IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 15 August 2022, accepted 26 August 2022, date of publication 29 August 2022, date of current version 16 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3202894

RESEARCH ARTICLE

Comparing SSALEO as a Scalable Large Scale Global Optimization Algorithm to High-Performance Algorithms for Real-World Constrained Optimization Benchmark

MOHAMMED QARAAD[1,2], SOUAD AMJAD[1], NAZAR K. HUSSEIN[3], SEYEDALI MIRJALILI[4,5], (Senior Member, IEEE), NADHIR BEN HALIMA[6], AND MOSTAFA A. ELHOSSEINI[7,8], (Member, IEEE)

[1]TIMS, FS, Abdelmalek Essaadi University, Tétouan 93000, Morocco
[2]Department of Computer Science, Faculty of Science, Amran University, Amran, Yemen
[3]Department of Mathematics, College of Computer Sciences and Mathematics, Tikrit University, Tikrit 34001, Iraq
[4]Centre for Artificial Intelligence Research and Optimisation, Torrens University Australia, Brisbane, Fortitude Valley, QLD 4006, Australia
[5]Yonsei Frontier Laboratory, Yonsei University, Seoul 03722, South Korea
[6]Mediterranean Institute of Technology, South Mediterranean University, Tunis 99628, Tunisia
[7]College of Computer Science and Engineering, Taibah University, Yanbu 46421, Saudi Arabia
[8]Computers and Control Systems Engineering Department, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt

Corresponding authors: Mostafa A. Elhosseini (melhosseini@mans.edu.eg) and
Mohammed Qaraad (mohammedalimohammed.qaraad@uae.ac.ma)

ABSTRACT The Salp Swarm Algorithm (SSA) outperforms well-known algorithms such as particle swarm optimizers and grey wolf optimizers in complex optimization challenges. However, like most meta-heuristic algorithms, SSA suffers from slow convergence and stagnation in the best local solution. In this study, a Salp swarm algorithm (SSA) is combined with a local escaping operator (LEO) to overcome some inherent limitations of the original SSA. SSALEO is a novel search technique that accounts for population diversity, the imbalance between exploitation and exploration, and the SSA algorithm's premature convergence. By implementing LEO in SSALEO, the search slowdown in SSA is eliminated, and the local search efficiency of swarm agents is improved. The proposed SSALEO method is tested using the CEC 2017 benchmark with 50 and 100 decision variables, seven CEC2008lsgo test functions with 200, 500, and 1000 decision variables, and its performance was compared to other metaheuristic algorithms (MAs) and advanced algorithms, including seven Salp swarm variants. The comparisons show that SSA greatly benefits from LEO by enhancing the quality and accelerating its solutions' convergence rate. The SSALEO was then assessed using a benchmark set of seven well-known constrained design challenges in various engineering domains defined in the CEC 2020 conference benchmark. Friedman and Wilcoxon rank-sum statistical tests are also used to examine the results. According to experimental data and statistical tests, the SSALEO algorithm is very competitive and often superior to the algorithms used in the studies. Further, the proposed approach can be viewed as a special LSGO optimizer whose performance exceeds that of specialized state-of-the-art algorithms like CMA-ES and SHADE.

INDEX TERMS Hybridization, global optimization, meta-heuristic, swarm intelligence, evolutionary algorithms, large-scale global optimization, salp swarm algorithm, engineering design problems.

I. INTRODUCTION

In real-world applications, challenges such as reducing time, energy, costs, and errors or optimizing efficiency, performance, and quality can be classified as optimization

The associate editor coordinating the review of this manuscript and approving it for publication was Bilal Alatas.

problems [1]. The complexity of problems necessitates new solutions. Metaheuristic algorithms (MAs) are new optimization algorithms that several researchers have talked about recently. Many optimizers for hard problems in the real world have also been made. Such algorithms explore the feature space randomly to find the best solution from various possibilities, mostly inspired by nature. In order to solve challenges relating to global optimization, traditional mathematical methodologies were applied to solve global optimization issues in multiple domains before developing meta-heuristic algorithms [2], [3]. When it comes to dealing with circumstances that are multimodal, discontinuous, and non-convex, these strategies encounter several significant obstacles. As a result, meta-heuristic algorithms based on biological evolution and insect/bird behavior are being created and used to difficult optimization problems [4]. Because meta-heuristics are not accurate algorithms, there is no guarantee that the optimum solution for an optimization issue will be found. However, even though computational resources are restricted, they can identify reasonably good answers in a reasonable amount of time [5].

In the literature, metaheuristics are classified as trajectory-based or population-based. Trajectory-based algorithms use a single solution to optimize the search space. In addition, randomization and particular guidance, such as the greedy criteria in a limited number of iterations, are typically utilized to change and promote the solution. Simulated Annealing (SA) [6], Tabu Search (TS) [7], and Iterated Local Search (ILS) [4] are the most common algorithms in this area. The advantages of algorithms in this category are fast convergence speed and minimal computing cost.

On the other hand, population-based algorithms iteratively build and improve numerous possible solutions within the limits of the problem. Knowledge transfer (information sharing), collaboration, and interaction among candidate solutions are all significantly emphasized in population-based algorithms. A population-based algorithm may fall under several categories: biology-based, social-based (swarm-based), chemical-based, physics-based, music-based, mathematically-based, sports-based, plant-based, and water-based [8]. Classifying their combinations as belonging to a hybrid category is also possible. PSO [8], ABC [9], Ant Colony Optimization [10], [11], DE [12], GA [13], GWO [14], CSO [15], and BAT [16] are all examples of biologically based algorithms. The Evolutionary Centers Algorithm (ECA) [17] and Electromagnetic Field Optimization (EFO) algorithm [18] are both examples of physics-based algorithms. The social group optimization SGO algorithm [19] is an example of a socially based algorithm. The Artificial Chemical Reaction Optimization algorithm ACROA [20] is an example of a chemistry-based algorithm. Melody search [21] is an example of a musically based. Recent research has led to the development of chaotic league championship algorithms for use in sport-based algorithms and models [22].

A population-based algorithm can avoid local solutions and exhibit better exploratory behavior. They are, however, computationally more expensive and necessitate information sharing between numerous solutions.

Population-based algorithms employ a variety of random operations such as crossovers to boost exploration and exploitation abilities [23], mutation [24], and selection [24]. Because of the benefits outlined above, population-based metaheuristics are quite popular and frequently employed today. Several MAs have thus been created for use in biomedicine [25], bioinformatics [26], cheminformatics [27], feature selection [28], engineering issues [29], [30], pattern recognition, text clustering [31], and wireless sensor networks [32]. On the other hand, all meta-heuristic (MA) algorithms need to strike the equilibrium between the exploration and exploitation stages. If they don't, the solutions either don't converge or become stuck in local optima [33], [34]. Such issues can arise as a result of randomization during the solution-finding process.

In 2017 [35], Mirjalili *et al.* came up with the idea for a contemporary population-based metaheuristic search algorithm and called it the Salp Swarm Algorithm (SSA). This metaheuristic algorithm attempts to simulate the behavior of deep-sea salps, namely their swarming and foraging patterns. Even though the mathematics that underpins SSA is rather straightforward, it may be more effective than other contemporary algorithms at solving challenging engineering optimization issues. These include GWO, ABC, CSA, and others [23].

The SSA algorithm has fewer parameters and a simple implementation [23]. The SSA demonstrated solving both large and minor issues [36]. In addition, SSA is distinguished by its adaptability and stochasticity [23]. However, the SSA, in addition to other optimization methods, has two disadvantages. In the first place, the convergence speed is insufficient to generate accurate solutions. Another drawback is that it lacks the exploratory possibilities of evolutionary algorithms that use crossover operators. This is a significant limitation. These issues frequently arise in most optimization strategies, particularly in complicated and high-dimensional situations [37]. As a result, various attempts have been made to solve the problem [38].

Researchers have devised several different modified SSA versions that improve the standard SSA's efficiency, eliminate any faults, and expand its possibilities despite any inherent restrictions it may have. This paper presents a modified Salp Optimization (SSALEO) version based on a local escaping operator (LEO). Modifying nature-inspired algorithms is a popular way to address those faults by strengthening the inventive optimizer's exploitation and exploration capabilities. For example, unknown search regions can be visited, and the local optima problem can be avoided using the new mathematical technique of the "local escaping operator," which is used in local searches to find an effective solution [39]. The suggested method has been validated using a series of test functions. Firstly, we demonstrate the implications of the proposed improvements on SSA. Next, SSALEO performance is evaluated and validated using the CEC'2017 test suite with 50 and 100 dimensions and the CEC2008lsgo benchmark test functions with 200, 500, and 1000 dimensions, respectively. Later, the SSALEO's applicability was validated by tackling seven common engineering design difficulties. The results of the experiments show that SSALEO has a successful optimization capability. This paper's key contributions are as follows:

- The problems of population variety, an unbalanced relationship between exploitation and exploration, and the premature convergence of Salp swarm algorithms inspired the development of SSALEO.
- The CEC'2017 and CEC'2008LSGO benchmarks were utilized to verify the suggested strategy's validity (SSALEO).
- The SSALEO was assessed using the engineering benchmark defined in the CEC 2020 conference.
- SSALEO is a scalable LSGO optimizing technique that outperforms its competitors.
- Friedman and Wilcoxon rank-sum tests determine whether the variations in algorithm performance are statistically significant.
- SSALEO outperforms various SSA variants and other sophisticated algorithms in Wilcoxon rank-sum and Friedman tests.

The remainder of this work is organized as follows. The second section discusses the large-scale global optimization problem, while Section 3 focuses on SSALEO's related work. Section 4 contains some preliminary information and theoretical background information. The principles of LEO and SSA algorithms are discussed in this section. Section 5 goes into great depth about the SSALEO. The algorithm's results are analyzed and explained in Section 6, which uses CEC 2017 benchmark functions with 50 and 100 decision variables and CEC 2008 benchmark functions with 200, 500, and 1000 decision variables. Seven well-known engineering design challenges were used in Section 7 to evaluate the proposed method. Section 8 explores the constraints and complexities of the problem. Finally, section 9 concludes with the conclusions.

II. LARGE-SCALE GLOBAL OPTIMIZATION

In real-world problems, large-scale global optimization efficiently handles many decision variables. However, as the number of decision variables and the problem's multimodality have increased, finding solutions to this set of challenges has become increasingly challenging. In addition, the local search space is typically limited, which makes it more difficult to identify the optimal answer for the entire global best solution. Several population-based meta-heuristics have been created in recent decades to address these challenges [40]. However, the downside to these solutions is that they dramatically increase simulation complexity while degrading performance. As a result, various algorithms for dealing with LSGO problems have been created. Currently, the solutions can be classified into two types [40]: Non-Decomposition methods and cooperative coevolution (CC) methods based on the dimension decomposition optimization strategy.

Potter and De Jong [41] proposed the Cooperative Coevolution (CC) approach (1994). While when applying the CC method, the LSGO problem is broken down into a series of low-dimensional problems. The solutions to those problems are combined to form a high-dimensional optimization challenge. Following that, researchers classified CC approaches into two types based on variable grouping strategies for LSGO problems: static and dynamic grouping methods. First, Potter used static grouping-based CC approaches on the evolutionary algorithm to produce a decent solution, the first CC algorithm to address LSGO problems. Then, by mixing the solutions from each subcomponent, the n-dimensional solution is created. Next, Yang *et al.* [42], [43], [44] solved LSGO issues with 500 and 1000 dimensions using a DE-based cooperative coevolving (CC) technique dubbed DECC-G, employing random grouping of decision variables. The multilevel CC approach, also known as MLCC, uses decomposers with an adjustable group size depending on their performance. Other similar algorithms in the literature are CCPSO [45] and CC-CMA-ES [46].

Non-Decomposition-based algorithms, on the other hand, avoid the divide-and-conquer strategy in favor of a range of successful methods for improving algorithm performance. The most frequent methodologies are local search-based [47], [48], evolutionary computation-based [49], [50], and swarm intelligence-based approaches [51] are the most common categories. For instance: a modified CSO (MCSO) [52] algorithm with two-thirds of search agents updated by a competitive try criterion is proposed. The MCSO was chosen to address large-scale optimization issues, and the findings revealed that it outperformed state-of-the-art algorithms.

This paper presents a modified PSO based on a population-based approach to address the LSGO problem [53]. The whale optimization algorithm (WOA) [54] uses quadratic interpolation to handle large-scale issues, which aids in increasing the algorithm's exploitation capabilities [55]. Cano and GarciaMartinez [56] tackle 100 million dimension issues in large-scale global optimization using an evolutionary computation technique and a modern GPU. Cano, Garcia-Martinez, and Ventura [57] also published a MapReduce implementation of the MA-SW-Chains algorithm as a new method version. The approaches solve 10 million dimensions of CEC functions for the first time.

In [58], the authors combine the MA-SW-Chains algorithm with the Local search strategy to create a high-performance memetic solution for high-dimensional issues. Furthermore, a modified SCA called DSCA

Although it can boost optimizations on a massive scale, CC comes with several main limitations:

- Its performance is influenced by the decomposition strategy used.

AQ:6

TABLE 1. Modifications and hybridizations to the SSA.

Approach	Enhancement Type and Year	Problem	methodology
An enhanced version of the salp swarm technique that is based on the simplex method SMSSA [69]	Modified version, 2018	Four optimization problem	Applying the simplex approach ensures that the positions of the worst Salp are always up to date after each iteration. This simplex technique is a random variation strategy that broadens the population and increases the algorithm's ability to search locally in a given environment.
An enhanced version of the salp-swarm method ISSA for feature selection [63]	Modified version, 2018	Feature selection (Twenty-three benchmark)	A new control parameter known as the inertia weight ∈ [0,1] has been included to maximize the best solution found so far and quicken the rate at which convergence occurs during the search. In addition, it strikes a healthy balance between the powers of exploitation and exploration.
A method that is a combination of salp swarming and simulated annealing. SSA-SA [70]	Hybrid version, 2018	Feature selection (16 benchmarks)	Simulated annealing is incorporated within the Salp swarm algorithm to improve the final solution by substituting the original SSA solution with the SA-enhanced solution.
SSA-HJ combines salp swarm and Hooke-Jeeves. [71]	Hybrid version, 2018	optimal design of CMOS	The Hooke-Jeeves algorithm is combined with the Salp swarm algorithm. The leader salp is the starting point for the HJ algorithm once the SSA implementation is complete. The HJ algorithm fine-tunes the global solution to improve optimization efficiency. Finally, the solution produced by the HJ algorithm is the best.
CBSSA1–CBSSA4 [72]	Modified version, 2018	Feature selection(twelve benchmarks)	They employed chaotic sequence maps (circular, logistic, piecewise, and tent) to replace the uniformly distributed random c3 variable in the Leader position update equation rather than utilizing random numbers. This allowed them to predict the leader's position more accurately.
Chaotic Binary SSA (CBSSA) [73]	Modified version, 2018	30 graphs DIMACS graph coloring benchmarks	The chaotic logistic map replaces the SSA mathematical model's random variables (c2, c3).
Hybrid PSO utilizing the Salp SSA (also known as SSAPSO). [74]	Hybrid version, 2019	fifteen benchmark functions. Feature selection (different dataset)	To keep the population current, either the SSA or the PSO approach may be utilized, depending on the likelihood of the fitness function. If the probability of fitness function for the current solution is more than 0.5, the SSA will be implemented; otherwise, the PSO will be used.
Memetic SSA (MSSA) [75]	Modified version, 2019	Thirteen benchmark functions maximum power point tracking (MPPT) of PV systems under PSC.	The MSSA is an extension of the SSA that includes other confident salp chains, allowing considerable exploration and exploitation under the same computing structure. In addition, a virtual population-based regroup method improves convergence durability for the global controller across multiple salp chains.
Integrates SSA with SCA (HSSASCA) [76]	Hybrid version, 2019	twenty-two benchmark functions and three engineering design issues	SSA's exploration and exploitation propensity was improved by including the sine and cosine functions in the location update equation in this integration.
Chaotic SSA (CSSA) [77]	Modified version, 2019	Evaluates the ideal amount of controllers and switches in big SDN networks.	The method dynamically assesses the optimal number of controllers and controller-switch links in large-scale SDN networks. Parallel salps iterate to find the optimal answer. The ideal solution has the right amount of controllers and switches. To find the best responses, two algorithms are nested.
An enhanced version of the SSA using space transformation search STS-SSA [78]	Modified version, 2020	CEC-2017 test suite and trains the NN network	STS increases SSA's performance. Modified SSA calculates candidate solutions simultaneously in the existing search space and a newly transferred search space. The best candidate solution is used to evaluate the following phase.

TABLE 1. *(Continued.)* **Modifications and hybridizations to the SSA.**

An enhanced version of the SSA based on LSA and OBL (ISSA_OBL) [79]	Hybrid version, 2020	18 benchmark datasets, Feature selection	Opposition Based Learning (OBL) was utilized during the initialization phase of SSA to improve its population diversity in the search space. Then a new Local Search Algorithm (LSA) was applied at the end of each iteration of SSA to enhance its exploitation.
TVBSSA[80].	Modified version, 2020	Feature selection(20 datasets)	Time-based and dynamic, TVBSSA's leadership hierarchy. This strategy increases leaders and decreases followers across iterations. In each iteration, several leaders are picked while the remainder of the population is dispersed as follower salps.
Dynamic salp swarm algorithm DSSA [28]	Modified version, 2021	23 benchmark datasets, Feature selection	The first enhancement is the creation of a new equation for updating the Salps' position. Singer's chaotic map is used to control this new equation. The first enhancement is to broaden the range of SSA solutions. The second improvement is creating a new local search algorithm (LSA) to better SSA exploitation.
Improved SSA based on FFA (SSAFA) [81]	Hybrid version, 2021	Unrelated parallel machine scheduling problem (UPMSP)	A population is updated using the SSA or FA method operators based on the likelihood of the fitness function. If the probability of fitness function for the current solution is greater than 0.5, the FA is used; otherwise, the SSA is used.
An enhanced version of the WOA – SSA (IWOSSA) [82]	Hybrid version, 2021	23 test suite benchmark	Depending on p, the population is updated using SSA or WOA (where p is an arbitrary number in [0,1]). If the present solution's p-value is greater than 0.5, the SSA is used; otherwise, the WOA is utilized.
Enhanced Salp swarm based on Quadratic interpolation and local escape operator QSSALEO [83]	Hybrid version, 2022	Cec2017, cec2008, 19 benchmark datasets, Feature selection	Using quadratic interpolation and a local escape operator, a QSSALEO is offered as a solution to address the underlying problems with SSA (LEO). Quadratic interpolation around the best search agent helps improve QSSALEO's exploitation ability and solution correctness. In contrast, the local escaping operator uses random operators to escape local optima in place of quadratic interpolation around the best search agent.

- Increasing the number of interrelated components will reduce its effectiveness.
- EA's evolutionary algorithms decide efficiency for non-separable optimization problems.
- There is a significant amount of computational complexity involved.

Since CCEAs can only handle a finite number of large-scale optimization problems, researchers are working to create new search algorithms for traditional EAs that can better utilize the finite number of FEs. Consequently, swarm intelligence, a non-decomposition approach, is used in this study to handle large-scale difficulties.

III. RELATED WORK

Various academic fields have been using natural-inspired algorithms to solve multiple problems. However, the No-Free Lunch Theorem claims that no single optimization strategy can solve all optimization issues [59]. Consequently, it is challenging to build novel optimization methods to address problems in real-world applications. As a result, combining fundamental meta-algorithms to create novel optimization algorithms is becoming increasingly popular. Furthermore, it's possible to mix the best features of existing algorithms to develop new ones that are more efficient and accurate through hybridization.

Since its initial release in 2017, the SSA method has been updated to accommodate many modifications produced by researchers to address and solve a wide range of optimization challenges. For the purpose of global optimization, Fan *et al.* [53] suggested combining two different algorithms known as the Whale Optimization Algorithm (WOA) and the Salp Swarm Algorithm (SSA) [60]. Likewise, Qaraad *et al.* [61] created a new hybrid methodology named SSAGWO by updating Salp followers' locations using the GWO mechanism. Finally, SSAGWO was applied to deal with the feature selection problem.

The Salp Swarm Algorithm based on the Levy Flight and Sine Cosine Operator algorithms were created by Zhang *et al.* [62]. LSC-SSA performed an outstanding job compared to the work done by the other optimizers. CMSSA was developed by Mirjalili *et al.* [35] and is an improvement on fundamental SSA exploitative mechanisms. In CMSSA, chaotic exploitative processes are combined with a "shrinking" mode to improve basic mechanisms for exploiting SSA. However, the proposed method has high computational complexity due to the fact that the number of function evaluations for a single iteration is equal to the square root of the number of evaluations performed by the initial SSA. In order to make adjustments to the current best and follower positions, Hegazy *et al.* [63] applied an inertia weight to SSA. The newly developed approach improved the success rate of SSA when applied to the problem of feature selection. Aljarah *et al.* [64] tried to improve the structure of the initial SSA by dividing the Salp chain into sub-chains and implementing asynchronous updating rules. In their study [23], Faris *et al.* presented and assessed eight different binary SSA variations by employing eight other SSA transfer functions.

In addition, the average operator was replaced with the crossover operator so that the algorithm would have a higher degree of global searchability. Chen *et al.* [65] suggested a new configuration of the SSA that would have many leaders. Finally, it was recommended to include a piecewise equation in the method to facilitate the convergence process.

Nevertheless, similar to other optimization methods, SSA is plagued by several problems, the most notable of which are local optima and population diversity problems. Therefore, significant adjustments were made to the SSA's algorithm (see Table 1). In addition, multiple local search approaches have been hybridized with various optimization algorithms in the literature to make them relevant for datasets of diverse dimensions. This is something that has been done in many different ways. As an instance, Houssein *et al.* [66] enhanced the performance of the Archimedes optimization process by employing the local escaping operator (LEO) operator (AOA). The tunicate swarm algorithm (TSA) was enhanced by Houssein *et al.* [67] by the addition of a local escape operator (LEO) operator, which increased the swarm agent convergence rate and improved the local search efficiency. Marine Predators Algorithm (MPA) was enhanced by M.Oszust [68] with the help of a Local Escaping Operator so that it could achieve global optimization. In Table 1, the SSA is modified and hybridized with various metaheuristic algorithms in chronological order, beginning with the oldest of the changes.

After considering everything that has been discussed so far, one can come to the following conclusions:

- A substantial amount of literature about the SSA algorithm has been published.
- The salp swarm algorithm has previously been shown to perform well in various situations. However, even though SSA is more effective than other optimizers with a long track record, local solutions may still be a barrier.
- Several distinct SSA algorithm versions have been produced by altering the SSA mechanism to enhance convergence speed, avoid optimal solution, and maintain a balance between exploratory and exploitative operations.
- Complex optimization algorithms often make use of diversity approaches to improve search quality. This is accomplished by minimizing the harmful effects of genetic drift, which is the root cause of diversity loss in bio-inspired algorithms.

These findings suggest that it is in the algorithm's favor to search for many optimal solutions during the earlier stages of the process. However, according to the previous discussion and the recommended changes in Table 1, the essential issues with SSA are premature convergence, being locked in local optimums, low population variation, and a poor balance between exploration and exploitation. Because of this, an improved SSA approach (SSALEO) is proposed in this study to address SSA's shortcomings. Following the works described above, SSALEO varies from them in the following ways:

- With the use of an efficient operator, this study has overcome the shortcomings of the classic SSA, such as (1) not getting stuck in local optima, (2) keeping exploration and exploitation in balance, and (3) increasing convergence speed.
- A novel mathematical technique called a local escaping operator (LEO) is a local search for developing an effective solution that intends to explore unobserved search regions and escape from the local optimal problem.
- High-dimensional and engineering design-constrained test functions were used to evaluate the proposed SSALEO algorithm. According to the statistical test analysis, research shows that the proposed method effectively deals with those issues. As a result, the results reveal that the new method is superior in most situations.

IV. PRELIMINARIES

The local escape operator and fundamental Salp swarm algorithm (SSA), as well as its analogies and mathematical models, are discussed in this section.

A. LOCAL ESCAPE OPERATOR (LEO)

In Ahmadianfar *et al.* [39], the LEO is proposed as a local search algorithm that is utilized to improve the ability of an optimization method, especially the Gradient-based Optimizer (GBO), to explore new search regions that are required in difficult real-world challenges. LEOs improve the overall quality of solutions by maintaining their positions according to a set of criteria. The behavior of the algorithm's convergence is enhanced as a direct result of this feature, which prevents the algorithm from being trapped in local optima. LEO creates high-quality alternative solutions (X_{LEO}) by combining many different solutions, such as the best position X_{best}, two randomly created solutions X_{r1}^m and X_{r2}^m two randomly selected solutions $X1_n^m$ and $X2_n^m$, and a new randomly generated solution X_k^m. This allows LEO to develop solutions that perform exceptionally well. As a consequence of this, the value X_{LEO} may be determined by utilizing Equations (1) and (2), which, in mathematical terms, can be expressed as follows:

$$\begin{aligned}
&\boldsymbol{if}\ r_2 < \mathbf{0.5}\\
&X_{LEO}^m = X_n^{m+1} + f_1 \times \left(u_1 \times X_{best} - u_2 \times X_k^m\right)\\
&\qquad + f_2 \times \rho_1 \times \left(u_3 \times \left(X2_n^m - X1_n^m\right) + u_2\right.\\
&\qquad \left.\times (X_{r1}^m - X_{r2}^m)\right)/2\\
&\qquad \times X_n^{m+1} = X_{LEO}^m
\end{aligned} \tag{1}$$

$$\begin{aligned}
&\boldsymbol{Else}\ X_{LEO}^m = X_{best} + f_1 \times \left(u_1 \times X_{best} - u_2 \times X_k^m\right)\\
&\qquad +, f_2 \times \rho_1 \times \left(u_3 \times (X2_n^m - X1_n^m\right) + u_2\\
&\qquad \times (X_{r1}^m - X_{r2}^m))/2 \times\\
&X_n^{m+1} = X_{LEO}^m
\end{aligned} \tag{2}$$

X_n^m represents the current salp position, X_{best} is the location of the food with the highest total score, r_2 is a random number in the range [0,1], and both $f1$ and $f2$ are random numbers drawn from a uniform distribution in the range $[-1, 1]$. X_{r1}^m and X_{r2}^m are two random solutions that were obtained from the population of the Salp. $X1_n^m$ and $X2_n^m$ are two alternative solutions that were generated at random from the current population of the Salp, as shown in Equation (3):

$$X1_n^m, X2_n^m = lb + \text{rand}(0, 1) \times (ub - lb) \tag{3}$$

where lb and ub signify the lower and upper limits, respectively. *rand* is a random number in the range [0,1]. In addition, n and m represent the coordinates of the solution, where n can range from 1 to N and m can range from 1 to Dim. Additionally, u_1, u_2, and u_3 are three variables that are generated by random processes in the following manner:

$$u_1 = L_1 \times 2 \times \text{rand} + (\text{ub} - L_1) \tag{4}$$

$$u_2 = L_1 \times \text{rand} + (\text{ub} - L_1) \tag{5}$$

$$u_3 = L_1 \times rand + (\text{ub} - L_1) \tag{6}$$

L_1 represents a binary parameter ($L_1 = 1 if \mu_1 < 0.5$, *and* 0 *therwise*), μ_1 represents a number in the range 0 and 1.

In addition, ρ_1 is implemented to maintain a healthy equilibrium between the searching processes of exploration and exploitation, and it can be described as follows:

$$\rho_1 = \alpha \times (2 \times \text{rand} - 1) \tag{7}$$

$$\alpha = \left|\beta \times \sin\left(\frac{3\pi}{2} + \sin\left(\beta \times \frac{3\pi}{2}\right)\right)\right| \tag{8}$$

$$\beta = \beta_{min}\left(\beta_{max} - \beta_{min}\right) \times \left(1 - \left(\frac{T}{T_{max}}\right)^3\right)^2 \tag{9}$$

The current iteration is denoted by T, while the maximum number of iterations is denoted by $Tmax$. The parameters β_{min} and β_{max} are configured to have values of 0.2 and 1.2, respectively. ρ_1 varies as a function of the sine function α to maintain a healthy equilibrium between the exploration and exploitation processes.

The following strategy is suggested as a means of locating the valueX_k^m in Equation. (1) and Equation. (2):

$$X_k^m = \begin{cases} X_{rand} & if\, \mu_2 < 0.5 \\ X_p^m & otherwise \end{cases} \tag{10}$$

where μ_2 is a number between 0 and 1, X_p^m is an illustration of a solution chosen randomly from the salps population (p ranges from 1 to N). X_{rand} is a new solution that may be found by following the above equation (11).

$$X_{rand} = \text{lb} + \text{rand}(0, 1) \times (\text{ub} - \text{lb}) \tag{11}$$

Eq. (10) can be written as follows:

$$X_k^m = L_2 \times X_p^m + (1 - L_2) \times X_{rand} \tag{12}$$

where L_2 is a binary parameter with a value of either 0 or 1, depending on the context. If the parameter μ_2 is less than 0.5, then the value of L_1 has a value of 1, and if it is greater than 0.5, then the value of L_1 is 0.

B. SALP SWARM ALGORITHM (SSA)

Mirjalili *et al.* [35] developed the Salp swarm algorithm (SSA), which is one of the most recently published swarm optimization methods. The SSA algorithm's core idea is to emulate the swarming behavior of salps in the water using the salps chain concept. Salps are barrel-shaped organisms that belong to the Salpidae family. Furthermore, the tissues and movements of salps are similar to jellyfish [84]. During their lives in the water, salps display a peculiar swarming behavior called a "salp chain" activity. This activity, which can be exploited in the salps' motions as they look for food, can be observed throughout their lives.

The members of SSA can be broken down into two categories: leaders and followers. The leader of the Salps chain is responsible for determining movement directions, selecting food places, leading the SSA chain to the food, and regularly updating the sites. The term "followers" is used to refer to the remaining members of the population. Each follows the leader in turn to establish the chain structure. Each salps point in the search space is characterized by n dimensions, where n represents the number of variables involved in the problem. In addition, the food supply denoted by the letter F is a metaphor for the salps' search aim. The following is one possible representation of this process:

$$x_j^1 = \begin{cases} F_j + r_1\left(\left(ub_j - lb_j\right) r_2 + lb_j\right) & r_3 \geq 0.5 \\ F_j - r_1\left(\left(ub_j - lb_j\right) r_2 + lb_j\right) & r_3 < 0.5 \end{cases} \tag{13}$$

x_j^1 represents the chain Salps leader position with the jth dimension. Fj stands for the food position with the jth dimension, ubj and lbj stand for the upper and lower bounds of Salps position components, respectively. r2 and r3 are two scalars that have been chosen at random from the range [0,1]. During the iteration process, the most important control parameter to pay attention to is $r1$, which is what stabilizes the exploration and exploitation phases. The following is the expression for the variable $r1$:

$$r_1 = 2e^{-\left(\frac{4t}{\mathrm{T}}\right)^2} \tag{14}$$

where the numbers t and T respectively signify the current number of iterations and the maximum number of possible iterations. The following equation is used to calculate an update to the Salps chain of followers' positions in such a way that $i \geq 2$:

$$x_j^i = \frac{1}{2}\left(x_j^i + x_j^{i-1}\right) \tag{15}$$

Following Isaac Newton's theory of motion:

$$x_j^i = \frac{1}{2}k \times \text{time}^2 + s_0 \times \text{time} \tag{16}$$

where, x_j^i is the position of the i-th follower in the j-th dimension, t denotes the time, s_0 is the initial speed, and $k = \frac{s_{final}}{s_0}$ where

$$s = (x - x_0)/\text{time} \tag{17}$$

V. THE PROPOSED ALGORITHM

When attempting to solve optimization problems, it is sometimes challenging to prevent the algorithm from getting stuck in the value of the local optima. The agent must search the solution space thoroughly to avoid local optimization. Consequently, an original SSA does an excellent job of striking a balance between exploration and exploitation. On the other side, the SSA has the following problems:

- The leader's position update model causes a decrease in the effectiveness of the food search throughout repeated iterations, which results in a stagnation effect.
- Within the SSA mathematical paradigm framework, there is no logical transition from exploiting to discovering.
- The resolution of low-dimensional optimization issues is the primary focus of the great majority of the SSA-enhanced approaches now in use. It is not known, however, whether or not SSA is capable of effectively tackling high-dimensional optimization difficulties.

We use a low-level co-evolutionary heterogeneous hybrid that blends SSA with LEO to evade the problems that SSA causes. The suggested algorithm, known as SSALEO, combines SSA with LEO. LEO is primarily used to increase the performance of the finest original SSA solutions. It is essential, particularly in human-aided systems, to combine a great number of theories and ideas originating from a variety of scientific fields. Through hybridization, it is possible to integrate the benefits of multiple algorithms to build improved versions with guaranteed performance and accuracy. According to [85], two algorithms can be hybridized in either a homogeneous or heterogeneous fashion, either at a high level or a low level, using a relay approach or a co-evolutionary strategy. The hybrid is low-level because we combine the capabilities of both techniques. However, it is co-evolutionary because we do not employ both methods sequentially. In other words, they run at the same time.

