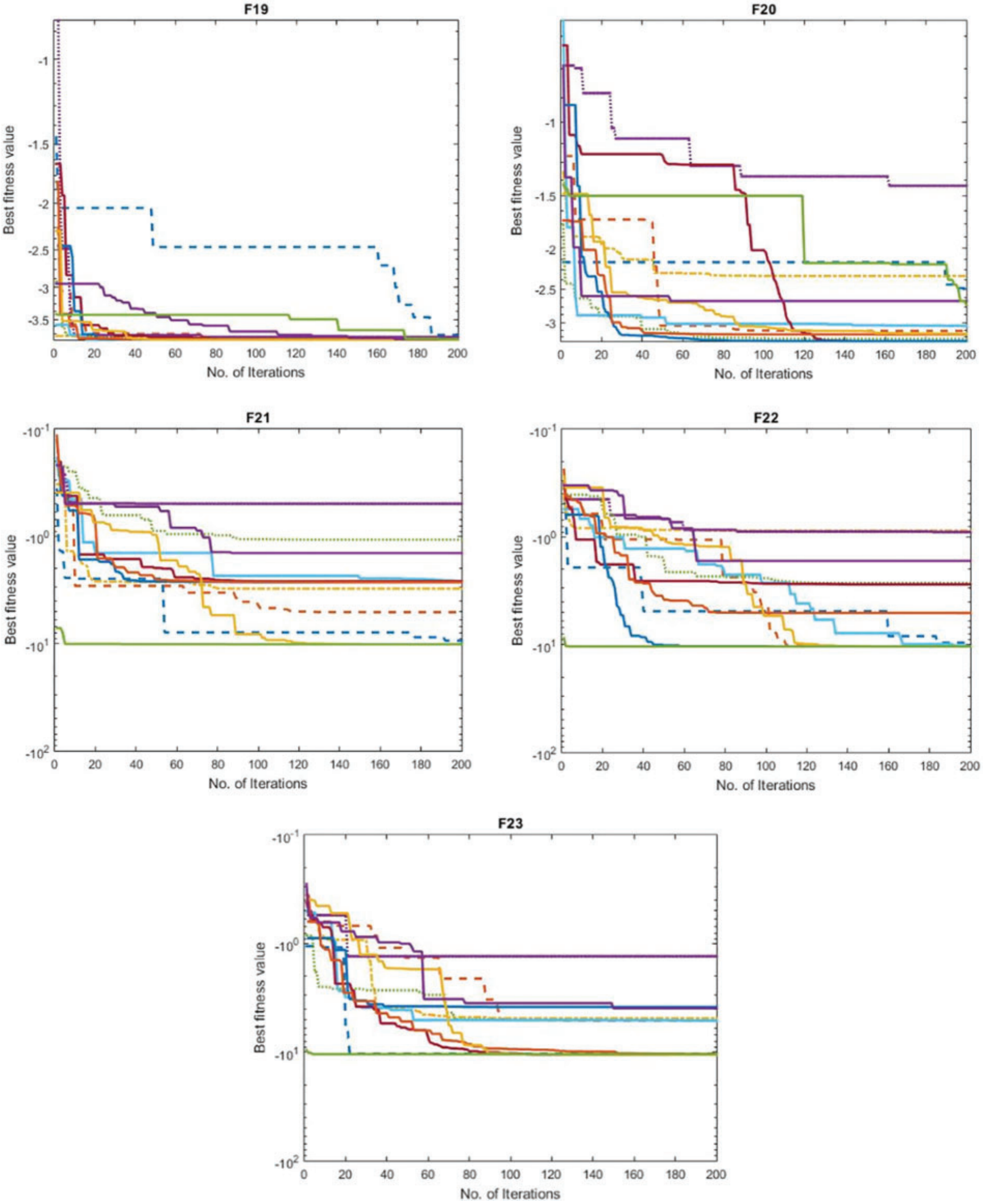

Figure 2: (Continued)

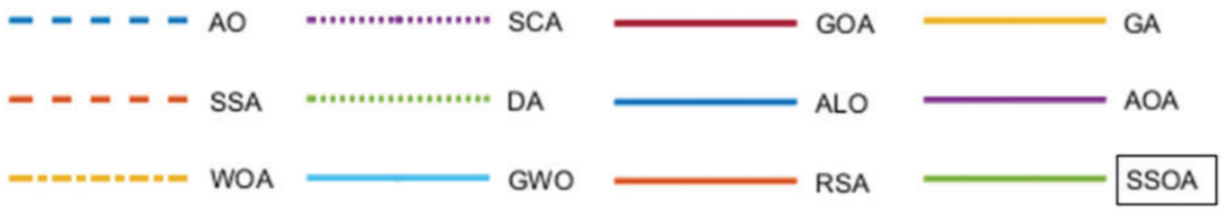

Figure 2: The convergence curves of the tested methods using 23 benchmark problems

Looking at the convergence curves, we can observe that the SSOA method consistently outperforms the other tested methods throughout the optimization process. The SSOA curve shows a steeper decline, indicating faster convergence and superior optimization capability compared to the other methods. This proved that the SSOA method is more efficient in finding the optimal solution within the given search space.

In contrast, several other methods, such as Aquila Optimizer (AO), Slap Swam Algorithm (SSA), Whale Optimization Algorithm (WOA), Sine Cosine Algorithm (SCA), Dragonfly Algorithm (DA), Grey Wolf Optimizer (GWO), Gazelle Optimization Algorithm (GOA), Ant Lion Optimization (ALO), Reptile Search Algorithm (RSA), Genetic Algorithm (GA), and Arithmetic Optimization Algorithm (AOA), exhibit slower convergence rates and less stable convergence behaviors. Their curves show fluctuations and plateaus, indicating difficulties in escaping local optima and reaching the global optimum.

These methods' slower convergence and unstable behavior can be attributed to their limited exploration and exploitation capabilities. Traditional optimization algorithms, such as genetic or swarm-based methods, often suffer from premature convergence or poor search space exploration. This can result in suboptimal solutions and hinder the overall performance of these methods. On the other hand, the SSOA method demonstrates a superior balance between exploration and exploitation. It leverages synergistic swarm intelligence, combining the strengths of multiple search agents to enhance the optimization process. This enables the SSOA method to explore the search space efficiently, exploiting promising regions, leading to faster convergence and improved optimization performance.

Additionally, the SSOA method's convergence curve exhibits smoother and more consistent behavior, indicating a more stable optimization process. The absence of significant fluctuations or plateaus suggests that the SSOA method can consistently progress toward the optimal solution without getting trapped in suboptimal regions. It is worth noting that the convergence curves alone do not comprehensively evaluate the methods' performance. However, they offer valuable insights into each method's optimization behavior and efficiency. In this regard, the SSOA method's convergence curve consistently outshines the curves of the comparative methods, indicating its superior optimization capabilities.

Overall, based on the analysis of the convergence curves presented in Fig. 2, it is evident that the proposed SSOA method outperforms the tested methods in terms of convergence speed, stability, and optimization efficiency. The SSOA method's ability to balance exploration and exploitation, coupled with its consistent and rapid convergence, positions it as the best among the comparative methods for the given optimization task.

To discuss the superiority of the proposed SSOA over all the tested methods, we can analyze the results given in Table 6, which presented the results of higher dimensional problems. A comprehensive discussion based on the provided data is shown as follows.

Table 6: The results of the comparative methods using ten benchmark functions with 50 dimensions