The SSALEO algorithm uses the LEO operator to stimulate the visitation of new regions while following the key phases of the traditional SSA. In addition, LEO increases the algorithm's search for global optima and its convergence rate, dynamically avoiding stagnation in local optima. In the next part, we will provide a comprehensive breakdown of the SSALEO implementation that has been suggested.

A. THE PRIMITIVE STEP OF THE SSALEO

Initialization is the first step in SSALEO, just like it is in the majority of optimization methods. This step involves establishing an initial population of (N) search agents. Each search agent has a dimension (Dim) in the search space, which is bound by the upper and lower limits, as shown in Eq. (18).

$$\vec{P} = \mathrm{lb} + \mathrm{rand}\,(N, Dim) \times (\mathrm{ub} - \mathrm{lb}) \tag{18}$$

Within the search space, each solution is constrained between the upper and lower bounds by a dimension known as Dim. $\vec{P}$ stands for the initial salps population, and N denotes the number of random solutions that can be generated at any time. The lower and upper bounds are indicated by lb and ub, respectively.

B. THE DIFFERENT SCENARIOS FOR THE SSALEO UPDATE

Two different sets of conditions determine the technique for updating the salp position. First, using equation (13), construct an agent solution based on the food position obtained up to this point, and store the results. During this phase, the initial SSA is completed as a matter of course. Then, in the second scenario, the solution is upgraded to improve efficiency by applying the LEO technique. This is done in the second scenario. The conditional nature of the LEO differentiation between the two paths is illustrated by Equations 1 and 2, respectively. If ($rand$ is less than 0.5), then the first path is selected as the one to take to continue the process of updating the solution, as shown in Eq (1). Otherwise, the second option, Equation (2), will be utilized to locate the new solution.

C. THE SSALEO SCENARIOS OF OPTIMIZATION

To enhance the overall quality of the succeeding solutions, it is necessary to perform this step at the beginning of each iteration to assess the vector of solutions produced in the preliminary phase. As a direct consequence of this, within the existing population, SSALEO determines the fitness value, denoted by the notation $Fitness(\vec{P})$ of each salp position. The best-scoring solution Xbest is determined, saved, and extracted at the updating stage.

D. CRITERIA FOR TERMINATION

After finishing all the optimization scenarios and iterating until all the stopping requirements are satisfied, the recommended SSALEO will find the best possible solution. Algorithm 1 provides the SSALEO algorithm's pseudocode, and Fig. 1 depicts a full flowchart.

E. COMPUTATIONAL COMPLEXITY

The computing complexity in practice should judge metaheuristics. The time required to initialize the population for the proposed SSALEO and other algorithms (such as SSA, GWO, PSO, CSO, SCA, and WOA) is O($n_o \times n_p$) time, where n_o represents the number of objectives and n_p denotes the size of the population. The time required to initialize the population for the proposed SSALEO. When it comes to search strategies, calculating the fitness of search agents takes $O(Maximum_{iterations} \times O_f)$ time, where O_f is the objective function for the problem at hand. The entire process requires time denoted by the notation $O(N)$. The suggested SSALEO algorithm has a computational complexity of $O(s)+ O(Maximum_{iterations} \times (S + S \times dim))$, where $O(S)$ is the number of search agents and dim is the dimension of the problem. The overall complexity of $O\,(Maximum_{iterations} \times S \times Dim)$ will vary according to this. The suggested SSALEO and several other algorithms are each given their typical execution times in Table 2.

TABLE 2. The length of the execution comparison (in seconds) of the performance of SSALEO and the other options in solving the benchmark functions for CEC2017, which each have 100 dimensions and are executed 30 times independently.

F	SSALEO	BAT	CSO	HHO	PSO	MFO	SCA	SSA	WOA
CEC_F1	118.210	192.030	220.680	22.540	255.410	286.040	159.790	195.090	232.360
CEC_F2	136.900	203.310	231.040	40.110	264.920	265.970	164.050	203.660	241.710
CEC_F3	160.830	213.220	243.130	63.430	275.400	248.280	171.370	216.400	253.710
CEC_F4	137.780	202.740	231.760	39.500	264.150	235.560	164.130	205.380	240.560
CEC_F5	205.160	236.260	265.650	111.370	296.770	269.590	186.980	238.280	274.570
CEC_F6	171.890	223.460	250.520	77.430	282.180	255.090	174.890	224.590	258.580
CEC_F7	261.360	265.440	295.710	171.250	324.940	298.010	203.900	268.150	304.510
CEC_F8	207.480	237.100	267.790	115.240	297.460	271.080	190.030	242.640	274.740
CEC_F9	229.780	253.580	279.100	138.140	307.930	355.740	197.320	251.640	288.410
CEC_F10	161.320	214.170	241.590	65.150	273.200	261.550	202.890	216.270	251.110
CEC_F11	170.510	221.600	248.090	73.520	279.760	253.950	175.080	220.520	256.380
CEC_F12	167.160	221.480	246.550	69.790	276.120	249.810	171.780	218.850	255.180
CEC_F13	164.100	219.230	222.320	67.820	269.440	248.480	174.080	218.760	252.520
CEC_F14	154.340	203.310	195.340	57.490	224.990	245.200	174.030	213.820	248.610
CEC_F15	118.210	191.190	261.910	108.650	298.650	268.260	187.230	235.140	271.900
CEC_F16	200.820	392.070	420.720	430.480	459.880	389.030	285.850	392.740	433.060
CEC_F17	512.640	208.260	237.220	50.530	270.220	213.170	167.630	210.420	243.080
CEC_F18	147.460	260.590	290.940	158.650	324.830	370.800	202.480	262.790	298.340
CEC_F19	252.110	384.610	414.250	410.160	448.760	426.970	279.440	385.950	421.500
CEC_F20	496.080	271.830	301.330	145.530	334.030	302.590	207.330	274.480	307.450
CEC_F21	271.240	251.050	336.940	205.290	369.180	339.880	233.220	308.370	344.850
CEC_F22	340.230	230.640	367.960	267.220	408.490	375.930	286.250	343.020	382.520
CEC_F23	376.220	277.390	307.840	159.440	343.850	310.410	209.760	278.030	318.530
CEC_F24	230.230	305.010	309.590	202.270	373.950	427.260	226.620	308.730	346.630
CEC_F25	274.460	415.910	323.830	383.510	390.930	450.930	291.030	360.930	423.340
CEC_F26	347.410	421.180	286.840	384.990	333.960	448.950	245.840	283.290	346.270
CEC_F27	344.960	322.740	265.080	319.640	310.950	411.950	224.610	245.290	292.500
CEC_F28	298.770	319.610	340.370	511.230	387.900	499.720	290.050	325.030	369.950
CEC_F29	425.790	304.860	201.270	247.360	229.030	313.210	204.140	220.810	261.240
AVG.	244.25690	264.27140	279.49520	175.78380	316.45790	315.2310	208.68280	261.00240	299.79690

Algorithm 1 Pseudocode ofSSALEO

Initialize the population matrix ($\vec{P}$) *according to upper and lower bounds (population size, dimensions).*
Evaluate the initial *Particles*
Sort the fitness values and set *FoodFitness* as the best salp's fitness and FoodPosition as the best salp's position
while (stopping condition is not hold)
  Compute r1 by Eq. (14)
  for (each Particle in (*Particles*))
    if ($i \leq N/2$) **then**
      Update the position of the leading Particle by Eq. (13)
    else
      Update the followers' Particles by Eq. (15)
    /* LEO strategy */
    if ($u_1 < 0.5$) **then**
      calculate X^m_{LEO} using Eq.(1)
    else
      calculate X^m_{LEO} using Eq.(2)
    Evaluate the New X^m_{LEO} and record them as *fitness*
    if (*Fitness*(X^m_{LEO}) < *Fitness*(CurrentSalpPosition)) **then**
      CurrentSalpPosition = X^m_{LEO}
      CurrentSalpPositionFitness = *Fitness*(X^m_{LEO})
      update the *FoodPosition* and *FoodFitness*
  end-for
end-while

As seen in Table 2, SSALEO was evaluated with eight other competitors in this subsection to assess their ability to compute time-intensive experiments included in the CEC 2017 benchmarks. Due to the time-consuming nature of the calculation method, it is essential that each participant carry out each function a total of thirty times and then report the outcomes in Table 2. In addition, the data in the table demonstrate that the computation of SSALEO takes a more extended time since the integration method, which requires a greater amount of processing resources, is utilized. On the other hand, SSALEO can beat certain algorithms while requiring less time. These algorithms include CSO, BAT, PSO, MFO, SCA, SSA, and GWO. SSALEO has substantial advantages over other algorithms, despite being rather time-consuming.

VI. EXPERIMENTAL RESULTS AND ANALYSIS

Using benchmark functions with various properties is a common approach while conducting tests on optimization algorithms with a stochastic nature. Benchmark functions have known global optima and mimic real-world optimization problems.

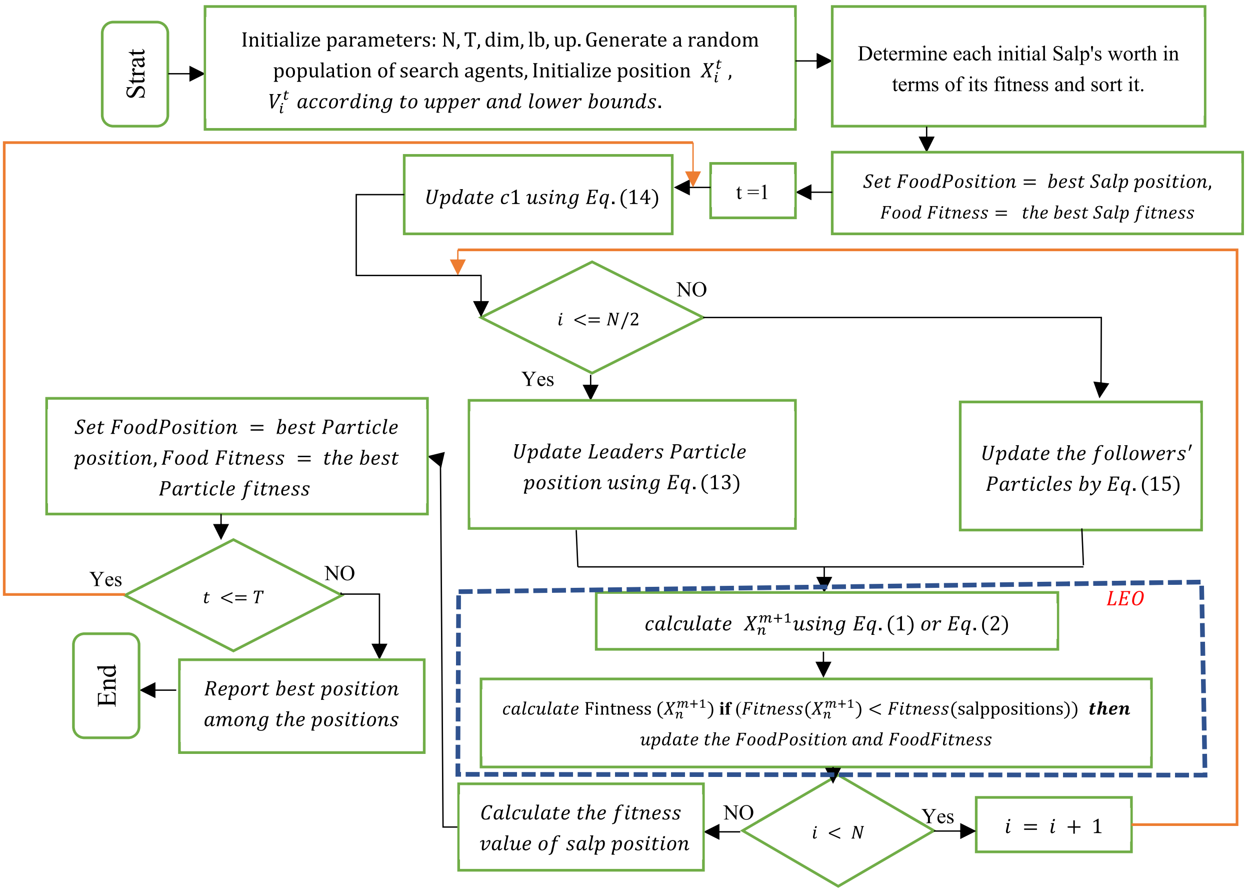

FIGURE 1. SSALEO flowchart.

Academics have used many different benchmark functions to evaluate algorithms. With the CEC2017 [86] benchmark functions with dimensions of 50 and 100 benchmark functions, and the LSGO issues with CEC2008 [87] with dimensions of 200, 500, and 1000, this section compares SSALEO performance (see Tables 3 and 4 for further information). The utilization of such sets aims to determine how robust SSALEOs are when resolving a diverse selection of benchmark functions. Comparisons are made between the performance of SSALEO and that of several innovative swarm intelligence algorithms, such as BAT, PSO, CSO, SCA [88], WOA [89], MFO [90], HHO [91], and SSA. The performance of SSALEO was then compared to the performance of other advanced algorithms such as RW-GWO [92], HIWOA [93], LJA [94], CPSO [95], WFOA [96], LNMRA [97], CLPSO [98], HIWOA [93], LJA [94], PPSO [99], PPSO_W [99], and HPSO_TVAC [100] as well as the convergence behavior of each. The Wilcoxon rank-sum and non-parametric Friedman tests assess the algorithms' overall efficacy.

A. EXPERIMENT SETUP

Due to the stochastic nature of swarm intelligence algorithms, they must be reviewed and compared objectively, with all experiments being carried out under comparable circumstances. Consequently, every algorithm was written in Python 3 and evaluated on a computer equipped with an Intel Core i3-7100 CPU operating at 3.90 GHz and 4 gigabytes of random access memory. The evaluations were conducted with CEC 2017 benchmark functions with 50 and 100 dimensions and CEC2018LSGO benchmark functions with 200, 500, and 1000 dimensions. These evaluations covered unimodal, multimodal, hybrid, and composite tasks. To guarantee consistency and fairness across all tests, we perform every experiment thirty times, with each function being treated independently. To produce metrics supported by sound statistics for each function, the experiment is carried out thirty times, and the population size (N) and the maximum number of iterations (Max iter) are each set to thirty and two thousand five hundred, respectively.

B. PERFORMANCE EVALUATION AND PARAMETER SETUP

This research uses the average, the median, and the standard deviation as three different descriptive metrics to evaluate the performance of the suggested algorithm. To calculate an average of the optimization results, the following formula

TABLE 3. CEC 2017 benchmark function.

Type	Fun	Function name	Fmin
U	$F1$	Shifted and Rotated Bent Cigar Function	100
U	$F2$	Shifted and Rotated Zakharov	300
M	$F3$	Shifted and Rotated Rosenbrock's	400
M	$F4$	Shifted and Rotated Rastrigin's	500
M	$F5$	Shifted and Rotated Expanded Scaffer's F6	600
M	$F6$	Shifted and Rotated Lunacek Bi_Rastrigin	700
M	$F7$	Shifted and Rotated Non Continuous Rastrigin's	800
M	$F8$	Shifted and Rotated Levy	900
M	$F9$	Shifted and Rotated Schwefel's	1000
H	$F10$	Hybrid Function 1 (N = 3)	1100
H	$F11$	Hybrid Function 2 (N = 3)	1200
H	$F12$	Hybrid Function 3 (N = 3)	1300
H	$F13$	Hybrid Function 4 (N = 4)	1400
H	$F14$	Hybrid Function 5 (N = 4)	1500
H	$F15$	Hybrid Function 6 (N = 4)	1600
H	$F16$	Hybrid Function 6 (N = 5)	1700
H	$F17$	Hybrid Function 6 (N = 5)	1800
H	$F18$	Hybrid Function 6 (N = 5)	1900
H	$F19$	Hybrid Function 6 (N = 6)	2000
C	$F20$	Composition Function 1 (N = 3)	2100
C	$F21$	Composition Function 2 (N = 3)	2200
C	$F22$	Composition Function 3 (N = 4)	2300
C	$F23$	Composition Function 4 (N = 4)	2400
C	$F24$	Composition Function 5 (N = 5)	2500
C	$F25$	Composition Function 6 (N = 5)	2600
C	$F26$	Composition Function 7 (N = 6)	2700
C	$F27$	Composition Function 8 (N = 6)	2800
C	$F28$	Composition Function 9 (N = 3)	2900
C	$F29$	Composition Function 10(N = 3)	3000
		Range $[-100, 100]$ D	

TABLE 4. CEC 2008lsgo benchmark function.

Fun	Function name	Characteristics
$F1$	Shifted Sphere	Separable
$F2$	Schwefel Problem	Non − Separable
$F3$	Shifted Rosenbrock	Non − Separable
$F4$	Shifted Rastrigin	Separable
$F5$	Shifted Griewank	Non Separable/ Separable
$F6$	Shifted Ackley	Separable
$F7$	Fast Fractal	Non − Separable

may be utilized:

$$\text{Mean} = \frac{1}{n}\sum_{i=1}^{n} S_i \tag{19}$$

For each optimization, S_i represents the final result. It is possible to calculate the standard deviation of the optimization

TABLE 5. Hyperparameter settings.

Algorithm	Parameter	Range/value
SSALEO	Coefficient (c1)	[2/e, 2]
MFO	T	[-1, 1]
	b	1
SSA	Coefficient (c1)	[2/e, 2]
BAT	Loudness, Pulse rate	0.5, 0.5
	Frequency minimum	0
	Frequency maximum	2
SCA	Convergence constant(r1)	[0, 2]
HHO	beta	1.5
PSO	Inertia weight (wmin, wmax)	0.04, 0.09
	Cognitive coefficient (c1, c2)	2
WOA	Convergence constant (a)	[0, 2]
	Coefficient (b)	1
CSO	phi	0
LNMRA	Step	0.001
	Beta	1
	Sigma_v	1
RW_GWO	Convergence constant (a)	[0, 2]
HIWOA	feedback_max	10
	Convergence constant (a)	[0, 2]
	Coefficient (b)	1
	Coefficient (p)	0.5
LJA	Sigma_v	1
	Multiplier	0.001
	beta	1
CPSO	Inertia weight (wmin, wmax)	0.2, 1.2
	Cognitive coefficient (c1, c2)	1.2
WFOA	B	1
PPSO	v_max	0.5
PSO_W	v_max	0.5
HPSO_TVA	Coefficient (ci)	0.5
C	Coefficient (cf)	0.0
CLPSO	(c_local, wmin, wmax)	1.49445, 0.4,0.9
ESSA	Coefficient (c1)	[2/e, 2]
	Coefficient (r1)	50 * random
ISSA	Coefficient (c1)	[2/e, 2]
	Cmax, Cmin	1 , 0.00003
IWSSA	Coefficient (c1)	[2/e, 2]
	Cmax, Cmin	1 , 0.00003
STS-SSA	Coefficient (c1)	[2/e, 2]
	Coefficient (r)	Random
HSSASCA	Coefficient (c1)	[2/e, 2]
	Coefficient (r)	2 * π *Random
ISSA_OBL	Coefficient (c1)	[2/e, 2]
	Max_local_iteration	10
TVSSA	Coefficient (c1)	[2/e, 2]
SHADE	Miu_f	0.5
	Miu_cr	0.5
DESAP_abs	Miu_f	0.5
	Miu_cr	0.5
large-scale	W	[0.1,0.9]
DSCA	σ	0.1
large-scale	α	Random[0,1]
QIWOA	β	Random[0,1]
Large-scale LM-CMA	sigma	0.2

TABLE 6. Results from 2500 iterations of the SSALEO vs standard techniques for unimodal functions.

Fun	D	Criteria	*CSO*	*SSA*	*PSO*	*WOA*	*BAT*	*HHO*	*SCA*	*MFO*	*SSALEO*
F1	50	Avg	1.802E+12	3.972E+09	1.523E+11	1.162E+10	1.358E+12	7.099E+11	5.275E+11	4.880E+11	**3.308E+03**
		Std	2.192E+11	2.537E+09	4.952E+10	4.472E+09	3.237E+10	7.922E+10	5.719E+10	1.969E+11	1.969E+11
		Med	1.779E+12	3.430E+09	1.549E+11	1.010E+10	7.765E+10	7.318E+11	5.219E+11	5.181E+11	5.181E+11
	100	Avg	4.618E+12	2.357E+11	1.125E+12	2.261E+11	3.489E+12	2.066E+12	1.871E+12	1.377E+12	**6.802E+03**
		Std	2.631E+11	4.572E+10	1.857E+11	4.430E+10	4.693E+11	1.210E+11	1.249E+11	5.616E+11	8.053E+03
		Med	4.584E+12	2.345E+11	1.154E+12	2.308E+11	3.454E+12	2.080E+12	1.855E+12	1.344E+12	1.344E+12
F2	50	Avg	7.261E+05	1.122E+05	1.630E+05	1.931E+05	4.182E+06	2.035E+05	1.457E+05	3.009E+05	**2.509E+03**
		Std	1.585E+06	2.506E+04	2.967E+04	6.001E+04	1.505E+04	2.757E+04	2.341E+04	1.047E+05	1.047E+05
		Med	3.806E+05	1.095E+05	1.616E+05	1.825E+05	8.602E+04	2.003E+05	1.420E+05	2.793E+05	2.793E+05
	100	Avg	9.529E+05	3.671E+05	4.689E+05	8.409E+05	5.765E+06	3.530E+05	4.085E+05	9.229E+05	**9.285E+04**
		Std	3.647E+05	5.821E+04	5.168E+04	1.550E+05	1.475E+07	9.945E+03	4.767E+04	1.626E+05	1.458E+04
		Med	8.541E+05	3.633E+05	4.553E+05	8.447E+05	1.174E+06	3.567E+05	4.085E+05	9.321E+05	9.321E+05
Rank	50	W/T/L	00/00/02	00/00/02	00/00/02	00/00/02	00/00/02	00/00/02	00/00/02	00/00/02	02/00/00
	100	W/T/L	00/00/02	00/00/02	00/00/02	00/00/02	00/00/02	00/00/02	00/00/02	00/00/02	02/00/00

outcomes using the formula below:

$$std = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(S_i - \text{Mean})^2 \ldots} \tag{20}$$

Intuitively, the average number can be interpreted as a reflection of the algorithm's effectiveness in optimizing its performance and its ability to avoid making computational errors. Standard deviation is a measure of dispersion, and the lower it is, the more durable and strong the algorithm will be. Table 5 contains the SSALEO and other algorithms' parameter values obtained from their corresponding research articles. These parameter selections ensure a fair comparison because they maximize the performance of each method. The findings shown in bold are the most significant.

C. PERFORMANCE EVALUATION

To evaluate the SSALEO method, its results are compared to several other cutting-edge metaheuristics techniques. The statistical features (mean, standard deviation, and median) of 30 runs are presented in Tables 6-9, with bold letters denoting the runs that produced the best results. The final rows of each of these tables provide additional information regarding the number of victories (W), ties (T), and losses (L) achieved by each algorithm. When comparing SSALEO's performance to other algorithms, various functions from the CEC 2017 benchmark with 50 and 100 dimensions were used. These functions were used to evaluate how well SSALEO explored, exploited, and escaped from local optimums. Last but not least, the overall efficacy (OE) of SSALEO is compared to that of various other methods.

1) ANALYSIS FOR EXPLOITATION AND EXPLORATION

Because they are unimodal, the F1 and F2 functions can be utilized to calculate the algorithm's exploitation ability. The proposed SSALEO technique is then empirically evaluated using these benchmark functions on dimensions 50 and 100. To demonstrate the suggested SSALEO's exploitation capabilities, the testing results for both the SSALEO and competing algorithms are presented in Table 6, which can be found here. The evaluation of the exploratory capacity of optimization algorithms is a strong suit for F3-F9 in particular. They have multiple local optima that increase in size exponentially as the dimension increases. The proposed SSALEO approach, as indicated in Table 7 , can solve these benchmark functions on dimensions 50 and 100. The results in Table 6 and 7 demonstrate that including an SSA in LEO results in a higher convergence rate for the algorithm, avoiding a standstill in local optima dynamically and greatly enhancing exploration and exploitation.

2) ANALYSIS OF ESCAPE ABILITY FROM LOCAL OPTIMA

To prevent the algorithm from becoming trapped in a local optimal solution, the hybrid and composite functions are essential in determining how much exploration and exploitation should happen hand in hand. For hybrid functions F10–F19, the proposed SSALEO outperforms competition techniques, as shown in Table 8. In addition, the SSALEO strategy outperforms the other competing approaches in the composite optimization benchmark functions F20–F29, as seen in Table 9. Combining an SSA with a LEO, as the findings in Tables 8 and 9, increase the algorithm's convergence rate and ensures that exploration and exploitation are appropriately balanced.

3) OVERALL EFFECTIVENESS (OE)

We evaluate the overall effectiveness (OE) [101] of the SSALEO compared to that of its competitors in this section by looking at the findings of those competitors in Tables 6–9. Equation (21) shows the OE of the comparison algorithms, where N and L are the total numbers of test functions and losses for each strategy. Table 10 reveals that the SSALEO is the most effective method for all test functions involving

TABLE 7. Results from 2500 iterations of the SSALEO vs standard techniques for multimodal functions.

Fun	D	Criteria	*CSO*	*SSA*	*PSO*	*WOA*	*BAT*	*HHO*	*SCA*	*MFO*	*SSALEO*
F3	50	Avg	6.454E+04	8.077E+02	4.753E+03	1.154E+03	5.232E+04	2.114E+04	8.630E+03	5.359E+03	**5.647E+02**
		Std	1.402E+04	9.771E+01	1.296E+03	1.460E+02	4.641E+02	4.230E+03	1.756E+03	3.475E+03	3.475E+03
		Med	6.602E+04	7.918E+02	4.930E+03	1.123E+03	1.190E+03	2.077E+04	8.607E+03	4.342E+03	4.342E+03
	100	Avg	1.942E+05	3.583E+03	2.536E+04	4.469E+03	1.381E+05	6.554E+04	3.857E+04	3.535E+04	**7.222E+02**
		Std	3.139E+04	1.059E+03	3.986E+03	9.564E+02	3.221E+04	9.588E+03	6.727E+03	1.472E+04	4.065E+01
		Med	1.911E+05	3.450E+03	2.566E+04	4.359E+03	1.314E+05	6.742E+04	3.837E+04	3.418E+04	3.418E+04
F4	50	Avg	1.481E+03	8.506E+02	9.740E+02	9.764E+02	1.176E+03	9.511E+02	1.092E+03	9.908E+02	**8.241E+02**
		Std	6.610E+01	7.137E+01	6.156E+01	7.472E+01	4.057E+01	3.603E+01	3.351E+01	9.151E+01	9.151E+01
		Med	1.486E+03	8.457E+02	9.729E+02	9.589E+02	7.339E+02	9.484E+02	1.095E+03	9.703E+02	9.703E+02
	100	Avg	2.677E+03	1.544E+03	1.793E+03	1.732E+03	2.151E+03	1.694E+03	1.970E+03	1.873E+03	**1.294E+03**
		Std	1.150E+02	1.023E+02	7.583E+01	1.425E+02	1.643E+02	5.982E+01	6.461E+01	1.622E+02	6.085E+01
		Med	2.683E+03	1.523E+03	1.801E+03	1.732E+03	2.158E+03	1.687E+03	1.985E+03	1.870E+03	1.870E+03
F5	50	Avg	7.741E+02	6.839E+02	6.848E+02	7.185E+02	7.118E+02	7.021E+02	6.998E+02	6.894E+02	**6.667E+02**
		Std	1.194E+01	1.113E+01	8.197E+00	1.568E+01	1.041E+01	6.522E+00	5.907E+00	1.282E+01	1.282E+01
		Med	7.734E+02	6.834E+02	6.856E+02	7.190E+02	6.340E+02	7.025E+02	6.981E+02	6.871E+02	6.871E+02
	100	Avg	7.678E+02	6.909E+02	7.041E+02	7.110E+02	7.114E+02	6.936E+02	7.138E+02	7.023E+02	**6.717E+02**
		Std	8.654E+00	6.891E+00	6.701E+00	1.162E+01	1.231E+01	4.320E+00	6.984E+00	1.044E+01	5.416E+00
		Med	7.686E+02	6.913E+02	7.036E+02	7.073E+02	7.092E+02	6.939E+02	7.141E+02	7.011E+02	7.011E+02
F6	50	Avg	4.077E+03	1.483E+03	1.515E+03	1.791E+03	3.003E+03	1.750E+03	1.707E+03	2.173E+03	**1.096E+03**
		Std	2.826E+02	1.626E+02	7.016E+01	9.950E+01	1.004E+02	5.799E+01	7.866E+01	5.200E+02	5.200E+02
		Med	4.111E+03	1.435E+03	1.496E+03	1.780E+03	1.098E+03	1.773E+03	1.714E+03	2.121E+03	2.121E+03
	100	Avg	9.061E+03	3.304E+03	3.181E+03	3.568E+03	6.129E+03	3.336E+03	3.715E+03	5.403E+03	**2.091E+03**
		Std	5.728E+02	1.799E+02	1.606E+02	1.834E+02	1.153E+03	9.997E+01	1.548E+02	9.758E+02	3.068E+02
		Med	9.076E+03	3.317E+03	3.210E+03	3.572E+03	5.834E+03	3.356E+03	3.698E+03	5.675E+03	5.675E+03
F7	50	Avg	1.772E+03	1.185E+03	1.250E+03	1.253E+03	1.746E+03	1.206E+03	1.406E+03	1.403E+03	**1.149E+03**
		Std	7.072E+01	7.845E+01	5.276E+01	5.448E+01	8.739E+01	3.710E+01	3.010E+01	7.913E+01	7.913E+01
		Med	1.774E+03	1.179E+03	1.246E+03	1.255E+03	1.037E+03	1.200E+03	1.411E+03	1.373E+03	1.373E+03
	100	Avg	3.134E+03	1.967E+03	2.134E+03	2.094E+03	3.010E+03	2.017E+03	2.339E+03	2.570E+03	**1.735E+03**
		Std	1.110E+02	1.107E+02	9.283E+01	1.228E+02	1.707E+02	6.178E+01	7.515E+01	1.904E+02	8.334E+01
		Med	3.110E+03	1.989E+03	2.095E+03	2.098E+03	2.965E+03	2.015E+03	2.342E+03	2.561E+03	2.561E+03
F8	50	Avg	7.208E+04	1.486E+04	1.990E+04	2.848E+04	1.649E+04	1.488E+04	2.573E+04	1.867E+04	**7.473E+03**
		Std	1.002E+04	3.043E+03	3.964E+03	8.005E+03	4.102E+03	1.171E+03	4.033E+03	5.450E+03	5.450E+03
		Med	7.268E+04	1.459E+04	2.026E+04	2.641E+04	1.189E+04	1.463E+04	2.542E+04	1.782E+04	1.782E+04
	100	Avg	1.631E+05	3.950E+04	6.715E+04	5.839E+04	3.398E+04	3.209E+04	8.236E+04	5.090E+04	**2.022E+04**
		Std	1.543E+04	5.132E+03	9.643E+03	1.183E+04	6.300E+03	2.894E+03	8.021E+03	7.859E+03	2.604E+03
		Med	1.645E+05	3.988E+04	6.611E+04	5.779E+04	3.335E+04	3.143E+04	8.189E+04	5.055E+04	5.055E+04
F9	50	Avg	1.570E+04	8.175E+03	1.457E+04	1.194E+04	1.147E+04	1.140E+04	1.502E+04	8.900E+03	**7.931E+03**
		Std	5.936E+02	8.483E+02	7.082E+02	1.400E+03	2.588E+03	1.303E+03	3.663E+02	1.228E+03	1.228E+03
		Med	1.578E+04	8.008E+03	1.469E+04	1.196E+04	7.076E+03	1.111E+04	1.506E+04	8.895E+03	8.895E+03
	100	Avg	3.342E+04	2.000E+04	3.214E+04	2.595E+04	2.674E+04	2.582E+04	3.221E+04	1.880E+04	**1.578E+04**
		Std	6.980E+02	1.647E+03	5.891E+02	2.259E+03	2.088E+03	1.789E+03	5.919E+02	2.304E+03	1.646E+03
		Med	3.346E+04	2.008E+04	3.213E+04	2.589E+04	2.718E+04	2.579E+04	3.238E+04	1.934E+04	1.934E+04
Rank	50	W/T/L	00/00/07	00/00/07	00/00/07	00/00/07	00/00/07	00/00/07	00/00/07	00/00/07	07/00/00
	100	W/T/L	00/00/07	01/00/06	00/00/07	00/00/07	00/00/07	00/00/07	00/00/07	00/00/07	06/00/01

50 and 100 dimensions, as it achieves an OE of 76.94 percent.

$$OE = \left(\frac{N-L}{N}\right) \times 100 \tag{21}$$

D. CONVERGENCE ANALYSIS

Fig 2 illustrates the convergence behavior of the existing techniques and the suggested SSALEO algorithm when applied to CEC2017 with 100 dimensions. According to the findings, the SSALEO performs better than other conventional algorithms because it aggressively explores the search space during the early iterations before gradually convergent to the global optimum over the future iterations. In addition, as can be observed in Fig 2, the convergent speed of the SSALEO technique is quite close to that of the other strategies. Furthermore, the convergence curves illustrate SSALEO's ability to balance exploration and exploitation in hybrid and composite functions while avoiding reaching

TABLE 8. Results from 2500 iterations of the SSALEO vs standard techniques for hybrid functions.

Fun	D	Criteria	*CSO*	*SSA*	*PSO*	*WOA*	*BAT*	*HHO*	*SCA*	*MFO*	*SSALEO*
F10	50	Avg	4.977E+04	2.524E+03	4.505E+03	2.783E+03	9.222E+04	1.637E+04	8.829E+03	2.228E+04	**1.372E+03**
		Std	1.426E+04	5.261E+02	1.191E+03	6.232E+02	1.923E+03	2.679E+03	1.956E+03	1.646E+04	1.646E+04
		Med	4.817E+04	2.402E+03	4.120E+03	2.677E+03	4.662E+03	1.719E+04	8.579E+03	1.725E+04	1.725E+04
	100	Avg	7.992E+05	6.706E+04	1.168E+05	1.521E+05	2.214E+06	2.649E+05	1.200E+05	1.727E+05	**2.784E+03**
		Std	1.864E+06	1.481E+04	1.748E+04	7.024E+04	6.540E+06	8.548E+04	2.047E+04	9.759E+04	2.818E+02
		Med	4.337E+05	6.856E+04	1.196E+05	1.350E+05	5.923E+05	2.352E+05	1.244E+05	1.781E+05	1.781E+05
F11	50	Avg	8.715E+11	2.843E+09	3.763E+10	6.611E+09	7.070E+11	3.838E+11	1.224E+11	6.277E+10	**2.170E+08**
		Std	2.132E+11	2.641E+09	1.675E+10	3.649E+09	1.301E+10	1.112E+11	2.931E+10	4.455E+10	4.455E+10
		Med	8.373E+11	1.834E+09	3.448E+10	5.755E+09	5.699E+09	3.967E+11	1.243E+11	4.834E+10	4.834E+10
	100	Avg	2.331E+12	2.004E+10	2.434E+11	3.298E+10	2.046E+12	1.161E+12	6.433E+11	4.275E+11	**1.257E+09**
		Std	3.823E+11	1.024E+10	6.763E+10	1.166E+10	3.434E+11	1.960E+11	1.067E+11	2.547E+11	5.263E+08
		Med	2.388E+12	1.745E+10	2.296E+11	3.018E+10	2.077E+12	1.155E+12	6.304E+11	4.361E+11	4.361E+11
F12	50	Avg	5.714E+11	**1.136E+05**	5.642E+09	1.795E+08	4.989E+11	1.700E+11	4.212E+10	2.332E+10	1.166E+05
		Std	2.047E+11	6.688E+04	3.459E+09	2.499E+08	1.131E+09	1.173E+11	1.563E+10	2.875E+10	2.875E+10
		Med	5.895E+11	9.423E+04	4.518E+09	1.125E+08	1.139E+09	1.409E+11	3.750E+10	7.490E+09	7.490E+09
	100	Avg	7.046E+11	1.524E+05	4.485E+10	5.507E+08	5.809E+11	2.978E+11	1.281E+11	9.287E+10	**6.953E+04**
		Std	1.292E+11	1.410E+05	1.536E+10	2.991E+08	1.360E+11	4.924E+10	2.388E+10	6.045E+10	2.714E+04
		Med	7.308E+11	9.138E+04	4.148E+10	4.556E+08	5.975E+11	2.820E+11	1.311E+11	8.116E+10	8.116E+10
F13	50	Avg	1.342E+08	7.774E+05	1.179E+06	2.292E+06	1.390E+08	3.041E+07	4.860E+06	2.360E+06	**1.173E+05**
		Std	9.520E+07	7.672E+05	1.201E+06	1.707E+06	8.484E+05	3.166E+07	3.373E+06	4.014E+06	4.014E+06
		Med	1.040E+08	5.727E+05	6.766E+05	2.127E+06	6.438E+05	1.999E+07	3.737E+06	1.033E+06	1.033E+06
	100	Avg	2.689E+08	1.015E+07	1.276E+07	1.029E+07	2.436E+08	3.207E+07	3.700E+07	2.777E+07	**5.746E+05**
		Std	1.285E+08	7.267E+06	8.654E+06	4.329E+06	1.670E+08	1.779E+07	1.534E+07	3.425E+07	2.698E+05
		Med	2.455E+08	9.230E+06	9.528E+06	1.029E+07	2.103E+08	2.967E+07	3.385E+07	1.473E+07	1.473E+07
F14	50	Avg	1.547E+11	6.185E+04	1.030E+08	2.669E+07	1.083E+11	1.938E+10	6.003E+09	1.531E+09	**4.542E+04**
		Std	6.306E+10	3.138E+04	9.247E+07	4.638E+07	1.330E+09	1.468E+10	2.984E+09	2.910E+09	2.910E+09
		Med	1.464E+11	5.368E+04	6.167E+07	6.974E+06	5.678E+06	1.708E+10	5.575E+09	3.652E+05	3.652E+05
	100	Avg	3.141E+11	8.790E+04	2.040E+09	9.488E+07	2.981E+11	1.247E+11	4.045E+10	2.984E+10	**5.522E+04**
		Std	7.472E+10	5.068E+04	1.144E+09	1.103E+08	7.629E+10	3.283E+10	1.013E+10	2.845E+10	2.464E+04
		Med	3.150E+11	7.396E+04	1.743E+09	5.970E+07	2.976E+11	1.273E+11	4.085E+10	2.127E+10	2.127E+10
F15	50	Avg	1.128E+04	4.004E+03	4.502E+03	5.516E+03	1.005E+04	7.402E+03	5.859E+03	4.471E+03	**3.962E+03**
		Std	1.748E+03	5.624E+02	6.527E+02	8.221E+02	5.171E+02	1.753E+03	4.389E+02	5.127E+02	5.127E+02
		Med	1.087E+04	3.842E+03	4.574E+03	5.421E+03	3.145E+03	7.033E+03	5.916E+03	4.468E+03	4.468E+03
	100	Avg	3.115E+04	8.345E+03	1.185E+04	1.382E+04	2.548E+04	1.856E+04	1.377E+04	8.614E+03	**6.961E+03**
		Std	5.042E+03	9.769E+02	1.048E+03	1.868E+03	4.655E+03	3.387E+03	8.264E+02	9.457E+02	9.044E+02
		Med	2.938E+04	8.336E+03	1.164E+04	1.338E+04	2.582E+04	1.778E+04	1.365E+04	8.634E+03	8.634E+03
F16	50	Avg	1.184E+05	3.658E+03	3.427E+03	4.193E+03	1.482E+05	5.216E+03	4.710E+03	4.535E+03	**3.402E+03**
		Std	1.634E+05	3.799E+02	3.704E+02	4.726E+02	2.495E+02	1.085E+03	2.911E+02	1.526E+03	1.526E+03
		Med	7.325E+04	3.568E+03	3.375E+03	4.113E+03	2.855E+03	4.952E+03	4.716E+03	4.182E+03	4.182E+03
	100	Avg	3.547E+07	6.352E+03	7.671E+03	9.070E+03	2.913E+07	1.246E+06	3.178E+04	1.396E+04	**5.870E+03**
		Std	3.877E+07	7.095E+02	7.814E+02	1.455E+03	2.968E+07	1.707E+06	3.442E+04	8.849E+03	7.443E+02
		Med	2.106E+07	6.446E+03	7.626E+03	8.740E+03	1.565E+07	7.740E+05	1.758E+04	1.028E+04	1.028E+04
F17	50	Avg	3.569E+08	5.877E+06	7.889E+06	1.633E+07	4.527E+08	7.121E+07	2.754E+07	1.294E+07	**9.575E+05**
		Std	2.342E+08	4.167E+06	5.373E+06	1.274E+07	1.130E+07	4.117E+07	1.397E+07	1.501E+07	1.501E+07
		Med	3.408E+08	4.547E+06	6.203E+06	1.167E+07	3.500E+06	5.392E+07	2.351E+07	9.159E+06	9.159E+06
	100	Avg	6.341E+08	9.402E+06	1.277E+07	6.782E+06	5.800E+08	4.460E+07	6.887E+07	1.558E+07	**9.973E+05**
		Std	2.607E+08	6.544E+06	5.676E+06	3.344E+06	3.754E+08	2.905E+07	2.736E+07	2.202E+07	3.810E+05
		Med	5.907E+08	6.943E+06	1.191E+07	5.771E+06	4.783E+08	3.940E+07	6.400E+07	8.656E+06	8.656E+06
F18	50	Avg	7.482E+10	1.854E+07	1.567E+08	2.907E+07	**1.795E+04**	7.659E+09	3.744E+09	8.190E+08	5.178E+06
		Std	2.695E+10	1.984E+07	2.194E+08	6.049E+07	8.186E+07	8.203E+09	1.666E+09	2.304E+09	2.304E+09
		Med	7.234E+10	8.922E+06	6.868E+07	1.058E+07	5.009E+06	6.083E+09	3.407E+09	3.988E+07	3.988E+07
	100	Avg	3.435E+11	9.470E+07	6.097E+09	1.340E+08	2.906E+11	1.260E+11	3.739E+10	2.363E+10	**3.512E+07**
		Std	9.166E+10	9.252E+07	2.139E+09	9.164E+07	8.846E+10	3.809E+10	1.217E+10	2.550E+10	2.636E+07
		Med	3.490E+11	5.832E+07	6.007E+09	1.107E+08	3.066E+11	1.233E+11	3.662E+10	1.716E+10	1.716E+10
F19	50	Avg	4.480E+03	3.212E+03	3.812E+03	3.765E+03	4.237E+03	3.522E+03	4.007E+03	3.843E+03	**3.036E+03**
		Std	2.392E+02	2.869E+02	3.241E+02	3.512E+02	4.889E+02	2.806E+02	1.611E+02	2.922E+02	2.922E+02

TABLE 8. *(Continued.)* **Results from 2500 iterations of the SSALEO vs standard techniques for hybrid functions.**

		Med	4.499E+03	3.235E+03	3.884E+03	3.813E+03	3.144E+03	3.596E+03	4.007E+03	3.797E+03	3.797E+03
	100	Avg	8.150E+03	5.317E+03	7.362E+03	6.475E+03	6.527E+03	6.129E+03	7.480E+03	5.832E+03	**5.325E+03**
		Std	3.363E+02	4.961E+02	3.151E+02	6.275E+02	6.295E+02	4.514E+02	2.955E+02	4.978E+02	5.837E+02
		Med	8.178E+03	5.350E+03	7.378E+03	6.457E+03	6.365E+03	6.080E+03	7.518E+03	5.983E+03	5.983E+03
Rank	50	W/T/L	00/00/10	01/00/09	01/00/09	00/00/10	01/00/09	00/00/10	00/00/10	00/00/10	08/00/02
	100	W/T/L	00/00/10	00/00/10	00/00/10	00/00/10	00/00/10	00/00/10	00/00/10	00/00/10	00/00/10

a local optimum. Additionally, the graph demonstrates that all functions are stable under the SSALEO approach. In addition, the suggested method is superior to the others in that it can obtain the lowest average of the global solutions for the CEC benchmark functions in a shorter time after only a few evaluations. Finally, because of its rapid convergence, the suggested SSALEO algorithm is a candidate for application as an optimization strategy in treating problems that call for fast computation, such as online optimization.

E. STATISTICAL RESULTS TEST ANALYSIS

In the nonparametric statistical analysis, the suggested SSALEO is evaluated compared to other tried and true methods. On the other hand, because of the stochastic nature of the proposed algorithm, we had to employ a statistical test to validate the hypothesis that its results are statistically significant [102]. SSALEO's performance is compared to other methodologies using the Wilcoxon rank-sum and Friedman tests.

1) WILCOXON RANK-SUM TEST

The Wilcoxon rank-sum [103] is a nonparametric statistical test utilized to evaluate the SSALEO's performance compared to its rivals. Each sample is given a rank, and the sum of those ranks is calculated during the rank-sum experiment. There is no significant difference in the overall performance of the algorithms evaluated while utilizing the benchmark functions, according to the null hypothesis (H0), which is tested at a significance level of 0.05. The p-values for dimensions 50 and 100 are shown in Table 11, focusing on the p-values that are more than 0.05. As a consequence, the null hypothesis is refuted for most functions, and in comparison to the other algorithms, SSALEO generates statistically significant results.

2) NON-PARAMETRIC FRIEDMAN TEST

The non-parametric Friedman test [104] is utilized to evaluate various algorithms and ascertain whether the proposed SSALEO's findings are considerably distinct from its competitors. First, each tactic is assessed on its own and then graded from best to worst, with 1 and 2 indicating the greatest first and second results, respectively, and k predicting the worst possible outcomes. After that, the average rank is utilized to establish the final rank for each algorithm. The Friedman test is then carried out with the equation mentioned above (22), in which k represents the number of swarm intelligence algorithms, Rj represents the average rank of algorithm j, and n represents the total number of swarm intelligence algorithms. Table 12 displays the results of the Friedman test for various dimensions of 50 and 100. These findings demonstrate that SSALEO beats its competitors and ranks top among other algorithms.

$$F_f = \frac{12n}{k\,(k+1)}\left[\sum\nolimits_j R_j^2 - \frac{k\,(k+1)^2}{4}\right]\ldots \tag{22}$$

F. RUN-TIME ANALYSIS

Although the computational complexity of the proposed algorithm has already been discussed in the previous sections, this section will explain how long it takes to complete the CEC2017 benchmark routines. The conclusions of SSALEO were evaluated and contrasted with those of its rival organizations. As part of the labor-intensive computing method, competitors must complete each benchmark thirty times and record their outcomes in Table 2. Regarding performance and timing, SSALEO can outperform and surpass SCA, MFO, SSA, BAT, CSO, WOA, and PSO. In addition, when compared to other algorithms, SSALEO has a higher overall efficiency than the others.

G. QUALITATIVE METRICS ANALYSIS

In this work, a Salp swarm algorithm (SSA) is combined with a local escape operator (LEO) to overcome the inherent limitations of the original SSA. SSALEO is a unique search technique that considers population diversity, the imbalance of exploitation and exploration, and the SSA algorithm's premature convergence. Implementing LEO in SSALEO eliminates the search slowness in SSA and improves the local search efficiency of swarm agents. Consequently, the primary focus of this section will be on the impact of the new method on the SSA. We choose five of the 29 functions from CEC 2017 as examples since they are more indicative of the whole and have a stronger impact. These examples include both unimodal and multimodal functions. These specific models are designated as the $F1, F3, F5, F9,$ *and* $F21$.

To intuitively analyze the position and fitness fluctuations of SSALEO while it is foraging, the qualitative analysis that SSALEO performed in handling unimodal and multimodal functions is displayed in Figure 3. The figure consists of three significant indicators: the SSALEO's first dimension trajectory, the projected exploration and exploitation phases of the SSALEO, and the SSALEO's average global best

TABLE 9. Results from 2500 iterations of the SSALEO vs standard techniques for composite functions.

Fun	D	Criteria	*CSO*	*SSA*	*PSO*	*WOA*	*BAT*	*HHO*	*SCA*	*MFO*	*SSALEO*
F20	50	Avg	3.427E+03	2.634E+03	2.813E+03	2.959E+03	3.151E+03	3.035E+03	2.908E+03	2.788E+03	**2.638E+03**
		Std	1.339E+02	6.514E+01	4.596E+01	1.013E+02	5.125E+01	8.516E+01	4.460E+01	7.420E+01	7.420E+01
		Med	3.413E+03	2.646E+03	2.819E+03	2.953E+03	2.530E+03	3.014E+03	2.903E+03	2.781E+03	2.781E+03
	100	Avg	5.221E+03	3.489E+03	3.889E+03	4.235E+03	4.984E+03	4.576E+03	4.048E+03	3.761E+03	**3.370E+03**
		Std	2.371E+02	1.552E+02	1.080E+02	1.535E+02	2.372E+02	2.392E+02	9.891E+01	1.518E+02	2.034E+02
		Med	5.229E+03	3.477E+03	3.885E+03	4.210E+03	5.022E+03	4.536E+03	4.042E+03	3.743E+03	3.743E+03
F21	50	Avg	1.753E+04	1.046E+04	1.593E+04	1.327E+04	1.390E+04	1.340E+04	1.665E+04	1.049E+04	**9.154E+03**
		Std	6.786E+02	1.850E+03	1.745E+03	1.324E+03	2.528E+03	1.208E+03	4.319E+02	1.010E+03	1.010E+03
		Med	1.746E+04	1.031E+04	1.639E+04	1.348E+04	8.790E+03	1.333E+04	1.674E+04	1.068E+04	1.068E+04
	100	Avg	3.570E+04	2.277E+04	3.444E+04	2.934E+04	2.901E+04	2.881E+04	3.462E+04	2.074E+04	**1.993E+04**
		Std	5.994E+02	3.944E+03	9.004E+02	1.442E+03	2.140E+03	1.822E+03	4.825E+02	1.812E+03	1.418E+03
		Med	3.570E+04	2.305E+04	3.454E+04	2.931E+04	2.952E+04	2.889E+04	3.470E+04	2.074E+04	2.074E+04
F22	50	Avg	4.941E+03	**3.165E+03**	3.443E+03	3.721E+03	4.618E+03	4.234E+03	3.590E+03	3.231E+03	3.257E+03
		Std	3.582E+02	9.977E+01	8.097E+01	1.883E+02	9.548E+01	2.132E+02	7.492E+01	7.564E+01	7.564E+01
		Med	4.972E+03	3.147E+03	3.451E+03	3.751E+03	2.989E+03	4.207E+03	3.578E+03	3.219E+03	3.219E+03
	100	Avg	7.841E+03	4.052E+03	4.883E+03	5.034E+03	6.516E+03	6.155E+03	5.047E+03	**3.957E+03**	4.263E+03
		Std	7.697E+02	1.924E+02	1.603E+02	2.216E+02	3.222E+02	4.134E+02	1.198E+02	1.533E+02	2.415E+02
		Med	7.832E+03	4.022E+03	4.898E+03	5.037E+03	6.491E+03	6.006E+03	5.038E+03	3.954E+03	3.954E+03
F23	50	Avg	5.453E+03	3.294E+03	3.664E+03	3.776E+03	4.864E+03	4.496E+03	3.775E+03	**3.246E+03**	3.443E+03
		Std	5.384E+02	9.187E+01	7.984E+01	1.723E+02	1.376E+02	2.405E+02	6.064E+01	5.091E+01	5.091E+01
		Med	5.434E+03	3.270E+03	3.653E+03	3.773E+03	3.165E+03	4.486E+03	3.768E+03	3.243E+03	3.243E+03
	100	Avg	1.358E+04	4.789E+03	6.537E+03	6.260E+03	1.032E+04	9.445E+03	6.884E+03	**4.592E+03**	5.113E+03
		Std	1.151E+03	2.263E+02	4.259E+02	4.944E+02	9.426E+02	8.380E+02	2.341E+02	2.120E+02	3.304E+02
		Med	1.364E+04	4.819E+03	6.486E+03	6.232E+03	1.026E+04	9.267E+03	6.929E+03	4.543E+03	4.543E+03
F24	50	Avg	3.193E+04	3.312E+03	5.748E+03	3.487E+03	2.509E+04	1.017E+04	7.382E+03	6.354E+03	**3.066E+03**
		Std	6.108E+03	8.815E+01	6.977E+02	1.335E+02	4.286E+02	9.735E+02	7.762E+02	3.849E+03	3.849E+03
		Med	3.259E+04	3.290E+03	5.652E+03	3.462E+03	3.608E+03	1.011E+04	7.134E+03	4.698E+03	4.698E+03
	100	Avg	7.226E+04	6.001E+03	1.180E+04	5.585E+03	4.748E+04	1.972E+04	1.795E+04	1.252E+04	**3.411E+03**
		Std	1.133E+04	6.119E+02	1.334E+03	4.327E+02	9.318E+03	1.592E+03	2.090E+03	4.562E+03	6.852E+01
		Med	7.154E+04	6.000E+03	1.169E+04	5.534E+03	4.604E+04	1.981E+04	1.746E+04	1.161E+04	1.161E+04
F25	50	Avg	2.574E+04	8.186E+03	1.085E+04	1.398E+04	2.165E+04	1.492E+04	1.282E+04	9.048E+03	**6.249E+03**
		Std	2.820E+03	2.675E+03	7.221E+02	1.215E+03	7.765E+02	7.011E+02	4.847E+02	8.458E+02	8.458E+02
		Med	2.562E+04	8.475E+03	1.081E+04	1.417E+04	6.629E+03	1.479E+04	1.277E+04	8.941E+03	8.941E+03
	100	Avg	7.546E+04	2.530E+04	2.923E+04	3.408E+04	6.918E+04	4.500E+04	3.764E+04	2.067E+04	**1.961E+04**
		Std	5.348E+03	4.351E+03	2.092E+03	3.802E+03	1.012E+04	2.540E+03	2.281E+03	1.936E+03	7.429E+03
		Med	7.575E+04	2.565E+04	2.881E+04	3.385E+04	6.804E+04	4.505E+04	3.723E+04	2.098E+04	2.098E+04
F26	50	Avg	8.007E+03	3.878E+03	4.606E+03	4.286E+03	**3.200E+03**	6.351E+03	4.579E+03	3.646E+03	3.916E+03
		Std	1.113E+03	1.688E+02	1.867E+02	4.691E+02	1.018E+02	9.094E+02	1.743E+02	1.230E+02	1.230E+02
		Med	8.130E+03	3.864E+03	4.684E+03	4.151E+03	3.640E+03	6.339E+03	4.608E+03	3.658E+03	3.658E+03
	100	Avg	1.535E+04	4.561E+03	6.511E+03	5.236E+03	**3.200E+03**	1.208E+04	7.791E+03	4.160E+03	4.163E+03
		Std	1.432E+03	3.047E+02	5.675E+02	7.152E+02	8.459E-05	1.755E+03	4.410E+02	2.111E+02	3.296E+02
		Med	1.558E+04	4.497E+03	6.428E+03	5.039E+03	3.200E+03	1.225E+04	7.798E+03	4.106E+03	4.106E+03
F27	50	Avg	1.849E+04	3.831E+03	5.611E+03	4.264E+03	**3.300E+03**	9.891E+03	7.310E+03	8.507E+03	3.299E+03
		Std	2.195E+03	2.585E+02	5.386E+02	2.741E+02	5.211E+02	8.839E+02	6.693E+02	1.117E+03	1.117E+03
		Med	1.847E+04	3.751E+03	5.717E+03	4.245E+03	4.404E+03	9.849E+03	7.295E+03	8.865E+03	8.865E+03
	100	Avg	5.591E+04	7.751E+03	1.334E+04	7.187E+03	**3.300E+03**	2.330E+04	2.254E+04	2.047E+04	3.478E+03
		Std	5.302E+03	1.443E+03	1.805E+03	6.741E+02	7.431E-05	1.568E+03	1.785E+03	3.778E+03	4.507E+01
		Med	5.688E+04	7.911E+03	1.331E+04	7.025E+03	3.300E+03	2.346E+04	2.219E+04	2.041E+04	2.041E+04
F28	50	Avg	7.208E+05	6.262E+03	7.035E+03	8.313E+03	4.121E+05	2.819E+04	8.007E+03	5.752E+03	**6.257E+03**
		Std	1.062E+06	7.372E+02	8.103E+02	9.872E+02	3.547E+02	2.448E+04	8.454E+02	6.151E+02	6.151E+02
		Med	2.641E+05	6.215E+03	7.157E+03	8.264E+03	4.812E+03	1.969E+04	7.943E+03	5.622E+03	5.622E+03
	100	Avg	3.164E+06	1.260E+04	1.537E+04	1.650E+04	3.323E+06	1.641E+05	2.357E+04	4.639E+04	**1.081E+04**
		Std	2.341E+06	1.667E+03	2.083E+03	2.669E+03	4.187E+06	1.030E+05	6.140E+03	1.041E+05	7.850E+02
		Med	2.692E+06	1.269E+04	1.523E+04	1.597E+04	1.585E+06	1.320E+05	2.191E+04	1.253E+04	1.253E+04
F29	50	Avg	1.044E+11	5.302E+08	8.971E+08	5.825E+08	8.860E+10	1.247E+10	6.105E+09	1.866E+09	**1.799E+08**
		Std	4.397E+10	2.303E+08	5.753E+08	3.097E+08	2.399E+08	7.920E+09	1.983E+09	3.894E+09	3.894E+09

TABLE 9. *(Continued.)* Results from 2500 iterations of the SSALEO vs standard techniques for composite functions.

		Med	9.897E+10	5.117E+08	7.147E+08	5.299E+08	3.399E+08	9.554E+09	6.234E+09	1.181E+08	1.181E+08
	100	Avg	5.255E+11	1.798E+09	2.074E+10	2.898E+09	4.476E+11	2.123E+11	7.870E+10	3.784E+10	**3.865E+08**
		Std	9.543E+10	1.024E+09	7.127E+09	1.296E+09	1.014E+11	6.481E+10	1.681E+10	2.535E+10	1.657E+08
		Med	5.224E+11	1.593E+09	1.913E+10	2.536E+09	4.301E+11	2.088E+11	7.572E+10	3.375E+10	3.375E+10
Rank	50	W/T/L	00/00/10	01/00/09	00/00/10	00/00/10	02/00/08	00/00/10	00/00/10	01/00/09	04/00/06
	100	W/T/L	00/00/10	00/00/10	00/00/10	00/00/10	02/00/08	00/00/10	00/00/10	02/00/08	06/00/04

TABLE 10. Overall effectiveness OE of the SSALEO with traditional algorithms.

Dimensions	Criteria	CSO	*SSA*	*PSO*	*WOA*	*BAT*	*HHO*	*SCA*	*MFO*	*SSALEO*
50	*W/T/L*	0/0/29	2/0/27	0/0/29	0/0/29	3/0/26	0/0/29	0/0/29	1/0/28	**23/0/6**
	OE	0%	6.89%	0%	0%	10.34%	0%	0%	3.144%	**79.31%**
100	*W/T/L*	0/0/29	0/0/29	0/0/29	0/0/29	2/0/27	0/0/29	0/0/29	2/0/27	**25/0/4**
	OE	0%	0%	0%	0%	6.89%	0%	0%	6.89%	**86.20%**

fitness level. The behavior of SSALEO's position change as it pertains to the first dimension is depicted by the SSALEO trajectory. The variance trend of the SSALEO's average fitness shifts due to the iteration process, as seen by the average global best fitness. Finally, a graphical representation of the proposed SSALEO's exploration and exploitation stages is displayed during the iteration phase.

The first-dimensional trajectory of the first SSALEO can be utilized to illustrate various components of the SSALEO, demonstrating the important exploratory aspect of the SSALEO. The SSALEO particle's ability to search for the perfect solution quickly and precisely can be ensured by the fast oscillation during the prophase and the moderate oscillation that arises during the anaphase [101]. As can be observed in figure 3, the location curve for the SSALEO has a very significant amplitude in the early iteration process. This amplitude can reach up to fifty percent of the exploration space (b). If the function is smooth, the amplitude of the SSALEO particle's position will begin to drop later in the iteration time. However, if the function's amplitude changes significantly, the position amplitude will also change. This exemplifies the adaptability and resiliency of SSALEO in several different tasks. The differences can be categorized as anything from significant to negligible when seen as a whole. Early variations in the SSALEO show a high level, indicative of the system's strong search capability. Later modifications, on the other hand, are made more slowly but may still be observed. This demonstrates that SSALEO is constantly working toward the optimal solution and is the best in this regard.

The SSALEO's proposed exploration and exploitation phases are illustrated in Figure 3 (c) to understand the exploration and exploitation trend better and search for the best solution. Every graphic contains a depiction of two curved lines. The process of the algorithm that involves exploration is represented by the blue curve, while the orange curve represents the process of the algorithm that involves exploitation. As shown in Figure 3, the proposed SSALEO starts with a high exploration ratio and a low exploitation ratio; however, it quickly transitions into an exploitation technique during the majority of the iterations in the majority of the selected functions. Consequently, the SSALEO that has been developed achieves a healthy equilibrium between exploitation and exploration.

The average global fitness curve depicts the variation tendency of SSALEO's fitness during the iterative technique in fig.3(d). If you look closely at SSALEO's average fitness curve, you'll notice that it sways a lot. This is because the average fitness value decreases over time, and the frequency of the oscillation is inversely proportional to the number of times it is run. This assures SSALEO will reach a conclusion quickly and conduct an exact search in the anaphase.

H. COMPARISON OF SSALEO WITH RECENT OPTIMIZATION ALGORITHMS

The suggested SSALEO has a superior search efficiency for determining the optimum solution to the challenges than traditional algorithms such as CSO, SSA, PSO, BAT, HHO, SCA, and MFO. However, these algorithms are examples of conventional search algorithms. Therefore, in the next sections, comparisons are made between the proposed algorithm and a wide variety of modern and sophisticated algorithms.

1) COMPARISON OF SSALEO WITH SOME RECENT ALGORITHMS

A comparison is made in this subsection between the efficiency of the proposed SSALEO and HIWOA [94], LJA [95], WFOA [97], RW-GWO [93], QSSALEO, and LNMRA [98]. Additionally, SSALEO is evaluated against other optimization methods, such as CPSO [96], PPSO [100], and PPSO_W [100]. In the tests that are detailed in this subsection, the CEC2017 test functions were used. For accurate comparisons, the population size (N) and the maximum number of iterations (Max iter) have been set at 30 and 2500, respectively. After running each method 30 times, the mean

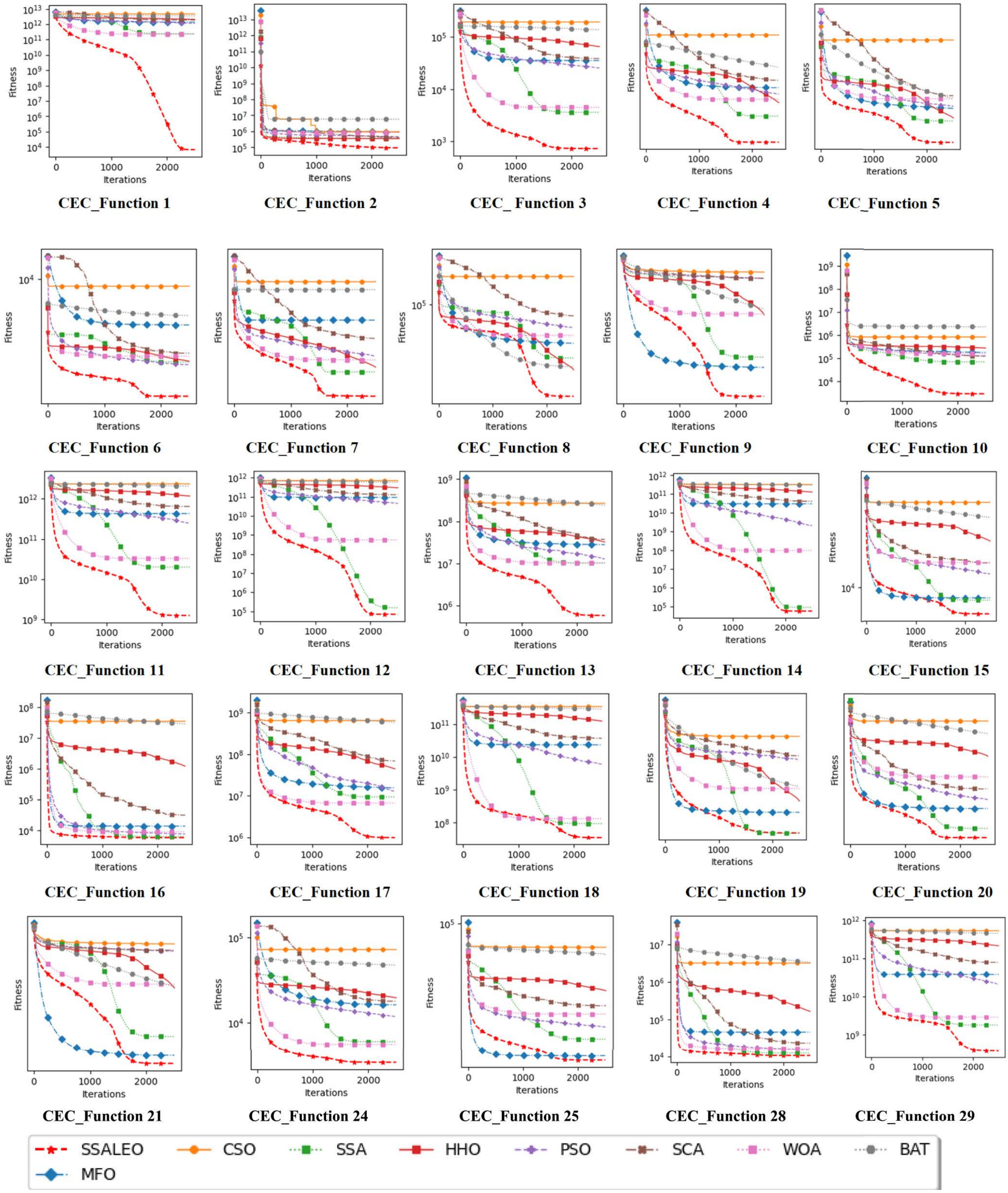

FIGURE 2. SSALEO convergence curves and other traditional algorithms during 2500 iterations.

optimization result, standard deviation, and median are compared in order to determine how well each one performs. The critically important parameters for each method are outlined in Table 5. Table 13 illustrates that the suggested SSALEO technique is on par with, if not better than, other algorithms already in use.