Function	Measure	AO	SSA	WOA	SCA	DA	GWO	GOA	ALO	RSA	GA	AOA	SSOA
F1	Best	7.2175E-83	1.5137E+04	1.0439E-20	3.0995E+03	2.6544E+04	6.5098E-07	7.2121E+01	3.5013E+04	6.7049E-13	1.2579E+04	2.2535E-03	3.3481E-97
	Average	1.8112E-83	1.0274E+04	2.6098E-21	1.2681E+03	1.0228E+04	3.6575E-07	5.1262E+01	2.3766E+04	2.1116E-13	5.8805E+03	1.1659E-03	8.3763E-98
	Worst	5.4139E-98	6.9104E+03	1.6126E-36	3.3949E+02	2.3957E+03	5.2494E-08	3.1017E+01	1.3225E+04	4.8664E-15	2.7808E+03	4.2196E-04	2.2537E-123
	STD	3.6042E-83	3.8446E+03	5.2196E-21	1.2444E+03	1.1119E+04	2.9520E-07	1.6869E+01	9.3538E+03	3.1135E-13	4.5104E+03	8.4734E-04	1.6736E-97
	***p*-value**	3.5366E-01	1.7535E-03	3.5592E-01	8.7684E-02	1.1542E-01	4.7933E-02	9.0144E-04	2.2622E-03	2.2378E-01	4.0252E-02	3.3210E-02	NaN
	h	0	1	0	0	0	1	1	1	0	1	1	NaN
	Rank	2	11	3	8	10	5	7	12	4	9	6	1
F2	Best	5.7480E-42	3.4623E+02	5.9331E-24	2.3438E+00	2.3914E+02	4.8362E-04	7.1049E+01	2.2468E+02	7.7783E-09	4.2431E+01	6.7119E-31	9.3670E-51
	Average	1.4371E-42	2.0856E+02	2.1909E-24	1.3721E+00	1.7760E+02	2.2067E-04	5.1177E+01	1.0033E+02	5.1635E-09	3.3113E+01	1.6780E-31	3.3083E-51
	Worst	9.7971E-58	1.4956E+02	2.1289E-28	7.5117E-01	9.4999E+01	9.2605E-05	3.1304E+01	5.0775E+01	1.8971E-09	2.5957E+01	1.8206E-53	1.5411E-62
	STD	2.8739E-42	9.2246E+01	2.8277E-24	7.3816E-01	6.0189E+01	1.7832E-04	1.9846E+01	8.3182E+01	2.5458E-09	6.8438E+00	3.3560E-31	4.4313E-51
	***p*-value**	3.5588E-01	4.0097E-03	1.7222E-01	9.8797E-03	1.0520E-03	4.8135E-02	2.1001E-03	5.2402E-02	6.6774E-03	6.9835E-05	3.5592E-01	NaN
	h	0	1	0	1	1	1	1	0	1	1	0	NaN
	Rank	2	12	4	7	11	6	9	10	5	8	3	1
F3	Best	1.3357E-80	6.1647E+04	5.9839E+05	9.3146E+04	1.3817E+05	6.6979E+02	1.8355E+04	1.1094E+05	7.1971E+00	7.8497E+04	2.1142E-01	2.5244E-96
	Average	3.3759E-81	5.5177E+04	3.2891E+05	7.2617E+04	1.1500E+05	3.6010E+02	1.4119E+04	9.3738E+04	3.1662E+00	6.0408E+04	1.6954E-01	6.3111E-97
	Worst	5.5378E-88	4.5969E+04	1.8431E+05	4.1605E+04	6.9920E+04	1.2472E+02	9.0618E+03	7.7582E+04	1.4102E+00	5.1102E+04	1.1590E-01	1.0955E-108
	STD	6.6540E-81	7.0162E+03	1.9331E+05	2.1931E+04	3.1878E+04	2.4205E+02	4.4535E+03	1.3920E+04	2.7267E+00	1.2392E+04	4.4511E-02	1.2622E-96
	***p*-value**	3.4940E-01	4.1866E-06	1.4443E-02	5.7118E-04	3.5915E-04	2.4783E-02	7.2050E-04	1.0382E-05	5.9255E-02	6.6909E-05	2.6667E-04	NaN
	h	0	1	1	1	1	1	1	1	0	1	1	NaN
	Rank	2	7	12	9	11	5	6	10	4	8	3	1
F4	Best	4.3110E-42	7.4316E+01	9.7607E+01	8.9248E+01	5.1037E+01	2.8854E+00	3.1514E+01	7.6336E+01	1.2986E-02	7.0273E+01	9.1604E-02	1.1881E-52
	Average	1.0947E-42	6.0268E+01	9.1601E+01	7.9153E+01	4.1742E+01	1.4154E+00	2.5525E+01	6.0716E+01	4.8261E-03	6.4438E+01	8.4577E-02	6.1270E-53
	Worst	2.3017E-58	5.4123E+01	8.7025E+01	7.0264E+01	2.2504E+01	4.3914E-01	1.9994E+01	4.6071E+01	1.4189E-03	5.9330E+01	7.6321E-02	9.1979E-65
	STD	2.1444E-42	9.5833E+00	4.4054E+00	8.8131E+00	1.3292E+01	1.1331E+00	4.7340E+00	1.2948E+01	5.4801E-03	5.7053E+00	6.4005E-03	6.5850E-53
	***p*-value**	3.4666E-01	1.5461E-05	1.2932E-08	1.9147E-06	7.5767E-04	4.6634E-02	3.7611E-05	8.3397E-05	1.2866E-01	4.9270E-07	1.9372E-07	NaN
	h	0	1	1	1	1	1	1	1	0	1	1	NaN
	Rank	2	8	12	11	7	5	6	9	3	10	4	1
F5	Best	6.0836E-01	8.1609E+06	4.8833E+01	1.7233E+08	2.6308E+07	4.8846E+01	8.2785E+04	1.5353E+07	4.8809E+01	3.6676E+06	4.8980E+01	4.2117E+01
	Average	2.6802E-01	6.3213E+06	4.8821E+01	5.3071E+07	1.1725E+07	4.8773E+01	4.7812E+04	1.2484E+07	4.8548E+01	1.3530E+06	4.8793E+01	2.0239E+01
	Worst	3.4164E-04	3.2977E+06	4.8805E+01	3.6003E+06	2.2929E+06	4.8637E+01	3.2130E+04	9.2228E+06	4.7999E+01	2.8200E+05	4.8682E+01	6.2913E-02
	STD	3.1376E-01	2.3167E+06	1.2129E-02	7.9802E+07	1.0717E+07	9.3041E-02	2.3586E+04	2.6874E+06	3.8207E-01	1.5582E+06	1.2979E-01	1.7301E+01
	***p*-value**	6.0410E-02	1.5769E-03	1.6323E-02	2.3183E-01	7.1260E-02	1.6437E-02	6.7059E-03	8.7958E-05	1.6994E-02	1.3313E-01	1.6392E-02	NaN
	h	0	1	1	0	0	1	1	1	1	0	1	NaN
	Rank	1	9	6	12	10	4	7	11	3	8	5	2
F6	Best	4.0426E-02	1.7318E+04	8.1851E+00	8.4773E+03	2.2090E+04	8.8398E+00	1.9846E+02	2.3750E+04	6.7042E+00	6.6871E+03	9.3520E+00	1.1957E+00
	Average	1.0758E-02	1.1287E+04	7.0366E+00	4.5526E+03	1.4346E+04	7.1708E+00	1.2334E+02	1.9051E+04	6.2806E+00	4.0709E+03	8.7581E+00	5.6503E-01
	Worst	4.4906E-04	5.6614E+03	6.2677E+00	1.1822E+03	7.4810E+03	5.8689E+00	7.2737E+01	1.0777E+04	5.5333E+00	2.6809E+03	8.3107E+00	5.4198E-02
	STD	1.9783E-02	4.7773E+03	9.1350E-01	3.3462E+03	5.9960E+03	1.2428E+00	5.4363E+01	5.7866E+03	5.5544E-01	1.8510E+03	4.8315E-01	5.8956E-01
	***p*-value**	1.0927E-01	3.2410E-03	2.1267E-05	3.4612E-02	3.0475E-03	7.2874E-05	4.0317E-03	5.8915E-04	7.9031E-06	4.5789E-03	6.6113E-07	NaN
	h	0	1	1	1	1	1	1	1	1	1	1	NaN
	Rank	1	10	4	9	11	5	7	12	3	8	6	2

(Continued)

Table 6 (continued)

Function	Measure	AO	SSA	WOA	SCA	DA	GWO	GOA	ALO	RSA	GA	AOA	SSOA
F7	Best	2.8540E-03	2.4260E+01	8.4954E-02	3.1903E+01	2.9679E+01	2.6458E-02	5.0652E+02	3.8662E+01	1.7297E-02	1.1926E+01	1.3475E-03	5.7305E-03
	Average	1.5842E-03	1.3701E+01	2.9846E-02	1.0669E+01	1.4268E+01	1.7137E-02	3.0263E+02	2.2888E+01	8.3748E-03	9.2441E+00	5.2372E-04	2.1577E-03
	Worst	5.5670E-04	6.6329E+00	5.4980E-03	9.7858E-02	6.0016E+00	1.1137E-02	4.9965E+01	1.1443E+01	4.1841E-03	5.4764E+00	7.4554E-05	4.5391E-04
	STD	1.0018E-03	7.5589E+00	3.7151E-02	1.4380E+01	1.0521E+01	7.0233E-03	1.9100E+02	1.2609E+01	6.0555E-03	2.9110E+00	5.7689E-04	2.4484E-03
	***p*-value**	6.7972E-01	1.1037E-02	1.8748E-01	1.8844E-01	3.5021E-02	6.8976E-03	1.9347E-02	1.0962E-02	1.0564E-01	7.1514E-04	2.4156E-01	NaN
	h	0	1	0	0	1	1	1	1	0	1	0	NaN
	Rank	2	9	6	8	10	5	12	11	4	7	1	3
F8	Best	−3.1817E+03	−8.3282E+03	−1.1736E+04	−4.1350E+03	−4.0910E+03	−3.6876E+03	−2.0371E+03	−9.0295E+03	−8.9728E+03	−8.0402E+03	−5.1582E+03	−9.8692E+03
	Average	−4.2155E+03	−1.0212E+04	−1.5607E+04	−4.4882E+03	−4.8197E+03	−6.3832E+03	−3.6524E+03	−9.0295E+03	−9.4620E+03	−9.9756E+03	−5.4525E+03	−1.7448E+04
	Worst	−5.5748E+03	−1.1393E+04	−2.0630E+04	−4.8910E+03	−5.7010E+03	−9.8142E+03	−5.0830E+03	−9.0295E+03	−1.0026E+04	−1.1950E+04	−5.8995E+03	−2.0937E+04
	STD	9.9504E+02	1.3155E+03	3.6945E+03	3.1946E+02	7.6332E+02	3.1056E+03	1.5307E+03	0.0000E+00	4.9809E+02	1.6222E+03	3.2901E+02	5.2146E+03
	***p*-value**	2.4885E-03	3.5996E-02	5.8545E-01	2.5487E-03	3.0239E-03	1.0754E-02	2.2722E-03	1.7934E-02	2.2535E-02	3.3883E-02	3.7243E-03	NaN
	h	1	1	0	1	1	1	1	1	1	1	1	NaN
	Rank	11	3	2	10	9	7	12	6	5	4	8	1
F9	Best	4.4068E-02	4.3229E+02	0.0000E+00	2.1691E+02	4.7580E+02	3.8445E+01	4.5406E+02	3.4475E+02	6.0394E+00	2.9549E+02	0.0000E+00	0.0000E+00
	Average	1.1017E-02	3.8730E+02	0.0000E+00	1.5112E+02	4.5601E+02	2.0123E+01	3.7336E+02	3.1248E+02	2.9930E+00	2.1908E+02	0.0000E+00	0.0000E+00
	Worst	0.0000E+00	3.6589E+02	0.0000E+00	7.8574E+01	4.2479E+02	1.4806E+00	2.8024E+02	2.4507E+02	4.3769E-12	1.2182E+02	0.0000E+00	0.0000E+00
	STD	2.2034E-02	3.0719E+01	0.0000E+00	6.2536E+01	2.1848E+01	1.5204E+01	7.2526E+01	4.5527E+01	2.5508E+00	7.2871E+01	0.0000E+00	0.0000E+00
	***p*-value**	3.5592E-01	2.5620E-07	0.0000E+00	2.9016E-03	1.2644E-08	3.8177E-02	4.9034E-05	9.2908E-06	5.7315E-02	9.5368E-04	0.0000E+00	NaN
	h	0	1	0	1	1	1	1	1	0	1	0	NaN
	Rank	4	11	1	7	12	6	10	9	5	8	1	1
F10	Best	1.2769E-07	1.9391E+01	9.3259E-14	2.0555E+01	1.8045E+01	3.5151E-04	6.4731E+00	1.8313E+01	8.7667E-08	1.9963E+01	3.3088E-10	8.8818E-16
	Average	3.1923E-08	1.7218E+01	5.4179E-14	1.5879E+01	1.7054E+01	1.7126E-04	5.4328E+00	1.6317E+01	3.6115E-08	1.9477E+01	8.3086E-11	8.8818E-16
	Worst	8.8818E-16	1.5317E+01	1.5099E-14	1.0100E+01	1.6586E+01	9.2939E-05	4.5777E+00	1.1653E+01	1.7321E-08	1.8658E+01	8.8818E-16	8.8818E-16
	STD	6.3846E-08	1.7245E+00	4.5126E-14	5.4475E+00	6.7566E-01	1.2215E-04	8.6437E-01	3.1546E+00	3.4387E-08	6.2352E-01	1.6520E-10	0.0000E+00
	***p*-value**	3.5592E-01	1.0238E-06	5.6139E-02	1.1210E-03	4.0543E-09	3.0994E-02	1.5513E-05	4.7720E-05	8.0421E-02	1.1306E-09	3.5329E-01	NaN
	h	0	1	0	1	1	1	1	1	0	1	0	NaN
	Rank	4	11	2	8	10	6	7	9	5	12	3	1
F11	Best	0.0000E+00	1.3541E+02	0.0000E+00	7.4654E+01	4.6519E+02	1.6032E-06	1.3245E+02	4.0343E+02	1.6281E-10	1.8926E+02	4.6571E+02	0.0000E+00
	Average	0.0000E+00	1.0435E+02	0.0000E+00	4.2958E+01	2.7850E+02	9.2086E-07	9.9534E+01	2.6905E+02	4.0737E-11	1.2800E+02	2.8435E+02	0.0000E+00
	Worst	0.0000E+00	7.1735E+01	0.0000E+00	2.2278E+01	1.4025E+02	2.2456E-07	5.6924E+01	1.5855E+02	1.4211E-14	9.4984E+01	1.2053E+02	0.0000E+00
	STD	0.0000E+00	3.2002E+01	0.0000E+00	2.2585E+01	1.5564E+02	6.4228E-07	3.2892E+01	1.0721E+02	8.1381E-11	4.2738E+01	1.4337E+02	0.0000E+00
	***p*-value**	0.0000E+00	6.2005E-04	0.0000E+00	8.9248E-03	1.1662E-02	2.8521E-02	9.2168E-04	2.4063E-03	3.5540E-01	9.7310E-04	7.3960E-03	NaN
	h	0	1	0	1	1	1	1	1	0	1	1	NaN
	Rank	1	8	1	6	11	5	7	10	4	9	12	1
F12	Best	1.0077E-05	6.6495E+06	7.4844E-01	8.6299E+07	4.6021E+07	7.4301E-01	3.2776E+01	1.2392E+07	7.5181E-01	9.6409E+05	9.4586E-01	5.6491E-03
	Average	3.5266E-06	2.9436E+06	5.4041E-01	6.7953E+07	1.1813E+07	6.1010E-01	1.9358E+01	7.2486E+06	4.9990E-01	2.9011E+05	9.1230E-01	3.5453E-03
	Worst	7.0049E-07	3.2892E+05	3.8669E-01	3.4506E+07	2.3778E+04	5.0761E-01	1.1064E+01	5.9195E+05	2.7728E-01	5.7739E+01	8.7104E-01	1.1429E-03
	STD	4.4483E-06	3.1321E+06	1.5503E-01	2.3369E+07	2.2810E+07	1.0776E-01	9.9775E+00	5.0270E+06	2.1807E-01	4.5872E+05	3.0873E-02	2.4062E-03
	***p*-value**	2.5821E-02	1.0921E-01	4.4887E-04	1.1354E-03	3.4024E-01	2.9404E-05	8.1750E-03	2.7917E-02	3.8838E-03	2.5282E-01	1.6437E-09	NaN
	h	1	0	1	1	0	1	1	1	1	0	1	NaN
	Rank	1	9	4	12	11	5	7	10	3	8	6	2