TABLE 11. Wilcoxon rank-sum (P-value) of the SSALEO versus other standard techniques on CEC2017 with 50,and 100 dimensions.

Fun	D	CSO	SSA	PSO	WOA	BAT	HHO	SCA	MFO
F1	50	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F2	50	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F3	50	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F4	50	< 0.05	0.362954	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F5	50	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	100	< 0.05	< 0.05	0.7616	< 0.05	< 0.05	< 0.05	< 0.05	0.7616
F6	50	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F7	50	< 0.05	0.0547840	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F8	50	< 0.05	< 0.05	< 0.05	< 0.05	0.405	< 0.05	< 0.05	0.174
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F9	50	< 0.05	0.5186817	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F10	50	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F11	50	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F12	50	< 0.05	0.7264106	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F13	50	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F14	50	< 0.05	0.0547840	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F15	50	< 0.05	0.4792036	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F16	50	< 0.05	< 0.05	0.68025925	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F17	50	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F18	50	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F19	50	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	100	< 0.05	0.9689874	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F20	50	< 0.05	0.8702911	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F21	50	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	0.0653540
F22	50	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	0.3310724
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F23	50	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F24	50	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F25	50	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	0.0915426
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	0.2938502
F26	50	< 0.05	0.5702914	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	0.469596

TABLE 11. ***(Continued.)*** **Wilcoxon rank-sum (P-value) of the SSALEO versus other standard techniques on CEC2017 with 50,and 100 dimensions.**

F27	50	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F28	< 0.05	0.5288074	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F29	50	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	0.2403457
	100	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05

TABLE 12. **Summary of Freidman test results on CEC2017's test functions with dimensions 50 and 100.**

Algorithm	Dimension	Average Rank	Overall Rank
CSO	50	8.69	9
	100	8.74	9
SSA	50	2.44	2
	100	2.58	2
PSO	50	4.36	4
	100	4.71	5
WOA	50	4.57	5
	100	4.21	3
BAT	50	7.04	8
	100	7.10	8
HHO	50	6.22	7
	100	5.89	6
SCA	50	5.79	6
	100	6.10	7
MFO	50	4.13	3
	100	4.29	4
SSALEO	50	**1.65**	**1**
	100	**1.39**	**1**

It is important to note that the p-value of the Wilcoxon rank-sum test is used in this subsection to compare the effectiveness of the suggested approach to that of other currently used strategies. This is done to determine which technique is the most efficient. Table 14 presents the p-values, with an asterisk indicating more than 0.05 and underlined. As a consequence, the null hypothesis is rejected for most functions, and in comparison to other methodologies, SSALEO generates statistically significant results. Table 15 presents the findings of the Friedman test, with the SSALEO coming in first place for most of the functions. On display in Figure 4 is the convergence behavior of the suggested SSALEO and that of alternative techniques after a total of 2500 iterations. Again, the SSALEO is superior to other contemporary algorithms and can obtain better solutions in a shorter amount of time while maintaining a healthy balance between its exploration and exploitation capabilities. The findings presented in Table 13 demonstrate, in addition, that SSALEO performs better than other modern algorithms in more than half of the functions when the efficiency of the entire system is evaluated (OE).

2) COMPARISON OF SSALEO WITH SOME SSA'S VARIANTS

On the 29 CEC 2017 test functions, the SSALEO method that has been suggested will be evaluated by comparing to seven different enhanced SSA variants, which include ESSA [105], HSSASCA [76], ISSA_OBL [79], IWSSA [79], STS-SSA [78], and TVSSA [80]. These enhanced variations of the SSA highlight their major advantages by providing novel approaches that can be used to improve the standard version of the SSA. The population size of each algorithm is set to 30, and the maximum number of iterations that can occur is 25000. After 30 iterations, the performance of each algorithm is analyzed by comparing the mean optimization result, the standard deviation, and the median of the results. Finally, the statistics are compiled and presented in Table 16.

Table 5 provides a review of the important parameters that are involved in each methodology. The outcomes of the Wilcoxon signed-rank test and the Freidman test are presented in Tables 17 and 22, respectively. The convergence graphs of the techniques being considered are shown in Figure 4. Table 16 demonstrates that when dealing with unimodal functions F1 and F2, SSALEO performs better than other alternatives. The strategy that has been proposed raises the basic SSA's exploitation potential in comparison to the potential of different SSA variants. SSALEO provides the lowest solutions, including multimodal functions such as F3, F4, F5, F6, F7, F8, and F9. The results in Table 16 demonstrate that the proposed SSALEO algorithm can solve the benchmark functions. In addition, Table 16 illustrates that implementing the suggested method increases the capabilities of the initial SSA in terms of exploration and exploitation. In addition to being competitive in hybrid functions F10–F19 and composite functions F20–F29, the search agents that SSALEO built are also competitive in these areas. According to Table 16, the new search technique has also significantly improved the ability of the original SSA to find an appropriate mix of exploration and exploitation for the algorithm to avoid becoming trapped in a local optima.

When the overall efficacy of each improved SSA is compared, Table 16 shows that SSALEO performs better than the other enhanced SSAs in more than half of the functions (OE). The SSALEO is the most efficient algorithm for all test functions, with an overall effectiveness score of 62.06 percent. Figure 4 from the CEC2017 presents a visual representation of the convergence behavior exhibited by the proposed SSALEO as well as other approaches after 2500 iterations.

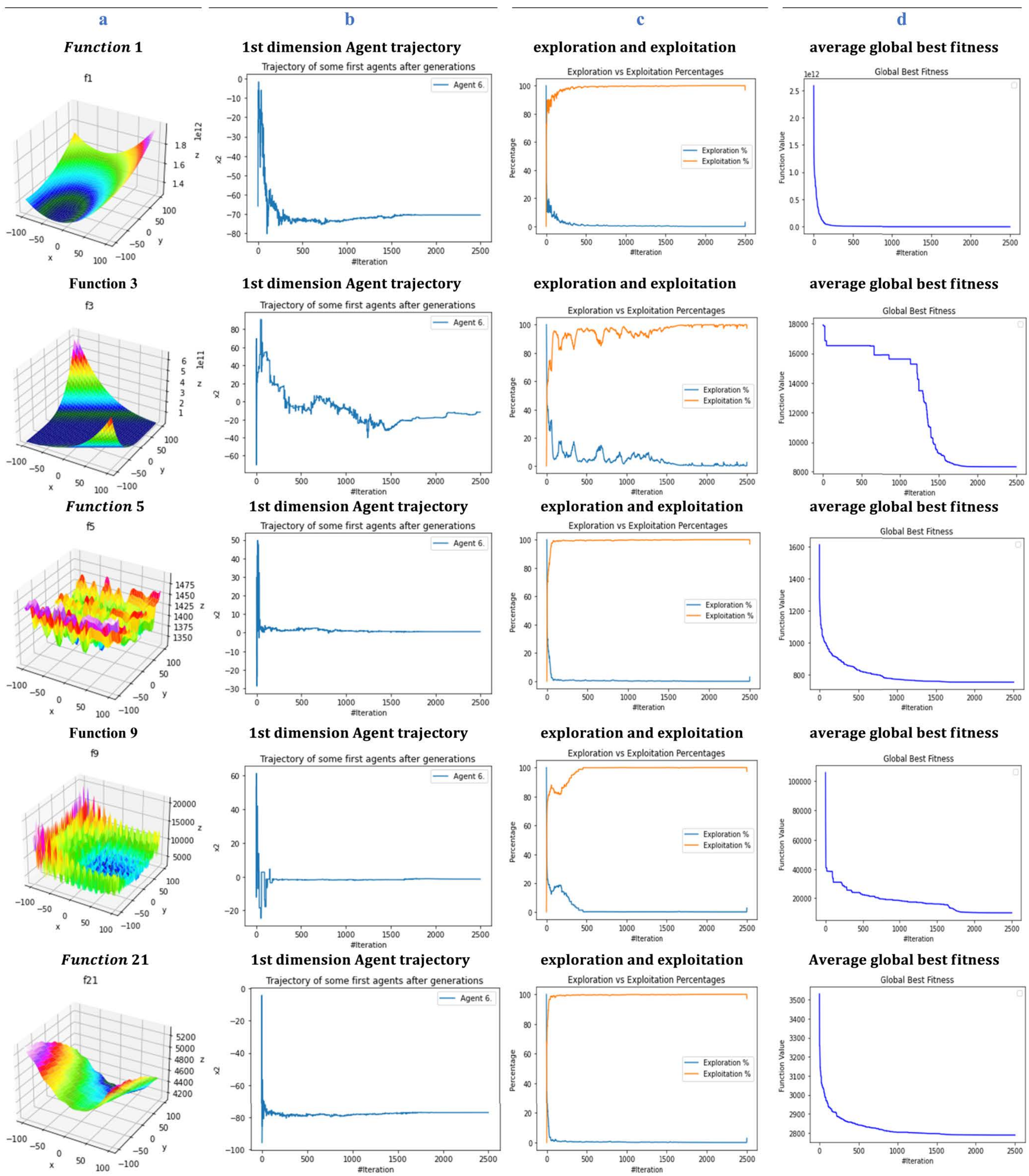

FIGURE 3. **(a) Illustration of CEC 2017 functions, (b) trajectory of SSALEO in the first dimension, (c) the exploration and exploitation phases (d) average global best fitness of SSALEO.**

According to the findings, the SSALEO is superior to the different techniques because it vigorously explores the search space in the early iterations before gradually converging to the global optimum in the later iterations.

In addition, as seen in Figure 5, the SSALEO strategies converge on the same solution nearly as soon as the other methods. In a similar vein, the convergence curves for hybrid and composite functions demonstrate that SSALEO has the

TABLE 13. Results from 2500 iterations of the SSALEO vs some recent techniques.

F	Cr.	RW-GWO	HI-WOA	LNIMRA	PPSO-W	PPSO	LJA	CPSO	WFOA	QSSALEO	SSALEO
F1	Avg	1.2826E+10	1.6391E+12	2.3821E+12	1.2364E+09	2.5678E+09	3.3161E+12	3.9739E+12	2.4077E+12	7.0852E+03	**6.8024E+03**
	Std	2.5666E+09	1.2174E+11	1.4659E+11	3.2377E+09	4.3714E+09	6.4745E+11	6.8253E+11	1.3965E+11	8.3782E+03	8.0534E+03
	Med	1.2966E+10	1.6468E+12	2.4127E+12	2.3634E+08	9.5675E+08	3.3186E+12	4.1577E+12	2.3875E+12	2.8532E+03	2.3875E+12
F2	Avg	8.5598E+05	3.5714E+05	2.9018E+05	1.8443E+05	1.6378E+05	1.3872E+06	7.7746E+05	2.6618E+12	1.7778E+05	**9.2850E+04**
	Std	3.0649E+05	5.0429E+03	2.2192E+04	9.2338E+04	2.7573E+04	4.8879E+05	1.0566E+05	9.5760E+12	1.6392E+04	1.4584E+04
	Med	8.0316E+05	3.5866E+05	2.9088E+05	1.5447E+05	1.5882E+05	1.2829E+06	7.5826E+05	1.0197E+11	1.8014E+05	1.0197E+11
F3	Avg	1.3544E+03	4.4207E+04	6.8519E+04	9.5582E+02	9.8675E+02	1.2025E+05	1.3225E+05	9.2216E+04	**6.9984E+02**	7.2221E+02
	Std	1.4357E+02	7.3501E+03	1.2322E+04	9.0829E+01	8.8774E+01	3.8870E+04	3.5729E+04	1.3375E+04	5.5763E+01	4.0646E+01
	Med	1.3207E+03	4.4123E+04	7.1146E+04	9.3974E+02	9.7118E+02	1.0967E+05	1.3868E+05	8.9029E+04	7.0314E+02	8.9029E+04
F4	Avg	1.5225E+03	1.8898E+03	1.9024E+03	1.4106E+03	1.3929E+03	2.2911E+03	2.3897E+03	2.1420E+03	1.3568E+03	**1.2940E+03**
	Std	8.8565E+01	9.1984E+01	4.7337E+01	7.8142E+01	6.8244E+01	1.3541E+02	1.3403E+02	5.4324E+01	6.1132E+01	6.0850E+01
	Med	1.5227E+03	1.8585E+03	1.9174E+03	1.3979E+03	1.3931E+03	2.2641E+03	2.3930E+03	2.1456E+03	1.3591E+03	2.1456E+03
F5	Avg	6.9407E+02	7.1189E+02	7.1032E+02	6.8254E+02	6.8396E+02	7.3670E+02	7.4808E+02	7.2645E+02	6.7542E+02	**6.7175E+02**
	Std	5.6574E+00	6.2374E+00	3.6556E+00	4.9362E+00	5.3424E+00	7.2137E+00	1.7824E+01	4.6667E+00	4.1852E+00	5.4157E+00
	Med	6.9501E+02	7.1212E+02	7.1058E+02	6.8180E+02	6.8582E+02	7.3693E+02	7.5036E+02	7.2664E+02	6.7605E+02	7.2664E+02
F6	Avg	3.0381E+03	3.8364E+03	3.7211E+03	3.2191E+03	3.2418E+03	5.3997E+03	8.6242E+03	4.1138E+03	2.6605E+03	**2.0915E+03**
	Std	2.0286E+02	7.4437E+01	1.0219E+02	1.6872E+02	1.4694E+02	1.1921E+03	9.1548E+02	7.4425E+01	2.6766E+02	3.0679E+02
	Med	3.0075E+03	3.8323E+03	3.7371E+03	3.2477E+03	3.2559E+03	5.1908E+03	8.6953E+03	4.1001E+03	2.6544E+03	4.1001E+03
F7	Avg	1.9029E+03	2.2224E+03	1.9930E+03	2.1037E+03	2.4357E+03	2.6432E+03	2.8217E+03	2.6946E+03	1.7705E+03	**1.7348E+03**
	Std	9.0182E+01	1.2155E+02	8.9275E+01	1.0482E+02	1.1724E+02	1.6530E+02	1.7689E+02	6.4812E+01	8.5746E+01	8.3337E+01
	Med	1.9011E+03	2.1773E+03	1.9671E+03	2.0927E+03	2.4222E+03	2.6452E+03	2.7956E+03	2.6842E+03	1.7898E+03	2.6842E+03
F8	Avg	5.0438E+04	6.2216E+04	6.1396E+04	2.7683E+04	2.8699E+04	1.0593E+05	1.2422E+05	7.6710E+04	2.3191E+04	**2.0220E+04**
	Std	5.1198E+03	5.6377E+03	2.4540E+03	3.2075E+03	3.3634E+03	1.8627E+04	2.1420E+04	3.5889E+03	1.2931E+03	2.6043E+03
	Med	4.9407E+04	6.2104E+04	6.1222E+04	2.7166E+04	2.8123E+04	1.0117E+05	1.2584E+05	7.6655E+04	2.3235E+04	7.6655E+04
F9	Avg	2.0045E+04	2.9687E+04	2.8554E+04	1.7785E+04	1.7797E+04	3.2342E+04	3.3735E+04	3.2200E+04	1.6476E+04	**1.5782E+04**
	Std	1.5046E+03	2.1381E+03	1.0252E+03	1.9474E+03	1.5941E+03	1.1262E+03	1.3839E+03	1.1264E+03	2.0991E+03	1.6461E+03
	Med	2.0288E+04	3.0077E+04	2.8353E+04	1.7982E+04	1.7818E+04	3.2236E+04	3.3738E+04	3.2297E+04	1.6071E+04	3.2297E+04
F10	Avg	4.8198E+04	1.8643E+05	1.1360E+05	9.6015E+03	5.6727E+03	5.2568E+05	4.5652E+05	2.8200E+05	3.2810E+03	**2.7838E+03**
	Std	1.8519E+04	2.7638E+04	2.0037E+04	1.2701E+04	4.4777E+03	2.5139E+05	1.5329E+05	9.1700E+04	4.4217E+02	2.8181E+02
	Med	4.3298E+04	1.8751E+05	1.1554E+05	3.9713E+03	4.5379E+03	5.0172E+05	4.4818E+05	2.6801E+05	3.1948E+03	2.6801E+05
F11	Avg	3.3429E+09	5.6840E+11	1.1400E+12	3.7617E+09	2.6390E+09	1.5910E+12	1.8683E+12	1.5435E+12	**1.1633E+09**	1.2575E+09
	Std	1.7755E+09	1.0856E+11	2.2303E+11	5.9728E+09	8.3166E+09	3.1554E+11	4.1655E+11	1.2718E+11	4.5152E+08	5.2634E+08
	Med	2.9391E+09	5.5767E+11	1.1992E+12	1.4228E+09	1.0344E+09	1.5962E+12	1.7599E+12	1.5446E+12	1.0805E+09	1.5446E+12
F12	Avg	8.3809E+07	1.2718E+11	2.9025E+11	5.0170E+08	2.9440E+07	4.4490E+11	4.5758E+11	3.7552E+11	7.0079E+04	**6.9531E+04**
	Std	1.3570E+08	2.7067E+10	7.7652E+10	1.4970E+09	1.6096E+08	1.1855E+11	1.5120E+11	5.3500E+10	2.2021E+04	2.7138E+04
	Med	2.1247E+07	1.2780E+11	2.8910E+11	6.6103E+04	5.0663E+04	4.5023E+11	4.6102E+11	3.7902E+11	6.4817E+04	3.7902E+11
F13	Avg	6.0618E+06	8.7333E+06	3.2071E+06	**4.4013E+05**	5.4733E+05	2.0604E+08	2.3637E+08	6.3340E+08	5.7937E+05	5.7457E+05
	Std	2.5074E+06	2.9851E+06	2.4666E+06	3.2313E+05	2.8898E+05	1.6724E+08	2.0024E+08	4.7810E+08	2.8852E+05	2.6978E+05
	Med	5.1722E+06	8.1864E+06	2.1926E+06	3.4512E+05	5.0482E+05	1.6668E+08	1.6989E+08	5.7065E+08	4.9483E+05	5.7065E+08
F14	Avg	8.5542E+07	5.0478E+10	9.8076E+10	3.9061E+04	**2.9343E+04**	2.0545E+11	2.3724E+11	3.0108E+11	5.4269E+04	5.5221E+04
	Std	2.9751E+08	1.2943E+10	4.0580E+10	1.5676E+04	1.3233E+04	7.4086E+10	1.0949E+11	3.8751E+10	1.9259E+04	2.4639E+04
	Med	6.0041E+06	5.0809E+10	9.0077E+10	3.6095E+04	2.8594E+04	1.9534E+11	2.1485E+11	2.9161E+11	5.7518E+04	2.9161E+11
F15	Avg	7.2032E+03	1.7109E+04	1.4590E+04	6.8893E+03	6.9244E+03	2.0659E+04	**5.5980E+03**	2.8509E+04	6.7610E+03	6.9612E+03
	Std	7.3170E+02	1.5789E+03	2.4214E+03	8.0828E+02	8.7708E+02	3.4590E+03	7.7167E+02	2.8217E+03	8.4687E+02	9.0439E+02
	Med	7.1983E+03	1.6775E+04	1.3927E+04	6.8286E+03	7.0973E+03	2.0139E+04	5.6380E+03	2.8580E+04	6.4861E+03	2.8580E+04
F16	Avg	7.8518E+03	3.4327E+04	2.6510E+05	6.3905E+03	6.3158E+03	9.3921E+06	**5.0204E+03**	4.5673E+07	5.8695E+03	5.8695E+03
	Std	1.8691E+03	3.7633E+04	4.9346E+05	7.1055E+02	7.6765E+02	2.1651E+07	6.0053E+02	4.0600E+07	7.4433E+02	7.4433E+02
	Med	7.1101E+03	2.0932E+04	6.2230E+04	6.2890E+03	6.3699E+03	2.2140E+06	4.9109E+03	2.4410E+07	6.0091E+03	2.4410E+07
F17	Avg	7.6161E+06	1.3649E+07	4.0715E+06	**7.7582E+05**	8.0066E+05	3.6618E+08	3.9148E+08	2.8295E+08	1.0434E+06	9.9734E+05
	Std	3.2855E+06	4.2834E+06	4.8480E+06	1.1239E+06	3.5574E+05	2.2811E+08	2.4555E+08	2.4888E+08	4.2304E+05	3.8100E+05
	Med	6.9028E+06	1.3124E+07	2.3515E+06	4.3015E+05	7.9885E+05	3.1671E+08	2.8720E+08	2.0392E+08	1.0368E+06	2.0392E+08
F18	Avg	4.0206E+07	4.8374E+10	8.9733E+10	1.3151E+06	**8.8499E+05**	1.8192E+11	2.3775E+11	2.9095E+11	2.5295E+07	3.5122E+07
	Std	2.9360E+07	1.4272E+10	4.1759E+10	1.7856E+06	1.3659E+06	8.4576E+10	7.4554E+10	5.3348E+10	1.6191E+07	2.6364E+07
	Med	3.4004E+07	4.5478E+10	9.8805E+10	5.5973E+05	3.5192E+05	1.7161E+11	2.2708E+11	2.7654E+11	2.2532E+07	2.7654E+11
F19	Avg	5.4831E+03	6.4290E+03	6.1897E+03	5.7024E+03	5.5662E+03	8.0982E+03	6.3286E+03	7.5695E+03	5.3247E+03	**5.3247E+03**
	Std	4.7856E+02	6.0036E+02	3.4409E+02	6.0038E+02	6.4658E+02	7.0482E+02	1.0973E+03	5.5112E+02	5.8373E+02	5.8373E+02

TABLE 13. *(Continued.)* Results from 2500 iterations of the SSALEO vs some recent techniques.

	Med	5.3538E+03	6.3082E+03	6.2708E+03	5.6488E+03	5.5009E+03	7.9573E+03	6.8623E+03	7.5359E+03	5.3097E+03	7.5359E+03
F20	Avg	3.5538E+03	4.2363E+03	4.2847E+03	3.8469E+03	3.8117E+03	4.6020E+03	4.5660E+03	5.0403E+03	3.3835E+03	**3.3703E+03**
	Std	1.6141E+02	1.3853E+02	1.9725E+02	1.6627E+02	2.2213E+02	2.1551E+02	2.0775E+02	1.8223E+02	1.7062E+02	2.0341E+02
	Med	3.5026E+03	4.2544E+03	4.2906E+03	3.7802E+03	3.7987E+03	4.6159E+03	4.5257E+03	5.0154E+03	3.4284E+03	5.0154E+03
F21	Avg	2.2548E+04	3.1416E+04	3.0977E+04	2.0633E+04	2.2094E+04	3.4261E+04	3.5623E+04	3.4334E+04	1.9998E+04	**1.9926E+04**
	Std	1.3579E+03	2.3722E+03	7.1274E+02	1.7641E+03	2.2735E+03	1.3473E+03	1.3925E+03	1.1052E+03	1.6320E+03	1.4179E+03
	Med	2.2539E+04	3.0850E+04	3.0895E+04	2.0211E+04	2.1823E+04	3.4079E+04	3.5828E+04	3.4620E+04	1.9794E+04	3.4620E+04
F22	Avg	4.3013E+03	5.0818E+03	5.4337E+03	5.2134E+03	5.1487E+03	5.4558E+03	7.2673E+03	7.4475E+03	**3.7426E+03**	4.2634E+03
	Std	1.5389E+02	3.1350E+02	2.8316E+02	4.4713E+02	5.1305E+02	2.9135E+02	6.6305E+02	2.7257E+02	2.1051E+02	2.4149E+02
	Med	4.2775E+03	5.0175E+03	5.4305E+03	5.2037E+03	5.0648E+03	5.4046E+03	7.3306E+03	7.4298E+03	3.7501E+03	7.4298E+03
F23	Avg	5.0791E+03	6.6390E+03	7.2367E+03	8.4074E+03	8.2110E+03	7.8571E+03	1.2647E+04	1.4218E+04	**4.2639E+03**	5.1134E+03
	Std	2.7372E+02	5.5394E+02	4.7922E+02	1.4618E+03	1.4229E+03	7.8506E+02	1.4041E+03	1.2733E+03	2.2007E+02	3.3041E+02
	Med	5.0622E+03	6.6090E+03	7.0590E+03	8.6242E+03	8.2001E+03	7.8607E+03	1.2586E+04	1.3850E+04	4.2191E+03	1.3850E+04
F24	Avg	3.9272E+03	1.5497E+04	2.4630E+04	3.5438E+03	3.6411E+03	4.3871E+04	6.3107E+04	2.7522E+04	3.3817E+03	**3.4113E+03**
	Std	1.3074E+02	1.3541E+03	2.3463E+03	9.0738E+01	7.4168E+01	1.4191E+04	1.6089E+04	3.0290E+03	5.8095E+01	6.8519E+01
	Med	3.9197E+03	1.5483E+04	2.4728E+04	3.5358E+03	3.6508E+03	3.7089E+04	6.0039E+04	2.7012E+04	3.3875E+03	2.7012E+04
F25	Avg	2.1940E+04	3.9307E+04	4.7986E+04	2.6744E+04	2.8645E+04	5.0103E+04	7.2470E+04	6.3601E+04	2.0203E+04	**1.9613E+04**
	Std	2.9514E+03	2.1362E+03	3.4971E+03	7.1788E+03	5.0767E+03	6.7687E+03	1.1321E+04	2.9866E+03	6.5080E+03	7.4286E+03
	Med	2.2418E+04	3.8918E+04	4.8768E+04	2.7506E+04	2.8881E+04	5.0761E+04	7.2188E+04	6.4249E+04	2.2485E+04	6.4249E+04
F26	Avg	**3.2000E+03**	7.6436E+03	6.0862E+03	4.6832E+03	4.4989E+03	1.0041E+04	1.3378E+04	1.8237E+04	3.9059E+03	4.1627E+03
	Std	4.3558E-04	7.7483E+02	7.1880E+02	6.6048E+02	5.6375E+02	1.7820E+03	1.7483E+03	1.8847E+03	2.3227E+02	3.2956E+02
	Med	3.2000E+03	7.8043E+03	6.0404E+03	4.5442E+03	4.3724E+03	9.7827E+03	1.3107E+04	1.8194E+04	3.8944E+03	1.8194E+04
F27	Avg	**3.3000E+03**	1.6101E+04	3.0763E+04	3.6791E+03	3.7102E+03	4.0964E+04	5.2879E+04	1.6101E+04	3.4652E+03	3.4780E+03
	Std	4.8835E-04	1.1571E+03	2.3650E+03	2.5100E+02	7.1728E+01	9.1270E+03	8.7330E+03	1.1571E+03	4.5841E+01	4.5068E+01
	Med	3.3000E+03	1.6147E+04	3.0692E+04	3.6120E+03	3.6990E+03	3.9560E+04	5.0614E+04	1.6147E+04	3.4550E+03	1.6147E+04
F28	Avg	**8.0853E+03**	3.3799E+04	3.9732E+04	1.0650E+04	1.0102E+04	9.2768E+05	1.0650E+04	3.3799E+04	1.0814E+04	1.0814E+04
	Std	1.1433E+03	1.6389E+04	2.7419E+04	1.3430E+03	9.3869E+02	1.5456E+06	1.3430E+03	1.6389E+04	7.8497E+02	7.8497E+02
	Med	7.8472E+03	2.9998E+04	2.8429E+04	1.0551E+04	1.0040E+04	3.7866E+05	1.0551E+04	2.9998E+04	1.0932E+04	2.9998E+04
F29	Avg	8.8386E+07	9.9196E+10	1.7515E+11	7.5903E+07	**2.4063E+07**	2.8058E+11	3.1133E+11	1.7515E+11	4.4147E+08	3.8653E+08
	Std	3.4667E+07	2.4649E+10	7.8347E+10	2.7578E+08	1.7551E+07	1.0408E+11	1.1842E+11	7.8347E+10	1.9013E+08	1.6572E+08
	Med	8.2467E+07	1.0185E+11	1.6746E+11	1.7482E+07	1.7967E+07	2.5120E+11	2.8544E+11	1.6746E+11	3.8878E+08	1.6746E+11
Rank	W/T/L	03/00/26	00/00/29	00/00/29	02/00/27	03/00/26	00/00/29	02/00/27	00/00/29	04/00/25	**15/00/14**
OE		10.34%	00.00%	00.00%	06.89%	10.34%	00.00%	06.89%	00.00%	13.79%	**51.72%**

potential to strike a balance between exploration and exploitation while avoiding becoming trapped in a local optima. Table 17 presents the p-values, emphasizing the p-values that are more than 0.05. As a direct consequence, the null hypothesis is refuted for most functions, and SSALEO generates statistically significant results compared to other methods. In conclusion, the results of the Friedman test are presented in Table 18, which shows that the SSALEO is superior to its rivals and ranks first compared to other algorithms.

I. SCALABILITY ANALYSIS OF SOLUTIONS TO HIGH-DIMENSIONAL FUNCTION OPTIMIZATION PROBLEMS

When solving optimization issues, it is not uncommon for functions of a high dimension and a huge scale. Because of this, the search space becomes more complicated, making the optimization process more difficult. Consequently, questions pertaining to various dimensions may be utilized to analyze the impact scalability has on the optimization efficiency of the suggested approach. Using SSALEO, it is possible to find solutions to problems with dimensions 200, 500, and 1000. Comparative studies may also use additional algorithms other than PSO and SSA, such as others. Table 19 shows the results of the calculations and parameter values for methods that comply with Section 6.2. According to the results presented in Table 19, SSALEO is the most efficient approach for performing the majority of functions when compared to other state-of-the-art algorithms. According to Table 23, the new integration technique boosted the effectiveness of the previous method in selecting the optimum combination of exploration and exploitation to avoid becoming entrapped within a local optima. When the overall performance of the algorithms is compared, Table 19 shows that SSALEO performs better than the other algorithms in more than half of the functions (OE). With an overall efficiency (OE) of 66.66 percent, the SSALEO methodology is the most successful strategy for all test functions with 200, 500, and 1000 dimensions.

The findings of the Friedman test are presented in Table 20, and the SSALEO ranks first for most of its functions. Table 21 contains the Wilcoxon rank-sum p-values, emphasizing the p-values that are more than 0.05 through underlining. As a consequence, the null hypothesis is refused for every function, and in comparison to other methodologies, the results that SSALEO produces are statistically significant.

TABLE 14. Wilcoxon rank-sum of the SSALEO vs.other advanced algorithms on CEC2017.

Fun	RW-GWO	HI-WOA	LNIMRA	PPSO-W	PPSO	LJA	CPSO	WFOA	QSSALEO
1	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	0.668898
2	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
3	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	0.06094
4	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
5	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
6	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
7	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
8	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
9	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	0.331072
10	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
11	< 0.05	< 0.05	< 0.05	0.100868	0.450706	< 0.05	< 0.05	< 0.05	0.488917
12	< 0.05	< 0.05	< 0.05	0.405412	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
13	< 0.05	< 0.05	< 0.05	< 0.05	0.460097	< 0.05	< 0.05	< 0.05	< 0.05
14	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
15	0.488917	< 0.05	< 0.05	0.338868	0.528807	< 0.05	< 0.05	< 0.05	< 0.05
16	< 0.05	< 0.05	< 0.05	< 0.05	0.129456	< 0.05	< 0.05	< 0.05	< 0.05
17	< 0.05	< 0.05	< 0.05	< 0.05	0.094571	< 0.05	< 0.05	< 0.05	0.441425
18	0.80952	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	0.183638
19	0.25303	< 0.05	< 0.05	< 0.05	0.168732	< 0.05	< 0.05	< 0.05	0.981389
20	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	0.635276
21	< 0.05	< 0.05	< 0.05	0.246631	< 0.05	< 0.05	< 0.05	< 0.05	0.559776
22	0.498735	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
23	0.088593	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
24	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	0.129456
25	0.726411	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	0.79749
26	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
27	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	0.308388
28	< 0.05	< 0.05	< 0.05	0.362954	< 0.05	< 0.05	0.362954	< 0.05	< 0.05
29	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	0.301061

In addition, it can be seen in Figure 6 that the SSALEO algorithm converges almost as quickly as the other algorithms do. As a consequence, the enhanced method might demonstrate excellent optimization accuracy and robustness when confronted with challenges of a significant magnitude. In addition, SSALEO has the potential to avoid dimensional catastrophe and has a high optimization efficiency even when tackling problems with high-dimensional functions.