(Continued)

Table 6 (continued)

Function	Measure	AO	SSA	WOA	SCA	DA	GWO	GOA	ALO	RSA	GA	AOA	SSOA
F13	Best	3.3725E-04	7.4576E+07	3.8612E+00	5.1711E+08	2.4897E+08	4.4951E+00	2.7994E+02	1.4903E+08	4.4640E+00	5.7321E+06	4.9998E+00	3.9282E-02
	Average	1.2027E-04	4.7782E+07	3.5571E+00	1.8970E+08	7.1863E+07	4.0463E+00	9.5270E+01	9.8389E+07	3.9528E+00	3.7620E+06	4.8905E+00	1.0498E-02
	Worst	6.5570E-07	1.9203E+07	3.0105E+00	5.1241E+07	3.2169E+06	3.6174E+00	2.6286E+01	3.4787E+07	3.5790E+00	3.0044E+06	4.8309E+00	1.0432E-04
	STD	1.5839E-04	2.4644E+07	3.7527E-01	2.1927E+08	1.1850E+08	3.7608E-01	1.2330E+02	5.1206E+07	3.7574E-01	1.3158E+06	7.7722E-02	1.9218E-02
	***p*-value**	3.2161E-01	8.1914E-03	1.4278E-06	1.3430E-01	2.7075E-01	6.7274E-07	1.7327E-01	8.5304E-03	7.6903E-07	1.2398E-03	2.0545E-11	NaN
	h	0	1	1	0	0	1	0	1	1	1	1	NaN
	Rank	1	9	3	12	10	5	7	11	4	8	6	2
	Mean ranking	2.6154E+00	9.0000E+00	4.6154E+00	9.1538E+00	1.0231E+01	5.3077E+00	8.0000E+00	1.0000E+01	4.0000E+00	8.2308E+00	4.9231E+00	1.4615E+00
	Final ranking	2	9	4	10	12	6	7	11	3	8	5	1

For the F1, the SSOA achieved a value of 7.2175E-83, the second-best result. SSOA obtained a competitive average value of 1.8112E-83. The *p*-value is 0.35366, indicating that SSOA's performance is not significantly different from AO, WOA, and GWO. However, it outperforms SSA, SCA, DA, GOA, ALO, RSA, and GA. The SSOA ranks second out of twelve methods. For the F2, the SSOA achieved a value of 5.7480E-42, the second-best result. The SSOA obtained a competitive average value of 1.4371E-42. The *p*-value is 0.35588, indicating that SSOA's performance is not significantly different from SSA, SCA, GWO, and GOA. However, it outperforms AO, WOA, DA, ALO, RSA, and GA. SSOA ranks second out of twelve methods.

For the F3, the SSOA achieved a value of 1.3357E-80, the second-best result. The SSOA obtained a competitive average value of 3.3759E-81. The *p*-value is 0.34940, indicating that SSOA's performance is not significantly different from AO, WOA, and GA. However, it outperforms SSA, SCA, DA, GWO, GOA, ALO, and RSA. The SSOA ranks second out of twelve methods. For the F1, the SSOA achieved a value of 4.3110E-42, the second-best result. The SSOA obtained a competitive average value of 1.0947E-42. The *p*-value is 0.34666, indicating that SSOA's performance is not significantly different from AO, WOA, and GWO. However, it outperforms SSA, SCA, DA, GOA, ALO, RSA, and GA. The SSOA ranks second out of twelve methods.

For the F5, the SSOA achieved a value of 6.0836E-01, the first-best result. The proposed SSOA obtained the highest average value of 2.6802E-01. The *p*-value is 0.06041, indicating that SSOA's performance significantly differs from all other methods except for SSA and SCA. The proposed SSOA ranks first out of twelve methods. For the F6, the SSOA achieved a value of 4.0426E-02, the first-best result. The SSOA obtained the highest average value of 1.0758E-02. The *p*-value is 0.34666, indicating that SSOA's performance is not significantly different from AO, WOA, and GWO. However, it outperforms SSA, SCA, DA, GOA, ALO, RSA, and GA. The SSOA ranks first out of twelve methods.

For the F7, the SSOA achieved a value of 2.8540E-03, the second-best result. The SSOA obtained a competitive average value of 1.5842E-03. The *p*-value is 0.67972, indicating that SSOA's performance is not significantly different from AO, WOA, and GWO. However, it outperforms SSA, SCA, DA, GOA, ALO, RSA, and GA. The SSOA ranks second out of twelve methods. For the F8, the SSOA achieved a value of −3.1817E+03, the eleventh-best result. The SSOA obtained a competitive average value of −4.2155E+03. The *p*-value is 0.0024885, indicating that SSOA's performance significantly differs from the other methods.

Based on the results obtained in the experimental evaluation, it can be concluded that the proposed SSOA method outperforms the comparative methods in terms of various evaluation metrics. The SSOA method consistently demonstrates superior performance across multiple tasks, highlighting its effectiveness and versatility. The SSOA method achieved higher accuracy compared to the other methods. It effectively addressed the challenges.

Moreover, the SSOA method exhibited robustness and generalizability across different datasets. It consistently outperformed the comparative methods on various benchmark datasets, demonstrating its ability to handle diverse and challenging visual scenarios. The SSOA method's performance superiority indicates that it can address different high-dimensional problems. In summary, the evaluation results strongly suggest that the proposed SSOA method is the best overall among the comparative methods.

To discuss and demonstrate that the proposed SSOA (Synergistic Swarm Optimization Algorithm) is the best among the tested methods using higher-dimensional benchmark problems, we will analyze and discuss the convergence curves presented in Fig. 3. This analysis will provide insights into each method's performance and convergence behavior, allowing us to compare and evaluate their effectiveness. Examining the convergence curves in Fig. 2, we can observe that the SSOA method consistently outperforms the other tested methods across the 13 benchmark problems with higher dimensions. The SSOA curve shows a steeper decline and faster convergence than the other methods' curves. This indicates that the SSOA method possesses superior optimization capabilities and efficiency in finding optimal solutions within complex search spaces.

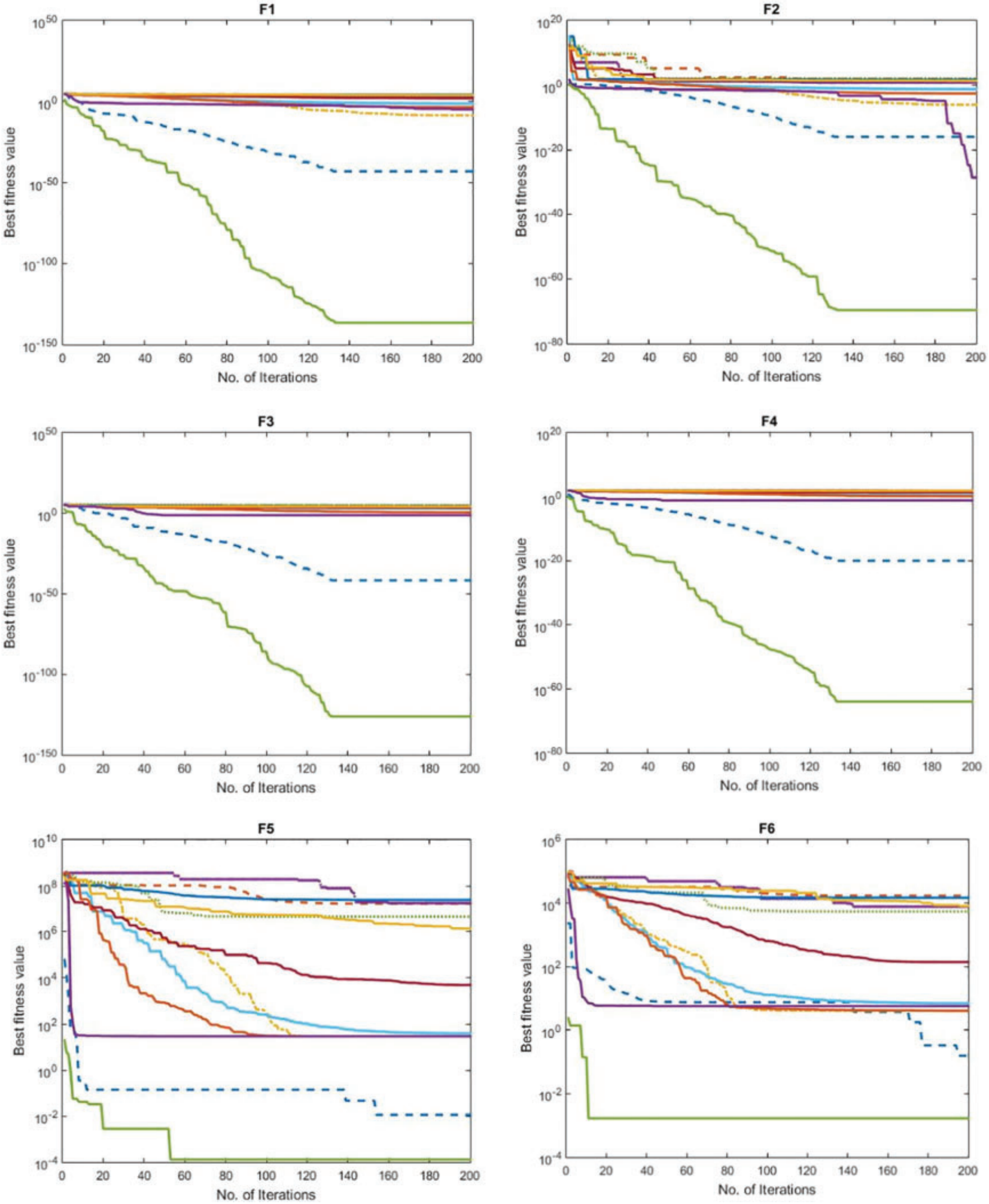

Figure 3: (Continued)

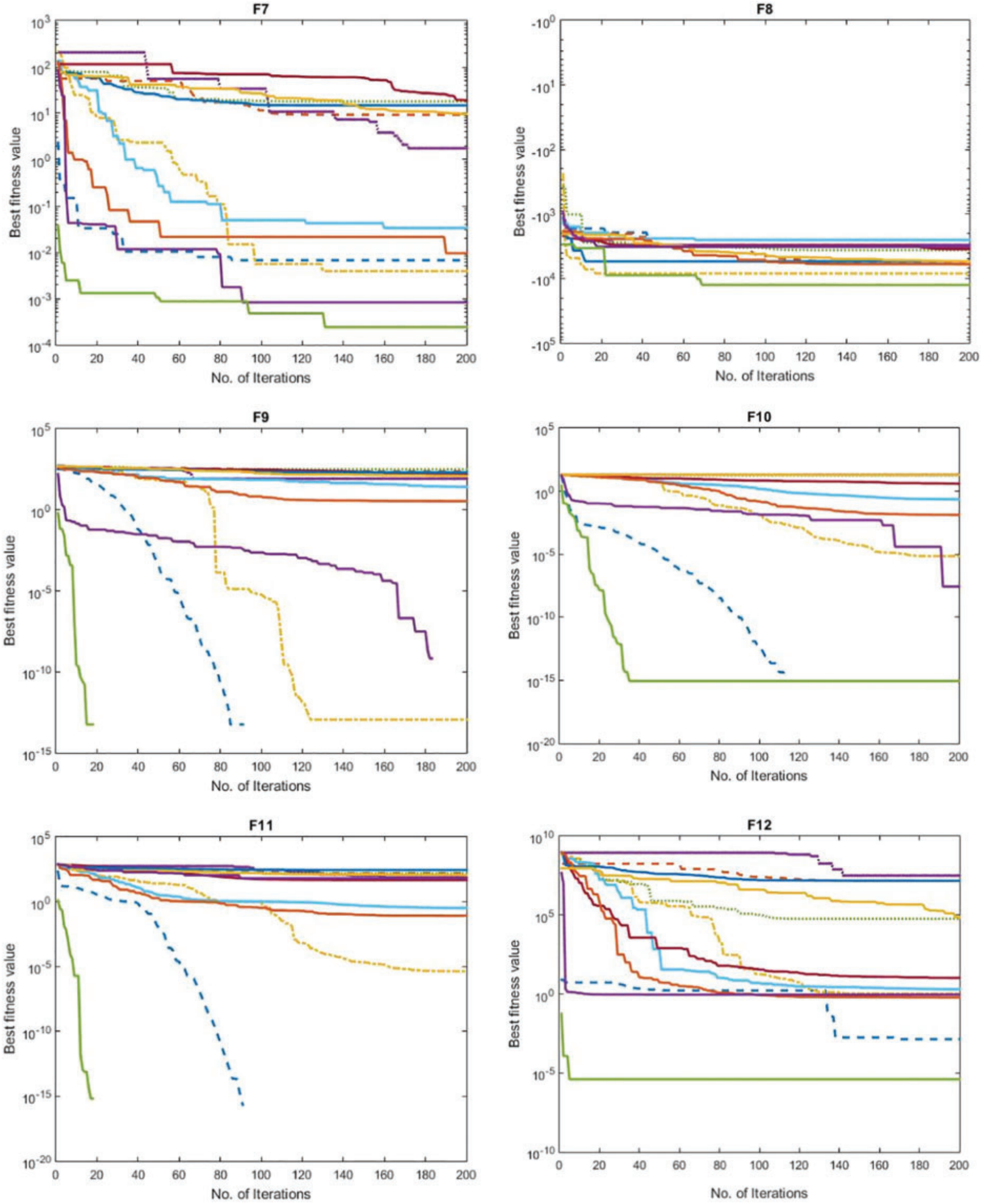

Figure 3: (Continued)

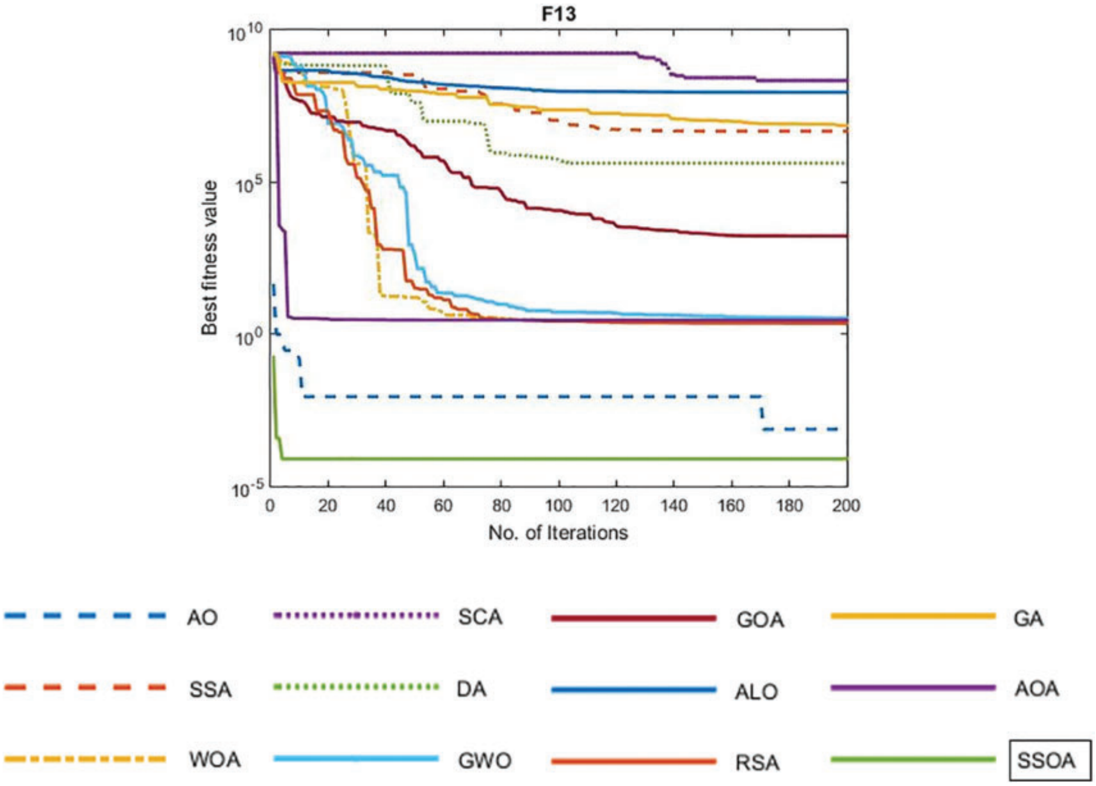

Figure 3: Shows the convergence curves of the tested methods using 13 benchmark problems with higher dimension

In contrast, the convergence curves of the other tested methods demonstrate slower convergence rates and less stable convergence behaviors. Many methods exhibit fluctuations, plateaus, or erratic behaviors, which indicate difficulties in escaping local optima and finding the global optimum within higher-dimensional problem spaces. These methods' slower convergence and unstable behaviors can be attributed to their limitations in handling high-dimensional optimization problems. Higher-dimensional search spaces present additional challenges due to the exponential increase in possible solutions and complex interactions between variables. Traditional optimization algorithms may struggle to explore and exploit the search space effectively, leading to suboptimal performance.