1) COMPARISON OF SSALEO WITH SOME ADVANCED ALGORITHMS ON CEC2008LSGO

For the purposes of comparative research, this section makes use of CMA-ES [106], LM-CMA [107], SHADE [108], DESAP_ABS [109], large-scale QIWOA [55], large-scale DSCA [110], and various other cutting-edge approaches. As a result of their superior performance in the CEC'2018 Competition (especially in more complex functions), these algorithms have been designated as the new LSGO standard. In addition, the parameter values of such algorithms are completely in line with the requirements of Section 6.2. The results of the Friedman test are presented in Table 22, demonstrating that the SSALEO ranks top for the majority of functions. The results of the computations are displayed in Table 23. Table 23 contains a collection of statistics that demonstrate that, in comparison to other advanced algorithms, SSALEO is the method that performs the majority of functions most effectively. In addition, the Wilcoxon rank-sum p-values are presented in Table 24, with p-values that are greater than 0.05 being highlighted.

Consequently, the null hypothesis cannot be accepted for any functions. Compared to other research approaches, the outcomes produced by SSALEO are statistically significant. As a consequence of this, one can conclude that the method that has been proposed can keep an excellent level of optimization accuracy and robustness even when dealing with issues that are on a large scale. The experiments' findings indicate that SSALEO can avoid dimensional catastrophe and possesses a high optimization efficiency when solving problems involving functions with a high dimension.

VII. SSALEO FOR ENGINEERING DESIGN PROBLEMS

Engineering design challenges such as [111], [112], [113], [114] are frequently solved using optimization approaches.

This section first describes the benchmark set of seven well-known constrained design challenges in various engineering domains defined in the CEC 2020 conference

TABLE 15. Friedman test result of the SSALEO vs other advanced algorithms on CEC2017.

Fun	RW-GWO	HI-WOA	LNIMRA	PPSO-W	PPSO	LJA	CPSO	WFOA	QSSALEO	SSALEO
1	4.931	6.000	7.586	3.138	3.931	9.207	7.552	9.655	1.448	**1.552**
2	7.759	5.931	4.897	2.897	2.828	8.793	9.828	7.621	3.448	**1.000**
3	4.966	6.069	7.069	3.414	3.621	9.172	8.241	9.448	1.345	**1.655**
4	4.724	6.345	6.655	3.276	3.241	9.138	8.103	9.724	2.483	**1.310**
5	4.897	6.724	6.345	3.276	3.448	9.172	8.138	9.621	1.931	**1.448**
6	3.310	6.862	6.172	4.138	4.345	8.897	8.138	9.931	2.103	**1.103**
7	3.103	5.828	4.034	5.034	7.034	8.448	8.862	9.517	1.759	**1.379**
8	5.103	6.414	6.483	3.310	3.552	9.207	8.000	9.793	2.034	**1.103**
9	4.483	7.034	6.276	3.138	3.276	8.621	8.517	9.552	2.241	**1.862**
10	5.000	7.069	5.966	3.310	3.552	9.345	8.172	9.414	1.931	**1.241**
11	4.552	6.034	7.069	3.207	2.483	8.897	8.655	9.345	2.241	**2.517**
12	4.828	6.034	7.310	2.897	1.966	9.069	8.586	9.000	2.759	**2.552**
13	6.034	6.690	5.069	2.069	2.552	8.759	9.379	8.862	2.690	**1.897**
14	5.000	6.138	6.966	2.345	1.724	8.448	9.690	8.759	2.897	**2.034**
15	4.414	8.069	7.172	3.724	3.862	8.793	9.966	1.517	3.414	**2.069**
16	5.345	7.207	7.759	4.069	3.931	9.069	9.897	1.621	2.966	**3.138**
17	5.897	6.828	4.828	1.828	2.414	9.172	8.621	9.207	3.172	**3.034**
18	4.103	6.207	7.000	1.690	1.448	8.207	9.552	9.034	3.655	**2.103**
19	3.552	6.552	5.724	4.448	3.483	9.655	9.069	6.414	3.000	**3.103**
20	2.897	6.310	6.793	4.483	4.345	8.241	9.931	8.414	1.897	**1.690**
21	4.241	6.862	6.552	2.759	3.690	8.345	8.655	9.586	2.138	**2.172**
22	2.655	5.207	6.828	5.690	5.414	6.793	9.586	9.414	1.069	**2.345**
23	2.655	4.586	5.448	6.897	6.724	6.379	9.724	9.241	1.000	**1.345**
24	4.966	6.000	7.241	3.034	3.793	9.207	7.793	9.759	1.414	**1.793**
25	2.241	5.966	7.276	4.000	4.345	7.724	9.241	9.690	2.138	**2.379**
26	1.000	7.069	5.966	4.345	4.138	7.966	9.897	9.000	2.379	**2.241**
27	1.000	6.500	8.172	4.172	4.793	8.966	6.500	9.862	2.345	**2.690**
28	1.172	7.948	8.103	3.983	3.483	10.000	7.948	3.983	4.172	**2.207**
29	2.862	6.310	7.672	1.724	1.552	9.103	7.672	9.241	4.517	**3.345**
Avg.	4.058	6.441	6.567	3.527	3.620	8.717	8.756	8.490	2.434	**2.011**
Rank	5.000	6.000	7.000	3.000	4.000	9.000	10.000	8.000	2.000	**1.000**

benchmark set of real-world problems (CEC2020) [115]. Then, This section assesses the applicability of the SSALEO for dealing with engineering difficulties and challenges.

A. ENGINEERING DESIGN CHALLENGES BENCHMARK

Constraint optimization challenges naturally arise in optimal engineering design, where exact design restrictions must be accounted for in minimizing or maximizing the cost function. The SSALEO algorithm is applied to seven well-known restricted design problems in various engineering domains, benchmarked for real-world optimization by the 2020 Competitions on Evolutionary Computation CEC 2020 [115]. Table 29 provides a brief explanation of different design problems.

B. NUMERICAL PERFORMANCE EVALUATION

Because they have various natural limitations, the constrained violation handles these confined engineering design problems. In this regard, the outcomes of multiple metaheuristics in dealing with these design instances were obtained from the literature to make fair assessments. The proposed method was run 30 times in a row. With a population size of 20, the maximum number of iterations is 2500. This section presents the numerical results of SSALEO for the previously specified engineering design challenges.

1) TENSION/COMPRESSION SPRING DESIGN (CASE 1) PROBLEM

This tension/compression spring design problem aims to minimize the spring's weight f(x) while considering limitations such as minimum deflection, shear stress, surge frequency, outside diameter limits, and design factors. The mean coil diameter D (x2), the wire diameter d (x1), and the number of active coils P are the design factors (x3) (see Fig. 7d). This problem's mathematical formulation is as follows:

$$\min f(x) = (x_3 + 2)\, x_2 x_1^2$$

$$s.t.\ g_1(x) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \le 0$$

$$g_2(x) = \frac{4x_2^2 - x_1 x_2}{12566\left(x_2 x_1^3 - x_1^4\right)} + \frac{1}{5108 x_1^2} - 1 \le 0$$

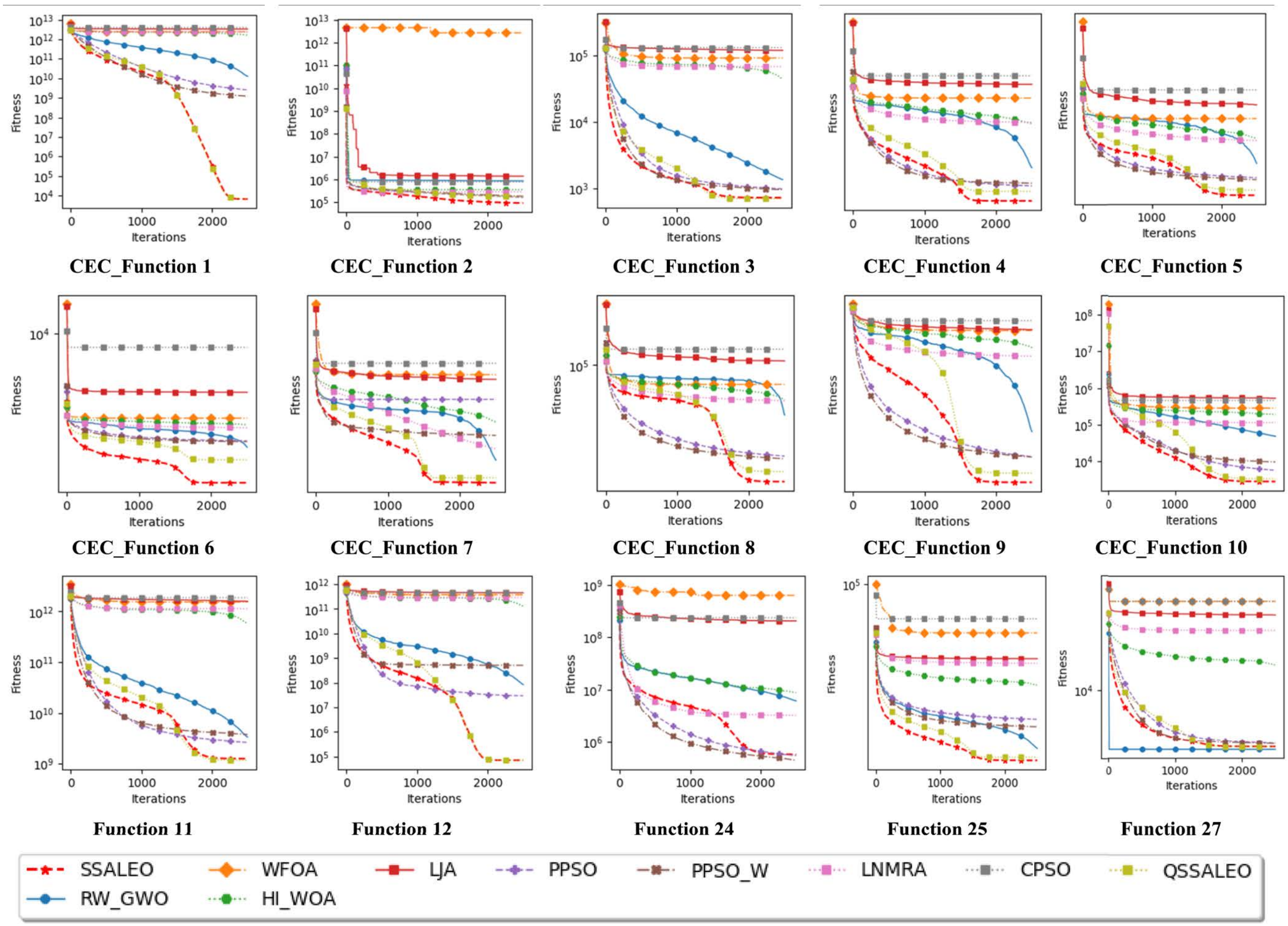

FIGURE 4. SSALEO Convergence curves and other recent algorithms during 2500 iterations.

$$g_3(x) = 1 - \frac{140.45x_1}{x_2^2 x_3} \leq 0$$

$$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \leq 0$$

where $0.05 \leq x1 \leq 2$, $0.25 \leq x2 \leq 1.3$, $2 \leq x3 \leq 15$.

A cost-optimal tension/compression spring design problem was solved using SSALEO and other state-of-the-art algorithms in Table 30. Based on the findings, the proposed algorithm is more efficient and solves hybrid decision variables faster than others.

2) PRESSURE VESSEL DESIGN

A pressure vessel design problem involves minimizing a pressure vessel's cost f(x) while accounting for materials, fabrication, and welding costs. The Ts (shell thickness), Th (head thickness), R (inner radius), and L (length) are the four design variables (x4, length of the cylindrical section of the vessel, not including the head) (see Fig. 7b). For example, the possible thickness of rolled steel plates is represented by continuous variables by Ts, Th, and R and L. Therefore, the problem of pressure vessel design can be described as follows:

$$\min f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

Subject to :

$$g_1(x) = -x_1 + 0.0193x_3 \leq 0$$

$$g_2(x) = -x_2 + 0.00954x_3 \leq 0$$

$$g_3(x) = -\Pi x_3^2 x_4 - \frac{4}{3}\Pi x_3^2 + 1296000 \leq 0$$

$$g_4(x) = x_4 - 240 \leq 0$$

where $1 \leq x1 \leq 99$, $1 \leq x2 \leq 99$, $10 \leq x3 \leq 200$, $10 \leq x4 \leq 200$.

A cost-optimal pressure vessel design problem was solved using SSALEO and other state-of-the-art algorithms in Table 30. Based on the findings, the proposed algorithm is more efficient and solves hybrid decision variables faster than others.

TABLE 16. Results from 2500 iterations of the SSALEO vs some enhanced SSA's techniques.

F	Criteria	ESSA	HSSASCA	ISSA	ISSA_OBL	IWSSA	STS – SSA	TVSSA	SSALEO
F1	Avg	8.82721E+10	2.36073E+12	1.28370E+04	1.56890E+11	2.23962E+12	2.66659E+12	3.76065E+05	**6.80241E+03**
	Std	1.92773E+10	1.54906E+11	1.69350E+04	3.98512E+10	1.03827E+11	9.44668E+10	1.99624E+06	8.05344E+03
	Med	8.57092E+10	2.35757E+12	6.43584E+03	1.51034E+11	2.22065E+12	2.68797E+12	6.62919E+03	3.69813E+03
F2	Avg	3.00589E+05	1.03767E+06	2.79746E+05	2.54334E+05	3.44867E+05	3.51127E+05	3.11099E+05	**9.28499E+04**
	Std	1.52124E+04	3.01464E+05	1.44296E+04	2.03125E+04	1.29930E+04	1.21220E+04	5.29212E+04	1.45836E+04
	Med	3.01297E+05	9.95831E+05	2.80694E+05	2.52952E+05	3.46554E+05	3.50949E+05	3.11261E+05	8.82713E+04
F3	Avg	1.93101E+03	5.64746E+04	7.46691E+02	2.36504E+03	6.59876E+04	1.03046E+05	7.66809E+02	**7.22206E+02**
	Std	2.85889E+02	1.23648E+04	4.31308E+01	5.83878E+02	5.48746E+03	1.07842E+04	5.43627E+01	4.06461E+01
	Med	1.90932E+03	5.39172E+04	7.52016E+02	2.19776E+03	6.72143E+04	1.01020E+05	7.62920E+02	7.23360E+02
F4	Avg	1.46989E+03	1.98905E+03	1.25676E+03	1.41609E+03	2.09685E+03	2.13403E+03	1.33281E+03	**1.29400E+03**
	Std	7.08687E+01	6.05012E+01	6.99477E+01	7.33370E+01	3.61070E+01	2.85552E+01	1.04372E+02	6.08496E+01
	Med	1.45915E+03	1.98856E+03	1.24920E+03	1.40437E+03	2.09807E+03	2.13752E+03	1.33526E+03	1.30292E+03
F5	Avg	6.88771E+02	7.21330E+02	6.72510E+02	6.86542E+02	7.26653E+02	7.28618E+02	6.76847E+02	**6.71749E+02**
	Std	6.71115E+00	7.38960E+00	7.81990E+00	5.08497E+00	2.32678E+00	2.47131E+00	5.60614E+00	5.41573E+00
	Med	6.88540E+02	7.20611E+02	6.72121E+02	6.87169E+02	7.26983E+02	7.28533E+02	6.76398E+02	6.71463E+02
F6	Avg	2.38470E+03	3.91454E+03	2.48889E+03	3.46387E+03	3.83356E+03	4.02601E+03	2.12673E+03	**2.09145E+03**
	Std	1.34275E+02	1.00803E+02	8.94545E+02	2.20154E+02	8.15918E+01	4.66687E+01	1.90212E+02	3.06788E+02
	Med	2.39550E+03	3.92365E+03	2.07005E+03	3.48863E+03	3.83819E+03	4.04003E+03	2.09539E+03	2.06043E+03
F7	Avg	1.75540E+03	2.43433E+03	1.86815E+03	1.90213E+03	2.53935E+03	2.61588E+03	1.89507E+03	**1.73480E+03**
	Std	6.57925E+01	8.77108E+01	3.66445E+02	8.39208E+01	4.86078E+01	3.57096E+01	3.83317E+02	8.33375E+01
	Med	1.74807E+03	2.43806E+03	1.75854E+03	1.88840E+03	2.53743E+03	2.62014E+03	1.73031E+03	1.73873E+03
F8	Avg	3.91152E+04	8.85148E+04	2.35103E+04	3.21074E+04	8.25983E+04	8.29901E+04	2.60131E+04	**2.02201E+04**
	Std	4.08878E+03	1.73041E+04	2.53894E+03	3.79934E+03	3.46289E+03	3.64714E+03	3.28399E+03	2.60427E+03
	Med	3.89212E+04	8.63448E+04	2.43702E+04	3.22773E+04	8.36752E+04	8.31473E+04	2.69356E+04	2.11122E+04
F9	Avg	1.94046E+04	3.26593E+04	**1.56341E+04**	1.87422E+04	3.23510E+04	3.24222E+04	1.55679E+04	1.57822E+04
	Std	1.00838E+03	1.40132E+03	1.49272E+03	1.43065E+03	6.31424E+02	5.48043E+02	1.38047E+03	1.64606E+03
	Med	1.91811E+04	3.25593E+04	1.52071E+04	1.91340E+04	3.25126E+04	3.24075E+04	1.56205E+04	1.55023E+04
F10	Avg	7.48795E+04	2.04425E+05	1.00170E+04	3.58459E+04	1.85018E+05	2.50445E+05	5.34988E+03	**2.78380E+03**
	Std	2.70741E+04	6.63959E+04	4.19204E+03	8.53009E+03	3.16121E+04	6.23408E+04	1.12219E+03	2.81812E+02
	Med	6.86738E+04	1.82520E+05	8.85007E+03	3.67990E+04	1.82264E+05	2.41069E+05	5.40083E+03	2.77664E+03
F11	Avg	8.62794E+09	1.19902E+12	2.87191E+09	9.23882E+09	1.05903E+12	1.76537E+12	2.43618E+09	**1.25746E+09**
	Std	3.76265E+09	2.19123E+11	1.09806E+09	3.02379E+09	1.18845E+11	1.16733E+11	1.19542E+09	5.26337E+08
	Med	7.62922E+09	1.18930E+12	2.68997E+09	9.39416E+09	1.08228E+12	1.78112E+12	2.28222E+09	1.29669E+09
F12	Avg	5.12748E+08	3.41707E+11	9.54372E+04	3.84972E+10	2.62341E+11	4.55796E+11	7.63850E+04	**6.95308E+04**
	Std	2.71665E+08	8.21440E+10	3.18959E+04	1.17466E+11	3.30113E+10	5.43670E+10	2.33294E+04	2.71385E+04
	Med	4.99874E+08	3.28496E+11	9.23537E+04	4.30435E+04	2.62064E+11	4.72494E+11	7.54998E+04	6.55235E+04
F13	Avg	1.29940E+07	5.25182E+07	2.26695E+06	5.67079E+06	5.76537E+07	1.05448E+08	1.56250E+06	**5.74569E+05**
	Std	4.38581E+06	3.41669E+07	1.24874E+06	1.48325E+06	1.44581E+07	3.84361E+07	6.77150E+05	2.69776E+05
	Med	1.24374E+07	3.86649E+07	2.04192E+06	5.32582E+06	5.69185E+07	9.77902E+07	1.44944E+06	5.46355E+05
F14	Avg	1.04384E+08	1.60067E+11	1.02836E+05	**2.06353E+04**	1.11445E+11	2.45148E+11	6.78334E+04	5.52213E+04
	Std	7.33814E+07	6.09628E+10	4.86787E+04	7.18438E+03	2.19583E+10	2.86095E+10	2.21729E+04	2.46389E+04
	Med	7.97241E+07	1.47064E+11	9.82097E+04	1.84452E+04	1.12331E+11	2.43603E+11	6.82550E+04	4.89657E+04
F15	Avg	6.98857E+03	1.62443E+04	**6.24618E+03**	9.48615E+03	1.79766E+04	2.47138E+04	6.52688E+03	6.96124E+03
	Std	8.32580E+02	2.08309E+03	8.27110E+02	1.05377E+03	1.33029E+03	1.98122E+03	8.19347E+02	9.04388E+02
	Med	7.18435E+03	1.62837E+04	6.26994E+03	9.48199E+03	1.80669E+04	2.50133E+04	6.56705E+03	7.00875E+03
F16	Avg	6.08695E+03	1.73395E+06	**5.44332E+03**	6.12407E+03	3.96830E+05	9.08787E+06	5.77479E+03	5.86955E+03
	Std	5.92377E+02	2.39937E+06	5.56422E+02	6.60291E+02	2.40581E+05	4.60695E+06	7.07373E+02	7.44334E+02
	Med	6.09321E+03	7.68133E+05	5.41932E+03	6.16633E+03	3.01393E+05	9.54454E+06	5.69064E+03	6.00912E+03
F17	Avg	1.01823E+07	5.59166E+07	2.97195E+06	3.50069E+06	1.03814E+08	2.14276E+08	2.57088E+06	**9.97336E+05**
	Std	4.36058E+06	3.48575E+07	2.04497E+06	9.37612E+05	2.54436E+07	7.32353E+07	1.27159E+06	3.81001E+05
	Med	9.13876E+06	4.68008E+07	2.57828E+06	3.29066E+06	1.02047E+08	2.06923E+08	2.54298E+06	8.67002E+05
F18	Avg	9.67738E+07	1.57312E+11	5.61331E+07	**2.23620E+06**	1.05505E+11	2.45107E+11	4.12473E+07	3.51220E+07
	Std	1.30087E+08	4.59835E+10	3.65525E+07	1.96842E+06	2.09090E+10	2.28047E+10	3.17475E+07	2.63644E+07
	Med	6.19221E+07	1.58522E+11	6.10577E+07	1.81848E+06	1.08816E+11	2.44068E+11	3.87168E+07	2.96132E+07
F19	Avg	5.50007E+03	7.71294E+03	5.05474E+03	5.20605E+03	7.57372E+03	7.48405E+03	**5.04204E+03**	5.32467E+03
	Std	5.76582E+02	6.07268E+02	5.56799E+02	5.06919E+02	2.20414E+02	2.68883E+02	5.05971E+02	5.83729E+02

TABLE 16. ***(Continued.)*** **Results from 2500 iterations of the SSALEO vs some enhanced SSA's techniques.**

	Med	5.44102E+03	7.70479E+03	4.96823E+03	5.19728E+03	7.62378E+03	7.54961E+03	4.96005E+03	5.30973E+03
F20	Avg	3.26345E+03	4.32480E+03	**3.03689E+03**	3.66274E+03	4.33380E+03	5.36970E+03	3.22928E+03	3.37034E+03
	Std	1.09178E+02	1.73175E+02	1.16322E+02	1.68808E+02	1.21865E+02	1.68386E+02	1.51248E+02	2.03406E+02
	Med	3.26593E+03	4.32273E+03	3.02347E+03	3.68153E+03	4.35301E+03	5.38988E+03	3.18417E+03	3.40538E+03
F21	Avg	2.30493E+04	3.46391E+04	2.58049E+04	2.33669E+04	3.47465E+04	3.48953E+04	2.61716E+04	**1.99264E+04**
	Std	1.60026E+03	1.07545E+03	8.13780E+03	1.69608E+03	4.95837E+02	5.30081E+02	7.88611E+03	1.41787E+03
	Med	2.33675E+04	3.45153E+04	2.01910E+04	2.32494E+04	3.47284E+04	3.50275E+04	2.17834E+04	1.99739E+04
F22	Avg	3.54290E+03	5.47862E+03	**3.51873E+03**	4.67527E+03	5.27030E+03	6.83108E+03	3.67394E+03	4.26339E+03
	Std	5.76873E+01	2.77927E+02	1.61339E+02	2.82766E+02	1.10523E+02	2.69423E+02	1.38418E+02	2.41488E+02
	Med	3.53737E+03	5.47610E+03	3.50196E+03	4.71482E+03	5.26476E+03	6.80037E+03	3.66926E+03	4.26333E+03
F23	Avg	4.31966E+03	7.67294E+03	**4.07667E+03**	5.63699E+03	7.50017E+03	1.17263E+04	4.18055E+03	5.11341E+03
	Std	1.26527E+02	6.57088E+02	1.40116E+02	4.84951E+02	3.23741E+02	6.06109E+02	1.67101E+02	3.30414E+02
	Med	4.31407E+03	7.56806E+03	4.04277E+03	5.54647E+03	7.54289E+03	1.17052E+04	4.13870E+03	5.00732E+03
F24	Avg	4.82284E+03	2.43078E+04	3.42343E+03	4.87868E+03	2.29369E+04	2.92866E+04	3.46139E+03	**3.41135E+03**
	Std	2.93674E+02	3.38099E+03	7.81586E+01	2.86470E+02	1.34680E+03	1.41599E+03	6.45600E+01	6.85187E+01
	Med	4.79581E+03	2.38344E+04	3.41277E+03	4.87789E+03	2.28720E+04	2.93665E+04	3.46981E+03	3.42099E+03
F25	Avg	**1.33257E+04**	4.23783E+04	1.35770E+04	2.46222E+04	4.74100E+04	5.69319E+04	1.41560E+04	1.96134E+04
	Std	5.43136E+03	3.19032E+03	1.35570E+03	4.70212E+03	1.78148E+03	1.75272E+03	5.00737E+03	7.42862E+03
	Med	1.10121E+04	4.20981E+04	1.35245E+04	2.57105E+04	4.73489E+04	5.70786E+04	1.53138E+04	2.17322E+04
F26	Avg	3.73747E+03	8.62299E+03	**3.64215E+03**	4.96743E+03	9.31803E+03	1.43477E+04	3.84484E+03	4.16265E+03
	Std	1.70874E+02	1.21231E+03	8.55931E+01	4.81215E+02	8.16229E+02	9.44854E+02	1.40926E+02	3.29561E+02
	Med	3.70899E+03	8.24758E+03	3.63276E+03	4.87265E+03	9.44008E+03	1.42822E+04	3.83785E+03	4.05309E+03
F27	Avg	5.57904E+03	2.74726E+04	3.48404E+03	5.99969E+03	2.86171E+04	3.66186E+04	3.52437E+03	**3.47801E+03**
	Std	6.23654E+02	3.85779E+03	4.12410E+01	5.75487E+02	1.61719E+03	1.25933E+03	4.47050E+01	4.50682E+01
	Med	5.54952E+03	2.88286E+04	3.49852E+03	5.97851E+03	2.88103E+04	3.66638E+04	3.52377E+03	3.46858E+03
F28	Avg	**7.54056E+03**	2.29376E+05	8.71114E+03	1.18645E+04	9.51392E+04	8.29937E+05	9.91754E+03	1.08140E+04
	Std	7.47246E+02	2.45131E+05	8.15288E+02	1.42249E+03	3.89640E+04	3.38618E+05	1.11377E+03	7.84970E+02
	Med	7.55484E+03	1.43638E+05	8.71957E+03	1.16039E+04	9.30063E+04	7.74752E+05	9.68836E+03	1.09317E+04
F29	Avg	7.23703E+08	2.80794E+11	7.53569E+08	1.26762E+09	1.93821E+11	3.95428E+11	8.28738E+08	**3.86526E+08**
	Std	2.83092E+08	7.01238E+10	2.98539E+08	5.20373E+08	4.06203E+10	3.63641E+10	4.45644E+08	1.65725E+08
	Med	8.53205E+08	2.85136E+11	8.01119E+08	1.19968E+09	1.92210E+11	4.02363E+11	6.69723E+08	3.48419E+08
Rank *W/T/L*		02/00/27	00/00/29	06/00/23	02/00/27	00/00/29	00/00/29	01/00/28	**18/0/11**
OE		06.89%	0.00%	20.68%	06.89%	0.00%	0.00%	03.44%	**62.06%**

3) THREE-BAR TRUSS DESIGN PROBLEM

It is well known that the three-bar truss design problem is used in constrained engineering applications for testing newly discovered optimization techniques. Its primary objective is to reduce the weight of a truss with three stress levels. (see Fig. 7a). The following are the formulae for this optimization problem:

$$Minimize\, f\,(x) = \left(2\sqrt{2x_1} + x_2\right) l$$

$$Subject\ to:$$

$$g_1\,(x) = \frac{\left(\sqrt{2x_1} + x_2\right)}{\left(\sqrt{2x_1^2 + 2x_1 x_2}\right)} P \leq \sigma$$

$$g_2\,(x) = \frac{x_2}{\left(\sqrt{2x_1^2 + 2x_1 x_2}\right)} P \leq \sigma$$

$$g_3\,(x) = \frac{1}{\left(x_1 + \sqrt{2x_2}\right)} P \leq \sigma$$

$$0 \leq x_1, x_2 \leq 1,$$

$$\text{where } l = 100cm,\ P = 2kN/cm^2,\ \sigma = 2kN/cm^2$$

The SSALEO algorithm outperformed most other state-of-the-art algorithms to solve the three-bar truss problem using the optimal decision variables to get optimal truss weight, as shown in Table 27.

4) WELDED BEAM DESIGN

shear stress (τ), bending stress in the beam (σ), buckling load on the bar (Pc), end deflection of the shaft (δ), and side constraints are all factors that must be considered while designing a welded beam. h (x1), l (x2), t (x3), and b (x4) are the four design variables (see Fig. 7c). This problem can be expressed as follows:

$$\min f\,(x) = 1.10471x_1^2 x_2 + 0.04811 x_3 x_4\,(14.0 + x_2)$$

$$\text{s.t. } g_1\,(x) = \tau\,(x) - \tau_{max} \leq 0$$

$$g_2\,(x) = \sigma\,(x) - \sigma_{max} \leq 0$$

$$g_3\,(x) = x_1 - x_4 \leq 0$$

$$g_4\,(x) = 0.10471x_1^2 + 0.04811 x_3 x_4\,(14.0 + x_2) - 5.0 \leq 0$$

$$g_5\,(x) = 0.125 - x_1 \leq 0$$

$$g_6\,(x) = \delta\,(x) - \delta_{max} \leq 0$$

TABLE 17. Wilcoxon rank-sum of the SSALEO vs.some improved SSA's algorithms on CEC2017.

Fun	ESSA	HSSASCA	ISSA	ISSA_OBL	IWSSA	STS − SSA	TVSSA
1	< 0.05	< 0.05	0.279779348	< 0.05	< 0.05	< 0.05	0.331072403
2	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
3	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
4	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	0.121777891
5	< 0.05	< 0.05	0.968987463	< 0.05	< 0.05	< 0.05	< 0.05
6	< 0.05	< 0.05	0.234174458	< 0.05	< 0.05	< 0.05	0.272918441
7	0.173596344	< 0.05	0.388084142	< 0.05	< 0.05	< 0.05	0.498735035
8	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
9	< 0.05	< 0.05	0.821595142	< 0.05	< 0.05	< 0.05	0.993795993
10	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
11	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
12	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	0.279779348
13	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
14	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
15	0.749876972	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
16	0.518681718	< 0.05	< 0.05	0.286756545	< 0.05	< 0.05	0.199498378
17	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
18	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	0.646404177
19	0.234174458	< 0.05	0.110934253	0.47920362	< 0.05	< 0.05	0.088592954
20	0.058827192	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
21	< 0.05	< 0.05	0.129455778	< 0.05	< 0.05	< 0.05	< 0.05
22	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
23	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
24	< 0.05	< 0.05	0.668898187	< 0.05	< 0.05	< 0.05	< 0.05
25	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
26	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
27	< 0.05	< 0.05	0.216336622	< 0.05	< 0.05	< 0.05	< 0.05
28	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
29	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05

$$g_6(x) = P - P_c(x) \le 0$$

$$\text{where } \tau(x) = \sqrt{\left(\tau'^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2\right.}$$

$$\tau' = \frac{P}{2^{0.5}x_1x_2}$$

$$\tau'' = \frac{MR}{J}$$

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

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

$$J = 2\left\{2^{0.5}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)\left(\frac{x_1 + x_3}{2}\right)\right]\right\}$$

$$\sigma(x) = \frac{6PL}{x_4x_3^2}$$

$$\delta(x) = \frac{4PL^3}{Ex_3^3x_4}$$

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

where P = 6000lb, L = 14 in, E = 30 x 106 psi, G = 12 x 106 psi, τmax =13,600 psi, σmax = 30,000 psi, δmax = 0.25 in, $0.1 \le x1 \le 2$, $0.1 \le x2 \le 10$, $0.1 \le x3 \le 10$, $0.1 \le x4 \le 2$.

A cost-optimal welded beam structure design problem was solved using SSALEO and other state-of-the-art algorithms in Table 27. Based on the findings, the proposed algorithm is more efficient and solves hybrid decision variables faster than others.

5) MULTIPLE DISK CLUTCH BRAKE DESIGN PROBLEM

Multiple disk clutch brake problem is a Constrained mechanical design problem (see Fig. 7e); this problem's mathematical formulation is as follows:

$$Min\, f(x) = \pi\left(r_o^2 - r_i^2\right)t(Z + 1)\rho$$

Subject to:

$$g_1(x) = r_o - r_i - \Delta r \ge 0,$$

$$g_2(x) = l_{max} - (Z + 1)(t + \delta) \ge 0,$$

$$g_3(x) = p_{max} - p_{rz} \ge 0,$$

$$g_4(x) = p_{max}v_{srmax} - p_{rz}v_{sr} \ge 0,$$

$$g_5(x) = v_{srmax} - v_{sr} \ge 0,$$

TABLE 18. Friedman test result of the SSALEO vs some improved SSA's algorithms on CEC2017.