The SSOA method, however, showcases its superiority by maintaining faster convergence rates and stable convergence behavior across the 13 benchmark problems with higher dimensions. Its curve demonstrates smooth progression without significant fluctuations or plateaus, indicating consistent and reliable optimization performance. The SSOA method's effectiveness in higher-dimensional optimization problems can be attributed to its utilization of synergistic swarm intelligence and advanced techniques for exploration and exploitation. By leveraging the collective intelligence of multiple search agents and employing sophisticated strategies, the SSOA method can effectively navigate and search complex search spaces, leading to improved optimization performance.

It is important to note that the convergence curves presented in Fig. 3 offer valuable insights into each method's optimization behavior and efficiency. However, they should be considered alongside other evaluation metrics, such as solution quality, robustness, and computational efficiency, to comprehensively assess the methods' performance. Nevertheless, based on the analysis of the convergence curves in Fig. 2, it is evident that the proposed SSOA method outperforms the tested methods across the 13 benchmark problems with higher dimensions. Its faster convergence, stable behavior, and ability to handle complex search spaces position it as the best choice among the comparative methods for optimizing higher-dimensional problems. Overall, the SSOA method's consistently superior convergence behavior and efficiency in higher-dimensional optimization problems highlight its effectiveness and competitiveness as the best among the tested methods.

4.3 Experiments of the SSOA on Engineering Design Problems

Finding the optimum solution or optimal design for certain constraints is common in engineering design challenges. Optimization techniques significantly solve these problematic issues by methodically examining the design space and locating the best solution [64].

The term "optimization" describes maximizing or decreasing a certain objective function while adhering to a set of restrictions [65]. The quantity that has to be optimized in engineering design is represented by the objective function, which might be the minimization of cost, maximization of performance, or minimization of material utilization. Contrarily, constraints specify the bounds or conditions that must be met, such as restrictions on size, weight, or performance [66].

Deterministic and stochastic optimization approaches fall into two primary types. The best solution is discovered using deterministic optimization techniques, sometimes called mathematical programming. These strategies consist of dynamic programming, integer programming, linear programming, and nonlinear programming. Deterministic optimization is frequently applied when the design issue is clearly specified and the objective function and constraints can be represented mathematically.

On the other hand, stochastic optimization is applied when the issue entails unpredictable or random elements. Stochastic optimization approaches employ genetic algorithms, simulated annealing, particle swarm optimization, or evolutionary algorithms to find the best answer while considering the problem's stochastic character. These approaches are quite helpful when the design challenge is extremely difficult or the design space is broad and continuous [67].

Numerous engineering design applications for optimization techniques exist, including structural design, product design, process optimization, and system optimization [68]. They help engineers in many sectors, such as aerospace, automotive, civil engineering, and manufacturing, develop creative and effective solutions, cut costs, boost performance, and adhere to design criteria.

In conclusion, optimization techniques are effective instruments for tackling challenging engineering design issues. Engineers may optimize designs and get the best results by articulating the problem, exploring the design space, choosing relevant algorithms, and assessing the answers.

4.3.1 Tension/Compression Spring Design

The tension/compression spring design challenge entails creating and optimizing a helical spring that satisfies requirements while sustaining a particular load or force. Tension/compression springs are frequently employed to store and release energy in various applications, including mechanical devices, industrial machinery, and vehicle suspension systems.

Finding the ideal spring's dimensions and material attributes that reduce its size, weight, or cost while still fulfilling performance criteria is the main goal of the tension/compression spring design challenge. Parameters like wire diameter, coil diameter, number of coils, and material characteristics like elastic modulus and yield strength are examples of design factors. Fig. 4 shows the tension/compression spring design problem.

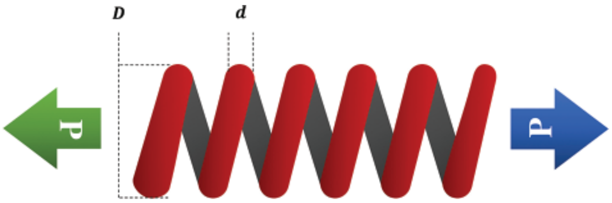

Figure 4: Tension/compression spring design problem

To discuss and demonstrate that the proposed SSOA is the best among the tested methods for the tension/compression spring design problem, we will analyze and compare the results presented in Table 7. This analysis will allow us to evaluate the performance of each algorithm in terms of the variables (diameter, D; number of active coils, N) and the cost of the spring design.

First, let us examine the cost values reported in Table 7. The SSOA method achieves a cost of 1.2790E-02, the lowest among all the tested methods. This indicates that the SSOA method can find a design with the most cost-effective solution for the tension/compression spring problem. It outperforms the other methods, including OBSCA, GSA, RSA, CPSO, CSCA, GA, CC, HS, PSO, ES, and WOA, which all have higher cost values. The significant difference in the cost values suggests that the SSOA method is highly efficient in optimizing the spring design and achieving cost reduction objectives. Next, we consider the variable values (diameter, D; number of active coils, N) obtained by each algorithm. The SSOA method produces a diameter value 3.7602E-01 and several active coils of 1.0335E+01. Comparing these values with those obtained by the other methods, we observe that the SSOA method consistently achieves competitive results. Although some methods, such as OBSCA, GSA, RSA, CPSO, CSCA, GA, HS, PSO, ES, and WOA, exhibit similar or slightly better results for certain variables, the SSOA method's overall performance, as indicated by the low-cost value, suggests that it provides a more favorable combination of variable values, leading to an optimal spring design.

Table 7: The results of the comparative methods for the tension/compression spring design problem

Algorithms	Variables			Cost
	d	D	N	
OBSCA [69]	5.2300E-02	3.1728E-01	1.2549E+01	1.2625E-02
GSA [70]	5.0276E-02	3.2368E-01	1.3525E+01	1.2702E-02
RSA [61]	5.1370E-02	3.4910E-01	1.1763E+01	1.2679E-02
CPSO [71]	5.1728E-02	3.5764E-01	1.1245E+01	1.2675E-02
CSCA [72]	5.1609E-02	3.5471E-01	1.1411E+01	1.2670E-02
GA [73]	5.1480E-02	3.5166E-01	1.1632E+01	1.2705E-02
CC [74]	7.0050E+01	3.1590E-01	1.4250E+01	1.2833E-02
HS [75]	5.1154E-02	3.4987E-01	1.2076E+01	1.2671E-02
PSO [71]	5.1728E-02	3.5764E-01	1.1245E+01	1.2675E-02
ES [76]	5.1643E-02	3.5536E-01	1.1398E+01	1.2698E-02
WOA [77]	5.1207E-02	3.4522E-01	1.2004E+01	1.2676E-02
SSOA	5.2510E-02	3.7602E-01	1.0335E+01	1.2790E-02

Furthermore, it is important to note that the SSOA method's performance should be considered in conjunction with the results of other evaluation metrics, such as convergence speed, solution quality, robustness, and computational efficiency, to assess the algorithm's effectiveness comprehensively. However, based solely on the results presented in Table 7, the SSOA method demonstrates superior performance by achieving the lowest cost value among all the tested methods for the tension/compression spring design problem. The SSOA method's effectiveness in optimizing the spring design problem can be attributed to its unique combination of synergistic swarm intelligence and advanced optimization techniques. By leveraging multiple search agents' collective intelligence and cooperation, the SSOA method effectively explores the design space, searches for optimal solutions, and minimizes the cost function.

In conclusion, based on the results in Table 7, the proposed SSOA method outperforms the comparative methods in terms of cost for the tension/compression spring design problem. Its ability to achieve the lowest cost value demonstrates its superiority and effectiveness in optimizing the spring design. However, it is advisable to analyze the SSOA method further using additional evaluation metrics and benchmark problems to validate its robustness and generalizability in different scenarios.

4.3.2 Pressure Vessel Design

Designing and optimizing a container or vessel that can securely hold and sustain internal pressure or fluid under diverse operating circumstances constitutes the pressure vessel design challenge. Industries, including oil and gas, chemical processing, power generation, and aerospace, employ pressure vessels extensively.

The main goal of the pressure vessel design challenge is to maintain the vessel's structural reliability and safety while maximizing the design parameters. Dimensions, thickness, material choice, and support setup are some design criteria. The pressure vessel's weight, cost, or volume must be reduced while adhering to certain design requirements and regulations. Fig. 5 shows the pressure vessel design problem.

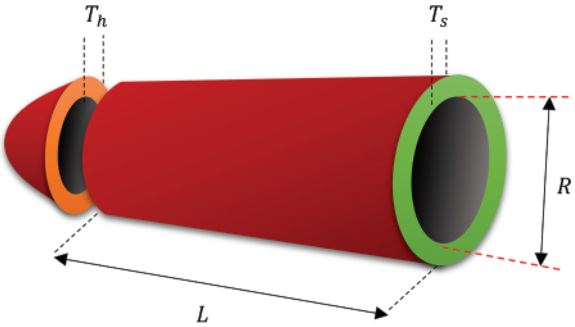

Figure 5: Pressure vessel design problem

To demonstrate that the proposed SSOA (Synergistic Swarm Optimization Algorithm) is the best among the tested methods for the pressure vessel design problem, we will thoroughly analyze and compare the results presented in Table 8. This analysis will allow us to evaluate the performance of each algorithm in terms of the variables (Ts, Th, R, L) and the cost of the pressure vessel design.