Fun	ESSA	HSSASCA	ISSA	ISSA_OBL	IWSSA	STS − SSA	TVSSA	SSALEO
1	4.03	6.76	2.14	4.97	6.24	8.00	2.03	**1.83**
2	4.14	8.00	3.24	2.38	6.07	6.48	4.69	**1.00**
3	4.14	6.14	2.00	4.86	6.86	8.00	2.34	**1.66**
4	4.52	6.03	1.79	3.93	7.21	7.76	2.66	**2.10**
5	4.38	6.34	2.00	4.28	7.10	7.55	2.59	**1.76**
6	3.45	6.62	3.22	4.86	5.97	7.74	2.14	**2.00**
7	2.69	5.72	3.17	4.28	6.59	7.86	3.38	**2.31**
8	5.00	7.21	2.00	3.86	6.83	6.97	2.90	**1.24**
9	4.55	7.07	2.17	4.00	6.86	7.07	2.03	**2.24**
10	4.93	6.72	2.93	4.07	6.66	7.62	2.07	**1.00**
11	4.34	6.79	2.59	4.66	6.21	8.00	2.17	**1.24**
12	4.90	6.86	3.07	2.10	6.00	7.97	2.59	**2.52**
13	4.97	6.55	2.62	4.03	6.72	7.69	2.24	**1.17**
14	5.00	6.90	3.52	1.10	6.24	7.86	3.14	**2.24**
15	2.86	6.17	1.86	4.97	6.83	8.00	2.38	**2.93**
16	3.48	6.83	1.93	3.52	6.24	7.93	2.66	**3.41**
17	4.97	6.14	2.90	3.28	6.97	7.90	2.69	**1.17**
18	4.14	6.93	3.66	1.14	6.14	7.93	3.14	**2.93**
19	3.59	7.28	2.69	2.86	6.93	6.79	2.83	**3.03**
20	2.93	6.59	1.34	4.76	6.41	8.00	2.55	**3.41**
21	3.38	6.28	3.62	3.62	6.38	6.97	3.90	**1.86**
22	1.79	6.76	1.52	4.97	6.21	8.00	2.72	**4.03**
23	2.69	6.59	1.45	4.83	6.41	8.00	1.86	**4.17**
24	4.45	6.69	1.86	4.55	6.41	7.90	2.45	**1.69**
25	2.14	6.10	2.14	4.52	6.90	8.00	2.55	**3.66**
26	1.93	6.41	1.55	4.90	6.59	8.00	2.83	**3.79**
27	4.28	6.55	1.83	4.72	6.45	8.00	2.55	**1.62**
28	1.21	6.69	2.07	4.62	6.38	7.93	3.14	**3.97**
29	3.12	6.97	2.95	4.38	6.10	7.93	3.03	**1.52**
Avg.	3.72	6.64	2.41	3.97	6.51	7.72	2.70	**2.33**
Rank	4.00	7.00	2.00	5.00	6.00	8.00	3.00	**1.00**

$$g_6(x) = T_{max} - T \geq 0,$$
$$g_7(x) = M_h - sM_s \geq 0,$$
$$g_8(x) = T \geq 0,$$

where

$$M_h = \frac{2}{3}\mu FZ\frac{r_o^3 - r_i^3}{r_o^2 - r_i^2},$$
$$p_{rz} = \frac{F}{\pi\left(r_o^2 - r_i^2\right)},$$
$$v_{sr} = \frac{2\pi n\left(r_o^3 - r_i^3\right)}{90\left(r_o^2 - r_i^2\right)},$$
$$T = \frac{I_z\pi n}{30\left(M_h + M_f\right)}.$$

And $\Delta r = 20$ mm, $t_{max} = 3$mm, $t_{max} = 1.5$ mm, $l_{max} = 30$ mm, $Z_{max} = 10$, $v_{srmax} = 10$m/s, $\mu = 0.5$, $s = 1.5$, $M_s = 40$Nm, $M_f = 3$Nm, $n = 250$ rpm, $p_{max} = 1$MPa, $I_z = 55$Kgmm^2, $T_{max} = 15s$, $F_{max} = 1000$N, $r_{imin} = 55$mm, $r_{omax} = 110$mm.

The SSALEO algorithm outperformed most of the other state-of-the-art algorithms to solve the Multiple disk clutch brake problem using the optimal decision variables to get optimal truss weight, as shown in Table 28.

6) WEIGHT MINIMIZATION OF A SPEED REDUCE

The speed reducer design problem for minimizing the weights of the speed reducer (see Fig. 7f) is subject to constraints on bending stress of the gear teeth, surface stress, transverse deflections of the shafts, and stresses in the beams. The parameters $x1, x2, \ldots\ldots, x7$ represent the face width (b), the module of the teeth (m), number of the teeth in the pinion (z), length of the first shaft between bearings ($l1$), length of the second shaft between bearings ($l2$), and the diameter of the first shaft ($d1$) and the second shaft ($d2$), respectively.

$$\min f(\mathrm{x}) = 0.7854\mathrm{x}_1\mathrm{x}_2^2\left(3.3333\mathrm{x}_3^2 + 14.9334\mathrm{x}_3 - 43.0934\right) - 1.508\mathrm{x}_1\left(\mathrm{x}_6^2 + \mathrm{x}_7^2\right) + 7.4777\left(\mathrm{x}_6^3 + \mathrm{x}_7^3\right)$$

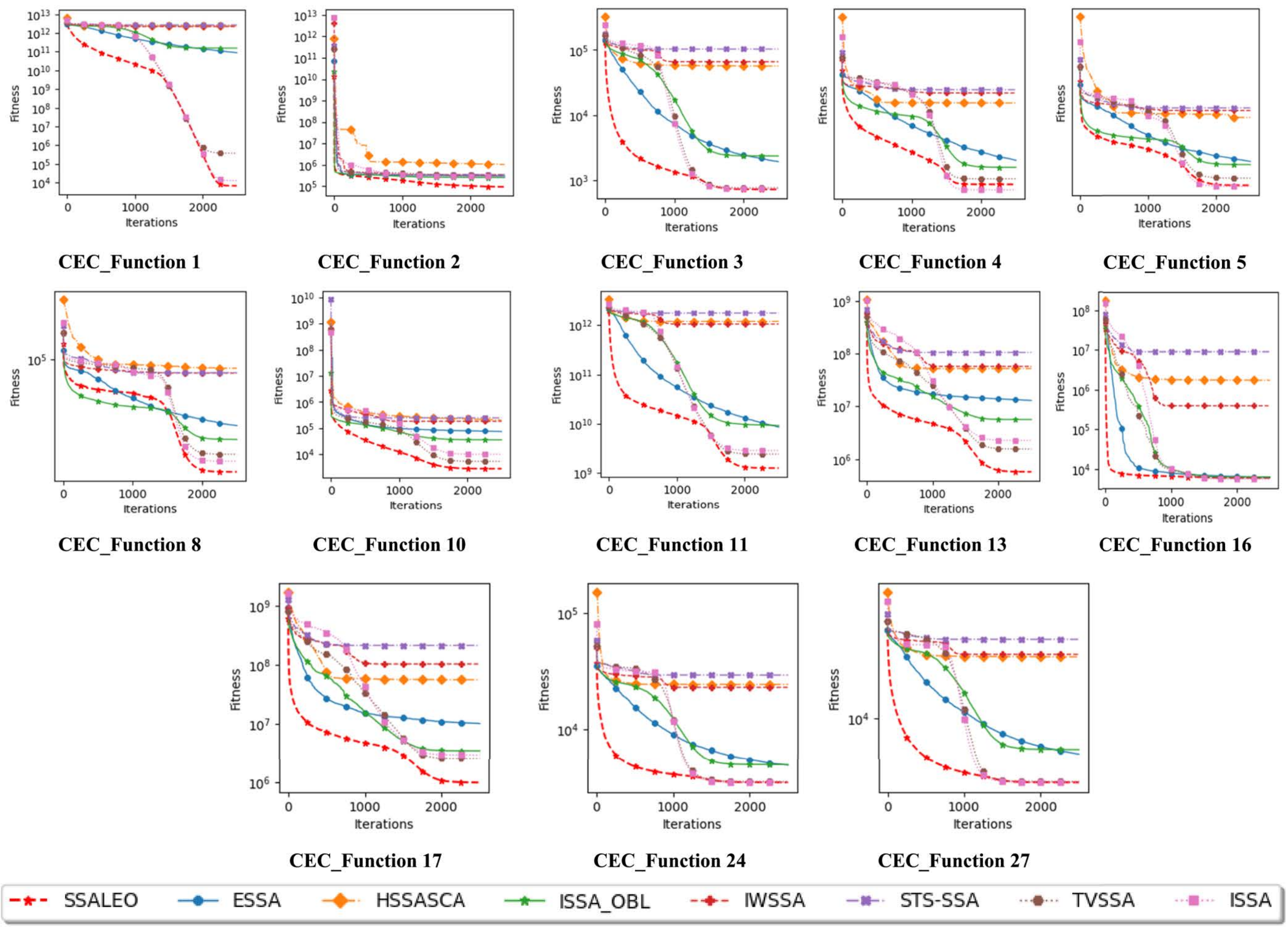

FIGURE 5. SSALEO Convergence curves and other SSA's Variants during 2500 iterations.

Subject to :

$$g_1(x) = \frac{27}{x_1 x_2^2 x_3} - 1 \le 0$$

$$g_2(x) = \frac{397.5}{x_1 x_2^2 x_3^2} - 1 \le 0$$

$$g_3(x) = \frac{1.93 x_4^3}{x_2 x_3 x_6^4} - 1 \le 0$$

$$g_4(x) = \frac{1.93 x_5^3}{x_2 x_3 x_7^4} - 1 \le 0$$

$$g_5(x) = \frac{\sqrt{\left(\frac{745 x_4}{x_2 x_3}\right)^2 + 16.9X10^6}}{110.0 x_6^3} - 1 \le 0$$

$$g_6(x) = \frac{\sqrt{\left(\frac{745 x_4}{x_2 x_3}\right)^2 + 157.5X10^6}}{85.0 x_6^3} - 1 \le 0$$

$$g_7(x) = \frac{x_2 x_3}{40} - 1 \le 0$$

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

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

$$g_{10}(x) = \frac{1.5 x_6 + 1.9}{x_4} - 1 \le 0$$

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

where $2.6 \le x1 \le 3.6$, $0.7 \le x2 \le 0.8$, $17 \le x3 \le 28$, $7.3 \le x4 \le 8.3$, $7.8 \le x5 \le 8.3$, $2.9 \le x6 \le 3.9$, $5.0 \le x7 \le 5.5$.

The SSALEO algorithm outperformed most of the other state-of-the-art algorithms to solve the speed reducer design problem using the optimal decision variables to get optimal minimizing the weights of a speed reader, as shown in Table 29.

7) PROCESS DESIGN PROBLEM

It is a minimization issue, which may be expressed as follows.

$$\min f(x) = 5@period357854 x_1^2 + 40792.141 - 37.29329\, x_4 + 0.835689 x_4 x_3$$

TABLE 19. Comparison results of the SSALEO on CEC2008lsgos with traditional algorithms during 2500 iterations.

Fun	D	Criteria	CSO	SSA	PSO	WOA	GWO	SSALEO
F1	200	Avg	9.41447E+05	1.86466E+05	4.84406E+05	1.20676E+05	1.88453E+05	**-4.50000E+02**
		Std	4.47631E+04	2.47443E+04	3.11655E+04	1.53577E+04	2.11420E+04	1.95613E-07
		Med	9.43571E+05	1.83967E+05	4.79739E+05	1.21832E+05	1.83550E+05	-4.50000E+02
	500	Avg	6.63500E+05	1.19699E+06	1.49758E+06	8.07477E+05	8.26436E+05	**-4.39092E+02**
		Std	4.86521E+04	5.37622E+04	5.43671E+04	3.41124E+04	4.86964E+04	3.89931E+00
		Med	6.58734E+05	1.19844E+06	1.49217E+06	8.07589E+05	8.23521E+05	-4.40015E+02
	1000	Avg	2.41456E+06	2.93200E+06	3.15452E+06	2.25118E+06	2.18275E+06	**1.54259E+04**
		Std	1.11230E+05	6.83467E+04	4.44000E+04	4.82855E+04	5.55002E+04	1.77507E+03
		Med	2.39656E+06	2.92502E+06	3.16067E+06	2.25695E+06	2.18020E+06	1.52354E+04
F2	200	Avg	-2.95191E+02	-3.49316E+02	**-3.96238E+02**	-3.75621E+02	-3.68273E+02	-3.66022E+02
		Std	4.68417E+00	3.59540E+00	1.95092E+00	8.57138E+00	1.50832E+00	2.15898E+00
		Med	-2.95008E+02	-3.49480E+02	-3.96357E+02	-3.74277E+02	-3.67812E+02	-3.65960E+02
	500	Avg	-3.15183E+02	-3.38130E+02	**-3.90372E+02**	-3.20365E+02	-3.55630E+02	-3.55765E+02
		Std	4.74999E+00	2.71316E+00	2.13121E+00	5.83231E+00	5.71280E-01	9.05805E-01
		Med	-3.15372E+02	-3.38643E+02	-3.90477E+02	-3.20112E+02	-3.55838E+02	-3.55627E+02
	1000	Avg	-3.05926E+02	-3.33946E+02	**-3.87202E+02**	-3.69835E+02	-3.52358E+02	-3.52716E+02
		Std	3.23466E+00	2.61506E+00	1.28394E+00	9.16499E+00	8.67306E-01	3.67162E-01
		Med	-3.05971E+02	-3.34178E+02	-3.87205E+02	-3.68042E+02	-3.52654E+02	-3.52748E+02
F3	200	Avg	7.15676E+11	4.58240E+10	1.18039E+11	1.59589E+10	3.90232E+10	**1.67060E+03**
		Std	6.82889E+10	8.96611E+09	1.84286E+10	3.02158E+09	8.71743E+09	1.23151E+03
		Med	7.25965E+11	4.72459E+10	1.20878E+11	1.60628E+10	3.80054E+10	1.40943E+03
	500	Avg	2.74798E+11	3.88128E+11	5.04232E+11	1.85605E+11	2.67730E+11	**3.65995E+05**
		Std	4.38043E+10	2.91775E+10	2.55513E+10	1.46824E+10	1.53898E+10	2.38085E+05
		Med	2.75128E+11	3.89438E+11	5.03726E+11	1.85677E+11	2.64510E+11	3.05483E+05
	1000	Avg	1.24300E+12	1.15242E+12	1.18520E+12	6.79871E+11	7.71725E+11	**1.01216E+09**
		Std	9.16572E+10	3.96255E+10	3.06239E+10	2.15793E+10	2.65896E+10	1.91667E+08
		Med	1.23500E+12	1.15053E+12	1.18479E+12	6.75919E+11	7.71073E+11	9.75496E+08
F4	200	Avg	3.91692E+03	2.19207E+03	2.71978E+03	2.19577E+03	1.53153E+03	**1.41044E+03**
		Std	1.39433E+02	1.04118E+02	1.29070E+02	1.48258E+02	1.24896E+02	5.96982E+01
		Med	3.92975E+03	2.17751E+03	2.70876E+03	2.19851E+03	1.53874E+03	1.41316E+03
	500	Avg	4.89396E+03	7.02541E+03	7.94480E+03	6.86701E+03	5.70529E+03	**3.87185E+03**
		Std	2.46952E+02	1.73898E+02	1.72197E+02	3.16860E+02	1.51257E+02	6.70093E+01
		Med	4.85618E+03	7.01558E+03	7.94437E+03	6.88213E+03	5.70827E+03	3.86776E+03
	1000	Avg	1.34893E+04	1.55147E+04	1.69095E+04	1.47615E+04	1.34250E+04	**8.69073E+03**
		Std	3.16745E+02	2.56674E+02	2.09591E+02	3.31070E+02	2.29643E+02	1.34024E+02
		Med	1.35020E+04	1.54098E+04	1.69069E+04	1.47504E+04	1.34133E+04	8.68958E+03
F5	200	Avg	7.57658E+03	1.38726E+03	3.43089E+03	8.05439E+02	1.24083E+03	**-1.79998E+02**
		Std	4.05594E+02	2.20849E+02	1.92131E+02	1.02604E+02	1.67020E+02	7.63254E-04
		Med	7.53768E+03	1.40279E+03	3.39431E+03	7.95720E+02	1.22701E+03	-1.79998E+02
	500	Avg	5.31005E+03	9.37326E+03	1.16501E+04	6.30289E+03	6.45561E+03	**-1.79123E+02**
		Std	4.37739E+02	4.87717E+02	3.20315E+02	2.70899E+02	3.69933E+02	1.65985E-01
		Med	5.31066E+03	9.39097E+03	1.16616E+04	6.27431E+03	6.39312E+03	-1.79163E+02
	1000	Avg	2.13033E+04	2.54891E+04	2.75882E+04	1.97767E+04	1.90584E+04	**-3.15769E+01**
		Std	7.80874E+02	8.30883E+02	4.44346E+02	5.07612E+02	4.40975E+02	1.51432E+01
		Med	2.13028E+04	2.55206E+04	2.76032E+04	1.99343E+04	1.91221E+04	-3.09241E+01
F6	200	Avg	-1.18739E+02	-1.19479E+02	-1.19347E+02	-1.20693E+02	**-1.20973E+02**	-1.20688E+02
		Std	2.69502E-02	4.27480E-02	9.03937E-02	1.48562E-02	3.99925E-01	1.45662E-03
		Med	-1.18737E+02	-1.19475E+02	-1.19335E+02	-1.20690E+02	-1.20963E+02	-1.20688E+02
	500	Avg	-1.20098E+02	-1.19228E+02	-1.19109E+02	-1.19812E+02	-1.20404E+02	**-1.20704E+02**
		Std	1.42874E-01	2.77015E-02	5.01581E-02	2.30793E-02	1.74640E-01	4.59695E-04
		Med	-1.20096E+02	-1.19230E+02	-1.19111E+02	-1.19811E+02	-1.20458E+02	-1.20704E+02
	1000	Avg	-1.19380E+02	-1.19113E+02	-1.18939E+02	-1.19474E+02	-1.20402E+02	**-1.20622E+02**
		Std	4.11822E-02	1.40751E-02	2.36246E-02	1.54084E-02	3.32611E-01	9.93389E-06
		Med	-1.19384E+02	-1.19112E+02	-1.18938E+02	-1.19473E+02	-1.20483E+02	-1.20622E+02
F7	200	Avg	-2.31102E+05	-3.62768E+05	-1.64424E+05	-4.16867E+05	**-5.39416E+05**	-4.00133E+05
		Std	1.32955E+04	3.96560E+04	2.08723E+02	5.31327E+04	4.46551E+04	3.93614E+04
		Med	-2.27775E+05	-3.62799E+05	-1.64386E+05	-4.13344E+05	-5.42449E+05	-3.93486E+05
	500	Avg	-7.84102E+05	-6.42985E+05	-4.30916E+05	-9.15178E+05	**-9.25915E+05**	-7.56513E+05
		Std	7.93991E+04	4.88904E+04	1.68187E+04	9.64777E+04	5.43955E+04	7.83293E+04
		Med	-7.90207E+05	-6.41969E+05	-4.33987E+05	-9.20699E+05	-9.19190E+05	-7.55646E+05
	1000	Avg	-1.24817E+06	-1.06619E+06	-7.25659E+05	**-1.96127E+06**	-1.45351E+06	-1.27721E+06
		Std	8.32144E+04	5.82091E+04	6.94568E+03	2.47030E+05	7.89016E+04	1.06644E+05
		Med	-1.24958E+06	-1.05764E+06	-7.26927E+05	-1.96659E+06	-1.44571E+06	-1.25153E+06
Rank	200	W/T/L	00/00/07	00/00/07	01/00/06	00/00/07	02/00/05	04/00/03
	500	W/T/L	00/00/07	00/00/07	01/00/06	00/00/07	01/00/06	05/00/02
	1000	W/T/L	00/00/07	00/00/07	01/00/06	01/00/06	00/00/07	05/00/02
	Overall OE		0.000%	00.000%%	14.280%	4.760%	14.280%	**66.660%**

TABLE 20. Friedman test result of the SSALEO vs.other traditional algorithms on CEC2008lsgo.

Algorithm	Dimension	Average Rank	Overall Rank
CSO	200	5.86	6
	500	3.08	3
	1000	4.20	4
SSA	200	3.89	4
	500	4.77	5
	1000	4.91	5
PSO	200	4.56	5
	500	5.28	6
	1000	5.15	6
WOA	200	2.34	2
	500	3.47	4
	1000	2.54	3
GWO	200	2.48	3
	500	2.84	2
	1000	2.50	2
SSALEO	**200**	**1.88**	**1**
	500	**1.55**	**1**
	1000	**2.21**	**1**

TABLE 21. Wilcoxon rank sum test result of the SSALEO vs other traditional algorithms on CEC2008lsgo.

Fun	D	CSO	SSA	PSO	WOA	GWO
F1	200	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	500	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	1000	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F2	200	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	500	< 0.05	< 0.05	< 0.05	< 0.05	<u>0.761699</u>
	1000	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F3	200	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	500	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	1000	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F4	200	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	500	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	1000	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F5	200	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	500	< 0.05	< 0.05	< 0.05	<u>05493</u>	< 0.05
	1000	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F6	200	< 0.05	< 0.05	< 0.05	<u>0.331072</u>	< 0.05
	500	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	1000	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F7	200	< 0.05	< 0.05	< 0.05	<u>0.125571</u>	< 0.05
	500	<u>0.141671</u>	< 0.05	< 0.05	< 0.05	< 0.05
	1000	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05

$$\begin{aligned}
\textit{Subject to}\ :\ g_1(x) &= -92 + a_3x_4x_2 + a_1 \\
&\quad + a_2x_4x_3 - a_4x_4x_3 \leq 0 \\
g_1\,(x) &= -110 + a_7x_4x_2 + a_5 \\
&\quad + a_6x_5x_3 - a_8x_1^2 \leq 0 \\
g_1(x) &= a_9 + a_{11}x_4x_1 + a_{10}x_4x_3 \\
&\quad - 25 + a_{12}x_1x_2 \leq 0
\end{aligned}$$

TABLE 22. Friedman test result of the SSALEO vs.other advanced algorithms on CEC2008lsgo.

Algorithm	Dimension	Average Rank	Overall Rank
PPSO	200	5.05	5
	500	4.59	5
	1000	4.21	4
PPSO_W	200	4.99	4
	500	3.48	3
	1000	3.17	2
DESAP-abs	200	3.34	3
	500	3.70	4
	1000	4.06	3
SHADE	200	3.20	2
	500	3.40	2
	1000	4.75	5
CMA-ES	200	5.26	6
	500	8.84	9
	1000	8.64	8
Large-scale LM-CMA	200	7.72	8
	500	6.82	7
	1000	6.96	7
Large-scale QIWOA	200	7.78	9
	500	7.31	8
	1000	6.97	9
Large-scale DSCA	200	5.81	7
	500	5.11	6
	1000	4.85	6
SSALEO	**200**	**1.86**	**1**
	500	**1.74**	**1**
	1000	**1.20**	**1**

With bounds:

$$\begin{aligned}
27 \leq x_1, x_2, x_3 &\leq 45 \\
x_4 &\in 78, 79, \ldots., 102\} \\
x_5 &\in 78, 79, \ldots., 102\}
\end{aligned}$$

where $a_1 to a_{12}$ and their values are listed in Table 30.

The SSALEO algorithm outperformed most of the other state-of-the-art algorithms to solve the speed reducer design problem using the optimal decision variables to get optimal minimizing the weights of process design, as shown in Table 31.

VIII. SSALEO FOR HIGH DIMENSIONAL DATA FEATURE SELECTION

Data mining and machine learning work more efficiently and effectively when dimensions are reduced [123]. Dimensionality reduction includes feature extraction and selection. Feature extraction creates a new set of attributes. A feature selection process eliminates superfluous or useless attributes to enhance learning efficiency [123]. FE approaches are less popular in machine learning than FS. Dimensionality reduction is crucial when learning about high-dimensional

TABLE 23. Comparison results of the SSALEO on CEC2008lsgos with advanced algorithms during 2500 iterations.

F.	D.	Cri.	PPSO	PPSO_W	DESAP-abs	SHADE	CMA-ES	Large-scale LM-CMA	Large-scale QIWOA	Large-scale DSCA	SSALEO
F1	200	Avg	2.67723E+03	5.40056E+03	-2.75121E+02	2.27078E+02	-4.39674E+02	7.09641E+05	1.16383E+06	6.50308E+05	**-4.50000E+02**
		Std	8.56053E+02	3.45015E+03	2.69432E+02	1.12988E+03	1.22157E+01	3.31634E+03	5.88773E+04	1.14243E+04	1.91633E-07
		Med	2.45246E+03	4.78277E+03	-3.69375E+02	-3.22867E+02	-4.44539E+02	7.09758E+05	1.17485E+06	6.50132E+05	-4.50000E+02
	500	Avg	1.33129E+05	7.54889E+04	1.21081E+05	1.33654E+05	3.92921E+06	1.75020E+06	3.03145E+06	1.68612E+06	**-4.39092E+02**
		Std	1.26192E+04	7.55302E+03	2.33644E+04	2.07153E+04	1.56345E+06	1.18318E+03	9.58400E+04	1.10611E+04	3.89931E+00
		Med	1.30973E+05	7.35224E+04	1.18493E+05	1.33243E+05	3.71338E+06	1.75007E+06	3.04107E+06	1.68861E+06	-4.40015E+02
	1000	Avg	1.00148E+06	7.60065E+05	9.24827E+05	9.08750E+05	4.23271E+07	3.38888E+06	6.18188E+06	3.33397E+06	**1.54259E+04**
		Std	3.64359E+04	5.23646E+04	6.47633E+04	6.20370E+04	8.43545E+05	1.38495E+03	1.75382E+05	1.22985E+04	1.77507E+03
		Med	9.98245E+05	7.55911E+05	9.22530E+05	9.04742E+05	4.22671E+07	3.38870E+06	6.20614E+06	3.33540E+06	1.52354E+04
F2	200	Avg	-3.60473E+02	-3.62456E+02	-3.57064E+02	-3.57772E+02	-3.32317E+02	-3.52660E+02	-3.19964E+02	-3.53686E+02	**-3.66022E+02**
		Std	2.13364E+00	2.30688E+00	3.86553E+00	2.91153E+00	4.90023E+01	3.17566E-01	2.64971E+01	2.34251E-01	2.15898E+00
		Med	-3.60398E+02	-3.62236E+02	-3.57085E+02	-3.58120E+02	-3.38324E+02	-3.52627E+02	-3.11711E+02	-3.53621E+02	-3.65960E+02
	500	Avg	-3.53759E+02	-3.54262E+02	-3.43151E+02	-3.42792E+02	1.20475E+02	-3.50757E+02	-3.04261E+02	-3.51362E+02	**-3.55765E+02**
		Std	8.05924E-01	6.35328E-01	2.50321E+00	2.68927E+00	2.07355E+01	1.23330E-01	2.45462E+01	2.82188E-01	9.05805E-01
		Med	-3.53529E+02	-3.54212E+02	-3.43146E+02	-3.43400E+02	1.20599E+02	-3.50771E+02	-3.02662E+02	-3.51238E+02	-3.55627E+02
	1000	Avg	-3.51804E+02	-3.52053E+02	-3.35916E+02	-3.34598E+02	1.69464E+02	-3.50200E+02	-2.87897E+02	-3.50447E+02	**-3.52716E+02**
		Std	3.89201E-01	4.11044E-01	2.26463E+00	2.45575E+00	1.89466E+01	1.04070E-01	2.64497E+01	5.45108E-02	3.67162E-01
		Med	-3.51807E+02	-3.52060E+02	-3.35891E+02	-3.34314E+02	1.69195E+02	-3.50193E+02	-2.78877E+02	-3.50460E+02	-3.52748E+02
F3	200	Avg	9.38313E+07	1.77352E+08	4.15728E+06	5.37904E+06	1.27028E+06	2.28729E+11	1.15027E+12	2.03900E+11	**1.67060E+03**
		Std	1.41342E+08	3.30954E+08	5.77312E+06	1.00395E+07	2.82489E+06	2.70170E+09	1.03055E+11	7.80542E+09	1.23151E+03
		Med	5.97508E+07	5.39904E+07	2.38440E+06	1.89543E+06	2.83976E+05	2.28992E+11	1.16254E+12	2.03239E+11	1.40943E+03
	500	Avg	1.84986E+10	7.00747E+09	2.20395E+10	2.36082E+10	7.82628E+13	6.29942E+11	3.42392E+12	5.98475E+11	**3.65995E+05**
		Std	2.92824E+09	1.27070E+09	5.93853E+09	9.83098E+09	6.88083E+13	9.20155E+08	2.52308E+11	5.80700E+09	2.38085E+05
		Med	1.78449E+10	6.70250E+09	2.24962E+10	2.17339E+10	4.64744E+13	6.29967E+11	3.44072E+12	5.99543E+11	3.05483E+05
	1000	Avg	2.35700E+11	1.44913E+11	2.65983E+11	2.72967E+11	5.03074E+14	1.27997E+12	7.35454E+12	1.24982E+12	**1.01216E+09**
		Std	2.17924E+10	1.36187E+10	2.85819E+10	2.37643E+10	2.72030E+13	1.04718E+09	4.76471E+11	6.93367E+09	1.91667E+08
		Med	2.37000E+11	1.46364E+11	2.61478E+11	2.71349E+11	5.07586E+14	1.27989E+12	7.52228E+12	1.25066E+12	9.75496E+08
F4	200	Avg	1.49429E+03	1.45366E+03	9.43151E+01	**9.11320E+01**	1.76836E+03	2.84602E+03	4.40701E+03	3.20377E+03	1.41044E+03
		Std	6.82082E+01	6.12147E+01	2.92671E+01	3.46131E+01	4.77365E+01	6.96089E+01	1.38971E+02	3.90448E+01	5.96982E+01
		Med	1.49176E+03	1.44494E+03	9.21228E+01	8.91806E+01	1.78302E+03	2.84735E+03	4.40023E+03	3.19833E+03	1.41316E+03
	500	Avg	5.16953E+03	4.87178E+03	**3.38261E+03**	3.44918E+03	1.46428E+04	7.93422E+03	1.20302E+04	8.50340E+03	3.87185E+03
		Std	1.10668E+02	1.12635E+02	1.32185E+02	1.23618E+02	3.60132E+03	9.05199E+01	3.60422E+02	6.77288E+01	6.70093E+01
		Med	5.16775E+03	4.84719E+03	3.39056E+03	3.48162E+03	1.52635E+04	7.94267E+03	1.20521E+04	8.51976E+03	3.86776E+03
	1000	Avg	1.24363E+04	1.18640E+04	9.61515E+03	9.68486E+03	1.15704E+05	1.67758E+04	2.46720E+04	1.75091E+04	**8.69073E+03**
		Std	2.04379E+02	1.93456E+02	8.59604E+02	8.87801E+02	2.27470E+03	1.10138E+02	5.76736E+02	7.36386E+01	1.34024E+02
		Med	1.24661E+04	1.18652E+04	9.47728E+03	9.64636E+03	1.16135E+05	1.67761E+04	2.47474E+04	1.75076E+04	8.68958E+03
F5	200	Avg	-1.51749E+02	-1.41683E+02	-1.77604E+02	-1.76849E+02	-1.78941E+02	5.47354E+03	9.41180E+03	4.90565E+03	**-1.79998E+02**
		Std	7.75099E+00	2.49952E+01	2.11595E+00	4.18669E+00	3.26564E-01	4.89619E+00	6.23769E+02	9.11084E+01	7.63254E-04
		Med	-1.52617E+02	-1.51422E+02	-1.78385E+02	-1.78063E+02	-1.79029E+02	5.47495E+03	9.40153E+03	4.92431E+03	-1.79998E+02
	500	Avg	1.02184E+03	4.83694E+02	8.51192E+02	8.49894E+02	3.20684E+04	1.38051E+04	2.53892E+04	1.32358E+04	**-1.79123E+02**
		Std	1.23337E+02	8.01745E+01	2.06541E+02	2.00183E+02	1.48241E+04	1.69611E+00	8.69293E+02	1.13895E+02	1.65985E-01
		Med	1.02559E+03	4.45034E+02	8.32378E+02	8.09551E+02	3.01526E+04	1.38046E+04	2.56591E+04	1.32510E+04	-1.79163E+02
	1000	Avg	8.60333E+03	5.86310E+03	7.97408E+03	7.73692E+03	3.80610E+05	2.99135E+04	5.46945E+04	2.93390E+04	**-3.15769E+01**
		Std	4.98719E+02	3.02618E+02	5.83882E+02	4.83914E+02	9.47069E+03	1.76983E+00	1.51757E+03	9.77270E+01	1.51432E+01
		Med	8.53873E+03	5.87774E+03	7.90700E+03	7.91416E+03	3.80254E+05	2.99134E+04	5.49442E+04	2.93369E+04	-3.09241E+01
F6	200	Avg	-1.20199E+02	-1.20174E+02	**-1.29473E+02**	-1.29117E+02	-1.18519E+02	-1.19304E+02	-1.19682E+02	-1.20706E+02	-1.20688E+02
		Std	3.04941E-01	2.91724E-01	1.50757E+00	1.60569E+00	1.89323E-02	4.37215E-02	4.88668E-01	1.52125E-02	1.45662E-03
		Med	-1.20021E+02	-1.20020E+02	-1.29268E+02	-1.29220E+02	-1.18515E+02	-1.19304E+02	-1.19981E+02	-1.20700E+02	-1.20688E+02
	500	Avg	-1.20135E+02	-1.20142E+02	**-1.25797E+02**	-1.25677E+02	-1.18428E+02	-1.19227E+02	-1.19741E+02	-1.20599E+02	-1.20704E+02
		Std	2.51087E-01	2.50208E-01	5.59690E-01	7.00558E-01	8.26991E-03	1.81520E-02	4.71897E-01	4.28952E-01	4.59695E-04
		Med	-1.20017E+02	-1.20017E+02	-1.25856E+02	-1.25711E+02	-1.18429E+02	-1.19225E+02	-1.20005E+02	-1.20711E+02	-1.20704E+02
	1000	Avg	-1.20120E+02	-1.20204E+02	-1.19833E+02	-1.19823E+02	-1.18383E+02	-1.19056E+02	-1.19633E+02	-1.20514E+02	**-1.20622E+02**
		Std	2.14074E-01	2.60545E-01	7.68115E-02	7.35832E-02	6.08336E-03	1.93918E-02	5.53791E-01	4.27335E-01	9.93389E-06
		Med	-1.20015E+02	-1.20015E+02	-1.19829E+02	-1.19824E+02	-1.18383E+02	-1.19058E+02	-1.19956E+02	-1.20626E+02	-1.20622E+02

data. Filter-based and wrapper-based approaches are used to identify features [124].