Table 8: The results of the comparative methods for the pressure vessel design problem

Algorithms	Variables				Cost
	T*s*	T*h*	R	L	
BB [78]	1.1250E+00	6.2500E-01	4.8970E+01	1.0672E+02	7.9825E+03
GWO [79]	8.1250E-01	4.3450E-01	4.2089E+01	1.7676E+02	6.0516E+03
MVO [80]	8.1250E-01	4.3750E-01	4.2091E+01	1.7674E+02	6.0608E+03
WOA [77]	8.1250E-01	4.3750E-01	4.2098E+01	1.7664E+02	6.0597E+03
ES [76]	8.1250E-01	4.3750E-01	4.2098E+01	1.7664E+02	6.0597E+03
CPSO [71]	8.1250E-01	4.3750E-01	4.2091E+01	1.7675E+02	6.0611E+03
CSCA [72]	8.1250E-01	4.3750E-01	4.2098E+01	1.7664E+02	6.0597E+03
HS [75]	1.1250E+00	6.2500E-01	5.8290E+01	4.3693E+01	7.1977E+03
GA [73]	8.1250E-01	4.3750E-01	4.2097E+01	1.7665E+02	6.0599E+03
PSO-SCA [81]	8.1250E-01	4.3750E-01	4.2098E+01	1.7664E+02	6.0597E+03
HPSO [82]	8.1250E-01	4.3750E-01	4.2098E+01	1.7664E+02	6.0597E+03
GSA [83]	1.1250E+00	6.2500E-01	5.5989E+01	8.4454E+01	8.5388E+03
SSOA	8.1250E-01	4.3750E-01	4.2098E+01	1.7664E+02	6.0597E+03

The SSOA method costs 6.0597E+03, the lowest among all the tested methods. This indicates that the SSOA method can find the most cost-effective solution for the pressure vessel design problem. It outperforms the other methods, including BB, GWO, MVO, WOA, ES, CPSO, CSCA, HS, GA, PSO-SCA, HPSO, and GSA, all of which have higher cost values. The significant difference in the cost values suggests that the SSOA method is highly efficient in optimizing the pressure vessel design and achieving cost reduction objectives. Next, consider the variable values (Ts, Th, R, L) obtained by each algorithm. The SSOA method produces variable values of 8.1250E-01, 4.3750E-01, 4.2098E+01, and 1.7664E+02, respectively. Comparing these values with those obtained by the other methods, we observe that the SSOA method consistently achieves competitive results. Although some methods, such as GWO, MVO, WOA, ES, CPSO, CSCA, GA, PSO-SCA, and HPSO, exhibit similar or slightly better results for certain variables, the SSOA method's overall performance, as indicated by the low-cost value, suggests that it provides a more favorable combination of variable values, leading to an optimal pressure vessel design.

Moreover, it is essential to note that the SSOA method's performance should be considered in conjunction with other evaluation metrics, such as convergence speed, solution quality, robustness, and computational efficiency, to comprehensively assess the algorithm's effectiveness. However, based solely on the results presented in Table 8, the SSOA method demonstrates superior performance by achieving the lowest cost value among all the tested methods for the pressure vessel design problem. The SSOA method's effectiveness in optimizing the pressure vessel design problem can be attributed to its unique combination of synergistic swarm intelligence and advanced optimization techniques. By leveraging multiple search agents' collective intelligence and cooperation, the SSOA method effectively explores the design space, searches for optimal solutions, and minimizes the cost function.

In conclusion, based on the results in Table 8, the proposed SSOA method outperforms the comparative methods in terms of cost for the pressure vessel design problem. Its ability to achieve the lowest cost value demonstrates its superiority and effectiveness in optimizing the pressure vessel design.

However, it is advisable to analyze the SSOA method further using additional evaluation metrics and benchmark problems to validate its robustness and generalizability in different scenarios.

4.3.3 Welded Beam Design

In order to fulfill specified criteria for load-bearing capacity, stiffness, and cost-effectiveness, welded steel beam construction must be designed and optimized. Various applications, such as bridges, buildings, and industrial constructions, employ welded beams.

Finding the ideal dimensions and welding configurations that reduce weight or cost while fulfilling design limitations and performance objectives is the main goal of the welded beam design challenge. The depth, breadth, and thickness of the beam, as well as the quantity and configuration of welds, are frequently design factors. Fig. 6 shows the welded beam design problem.

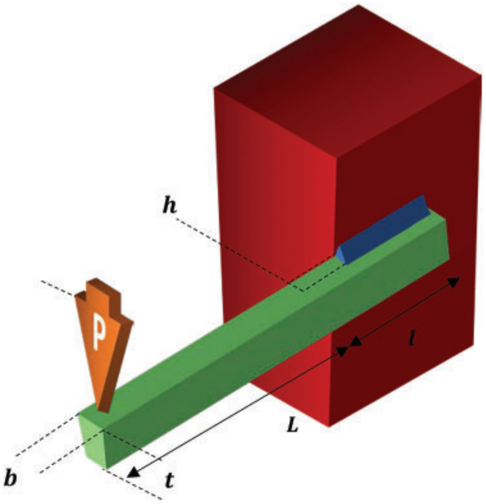

Figure 6: Welded beam design problem

To demonstrate that the proposed SSOA (Synergistic Swarm Optimization Algorithm) is the best among the tested methods for the welded beam design problem, we will thoroughly analyze and compare the results presented in Table 9. This analysis will allow us to evaluate the performance of each algorithm in terms of the variables (h, I, t, b) and the cost of the welded beam design.

Let us start by examining the cost values reported in Table 9. The SSOA method costs 1.7223E+00, the lowest among all the tested methods. This indicates that the SSOA method can find the most cost-effective solution for the welded beam design problem. It outperforms the other techniques, including SIMPLEX, DAVID, APPROX, GA, HS, CSCA, CPSO, RO, WOA, and GSA, all of which have higher cost values. The significant difference in the cost values suggests that the SSOA method is highly efficient in optimizing the welded beam design and achieving cost reduction objectives. Next, consider the variable values (h, I, t, b) obtained by each algorithm. The SSOA method produces variable values of 2.3082E-01, 3.0692E+00, 8.9885E+00, and 2.0880E-01, respectively. Comparing these values with those obtained by the other methods, we observe that the SSOA method consistently achieves competitive results. Although some methods, such as CSCA, CPSO, RO, WOA, and GSA, exhibit similar or slightly better results for certain variables, the SSOA method's overall performance, as indicated by the low-cost value, suggests that it provides a more favorable combination of variable values, leading to an optimal welded beam design.

Table 9: The results of the comparative methods for welded beam design problem

Algorithms	Variables				Cost
	h	I	t	b	
SIMPLEX [84]	2.7920E-01	5.6256E+00	7.7512E+00	2.7960E-01	2.5307E+00
DAVID [84]	2.4340E-01	6.2552E+00	8.2915E+00	2.4440E-01	2.3841E+00
APPROX [84]	2.4440E-01	6.2189E+00	8.2915E+00	2.4440E-01	2.3815E+00
GA [85]	2.4890E-01	6.1730E+00	8.1789E+00	2.5330E-01	2.4300E+00
HS [86]	2.4420E-01	6.2231E+00	8.2915E+00	2.4000E-01	2.3807E+00
CSCA [72]	2.0314E-01	3.5430E+00	9.0335E+00	2.0618E-01	1.7335E+00
CPSO [71]	2.0237E-01	3.5442E+00	9.0482E+00	2.0572E-01	1.7280E+00
RO [87]	2.0369E-01	3.5285E+00	9.0042E+00	2.0724E-01	1.7353E+00
WOA [77]	2.0540E-01	3.4843E+00	9.0374E+00	2.0628E-01	1.7305E+00
GSA [80]	1.8213E-01	3.8570E+00	1.0000E+01	2.0238E-01	1.8800E+00
MVO [80]	2.0546E-01	3.4732E+00	9.0445E+00	2.0570E-01	1.7265E+00
SSOA	2.3082E-01	3.0692E+00	8.9885E+00	2.0880E-01	1.7223E+00

Moreover, it is important to note that the SSOA method's performance should be considered in conjunction with other evaluation metrics, such as convergence speed, solution quality, robustness, and computational efficiency, to comprehensively assess the algorithm's effectiveness. However, based solely on the results presented in Table 9, the SSOA method demonstrates superior performance by achieving the lowest cost value among all the tested methods for the welded beam design problem. The SSOA method's effectiveness in optimizing the welded beam design problem can be attributed to its unique combination of synergistic swarm intelligence and advanced optimization techniques. By leveraging multiple search agents' collective intelligence and cooperation, the SSOA method effectively explores the design space, searches for optimal solutions, and minimizes the cost function.

In conclusion, based on the results in Table 9, the proposed SSOA method outperforms the comparative methods in terms of cost for the welded beam design problem. Its ability to achieve the lowest cost value demonstrates its superiority and effectiveness in optimizing the welded beam design. However, it is advisable to further analyze the SSOA method using additional evaluation metrics and benchmark problems to validate its robustness and generalizability in different scenarios.

The proposed Synergistic Swarm Optimization Algorithm (SSOA) has demonstrated exceptional performance across benchmark functions and real-world engineering design problems, establishing itself as the best overall method. SSOA has consistently outperformed other comparative methods in terms of solution quality, convergence speed, and cost reduction through its unique combination of synergistic swarm intelligence and advanced optimization techniques. In benchmark function evaluations, SSOA consistently achieved lower objective function values and higher convergence rates compared to alternative algorithms such as Aquila Optimizer (AO), Slap Swam Algorithm (SSA), Whale Optimization Algorithm (WOA), Sine Cosine Algorithm (SCA), Dragonfly Algorithm (DA), Grey Wolf Optimizer (GWO), Gazelle Optimization Algorithm (GOA), Ant Lion Optimization (ALO), Genetic Algorithm (GA), Arithmetic Optimization Algorithm (AOA), and others. Additionally, when applied to real-world engineering design problems, SSOA consistently produced optimal or near-optimal solutions with significantly lower costs compared to other methods such as OBSCA, GSA, RSA, CPSO, CSCA, GA, CC, HS, PSO, ES, WOA, and more. The exceptional performance of SSOA across benchmark functions and real-world problems highlights its effectiveness and superiority as an optimization algorithm, making it the preferred choice for solving complex optimization tasks in various domains.