The filter-based methods pick features that are acceptable based on statistics and data. Wrapper-based techniques use machine learning algorithms to find a near-optimal solution. They don't consider relevance when choosing subgroup attributes. Wrapper-based methods, in contrast, determine the optimum subset depending on classifier accuracy.

TABLE 23. *(Continued.)* **Comparison results of the SSALEO on CEC2008lsgos with advanced algorithms during 2500 iterations.**

F7	200	Avg	-1.62035E+05	-1.63961E+05	-1.50032E+05	-1.50782E+05	-7.38559E+04	-1.13269E+05	-1.93821E+05	-2.12997E+05	**-4.00133E+05**
		Std	8.34091E+04	9.39612E+04	1.76678E+04	1.18405E-10	3.87406E+03	2.77901E+02	1.72633E+04	2.07810E+04	3.93614E+04
		Med	-1.46806E+05	-1.46806E+05	-1.46806E+05	-1.50782E+05	-7.31485E+04	-1.13320E+05	-1.90662E+05	-2.07338E+05	-3.93486E+05
	500	Avg	-2.47637E+05	-2.79220E+05	-2.56923E+05	-4.44329E+05	-1.24142E+05	-2.27913E+05	-4.17585E+05	-4.17840E+05	**-7.56513E+05**
		Std	8.88041E-11	1.72986E+05	6.34216E+03	1.56074E+03	1.26437E+03	3.00426E+01	5.86834E+04	1.73424E+04	7.83293E+04
		Med	-2.47637E+05	-2.47637E+05	-2.55765E+05	-4.44044E+05	-1.24373E+05	-2.27907E+05	-4.12069E+05	-4.19029E+05	-7.55646E+05
	1000	Avg	-4.56663E+05	-4.34942E+05	-9.33121E+05	-3.89296E+05	-3.89296E+05	-3.75098E+05	-7.32177E+05	-6.97834E+05	**-1.27721E+06**
		Std	3.91823E+05	2.72852E+05	7.57056E+04	2.28364E+04	2.28364E+04	3.68102E+03	8.44885E+04	3.85230E+04	1.06644E+05
		Med	-3.85127E+05	-3.85127E+05	-9.57932E+05	-3.85127E+05	-3.85127E+05	-3.74426E+05	-7.13519E+05	-6.89440E+05	-1.25153E+06
Rank	200	W/T/L	00/00/07	00/00/07	01/00/06	01/00/06	00/00/07	00/00/07	00/00/07	00/00/07	05/00/02
	500	W/T/L	00/00/07	00/00/07	02/00/05	00/00/07	00/00/07	00/00/07	00/00/07	00/00/07	05/00/02
	1000	W/T/L	00/00/07	00/00/07	00/00/07	00/00/07	00/00/07	00/00/07	00/00/07	00/00/07	07/00/00
Overall OE			0.000%	0.000%	14.280%	4.760%	0.000%	0.000%	0.000%	0.000%	**80.950%**

TABLE 24. **Wilcoxon rank-sum test result of the SSALEO vs other advanced algorithms on CEC2008lsgo.**

Fun	D	PPSO	PPSO_W	DESAP-abs	SHADE	CMA-ES	Large-scale LM-CMA	Large-scale QIWOA	Large-scale DSCA
F1	200	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	500	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	1000	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F2	200	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	500	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	1000	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F3	200	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	500	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	1000	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F4	200	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	500	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	1000	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F5	200	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	500	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	1000	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F6	200	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	500	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	1000	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F7	200	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	500	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	1000	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05

Numerous studies have compared feature selection methods, such as [125] which evaluated eight typical approaches using three classifiers. Due to data's growth and complexity, problem-solving utilizing feature selection has expanded. Typical optimization approaches require examining all feasible subsets, making the application of AI for massive data sets challenging and expensive [126]. Several optimization techniques have already been created to address such difficulties, as old methods are inefficient [75]. In addition, several metaheuristic algorithms [72], [79], [125], [127], [128] have been used to solve feature selection problems.

This sub-section shows experimental findings on how SSALEO can be applied to problems in the real world, such as selecting features from high-dimensional data (with more than 2000 decision variables).

A. METHODOLOGY AND DATASET DESCRIPTION

A variety of meta-heuristic feature selection techniques are compared and contrasted with TBLSBCL in this section. In addition, various benchmark high-dimensional datasets are used to illustrate the effectiveness of algorithms in large feature spaces, including central nervous system (CNS) [129], ovarian cancer [130], and Colon cancer [131]. We've compiled a summary of the datasets we looked at in Table 32. In addition, for a fair comparison, all procedures use the same seed data. Table 5 contains the rest of the parameters for each method. A total of twenty repetitions of each procedure were performed on a machine equipped with 4 GB of RAM and an Intel Core i3 processor to eliminate the potential of chance. We start with a population of 10, then run for 50 iterations.

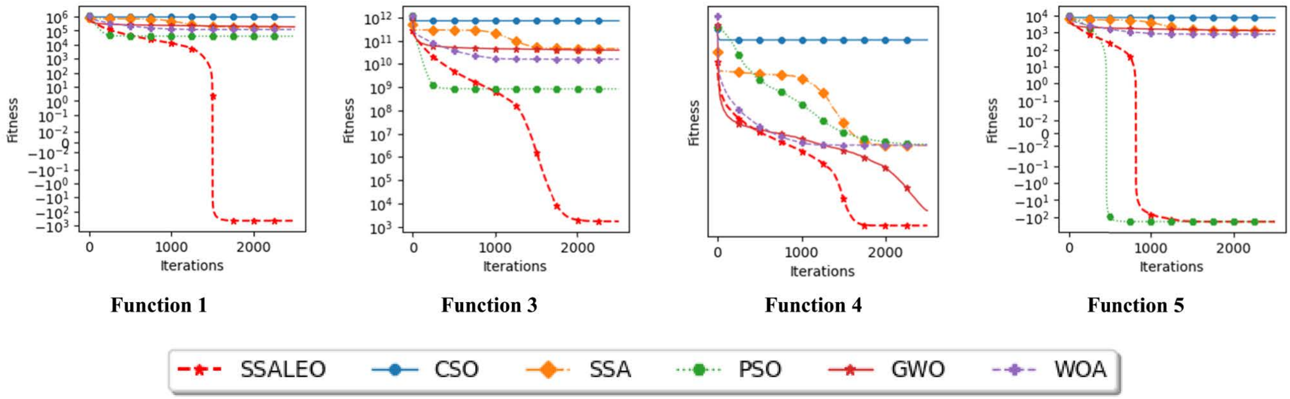

FIGURE 6. SSALEO Convergence curves and other traditional algorthms during 2500 iterations on CEC2008lsgo.

TABLE 25. The specifics of the seven real-world engineering design difficulties. D represents the total number of decision variables in the problem, g represents the number of inequality constraints, h represents the number of equality constraints, and $f(x^*)$ represents the best-known feasible objective function value.

Fun	Name	D	g	h	$f(x^*)$
F1	Tension/compression spring design (case 1)	3	3	0	1.2665232788E-02
F2	Pressure vessel design	4	4	0	5.8853327736E+03
F3	Three-bar truss design problem	2	3	0	2.6389584338E+02
F4	Welded beam design	4	5	0	1.6702177263E+00
F5	Multiple disk clutch brake design problem	5	7	0	2.3524245790E-01
F6	Weight Minimization of a Speed Reduce	7	11	0	2.9944244658E+03
F7	Process design Problem	5	3	0	2.6887000000E+04

TABLE 26. An examination of several existing studies in the literature in comparison to the suggested SSALEO method for Tension/compression spring design and pressure vessel design problems.

Algorithm	Tension/compression spring design				Pressure vessel design			
	Best	**Worst**	**Mean**	**Std.**	**Best**	**Worst**	**Mean**	**Std.**
IUDE[116], [117]	1.27E-02	1.27E-02	1.27E-02	1.08E-05	6.06E+03	6.06E+03	6.06E+03	6.16E+00
ϵMAgES[118]	1.27E-02	1.37E-02	1.27E-02	2.16E-04	6.06E+03	1.19E+04	7.38E+03	1.93E+03
iLSHADEϵ [119]	1.27E-02	1.78E-02	1.30E-02	1.06E-03	6.06E+03	1.49E+04	8.48E+03	3.14E+03
COLSHADE[120]	1.27E-02	1.27E-02	1.27E-02	1.08E-07	6.06E+03	6.09E+03	6.06E+03	8.53E+00
BiPopEpsMAgES [121]	1.27E-02	1.37E-02	1.27E-02	1.09E-04	6.06E+03	7.46E+03	6.17E+03	2.10E+02
SASS [122]	1.27E-02	1.27E-02	1.27E-02	2.62E-06	6.06E+03	8.96E+03	6.41E+03	6.28E+02
SSALEO	**0.01266**	**0.012**	**0.01270**	**2.2837E-05**	**4543**	**5271**	**4738**	**159.6421**

B. TRANSFER FUNCTION

Optimization strategies for Feature Selection are typically binary. No feature selection problem-solving techniques exist outside the binary range [0,1]. For this reason, it's important to develop a binary implementation of the optimization procedure. After investigating the process of transforming a continuous optimization algorithm into a binary one, Mirjalili and Lewis [132] discovered that using a Transfer Function (TF) could be helpful. The probability-based transfer function maps continuous data to binary 0s and 1s. In order to implement the s-shaped transfer function, we use Eq. (23-24).

$$X_{S2} = \frac{1}{1+e^{-x}} \,(23) \tag{23}$$

$$X_{Binary} = \begin{cases} 0, & X_{S2} < N_{random} \\ 1, & X_{S2} \geq N_{random} \end{cases} \tag{24}$$

The variable denotes the solution to the problem of selecting features X_{Binary} while the random number denotes the threshold N_{random}.

C. OBJECTIVE FUNCTION AND MEASURE OF PERFORMANCE

The objective function must be well thought out before constructing the optimization problem. Wrapper feature selection approaches, for instance, can be used to reduce the total amount of features while simultaneously increasing the precision of a learning process. These two competing

TABLE 27. An examination of several existing studies in the literature in comparison to the suggested SSALEO method for Three-bar truss design design and Welded beam design problems.

Algorithm	Three-bar truss design problem				Welded beam design			
	Best	Worst	Mean	Std.	Best	Worst	Mean	Std.
IUDE[116], [117]	2.64E+02	2.64E+02	2.64E+02	0.00E+00	1.67E+00	1.67E+00	1.67E+00	1.20E-16
ϵMAgES[118]	2.64E+02	2.64E+02	2.65E+02	2.88E+00	1.67E+00	1.85E+00	1.67E+00	3.95E-02
iLSHADEϵ [119]	2.64E+02	2.64E+02	2.64E+02	1.99E- 02	1.67E+00	1.67E+00	1.67E+00	7.59E-07
COLSHADE[120]	263.8958	263.896099	263.895865	5.8E− 14	1.67E+00	1.67E+00	1.67E+00	2.27E-16
BiPopEpsMAgES [121]	263.895843	263.896201	263.895903	8.71E− 05	1.67E+00	1.67E+00	1.67E+00	2.30E-03
SASS [122]	263.895843	263.896201	263.895903	5.69E-14	1.67E+00	1.79E+00	1.68E+00	2.01E-02
SSALEO	**263.463430**	**263.463430**	**263.463430**	**1.1563E-13**	**1.670438038**	1.814587608	1.715378845	0.045403417

TABLE 28. An examination of several existing studies in the literature compared to the suggested SSALEO method for multiple disk clutch break design problem.

Algorithm	Multiple disk clutch break design problem			
	Best	Worst	Mean	Std.
IUDE[116], [117]	0.235242	0.235242	0.235242	1.69E-16
ϵMAgES[118]	0.235242	0.235242	0.235242	1.69E-16
iLSHADEϵ [119]	0.235242	0.235242	0.235242	1.13E-16
COLSHADE[120]	0.235242	0.235242	0.235242	2.83E-17
BiPopEpsMAgES [121]	0.235242	0.235242	0.235242	5.84E-16
SASS [122]	0.235242	0.235242	0.235242	8.55E-07
SSALEO	**0.235242**	**0.235242**	**0.235242**	4.2E-08

TABLE 29. An examination of several existing studies in the literature in comparison to the suggested SSALEO method for speed reader design design

Algorithm	Speed reader design problem			
	Best	Worst	Mean	Std.
IUDE[116], [117]	2.99E+03	2.99E+03	2.99E+03	4.64E+13
ϵMAgES[118]	2.99E+03	2.99E+03	2.99E+03	4.64E+13
iLSHADEϵ [119]	2.99E+03	2.99E+03	2.99E+03	4.64E+13
COLSHADE[120]	2.99E+03	2.99E+03	2.99E+03	4.64E+13
BiPopEpsMAgES [121]	2.99E+03	2.99E+03	2.99E+03	4.64E+13
SASS [122]	2.99E+03	305E+03	3.00E+03	6.29E+00
SSALEO	**2992.63**	3010.928	3001.707	5.28610

TABLE 30. Process design problem constant.

$a_1 = 85.334407$	$a_5 = 80.51249$	$a_9 = 9.300961$
$a_2 = 0.00556858$	$a_6 = 0.0071317$	$a_{10} = 0.0047026$
$a_3 = 0.0006262$	$a_7 = 0.0029955$	$a_{11} = 0.0012547$
$a_4 = 0.00222053$	$a_8 = 0.0021813$	$a_{12} = 0.0019085$

objectives need to be factored into the objective function. This research employed equation (25) as an objective function since it simultaneously minimizes the selection ratio and the classification error rate (minimization).

$$Fitness = \rho Err\,(D) + \varphi \frac{|F|}{|T|} \tag{25}$$

TABLE 31. An examination of several existing studies in the literature in comparison to the suggested SSALEO method for process design design.

Algorithm	Process design problem			
	Best	Worst	Mean	Std.
IUDE[116], [117]	2.69E+04	2.69E+04	2.69E+04	1.11E-11
ϵMAgES[118]	2.69E+04	2.69E+04	2.69E+04	1.11E-11
iLSHADEϵ [119]	2.69E+04	2.69E+04	2.69E+04	1.11E-11
COLSHADE[120]	2.69E+04	2.69E+04	2.69E+04	1.11E-11
BiPopEpsMAgES [121]	2.69E+04	2.69E+04	2.69E+04	1.11E-11
SASS [122]	2.69E+04	2.69E+04	2.69E+04	1.11E-11
SSALEO	**2.66E+04**	**2.66E+04**	**2.66E+04**	6.10E-10

TABLE 32. Dataset overview.

ID	DS Name	No. of Samples	No. attributes	No. classes
D1	CNS	60	7129	2
D2	Colon	62	2000	2
D3	Ovarian	253	15155	2

where ρ and φ are constants that adjust the precision of the classification and the level of feature reduction, respectively. The result was generated using the k-Nearest Neighbor (k-NN) classifier with K = 5. The error rate, denoted by Err (D), represents the percentage of incorrect identifications in the recognized subset. $|T|$ is the total number of features and, $|F|$ is the size of the subset of features that have been determined. In this investigation, ρ is set to 0.99 [133], and $\varphi = 1-\rho$. Furthermore, Three criteria are utilized to evaluate the suggested strategy in comparison to existing ones: Accuracy in classification is determined by averaging the results of twenty iterations on the test dataset using the same set of features, as well as the fitness values and the average number of features for each of the methods used.

D. EVALUATION AND DISCUSSIONS OF CALCULATED FINDINGS

The effectiveness and efficiency of the proposed algorithm are assessed in this part. The results of the comparisons may be seen in Tables 33 and 34, with the most advantageous

TABLE 33. Analyzing and contrasting statistical data Calculating the average fitness over 20 iterations.

	GWO	ISSA	ESSA	TVBSSA	SSA	WOA	SSALEO
D1	0.48927	0.41443	0.49908	0.42580	0.43792	0.48448	**0.29376**
D2	0.45282	0.47686	0.46205	0.47668	0.49289	0.45736	**0.19073**
D3	0.39268	0.36455	0.38624	0.36281	0.36070	0.39283	**0.05976**

TABLE 34. Comparison of statistical results Accuracy of Classification,Selection Size over 20 iteration.

ID	GWO		ISSA		ESSA		TVBSSA		SSA		WOA		SSALEO	
	Accu.	#Fe	Accu.	#Fe	Accu.	#Fe	Accu.	#Fe	Accu.	#Fe	Accu.	#Fe	Accu.	#Fe
D1	0.639	3549.55	0.638	2716.000	0.629	3240.70	0.648	2384.40	0.673	1066.15	0.632	3779.150	**0.6100**	4159.05
D2	0.802	999.500	0.807	878.050	0.808	705.700	**0.818**	632.100	0.813	671.700	0.812	1018.300	0.7987	1159.65
D3	0.915	7573.10	0.928	1904.150	0.919	5324.30	0.934	2753.50	0.924	2936.45	0.923	7697.950	**0.9424**	7956.75

TABLE 35. Wilcoxon rank-sum and Freidman test SSALEO vs.other methods.

ID	SSALEO Vs GWO		SSALEO Vs ISSA		SSALEO Vs ESSA		SSALEO Vs TVBSSA		SSALEO Vs SSA		SSALEO Vs WOA		SSALEO	
	Wel.	Frie.	Wel.	Frie.	Wel.	Frie.	Wel.	Frie.	Wel.	Frie.	Wel.	Frie.	Wel.	Frie.
D1	<0.05	5.5789	<0.05	2.9737	<0.05	6.2632	<0.05	3.5526	<0.05	3.5789	<0.05	5.0526	----	1.0000
D2	<0.05	2.9211	<0.05	5.4737	<0.05	3.8947	<0.05	5.6579	<0.05	5.9737	<0.05	3.0789	----	1.0000
D3	<0.05	6.3158	<0.05	3.3684	<0.05	5.2632	<0.05	3.3684	<0.05	2.9474	<0.05	5.7368	----	1.0000

Wel: Wilcoxon rank-sum test Frie.: Friedman test

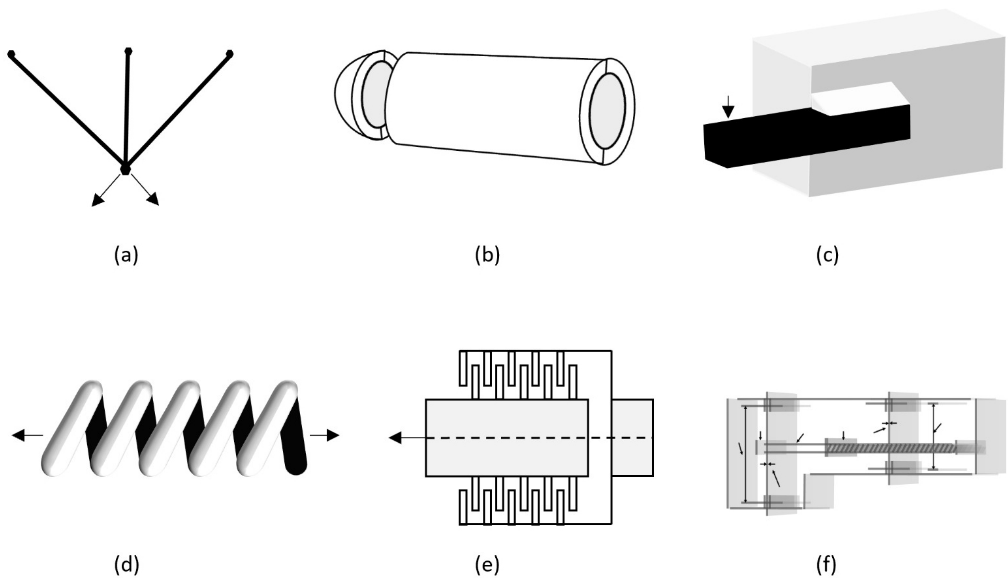

FIGURE 7. (a) Structure of three-bar truss design problem [134]. (b) Structure of Pressure vessel design problem [135]. (c) Welded beam design problem [136]. (d) Structure of tension/compression spring design problem [138]. (e) Multiple disk clutch brake design problem [137]. (f) The speed reducer problem [139].

results highlighted in bold. In table 30, Accu and # Fe stand for the average classification accuracy and the average number of the specified feature, respectively.

Table 33 demonstrates that SSALEO is superior to competing methods across three different datasets. Accuracy in feature selection is displayed in Table 34 for various methods. Extracted features from datasets employing other optimizers are tallied in Table 34. The following table shows the results of several statistical methods applied to each dataset in numerical form. When it comes to selecting features, SSALEO has the highest average accuracy. When compared to alternative methods, SSALEO yields the highest feature selection accuracy for these datasets. The new approach may be to blame since it strikes a better balance between discovery and exploitation.

The effectiveness of the suggested method is compared to that of other preceding algorithms using the Wilcoxon rank-sum test. Table 35 presents the p-values, with highlighted values denoting p-values larger than 0.05. As a result, SSALEO produces statistically significant results compared to other methods, and the null hypothesis is rejected for most functions. The significance of a difference between two or more sample methods is evaluated using the Friedman test's findings. The results of the Friedman tests on each dataset are shown in Table 35. They show that SSALEO performs better and achieves higher rankings than other algorithms.

IX. CONCLUSION AND FUTURE WORK

The Salp swarm algorithm (SSA) and local escape operator (LEO) are combined in this study. The SSALEO proposed is an innovative search strategy that can deal with a wide range of population issues, including an imbalance between exploitation and exploration and premature convergence of the SSA algorithm. The LEO strategy employed in SSALEO helps to reduce search deflation in SSA while also improving the convergence rate of swarm agents and the effectiveness of local search. The method that was suggested was primarily developed for use with large-scale problems involving global optimization. The SSALEO system was initially assessed by completing a series of standard benchmark tasks (29 CEC 2017 test suites with 50 and 100 decision variables and seven CEC2008 lsgo test suites with 200, 500, and 1000 decision variables). Following this, the SSALEO was put through its paces by competing against a benchmark set of seven well-known constrained design tasks from the CEC 2020 conference. These challenges were taken from a variety of engineering specialties.

Comparisons were made between the novel method, traditional SSA, and other cutting-edge metaheuristics. The proposed method outperforms the original SSA and several different algorithms in terms of convergence speed and the ability to break free from local optima. In addition, the proposed method achieves superior results in terms of solution efficiency in comparison to earlier metaheuristic algorithms. According to the findings, the suggested SSALEO algorithm is superior, and the method can be tested further on other problems that are relevant to the real world in subsequent research.

REFERENCES

[1] X.-S. Yang, "Nature-inspired optimization algorithms: Challenges and open problems," *J. Comput. Sci.*, vol. 46, Oct. 2020, Art. no. 101104, doi: 10.1016/j.jocs.2020.101104.

[2] A. Boubezoul and S. Paris, "Application of global optimization methods to model and feature selection," *Pattern Recognit.*, vol. 45, no. 10, pp. 3676–3686, Oct. 2012, doi: 10.1016/j.patcog.2012.04.015.

[3] B. Bharani and A. P. Prakash, "Optimization of a multiproduct economic production quantity problem with stochastic constraints using sequential quadratic programming," *Global J. Pure Appl. Math.*, vol. 12, no. 1, pp. 170–175, 2016. Accessed: Oct. 31, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705115001392

[4] H. R. Lourenço, O. C. Martin, and T. Stützle, "Iterated local search: Framework and applications," in *Handbook of Metaheuristics* (International Series in Operations Research & Management Science), vol. 272. New York, NY, USA: Springer, 2019, pp. 129–168.

[5] X. S. Yang and X. He, "Nature-inspired optimization algorithms in engineering: Overview and applications," in *Nature-Inspired Computation in Engineering* (Studies in Computational Intelligence), vol. 637. Cham, Switzerland: Springer, 2016, pp. 1–20.

[6] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983, doi: 10.1126/science.220.4598.671.

[7] F. Glover, "Tabu search—Part I," *ORSA J. Comput.*, vol. 1, no. 3, pp. 190–206, Aug. 1989, doi: 10.1287/ijoc.1.3.190.

[8] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE ICNN*, vol. 4, Nov./Dec. 1995, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.

[9] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *J. Glob. Optim.*, vol. 39, no. 3, pp. 459–471, 2007, doi: 10.1007/s10898-007-9149-x.

[10] A. Colorni, M. Dorigo, and V. Maniezzo. (1991). *Distributed Optimization by Ant Colonies*. Accessed: May 14, 2021. [Online]. Available: https://faculty.washington.edu/paymana/swarm/colorni92-ecal.pdf

[11] N. Kushwaha, M. Pant, S. Kant, and V. K. Jain, "Magnetic optimization algorithm for data clustering," *Pattern Recognit. Lett.*, vol. 115, pp. 59–65, Nov. 2018, doi: 10.1016/J.PATREC.2017.10.031.

[12] R. Storn and K. Price, "Minimizing the real functions of the ICEC'96 contest by differential evolution," in *Proc. IEEE Int. Conf. Evol. Comput.*, May 1996, pp. 842–844, doi: 10.1109/ICEC.1996.542711.

[13] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Mach. Learn.*, vol. 3, nos. 2–3, pp. 95–99, 1988, doi: 10.1023/A:1022602019183.

[14] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[15] R. Cheng and Y. Jin, "A competitive swarm optimizer for large scale optimization," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 191–204, Feb. 2015, doi: 10.1109/TCYB.2014.2322602.

[16] X.-S. S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)* (Studies in Computational Intelligence), vol. 284. Berlin, Germany: Springer, 2010, pp. 65–74, doi: 10.1007/978-3-642-12538-6_6.

[17] J.-A. Mejía-de-Dios and E. Mezura-Montes, "A new evolutionary optimization method based on center of mass," in *Decision Science in Action*. Singapore: Springer, 2019, pp. 65–74, doi: 10.1007/978-981-13-0860-4_6.

[18] H. Abedinpourshotorban, S. M. Shamsuddin, Z. Beheshti, and D. N. A. Jawawi, "Electromagnetic field optimization: A physics-inspired metaheuristic optimization algorithm," *Swarm Evol. Comput.*, vol. 26, pp. 8–22, Feb. 2016, doi: 10.1016/J.SWEVO.2015.07.002.

[19] A. Naik, S. C. Satapathy, A. S. Ashour, and N. Dey, "Social group optimization for global optimization of multimodal functions and data clustering problems," *Neural Comput. Appl.*, vol. 30, no. 1, pp. 271–287, Nov. 2016, doi: 10.1007/S00521-016-2686-9.

[20] B. Alatas, "ACROA: Artificial chemical reaction optimization algorithm for global optimization," *Expert Syst. Appl.*, vol. 38, no. 10, pp. 13170–13180, May 2011, doi: 10.1016/J.ESWA.2011.04.126.

[21] E. V. Altay and B. Alatas, "Randomness as source for inspiring solution search methods: Music based approaches," *Phys. A, Stat. Mech. Appl.*, vol. 537, Jan. 2020, Art. no. 122650, doi: 10.1016/J.PHYSA.2019.122650.

[22] H. Bingol and B. Alatas, "Chaotic league championship algorithms," *Arabian J. Sci. Eng.*, vol. 41, no. 12, pp. 5123–5147, May 2016, doi: 10.1007/S13369-016-2200-9.

[23] H. Faris, M. M. Mafarja, A. A. Heidari, I. Aljarah, A.-Z. Ala'M, S. Mirjalili, and H. Fujita, "An efficient binary Salp swarm algorithm with crossover scheme for feature selection problems," *Knowl.-Based Syst.*, vol. 154, pp. 43–67, Aug. 2018, doi: 10.1016/j.knosys.2018.05.009.

[24] Q. Tu, X. Chen, and X. Liu, "Multi-strategy ensemble grey wolf optimizer and its application to feature selection," *Appl. Soft Comput.*, vol. 76, pp. 16–30, Mar. 2019, doi: 10.1016/j.asoc.2018.11.047.

[25] A. Mostafa, E. H. Houssein, M. Houseni, A. E. Hassanien, and H. Hefny, "Evaluating swarm optimization algorithms for segmentation of liver images," in *Advances in Soft Computing and Machine Learning in Image Processing* (Studies in Computational Intelligence), vol. 730. Cham, Switzerland: Springer, 2018, pp. 41–62, doi: 10.1007/978-3-319-63754-9_3.

[26] F. A. Hashim, E. H. Houssein, K. Hussain, M. S. Mabrouk, and W. Al-Atabany, "A modified Henry gas solubility optimization for solving motif discovery problem," *Neural Comput. Appl.*, vol. 32, no. 14, pp. 10759–10771, Jul. 2020, doi: 10.1007/s00521-019-04611-0.

[27] E. H. Houssein, M. E. Hosney, D. Oliva, W. M. Mohamed, and M. Hassaballah, "A novel hybrid Harris hawks optimization and support vector machines for drug design and discovery," *Comput. Chem. Eng.*, vol. 133, Feb. 2020, Art. no. 106656, doi: 10.1016/j.compchemeng.2019.106656.

[28] M. Tubishat, S. Ja'afar, M. Alswaitti, S. Mirjalili, N. Idris, M. A. Ismail, and M. S. Omar, "Dynamic Salp swarm algorithm for feature selection," *Expert Syst. Appl.*, vol. 164, Feb. 2021, Art. no. 113873, doi: 10.1016/j.eswa.2020.113873.

[29] B. Niu and L. Li, "A novel PSO-DE-based hybrid algorithm for global optimization," in *Proc. Int. Conf. Intell. Comput.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 5227, 2008, pp. 156–163, doi: 10.1007/978-3-540-85984-0_20.

[30] R. Salgotra, U. Singh, S. Singh, G. Singh, and N. Mittal, "Self-adaptive Salp swarm algorithm for engineering optimization problems," *Appl. Math. Model.*, vol. 89, pp. 188–207, Jan. 2021, doi: 10.1016/j.apm.2020.08.014.

[31] L. Abualigah, B. Alsalibi, M. Shehab, M. Alshinwan, A. M. Khasawneh, and H. Alabool, "A parallel hybrid krill herd algorithm for feature selection," *Int. J. Mach. Learn. Cybern.*, vol. 12, no. 3, pp. 783–806, Mar. 2021, doi: 10.1007/s13042-020-01202-7.

[32] E. H. Houssein, M. R. Saad, K. Hussain, W. Zhu, H. Shaban, and M. Hassaballah, "Optimal sink node placement in large scale wireless sensor networks based on Harris' hawk optimization algorithm," *IEEE Access*, vol. 8, pp. 19381–19397, 2020, doi: 10.1109/ACCESS.2020.2968981.

[33] B. Morales-Castañeda, D. Zaldívar, E. Cuevas, F. Fausto, and A. Rodríguez, "A better balance in metaheuristic algorithms: Does it exist?" *Swarm Evol. Comput.*, vol. 54, May 2020, Art. no. 100671, doi: 10.1016/j.swevo.2020.100671.

[34] M. Črepinšek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: A survey," *ACM Comput. Surv.*, vol. 45, no. 3, pp. 1–33, Jun. 2013, doi: 10.1145/2480741.2480752.

[35] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Adv. Eng. Softw.*, vol. 114, pp. 163–191, Dec. 2017, doi: 10.1016/j.advengsoft.2017.07.002.

[36] M. ElGayyar, E. Emary, N. H. Sweilam, and M. Abdelazeem, "A hybrid Grey Wolf-bat algorithm for global optimization," in *Proc. Int. Conf. Adv. Mach. Learn. Technol. Appl.* (Advances in Intelligent Systems and Computing), vol. 723, 2018, pp. 3–12, doi: 10.1007/978-3-319-74690-6_1.