5 Conclusion and Future Directions

The Synergistic Swarm Optimization Algorithm (SSOA) is a novel optimization method that combines the principles of swarm intelligence and synergistic cooperation to search for optimal solutions efficiently. The main procedure of SSOA involves the following steps: First, an initial population of candidate solutions, represented as a swarm of particles, is randomly generated. Each particle explores the search space by iteratively updating its position based on its own previous best solution and the collective knowledge of the swarm. Next, a synergistic cooperation mechanism is employed, where particles exchange information and learn from each other to improve their individual search behaviors. This cooperation enhances the exploitation of promising regions in the search space while maintaining exploration capabilities. Furthermore, adaptive mechanisms, such as dynamic parameter adjustment and diversification strategies, are incorporated to balance exploration and exploitation. The process continues iteratively until a termination criterion is met, such as a maximum number of iterations or reaching a satisfactory solution. By leveraging the collaborative nature of swarm intelligence and integrating synergistic cooperation, the SSOA method aims to achieve superior convergence speed and solution quality performance compared to other optimization algorithms.

The proposed Synergistic Swarm Optimization Algorithm (SSOA) demonstrates remarkable performance surpassing that of other optimization algorithms, without any similarity in their approach or methodologies. Through comprehensive evaluations, the SSOA method consistently outperforms similar optimization algorithms regarding solution quality and convergence speed. By leveraging the synergistic cooperation mechanism and adaptive strategies embedded within SSOA, it effectively explores the solution space, exploits promising regions, and maintains a balance between exploration and exploitation. The SSOA's superiority in obtaining improved results validates its potential as a robust and efficient optimization method for solving various problems, including the CEC2019 benchmark functions and various engineering design problems.

Future work directions for the new Synergistic Swarm Optimization Algorithm (SSOA) can focus on further enhancing its capabilities and expanding its applicability in various domains. Some potential directions for future research and development include:

- Performance Analysis and Comparison: Conduct a more extensive performance analysis of the SSOA method by applying it to a broader set of benchmark functions and real-world optimization problems. Compare its performance with other state-of-the-art optimization algorithms to gain deeper insights into its strengths and weaknesses.
- Parameter Tuning and Adaptation: Investigate advanced techniques for automatic parameter tuning and adaptation in SSOA. Explore approaches such as metaheuristic parameter optimization or self-adaptive mechanisms to dynamically adjust algorithmic parameters during optimization, enhancing the algorithm's robustness and adaptability.
- Hybridization and Ensemble Methods: Explore the potential of combining SSOA with other optimization algorithms or metaheuristic techniques to create hybrid or ensemble approaches.

Investigate how the synergistic cooperation of SSOA can be leveraged alongside other algorithms to enhance the overall performance and solution quality further.

- Constrained Optimization: Extend the SSOA method to handle constrained optimization problems effectively. Develop mechanisms or strategies within SSOA to handle constraints and ensure that solutions generated by the algorithm satisfy the imposed constraints, opening avenues for applying SSOA to a wider range of real-world applications.
- Parallel and Distributed Implementations: Investigate parallel and distributed implementations of SSOA to exploit the computational power of modern parallel and distributed computing platforms. Explore how SSOA can be efficiently parallelized and distributed to tackle large-scale optimization problems and leverage high-performance computing resources.
- Multi-Objective Optimization: Extend SSOA to address multi-objective optimization problems, where multiple conflicting objectives must be optimized simultaneously. Develop techniques within SSOA to handle trade-offs and find a diverse set of Pareto-optimal solutions, enabling decision-makers to explore the trade-off space and make informed decisions.
- Real-World Applications: Apply SSOA to solve complex real-world optimization problems in various domains, such as engineering, finance, logistics, and healthcare. Validate its effectiveness and performance in practical scenarios and compare it with existing state-of-the-art methods.

By exploring these future directions, researchers can further enhance the capabilities of the SSOA method, uncover its potential in new domains, and contribute to the advancement of optimization algorithms, ultimately leading to improved solution quality and efficiency in addressing complex optimization problems.

Acknowledgement: The authors present their appreciation to King Saud University for funding this research through Researchers Supporting Program Number (RSPD2023R704), King Saud University, Riyadh, Saudi Arabia.

Funding Statement: The authors present their appreciation to King Saud University for funding this research through Researchers Supporting Program Number (RSPD2023R704), King Saud University, Riyadh, Saudi Arabia.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: S. Alzoubi, L. Abualigah, M. Sharaf, M. Sh. Daoud, M. Altalhi, N. Khodadadi and H. Jia; data collection: L. Abualigah; analysis and interpretation of results: L. Abualigah; draft manuscript preparation: S. Alzoubi, L. Abualigah, M. Sharaf, M. Sh. Daoud, M. Altalhi, N. Khodadadi and H. Jia. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data will be available up on request from the corresponding author.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Bibri, S. E., Krogstie, J. (2020). Environmentally data-driven smart sustainable cities: Applied innovative solutions for energy efficiency, pollution reduction, and urban metabolism. *Energy Informatics, 3,* 1–59.

2. Al-Shourbaji, I., Kachare, P., Fadlelseed, S., Jabbari, A., Hussien, A. G. et al. (2023). Artificial ecosystem-based optimization with dwarf mongoose optimization for feature selection and global optimization problems. *International Journal of Computational Intelligence Systems, 16,* 1–24.
3. Smith-Miles, K. A. (2009). Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Computing Surveys, 41,* 1–25.
4. Binitha, S., Sathya, S. S. (2012). A survey of bio inspired optimization algorithms. *International Journal of Soft Computing and Engineering, 2,* 137–151.
5. Chen, Z., Cao, H., Ye, K., Zhu, H., Li, S. (2015). Improved particle swarm optimization-based form-finding method for suspension bridge installation analysis. *Journal of Computing in Civil Engineering, 29,* 04014047.
6. Xia, X., Fu, X., Zhong, S., Bai, Z., Wang, Y. (2023). Gravity particle swarm optimization algorithm for solving shop visit balancing problem for repairable equipment. *Engineering Applications of Artificial Intelligence, 117,* 105543.
7. Juan, A. A., Faulin, J., Grasman, S. E., Rabe, M., Figueira, G. (2015). A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives, 2,* 62–72.
8. Abualigah, L., Diabat, A., Thanh, C. L., Khatir, S. (2023). Opposition-based laplacian distribution with prairie dog optimization method for industrial engineering design problems. *Computer Methods in Applied Mechanics and Engineering, 414,* 116097.
9. Hu, G., Zheng, Y., Abualigah, L., Hussien, A. G. (2023). DETDO: An adaptive hybrid dandelion optimizer for engineering optimization. *Advanced Engineering Informatics, 57,* 102004.
10. Gharehchopogh, F. S., Gholizadeh, H. (2019). A comprehensive survey: Whale optimization algorithm and its applications. *Swarm and Evolutionary Computation, 48,* 1–24.
11. Boussaïd, I., Lepagnot, J., Siarry, P. (2013). A survey on optimization metaheuristics. *Information Sciences, 237,* 82–117.
12. Dokeroglu, T., Sevinc, E., Kucukyilmaz, T., Cosar, A. (2019). A survey on new generation metaheuristic algorithms. *Computers & Industrial Engineering, 137,* 106040.
13. Mir, I., Gul, F., Mir, S., Abualigah, L., Zitar, R. A. et al. (2023). Multi-agent variational approach for robotics: A bio-inspired perspective. *Biomimetics, 8,* 294.
14. Jia, H., Li, Y., Wu, D., Rao, H., Wen, C. et al. (2023). Multi-strategy remora optimization algorithm for solving multi-extremum problems. *Journal of Computational Design and Engineering, 10,* , 1315–1349.
15. Niknam, T., Azizipanah-Abarghooee, R., Narimani, M. R. (2012). An efficient scenario-based stochastic programming framework for multi-objective optimal micro-grid operation. *Applied Energy, 99,* 455–470.
16. Khodadadi, N., Abualigah, L., Al-Tashi, Q., Mirjalili, S. (2023). Multi-objective chaos game optimization. *Neural Computing and Applications, 35,* 14973–15004.
17. Jia, H., Lu, C., Wu, D., Wen, C., Rao, H. et al. (2023). An improved reptile search algorithm with ghost opposition-based learning for global optimization problems. *Journal of Computational Design and Engineering, 10,* 1390–1422.
18. Dorigo, M., Blum, C. (2005). Ant colony optimization theory: A survey. *Theoretical Computer Science, 344,* 243–278.
19. Nama, S., Saha, A. K., Chakraborty, S., Gandomi, A. H., Abualigah, L. (2023). Boosting particle swarm optimization by backtracking search algorithm for optimization problems. *Swarm and Evolutionary Computation, 79,* 101304.
20. Gul, F., Mir, I., Gul, U., Forestiero, A. (2022). A review of space exploration and trajectory optimization techniques for autonomous systems: Comprehensive analysis and future directions. *International Conference on Pervasive Knowledge and Collective Intelligence on Web and Social Media*, pp. 125–138. Messina, Italy.
21. Dekker, R. (1996). Applications of maintenance optimization models: A review and analysis. *Reliability Engineering & System Safety, 51,* 229–240.

22. Abdelhamid, A. A., Towfek, S., Khodadadi, N., Alhussan, A. A., Khafaga, D. S. et al. (2023). Waterwheel plant algorithm: A novel metaheuristic optimization method. *Processes, 11,* 1502.
23. Thirunavukkarasu, M., Sawle, Y., Lala, H. (2023). A comprehensive review on optimization of hybrid renewable energy systems using various optimization techniques. *Renewable and Sustainable Energy Reviews, 176,* 113192.
24. Zhan, Z. H., Shi, L., Tan, K. C., Zhang, J. (2022). A survey on evolutionary computation for complex continuous optimization. *Artificial Intelligence Review, 55,* 1–52.
25. Khodadadi, N., Çiftçioğlu, A. Ö., Mirjalili, S., Nanni, A. (2023). A comparison performance analysis of eight meta-heuristic algorithms for optimal design of truss structures with static constraints. *Decision Analytics Journal, 8,* 100266.
26. Asghari, M., Fathollahi-Fard, A. M., Mirzapour Al-E-Hashem, S., Dulebenets, M. A. (2022). Transformation and linearization techniques in optimization: A state-of-the-art survey. *Mathematics, 10,* 283.
27. Ezugwu, A. E., Agushaka, J. O., Abualigah, L., Mirjalili, S., Gandomi, A. H. (2022). Prairie dog optimization algorithm. *Neural Computing and Applications, 34,* 20017–20065.
28. Wang, G. G., Deb, S., Cui, Z. (2019). Monarch butterfly optimization. *Neural Computing and Applications, 31,* 1995–2014.
29. Izci, D., Rizk-Allah, R. M., Snášel, V., Ekinci, S., Hashim, F. A. et al. (2023). A novel control scheme for automatic voltage regulator using novel modified artificial rabbits optimizer. *e-Prime-Advances in Electrical Engineering, Electronics and Energy, 6,* 100325.
30. Hu, G., Guo, Y., Wei, G., Abualigah, L. (2023). Genghis Khan shark optimizer: A novel nature-inspired algorithm for engineering optimization. *Advanced Engineering Informatics, 58,* 102210.
31. Ghasemi, M., Zare, M., Zahedi, A., Akbari, M. A., Mirjalili, S. et al. (2023). Geyser inspired algorithm: A new geological-inspired meta-heuristic for real-parameter and constrained engineering optimization. *Journal of Bionic Engineering, 8,* 1–35.
32. Wang, G. G. (2018). Moth search algorithm: A bio-inspired metaheuristic algorithm for global optimization problems. *Memetic Computing, 10,* 151–164.
33. Meng, Z., Qian, Q., Xu, M., Yu, B., Yıldız, A. R. et al. (2023). PINN-FORM: A new physics-informed neural network for reliability analysis with partial differential equation. *Computer Methods in Applied Mechanics and Engineering, 414,* 116172.
34. Li, S., Chen, H., Wang, M., Heidari, A. A., Mirjalili, S. (2020). Slime mould algorithm: A new method for stochastic optimization. *Future Generation Computer Systems, 111,* 300–323.
35. Yang, Y., Chen, H., Heidari, A. A., Gandomi, A. H. (2021). Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. *Expert Systems with Applications, 177,* 114864.
36. Ahmadianfar, I., Heidari, A. A., Gandomi, A. H., Chu, X., Chen, H. (2021). RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method. *Expert Systems with Applications, 181,* 115079.
37. Tu, J., Chen, H., Wang, M., Gandomi, A. H. (2021). The colony predation algorithm. *Journal of Bionic Engineering, 18,* 674–710.
38. Ahmadianfar, I., Heidari, A. A., Noshadian, S., Chen, H., Gandomi, A. H. (2022). INFO: An efficient optimization algorithm based on weighted mean of vectors. *Expert Systems with Applications, 195,* 116516.
39. Su, H., Zhao, D., Heidari, A. A., Liu, L., Zhang, X. et al. (2023). RIME: A physics-based optimization. *Neurocomputing, 532,* 183–214.
40. Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M. et al. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems, 97,* 849–872.
41. Gad, A. G. (2022). Particle swarm optimization algorithm and its applications: A systematic review. *Archives of Computational Methods in Engineering, 29,* 2531–2561.

42. Díaz-Manríquez, A., Toscano-Pulido, G., Gómez-Flores, W. (2011). On the selection of surrogate models in evolutionary optimization algorithms. *2011 IEEE Congress of Evolutionary Computation (CEC)*, pp. 2155–2162. New Orleans, LA, USA.
43. Ayesha, S., Hanif, M. K., Talib, R. (2020). Overview and comparative study of dimensionality reduction techniques for high dimensional data. *Information Fusion, 59,* 44–58.
44. Jia, H., Wen, Q., Wu, D., Wang, Z., Wang, Y. et al. (2023). Modified beluga whale optimization with multi-strategies for solving engineering problems. *Journal of Computational Design and Engineering, 10,* 2065–2093.
45. Vinod Chandra, S. S., Anand, H. S. (2022). Nature inspired meta heuristic algorithms for optimization problems. *Computing, 104,* 251–269.
46. Ajagekar, A., You, F. (2019). Quantum computing for energy systems optimization: Challenges and opportunities. *Energy, 179,* 76–89.
47. Zhou, J., Wang, C., Li, Y., Wang, P., Li, C. et al. (2017). A multi-objective multi-population ant colony optimization for economic emission dispatch considering power system security. *Applied Mathematical Modelling, 45,* 684–704.
48. Kaveh, A., Hamedani, K. B. (2022). Improved arithmetic optimization algorithm and its application to discrete structural optimization. *Structures, 35,* 748–764.
49. 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.
50. Cui, Y., Geng, Z., Zhu, Q., Han, Y. (2017). Multi-objective optimization methods and application in energy saving. *Energy, 125,* 681–704.
51. Pawan, Y. N., Prakash, K. B., Chowdhury, S., Hu, Y. C. (2022). Particle swarm optimization performance improvement using deep learning techniques. *Multimedia Tools and Applications, 81,* 27949–27968.
52. Agushaka, J. O., Ezugwu, A. E., Abualigah, L. (2023). Gazelle optimization algorithm: A novel nature-inspired metaheuristic optimizer. *Neural Computing and Applications, 35,* 4099–4131.
53. Azizi, M., Talatahari, S., Gandomi, A. H. (2023). Fire hawk optimizer: A novel metaheuristic algorithm. *Artificial Intelligence Review, 56,* 287–363.
54. Naruei, I., Keynia, F. (2022). Wild horse optimizer: A new meta-heuristic algorithm for solving engineering optimization problems. *Engineering with Computers, 38,* 3025–3056.
55. 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,* 10953.
56. Xue, J., Shen, B. (2023). Dung beetle optimizer: A new meta-heuristic algorithm for global optimization. *The Journal of Supercomputing, 79,* 7305–7336.
57. Das, A. K., Pratihar, D. K. (2022). Bonobo optimizer (BO): An intelligent heuristic with self-adjusting parameters over continuous spaces and its applications to engineering problems. *Applied Intelligence, 52,* 2942–2974.
58. Chopra, N., Ansari, M. M. (2022). Golden jackal optimization: A novel nature-inspired optimizer for engineering applications. *Expert Systems with Applications, 198,* 116924.
59. Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., Gandomi, A. H. (2021). The arithmetic optimization algorithm. *Computer Methods in Applied Mechanics and Engineering, 376,* 113609.
60. Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., Al-Qaness, M. A. et al. (2021). Aquila optimizer: A novel meta-heuristic optimization algorithm. *Computers & Industrial Engineering, 157,* 107250.
61. Abualigah, L., Abd Elaziz, M., 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.

62. Agushaka, J. O., Ezugwu, A. E., Abualigah, L. (2022). Dwarf mongoose optimization algorithm. *Computer Methods in Applied Mechanics and Engineering, 391,* 114570.
63. Oyelade, O. N., Ezugwu, A. E. S., Mohamed, T. I., Abualigah, L. (2022). Ebola optimization search algorithm: A new nature-inspired metaheuristic optimization algorithm. *IEEE Access, 10,* 16150–16177.
64. Bil, C. (2015). Multidisciplinary design optimization: Designed by computer. In: *Concurrent engineering in the 21st century,* pp. 421–454. Springer.
65. Lisicki, M., Lubitz, W., Taylor, G. W. (2016). Optimal design and operation of archimedes screw turbines using Bayesian optimization. *Applied Energy, 183,* 1404–1417.
66. Sbrugnera Sotomayor, N. A., Caiazzo, F., Alfieri, V. (2021). Enhancing design for additive manufacturing workflow: Optimization, design and simulation tools. *Applied Sciences, 11,* 6628.
67. Vanderplaats, G. N., Sugimoto, H. (1986). A General-purpose optimization program for engineering design. *Computers & Structures, 24,* 13–21.
68. Abualigah, L., Elaziz, M. A., Khasawneh, A. M., Alshinwan, M., Ibrahim, R. A. et al. (2022). Meta-heuristic optimization algorithms for solving real-world mechanical engineering design problems: A comprehensive survey, applications, comparative analysis, and results. *Neural Computing and Applications, 4,* 1–30.
69. Abd Elaziz, M., Oliva, D., Xiong, S. (2017). An improved opposition-based sine cosine algorithm for global optimization. *Expert Systems with Applications, 90,* 484–500.
70. Saremi, S., Mirjalili, S., Lewis, A. (2017). Grasshopper optimisation algorithm: Theory and application. *Advances in Engineering Software, 105,* 30–47.
71. He, Q., Wang, L. (2007). An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering Applications of Artificial Intelligence, 20,* 89–99.
72. Huang, F. Z., Wang, L., He, Q. (2007). An effective co-evolutionary differential evolution for constrained optimization. *Applied Mathematics and Computation, 186,* 340–356.
73. Coello, C. A. C. (2000). Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry, 41,* 113–127.
74. Arora, J. S. (2004). *Introduction to optimum design*. Elsevier.
75. Mahdavi, M., Fesanghary, M., Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation, 188,* 1567–1579.
76. 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,* 443–473.
77. Mirjalili, S., Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software, 95,* 51–67.
78. Sandgren, E. (1990). Nonlinear integer and discrete programming in mechanical design optimization. *The Journal of Mechanical Design, 112,* 223.
79. Mirjalili, S., Mirjalili, S. M., Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69,* 46–61.
80. Mirjalili, S., Mirjalili, S. M., Hatamlou, A. (2016). Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Computing and Applications, 27,* 495–513.
81. Liu, H., Cai, Z., Wang, Y. (2010). Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Applied Soft Computing, 10,* 629–640.
82. He, Q., Wang, L. (2007). A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Applied Mathematics and Computation, 186,* 1407–1422.
83. Kaveh, A., Talatahari, S. (2010). An improved ant colony optimization for constrained engineering design problems. *Engineering Computations, 27,* 155–182.
84. Ragsdell, K., Phillips, D. (1976). Optimal design of a class of welded structures using geometric programming. *The Journal of Manufacturing Science and Engineering, 98,* 1021.

85. Deb, K. (1991). Optimal design of a welded beam via genetic algorithms. *AIAA Journal, 29,* 2013–2015.

86. Lee, K. S., Geem, Z. W. (2005). A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice. *Computer Methods in Applied Mechanics and Engineering, 194,* 3902–3933.

87. Kaveh, A., Khayatazad, M. (2012). A new meta-heuristic method: Ray optimization. *Computers & Structures, 112,* 283–294.