[37] E. Cuevas, A. Echavarría, and M. A. Ramírez-Ortegón, "An optimization algorithm inspired by the states of matter that improves the balance between exploration and exploitation," *Appl. Intell.*, vol. 40, no. 2, pp. 256–272, 2014, doi: 10.1007/s10489-013-0458-0.

[38] L. Abualigah, M. Shehab, M. Alshinwan, and H. Alabool, "Salp swarm algorithm: A comprehensive survey," *Neural Comput. Appl.*, vol. 32, no. 15, pp. 11195–11215, Aug. 2020, doi: 10.1007/s00521-019-04629-4.

[39] I. Ahmadianfar, O. Bozorg-Haddad, and X. Chu, "Gradient-based optimizer: A new metaheuristic optimization algorithm," *Inf. Sci.*, vol. 540, pp. 131–159, Nov. 2020, doi: 10.1016/j.ins.2020.06.037.

[40] S. Mahdavi, M. E. Shiri, and S. Rahnamayan, "Metaheuristics in large-scale global continues optimization: A survey," *Inf. Sci.*, vol. 295, pp. 407–428, Feb. 2015, doi: 10.1016/j.ins.2014.10.042.

[41] M. A. Potter and K. A. Jong, "A cooperative coevolutionary approach to function optimization," in *Parallel Problem Solving From Nature—PPSN III* (Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 866. Cham, Switzerland: Springer, 1994, pp. 249–257.

[42] Z. Yang, K. Tang, and X. Yao, "Multilevel cooperative coevolution for large scale optimization," in *Proc. IEEE Congr. Evol. Comput., IEEE World Congr. Comput. Intell.*, Jun. 2008, pp. 1663–1670, doi: 10.1109/CEC.2008.4631014.

[43] Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Inf. Sci.*, vol. 178, no. 15, pp. 2985–2999, 2008, doi: 10.1016/j.ins.2008.02.017.

[44] Z. Yang, K. Tang, and X. Yao, "Differential evolution for high-dimensional function optimization," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2007, pp. 3523–3530, doi: 10.1109/CEC.2007.4424929.

[45] X. Li and X. Yao, "Cooperatively coevolving particle swarms for large scale optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 2, pp. 210–224, Apr. 2012, doi: 10.1109/TEVC.2011.2112662.

[46] J. Liu and K. Tang, "Scaling up covariance matrix adaptation evolution strategy using cooperative coevolution," in *Proc. Int. Conf. Intell. Data Eng. Automated Learn.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 8206, 2013, pp. 350–357, doi: 10.1007/978-3-642-41278-3_43.

[47] A. LaTorre, S. Muelas, and J.-M. Peña, "A MOS-based dynamic memetic differential evolution algorithm for continuous optimization: A scalability test," *Soft Comput.*, vol. 15, no. 11, pp. 2187–2199, Nov. 2011, doi: 10.1007/s00500-010-0646-3.

[48] S. Z. Zhao, J. J. Liang, P. N. Suganthan, and M. F. Tasgetiren, "Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization," in *Proc. IEEE Congr. Evol. Comput., IEEE World Congr. Comput. Intell.*, Jun. 2008, pp. 3845–3852, doi: 10.1109/CEC.2008.4631320.

[49] Z. Yang, K. Tang, and X. Yao, "Scalability of generalized adaptive differential evolution for large-scale continuous optimization," *Soft Comput.*, vol. 15, no. 11, pp. 2141–2155, Nov. 2011, doi: 10.1007/s00500-010-0643-6.

[50] J. Brest and M. S. Maučec, "Self-adaptive differential evolution algorithm using population size reduction and three strategies," *Soft Comput.*, vol. 15, no. 11, pp. 2157–2174, Nov. 2011, doi: 10.1007/s00500-010-0644-5.

[51] S.-T. Hsieh, T.-Y. Sun, C.-C. Liu, and S.-J. Tsai, "Solving large scale global optimization using improved particle swarm optimizer," in *Proc. IEEE Congr. Evol. Comput., IEEE World Congr. Comput. Intell.*, Jun. 2008, pp. 1777–1784, doi: 10.1109/CEC.2008.4631030.

[52] P. Mohapatra, K. N. Das, and S. Roy, "A modified competitive swarm optimizer for large scale optimization problems," *Appl. Soft Comput.*, vol. 59, pp. 340–362, Oct. 2017, doi: 10.1016/j.asoc.2017.05.060.

[53] S. T. Hsieh, T. Y. Sun, C. C. Liu, and S. J. Tsai, "Efficient population utilization strategy for particle swarm optimizer," *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 39, no. 2, pp. 444–456, Apr. 2009, doi: 10.1109/TSMCB.2008.2006628.

[54] M. Abdelazeem, M. Gobashy, M. H. Khalil, and M. Abdrabou, "A complete model parameter optimization from self-potential data using whale algorithm," *J. Appl. Geophys.*, vol. 170, Nov. 2019, Art. no. 103825, doi: 10.1016/J.JAPPGEO.2019.103825.

[55] Y. Sun, T. Yang, and Z. Liu, "A whale optimization algorithm based on quadratic interpolation for high-dimensional global optimization problems," *Appl. Soft Comput.*, vol. 85, Dec. 2019, Art. no. 105744, doi: 10.1016/j.asoc.2019.105744.

[56] A. Cano and C. Garcia-Martinez, "100 million dimensions large-scale global optimization using distributed GPU computing," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2016, pp. 3566–3573, doi: 10.1109/CEC.2016.7744241.

[57] A. Cano, C. García-Martínez, and S. Ventura, "Extremely high-dimensional optimization with MapReduce: Scaling functions and algorithm," *Inf. Sci.*, vols. 415–416, pp. 110–127, Nov. 2017, doi: 10.1016/j.ins.2017.06.024.

[58] M. Lastra, D. Molina, and J. M. Benítez, "A high performance memetic algorithm for extremely high-dimensional problems," *Inf. Sci.*, vol. 293, pp. 35–58, Feb. 2015, doi: 10.1016/j.ins.2014.09.018.

[59] S. Z. Mirjalili, S. Mirjalili, S. Saremi, H. Faris, and I. Aljarah, "Grasshopper optimization algorithm for multi-objective optimization problems," *Appl. Intell.*, vol. 48, no. 4, pp. 805–820, Apr. 2018, doi: 10.1007/s10489-017-1019-8.

[60] Q. Fan, Z. Chen, W. Zhang, and X. Fang, "ESSAWOA: Enhanced whale optimization algorithm integrated with Salp swarm algorithm for global optimization," *Eng. Comput.*, vol. 38, pp. 1–18, Oct. 2020, doi: 10.1007/s00366-020-01189-3.

[61] M. Qaraad, S. Amjad, N. K. Hussein, and M. A. Elhosseini, "Large scale Salp-based grey wolf optimization for feature selection and global optimization," *Neural Comput. Appl.*, vol. 34, no. 11, pp. 8989–9014, Feb. 2022, doi: 10.1007/S00521-022-06921-2.

[62] J. Zhang and J. S. Wang, "Improved Salp swarm algorithm based on Levy flight and sine cosine operator," *IEEE Access*, vol. 8, pp. 99740–99771, 2020, doi: 10.1109/ACCESS.2020.2997783.

[63] A. E. Hegazy, M. A. Makhlouf, and G. S. El-Tawel, "Improved Salp swarm algorithm for feature selection," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 32, no. 3, pp. 335–344, Mar. 2020, doi: 10.1016/j.jksuci.2018.06.003.

[64] I. Aljarah, M. Mafarja, A. A. Heidari, H. Faris, Y. Zhang, and S. Mirjalili, "Asynchronous accelerating multi-leader Salp chains for feature selection," *Appl. Soft Comput.*, vol. 71, pp. 964–979, Oct. 2018, doi: 10.1016/j.asoc.2018.07.040.

[65] T. Chen, M. Wang, X. Huang, and Q. Xie, "TDOA-AOA localization based on improved Salp swarm algorithm," in *Proc. 14th IEEE Int. Conf. Signal Process. (ICSP)*, Aug. 2018, pp. 108–112, doi: 10.1109/ICSP.2018.8652322.

[66] E. H. Houssein, B. E.-D. Helmy, H. Rezk, and A. M. Nassef, "An enhanced archimedes optimization algorithm based on local escaping operator and orthogonal learning for PEM fuel cell parameter identification," *Eng. Appl. Artif. Intell.*, vol. 103, Aug. 2021, Art. no. 104309, doi: 10.1016/j.engappai.2021.104309.

[67] E. H. Houssein, B. E.-D. Helmy, A. A. Elngar, D. S. Abdelminaam, and H. Shaban, "An improved tunicate swarm algorithm for global optimization and image segmentation," *IEEE Access*, vol. 9, pp. 56066–56092, 2021, doi: 10.1109/ACCESS.2021.3072336.

[68] M. Oszust, "Enhanced marine predators algorithm with local escaping operator for global optimization," *Knowl.-Based Syst.*, vol. 232, Nov. 2021, Art. no. 107467, doi: 10.1016/j.knosys.2021.107467.

[69] D. Wang, Y. Zhou, S. Jiang, and X. Liu, "A simplex method-based Salp swarm algorithm for numerical and engineering optimization," *IFIP Adv. Inf. Commun. Technol.*, vol. 538, pp. 150–159, Oct. 2018, doi: 10.1007/978-3-030-00828-4_16.

[70] M. Khamees, A. Albakry, and K. Shaker, "Multi-objective feature selection: Hybrid of Salp swarm and simulated annealing approach," in *Proc. Int. Conf. New Trends Inf. Commun. Technol. Appl.*, in Communications in Computer and Information Science, vol. 938, 2018, pp. 129–142, doi: 10.1007/978-3-030-01653-1_8.

[71] S. Asaithambi and M. Rajappa, "Swarm intelligence-based approach for optimal design of CMOS differential amplifier and comparator circuit using a hybrid Salp swarm algorithm," *Rev. Sci. Instrum.*, vol. 89, no. 5, May 2018, Art. no. 054702, doi: 10.1063/1.5020999.

[72] S. Ahmed, M. Mafarja, H. Faris, and I. Aljarah, "Feature selection using Salp swarm algorithm with chaos," in *Proc. 2nd Int. Conf. Intell. Syst., Metaheuristics Swarm Intell. (ISMSI)*, 2018, pp. 65–69, doi: 10.1145/3206185.3206198.

[73] Y. Meraihi, A. Ramdane-Cherif, M. Mahseur, and D. Achelia, "A chaotic binary Salp swarm algorithm for solving the graph coloring problem," in *Proc. Int. Symp. Modelling Implement. Complex Syst.*, in Lecture Notes in Networks and Systems, vol. 64, Dec. 2019, pp. 106–118, doi: 10.1007/978-3-030-05481-6_8.

[74] R. A. Ibrahim, A. A. Ewees, D. Oliva, M. A. Elaziz, and S. Lu, "Improved Salp swarm algorithm based on particle swarm optimization for feature selection," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 8, pp. 3155–3169, Aug. 2019, doi: 10.1007/s12652-018-1031-9.

[75] B. Yang, L. Zhong, X. Zhang, H. Shu, T. Yu, H. Li, L. Jiang, and L. Sun, "Novel bio-inspired memetic Salp swarm algorithm and application to MPPT for PV systems considering partial shading condition," *J. Cleaner Prod.*, vol. 215, pp. 1203–1222, Apr. 2019, doi: 10.1016/j.jclepro.2019.01.150.

[76] N. Singh, L. H. Son, F. Chiclana, and J.-P. Magnot, "A new fusion of Salp swarm with sine cosine for optimization of non-linear functions," *Eng. Comput.*, vol. 36, no. 1, pp. 185–212, Jan. 2020, doi: 10.1007/s00366-018-00696-8.

[77] A. A. Ateya, A. Muthanna, A. Vybornova, A. D. Algarni, A. Abuarqoub, Y. Koucheryavy, and A. Koucheryavy, "Chaotic Salp swarm algorithm for SDN multi-controller networks," *Eng. Sci. Technol., Int. J.*, vol. 22, no. 4, pp. 1001–1012, Aug. 2019, doi: 10.1016/j.jestch.2018.12.015.

[78] N. Panda and S. K. Majhi, "Improved Salp swarm algorithm with space transformation search for training neural network," *Arabian J. Sci. Eng.*, vol. 45, no. 4, pp. 2743–2761, Apr. 2020, doi: 10.1007/s13369-019-04132-x.

[79] M. Tubishat, N. Idris, L. Shuib, M. A. M. Abushariah, and S. Mirjalili, "Improved Salp swarm algorithm based on opposition based learning and novel local search algorithm for feature selection," *Expert Syst. Appl.*, vol. 145, May 2020, Art. no. 113122, doi: 10.1016/j.eswa.2019.113122.

[80] H. Faris, A. A. Heidari, A. M. Al-Zoubi, M. Mafarja, I. Aljarah, M. Eshtay, and S. Mirjalili, "Time-varying hierarchical chains of Salps with random weight networks for feature selection," *Expert Syst. Appl.*, vol. 140, Feb. 2020, Art. no. 112898, doi: 10.1016/j.eswa.2019.112898.

[81] A. A. Ewees, M. A. A. Al-qaness, and M. A. Elaziz, "Enhanced Salp swarm algorithm based on firefly algorithm for unrelated parallel machine scheduling with setup times," *Appl. Math. Model.*, vol. 94, pp. 285–305, Jun. 2021, doi: 10.1016/j.apm.2021.01.017.

[82] M. M. Saafan and E. M. El-Gendy, "IWOSSA: An improved whale optimization Salp swarm algorithm for solving optimization problems," *Expert Syst. Appl.*, vol. 176, Aug. 2021, Art. no. 114901, doi: 10.1016/j.eswa.2021.114901.

[83] M. Qaraad, S. Amjad, N. K. Hussein, and M. A. Elhosseini, "An innovative quadratic interpolation Salp swarm-based local escape operator for large-scale global optimization problems and feature selection," *Neural Comput. Appl.*, vol. 2022, pp. 1–59, Jun. 2022, doi: 10.1007/S00521-022-07391-2.

[84] N. Henschke, J. D. Everett, A. J. Richardson, and I. M. Suthers, "Rethinking the role of Salps in the ocean," *Trends Ecol. Evol.*, vol. 31, no. 9, pp. 720–733, Sep. 2016, doi: 10.1016/j.tree.2016.06.007.

[85] E.-G. Talbi, "A taxonomy of hybrid metaheuristics," *J. Heuristics*, vol. 8, no. 5, pp. 541–564, Jan. 2002, doi: 10.1023/A:1016540724870.

[86] R. Mallipeddi and P. Suganthan, "Problem definitions and evaluation criteria for the CEC 2010 competition on constrained real-parameter optimization," Nanyang Technol. Univ., Singapore, Tech. Rep., 24, Dec. 2013, Accessed: Aug. 8, 2021. [Online]. Available: http://www3.ntu.edu.sg/home/epnsugan/index_files/CEC10-Const/TR-April-2010.pdf

[87] X. Li, K. Tang, M. N. Omidvar, Z. Yang, K. Qin, and H. China, "Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization," Univ. Sci. Technol., Beijing, China, 2010, pp. 1–21. Accessed: Dec. 12, 2021. [Online]. Available: http://goanna.cs.rmit.edu.au/~xiaodong/cec13-lsgo/competition/cec2013-lsgo-benchmark-tech-report.pdf

[88] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016, doi: 10.1016/j.knosys.2015.12.022.

[89] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, Feb. 2016, doi: 10.1016/j.advengsoft.2016.01.008.

[90] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015, doi: 10.1016/j.knosys.2015.07.006.

[91] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019, doi: 10.1016/j.future.2019.02.028.

[92] S. Gupta and K. Deep, "A novel random walk Grey Wolf optimizer," *Swarm Evol. Comput.*, vol. 44, pp. 101–112, Feb. 2019, doi: 10.1016/j.swevo.2018.01.001.

[93] C. Tang, W. Sun, W. Wu, and M. Xue, "A hybrid improved whale optimization algorithm," in *Proc. IEEE 15th Int. Conf. Control Autom. (ICCA)*, Jul. 2019, pp. 362–367, doi: 10.1109/ICCA.2019.8900003.

[94] G. Iacca, V. C. dos Santos Junior, and V. V. de Melo, "An improved Jaya optimization algorithm with Lévy flight," *Expert Syst. Appl.*, vol. 165, Mar. 2021, Art. no. 113902, doi: 10.1016/j.eswa.2020.113902.

[95] B. Liu, L. Wang, Y.-H. Jin, F. Tang, and D. X. Huang, "Improved particle swarm optimization combined with chaos," *Chaos, Solitons Fractals*, vol. 25, no. 5, pp. 1261–1271, Sep. 2005, doi: 10.1016/j.chaos.2004.11.095.

[96] Y. Fan, P. Wang, A. A. Heidari, M. Wang, X. Zhao, H. Chen, and C. Li, "Boosted hunting-based fruit fly optimization and advances in real-world problems," *Expert Syst. Appl.*, vol. 159, Nov. 2020, Art. no. 113502, doi: 10.1016/j.eswa.2020.113502.

[97] G. Singh, U. Singh, and R. Salgotra, "Effect of parametric enhancements on naked mole-rat algorithm for global optimization," *Eng. Comput.*, vol. 38, pp. 1–29, Mar. 2021, doi: 10.1007/s00366-021-01344-4.

[98] J. J. Liang, A. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281–295, Jul. 2006, doi: 10.1109/TEVC.2005.857610.

[99] M. Ghasemi, E. Akbari, A. Rahimnejad, S. E. Razavi, S. Ghavidel, and L. Li, "Phasor particle swarm optimization: A simple and efficient variant of PSO," *Soft Comput.*, vol. 23, no. 19, pp. 9701–9718, Oct. 2019, doi: 10.1007/s00500-018-3536-8.

[100] M. Ghasemi, J. Aghaei, and M. Hadipour, "New self-organising hierarchical PSO with jumping time-varying acceleration coefficients," *Electron. Lett.*, vol. 53, no. 20, pp. 1360–1362, Sep. 2017, doi: 10.1049/el.2017.2112.

[101] M. H. Nadimi-Shahraki, S. Taghian, S. Mirjalili, and H. Faris, "MTDE: An effective multi-trial vector-based differential evolution algorithm and its applications for engineering design problems," *Appl. Soft Comput.*, vol. 97, Dec. 2020, Art. no. 106761, doi: 10.1016/j.asoc.2020.106761.

[102] E. G. Mahadevan, *Ammonium Nitrate Explosives for Civil Applications: Slurries, Emulsions and Ammonium Nitrate Fuel Oils*. Hoboken, NJ, USA: Wiley, 2009.

[103] F. Wilcoxon, "Individual comparisons by ranking methods," in *Breakthroughs in Statistics*. New York, NY, USA: Springer, 1992, pp. 196–202.

[104] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Dec. 2006. Accessed: Jan. 28, 2022. [Online]. Available: https://www.jmlr.org/papers/volume7/demsar06a/demsar06a.pdf

[105] M. H. Qais, H. M. Hasanien, and S. Alghuwainem, "Enhanced Salp swarm algorithm: Application to variable speed wind generators," *Eng. Appl. Artif. Intell.*, vol. 80, pp. 82–96, Apr. 2019, doi: 10.1016/j.engappai.2019.01.011.

[106] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)," *Evol. Comput.*, vol. 11, no. 1, pp. 1–18, Mar. 2003, doi: 10.1162/106365603321828970.

[107] I. Loshchilov, "LM-CMA: An alternative to L-BFGS for large-scale black box optimization," *Evol. Comput.*, vol. 25, no. 1, pp. 143–171, Mar. 2017, doi: 10.1162/EVCO_a_00168.

[108] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 71–78, doi: 10.1109/CEC.2013.6557555.

[109] J. Teo, "Exploring dynamic self-adaptive populations in differential evolution," *Soft Comput.*, vol. 10, no. 8, pp. 673–686, 2006, doi: 10.1007/s00500-005-0537-1.

[110] Y. Li, Y. Zhao, and J. Liu, "Dynamic sine cosine algorithm for large-scale global optimization problems," *Expert Syst. Appl.*, vol. 177, Sep. 2021, Art. no. 114950, doi: 10.1016/j.eswa.2021.114950.

[111] M. R. Ghasemi and H. Varaee, "A fast multi-objective optimization using an efficient ideal gas molecular movement algorithm," *Eng. Comput.*, vol. 33, no. 3, pp. 477–496, Jul. 2017, doi: 10.1007/s00366-016-0485-7.

[112] W. He, S. A. Bagherzadeh, M. Tahmasebi, A. Abdollahi, M. Bahrami, R. Moradi, A. Karimipour, M. Goodarzi, and Q.-V. Bach, "A new method of black-box fuzzy system identification optimized by genetic algorithm and its application to predict mixture thermal properties," *Int. J. Numer. Methods Heat Fluid Flow*, vol. 30, no. 5, pp. 2485–2499, Apr. 2020, doi: 10.1108/HFF-12-2018-0758.

[113] Z. Li, H. Shahrajabian, S. A. Bagherzadeh, H. Jadidi, A. Karimipour, and I. Tlili, "Effects of nano-clay content, foaming temperature and foaming time on density and cell size of PVC matrix foam by presented least absolute shrinkage and selection operator statistical regression via suitable experiments as a function of MMT content," *Phys. A, Stat. Mech. Appl.*, vol. 537, Jan. 2020, Art. no. 122637, doi: 10.1016/j.physa.2019.122637.

[114] P. Vasant, *Meta-Heuristics Optimization Algorithms in Engineering, Business, Economics, and Finance*. IGI Global, 2012, pp. 1–708, doi: 10.4018/978-1-4666-2086-5.

[115] A. Kumar, G. Wu, M. Z. Ali, R. Mallipeddi, P. N. Suganthan, and S. Das, "A test-suite of non-convex constrained optimization problems from the real-world and some baseline results," *Swarm Evol. Comput.*, vol. 56, Aug. 2020, Art. no. 100693, doi: 10.1016/j.swevo.2020.100693.

[116] A. K. Trivedi, L. Arya, S. K. Verma, R. K. Tyagi, and A. Hemantaranjan, "Corrigendum: Evaluation of barnyard millet diversity in central Himalayan region for environmental stress tolerance," *J. Agricult. Sci.*, vol. 155, pp. 1497–1507, Dec. 2017, doi: 10.1017/S0021859617000545.

[117] A. K. Trivedi, L. Arya, S. K. Verma, R. K. Tyagi, and A. Hemantaranjan, "Evaluation of barnyard millet diversity in central Himalayan region for environmental stress tolerance—Corrigendum," *J. Agric. Sci.*, vol. 156, no. 1, p. 2, Jan. 2018, doi: 10.1017/S0021859617000922.

[118] M. Hellwig and H.-G. Beyer, "A matrix adaptation evolution strategy for constrained real-parameter optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2018, pp. 1–8, doi: 10.1109/CEC.2018.8477950.

[119] Z. Fan, Y. Fang, W. Li, Y. Yuan, Z. Wang, and X. Bian, "LSHADE44 with an improved ϵ constraint-handling method for solving constrained single-objective optimization problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2018, pp. 1–8, doi: 10.1109/CEC.2018.8477943.

[120] J. Gurrola-Ramos, A. Hernandez-Aguirre, and O. Dalmau-Cedeno, "COLSHADE for real-world single-objective constrained optimization problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2020, pp. 1–8, doi: 10.1109/CEC48606.2020.9185583.

[121] M. Hellwig and H.-G. Beyer, "A modified matrix adaptation evolution strategy with restarts for constrained real-world problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2020, pp. 1–8, doi: 10.1109/CEC48606.2020.9185566.

[122] A. Kumar, S. Das, and I. Zelinka, "A self-adaptive spherical search algorithm for real-world constrained optimization problems," in *Proc. Genetic Evol. Comput. Conf. Companion*, Jul. 2020, pp. 13–14, doi: 10.1145/3377929.3398186.

[123] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 4, pp. 491–502, Apr. 2005, doi: 10.1109/TKDE.2005.66.

[124] V. R. Balasaraswathi, M. Sugumaran, and Y. Hamid, "Feature selection techniques for intrusion detection using non-bio-inspired and bio-inspired optimization algorithms," *J. Commun. Inf. Netw.*, vol. 2, no. 4, pp. 107–119, Dec. 2017, doi: 10.1007/s41650-017-0033-7.

[125] M. Mafarja and S. Abdullah, "A fuzzy record-to-record travel algorithm for solving rough set attribute reduction," *Int. J. Syst. Sci.*, vol. 46, no. 3, pp. 503–512, Feb. 2015, doi: 10.1080/00207721.2013.791000.

[126] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, nos. 1–2, pp. 273–324, Dec. 1997, doi: 10.1016/s0004-3702(97)00043-x.

[127] M. M. Kabir, M. Shahjahan, and K. Murase, "A new local search based hybrid genetic algorithm for feature selection," *Neurocomputing*, vol. 74, no. 17, pp. 2914–2928, Oct. 2011, doi: 10.1016/j.neucom.2011.03.034.

[128] H. Chen, W. Jiang, C. Li, and R. Li, "A heuristic feature selection approach for text categorization by using chaos optimization and genetic algorithm," *Math. Problems Eng.*, vol. 2013, pp. 1–6, Nov. 2013, doi: 10.1155/2013/524017.

[129] A. Berchuck *et al.*, "Microarray analysis of early stage serous ovarian cancers shows profiles predictive of favorable outcome," *Clin. Cancer Res.*, vol. 15, no. 7, pp. 2448–2455, 2009, doi: 10.1158/1078-0432.CCR-08-2430.

[130] A. Berchuck, E. S. Iversen, J. Luo, J. P. Clarke, H. Horne, D. A. Levine, J. Boyd, M. A. Alonso, A. A. Secord, M. Q. Bernardini, J. C. Barnett, T. Boren, S. K. Murphy, H. K. Dressman, J. R. Marks, and J. M. Lancaster, "Microarray analysis of early stage serous ovarian cancers shows profiles predictive of favorable outcome," *Clin. Cancer Res.*, vol. 15, no. 7, pp. 2448–2455, 2009, doi: 10.1158/1078-0432.CCR-08-2430.

[131] A. Frank and A. Asuncion. (2010). *UCI Machine Learning Repository*. Accessed: Apr. 2, 2021. [Online]. Available: http://archive.ics.uci.edu/ml

[132] S. Mirjalili and A. Lewis, "S-shaped versus V-shaped transfer functions for binary particle swarm optimization," *Swarm Evol. Comput.*, vol. 9, pp. 1–14, Apr. 2013, doi: 10.1016/j.swevo.2012.09.002.

[133] M. Abdel-Basset, D. El-Shahat, I. El-henawy, V. H. C. de Albuquerque, and S. Mirjalili, "A new fusion of Grey Wolf optimizer algorithm with a two-phase mutation for feature selection," *Expert Syst. Appl.*, vol. 139, Jan. 2020, Art. no. 112824, doi: 10.1016/j.eswa.2019.112824.

[134] S. Gupta, K. Deep, H. Moayedi, L. K. Foong, and A. Assad, "Sine cosine Grey Wolf optimizer to solve engineering design problems," *Eng. Comput.*, vol. 37, no. 4, pp. 3123–3149, Oct. 2021, doi: 10.1007/s00366-020-00996-y.

[135] Y. Belkourchia, L. Azrar, and E.-S.-M. Zeriab, "A hybrid optimization algorithm for solving constrained engineering design problems," in *Proc. 5th Int. Conf. Optim. Appl. (ICOA)*, Apr. 2019, pp. 1–7, doi: 10.1109/ICOA.2019.8727654.

[136] H. Garg, "Solving structural engineering design optimization problems using an artificial bee colony algorithm," *J. Ind. Manage. Optim.*, vol. 10, no. 3, pp. 777–794, Nov. 2014, doi: 10.3934/jimo.2014.10.777.

[137] S. Kaur, L. K. Awasthi, and A. L. Sangal, "HMOSHSSA: A hybrid meta-heuristic approach for solving constrained optimization problems," *Eng. Comput.*, vol. 37, no. 4, pp. 3167–3203, Oct. 2021, doi: 10.1007/s00366-020-00989-x.

[138] M. Dong, N. Wang, X. Cheng, and C. Jiang, "Composite differential evolution with modified Oracle penalty method for constrained optimization problems," *Math. Problems Eng.*, vol. 2014, pp. 1–15, Jan. 2014, doi: 10.1155/2014/617905.

[139] R. V. Rao and G. G. Waghmare, "Complex constrained design optimisation using an elitist teaching-learning-based optimisation algorithm," *Int. J. Metaheuristics*, vol. 3, no. 1, p. 81, 2014, doi: 10.1504/ijmheur.2014.058863.

MOHAMMED QARAAD received the B.Sc. degree from the Department of Computer Science, Mutah University, Jordan, in 2006, and the M.Sc. degree from the Department of Computer Science, University Abdelmalek Essaadi, in 2018, where he is currently pursuing the Ph.D. degree. His research interests include artificial intelligence (AI), machine learning, deep learning, non-invasive computer-assisted diagnosis systems, optimizing features selection techniques, classification high-dimension data, computer vision, and bio-inspired optimization techniques. He is also fascinated with the use of artificial intelligence to image processing, access control, robotics, unmanned aerial vehicles, and bioinformatics.

SOUAD AMJAD received the B.Sc. degree from the Faculty of Science, Cadi Ayyad University, Semlalia, Morocco, the M.Sc. degree from the Department of Science, Cadi Ayyad University, and the Ph.D. degree from the Faculty of Science, Cadi Ayyad University. He is currently an Assistant Professor with the College of Computer Science, University Abdelmalek Essaadi, Tétouan. Artificial Intelligence (AI), machine learning, deep learning, non-invasive computer-assisted diagnosis methods, and feature optimization are all terms used to describe AI. Among the key research, fields are selection strategies, categorization of high-dimensional data, and computer vision. He is also fascinated by bio-inspired optimization methods. He is also interested in how artificial intelligence can be used in image processing, access control, robotics, unmanned aerial vehicles, and bioinformatics.

NAZAR K. HUSSEIN received the B.Sc. degree from the Department of Mathematics, College of Computer Science and Mathematics, University of Mosul, Iraq, the M.Sc. degree from the Department of Mathematics, College of Education, Tikrit University, Tikrit, Iraq, and the Ph.D. degree in computational mathematics from the College of Computer Science and Mathematics, University of Mosul.

He is currently an Assistant Professor at the College of Computer Science and Mathematics, Tikrit University. His main research interests include numerical optimization, bio-inspired algorithm-based optimization, artificial intelligence (AI), machine learning, fuzzy logic, and neural networks. Moreover, he is also concerned with applying optimization in solving engineering and medical problems. He served as a member of the editorial board of local journals in Iraq.

SEYEDALI MIRJALILI (Senior Member, IEEE) is currently a Professor at the Center for Artificial Intelligence Research and Optimization, Torrens University, and is internationally recognized for his advances in nature-inspired artificial intelligence (AI) techniques. He is the author of more than 300 publications, including five books, 250 journal articles, 20 conference papers, and 30 book chapters. With over 40,000 citations and an H-index of 70, he is one of the most influential AI researchers in the world. From Google Scholar metrics, he is globally the most cited researcher in optimization using AI techniques, which is his main area of expertise. Since 2019, he has been is in the list of 1% highly-cited researchers and named as one of the most influential researchers in the world by Web of Science. In 2021, The Australian newspaper named him as the top researcher in Australia in three fields: artificial intelligence, evolutionary computation, and fuzzy systems. He is serving as an Associate Editor of leading AI journals, including *Neurocomputing*, *Applied Soft Computing*, *Advances in Engineering Software*, *Computers in Biology and Medicine*, *Healthcare Analytics*, *Applied Intelligence*, and IEEE Access.

NADHIR BEN HALIMA received the B.Sc. degree in computer engineering from the National School of Computer Sciences (ENSI), Manouba, Tunisia, the M.Sc. degree in communication networks engineering from the Sant'Anna School of Advanced Studies, Pisa, Italy, and the Ph.D. degree in information and communication technologies from the University of Trento, Trento, Italy. In 2009, he was a Visiting Researcher at the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC, USA. He is currently an Associate Professor at the Mediterranean Institute of Technology, South Mediterranean University, Tunis, Tunisia, and a Cybersecurity and Networking Expert. He is an educational leader and a consultant with more than 15 years of experience in higher education.

MOSTAFA A. ELHOSSEINI (Member, IEEE) received the B.Sc. degree from the Electronics Engineering Department, Mansoura University, Egypt, and the M.Sc. and Ph.D. degrees from the Computers and Systems Engineering Department, Mansoura University.

He is currently a Professor at the College of Computer Science and Engineering, Taibah University, Yanbu, Madinah, Saudi Arabia. He is the author of more than 70 journal articles. His main research interests include artificial intelligence (AI), machine learning and deep learning, non-invasive computer-assisted diagnosis systems, and computer vision. On top of that, he is interested in bio-inspired optimization algorithms. He is also interested in applying artificial intelligence in image processing, access control, robotics, unmanned aerial vehicle, and bioinformatics. He served as a member of the international program committees for numerous international conferences and journals.