Al-Rafidain Journal of Computer Sciences and Mathematics (RJCM)

www.csmj.mosuljournals.com

A Modified Sine Cosine Algorithm Based on a Novel Locally Weighted Method for Global Optimization Problems

Shahad Ibrahim Mohammed [1], *, Nazar K. Hussein[2] and Mohammed Abdulrazaq Alkahya[3]

[1] Department of Mathematics, College of Computer Sciences and mathematics, Tikrit University, Iraq
[2]Department of Mathematics, College of Computer Sciences and mathematics, Tikrit University
[3]College of Education for Pure Sciences, University of Mosul, Mosul, Iraq[3]
Email: Shahad.i.mohammed35380@st.tu.edu.iq

Article information

Article history:
Received: 31/7/2023
Accepted: 5/11/2023
Available online: 25/6/2024

Abstract

The Sine Cosine Algorithm (SCA), a recently discovered population-based optimization technique, is used to resolve optimization issues. In this research, the study proposes employing the LWSCA (Locally Weighted Sine Cosine Algorithm) as a hybrid approach to enhance the performance of the original SCA (Sine Cosine Algorithm) and mitigate its limitations. These limitations encompass restricted resolution, slow convergence rates, and difficulties in achieving global optimization when dealing with complex, multi-dimensional spaces. The fundamental idea underlying LWSCA is to incorporate the SCA algorithm with the locally weighted (LW) technique and mutation diagram. The hybridization process has two stages: An algorithm is initially changed by altering the fundamental equations to ensure greater effectiveness and accuracy. The second point is that when the LW local approach is used to create a new dependent site, it increases the randomness during the search process. This, in turn, raises the population variance of the optimizer being proposed, ultimately enhancing the overall effectiveness of the global search. The putative method's hybrid architecture is anticipated to significantly increase the potential for exploration and exploitation. By evaluating SCA's performance against IEEE CEC 2017 functions and contrasting it with a variety of different metaheuristic techniques, the usefulness of SCA is investigated. According to the experimental data gathered, the LWSCA's convergence, exploration, and exploitation tendencies have all greatly improved. According to the results, the suggested LWSCA method is a good one that performs better than SCA and other rival algorithms in most functions.

Keywords:
Population-based optimization, Sine Cosine optimizer, global Solutions, locally weight and metaheuristic

Correspondence:
Author : Shahad Ibrahim Mohammed
Email:Shahad.i.mohammed35380@st.tu.edu.iq

1. Introduction

Finding the ideal solution is one of the issues of improvement in scientific research that has attracted the attention of many scientists, and it was one of the hot spots in the issue of absolute judgment on the optimal solution. Post intuitive algorithms have become a dominant window modeling and optimization tool. The basic work of any descriptive algorithm is exploration and exploitation. It's worth thinking about because it's simple, doesn't call for difficult stages, avoids the need for localized fine-tuning, uses fewer parameters, and enables thorough global searches [1]. Metaheuristic algorithms based on solutions and descriptive Evolutionary algorithms draw their inspiration from the mechanisms of natural evolution. They employ mutation procedures and crossover operations to retain the most promising individuals while discarding the least fit ones within a population. This process begins by ensuring that the newly generated population surpasses the previous one, much like how genetic algorithms (GA) operate [2]. Additionally, there exist behavior-inspired algorithms that mimic the intelligent group dynamics found in biological swarms. Examples of such algorithms include the Seagull Optimization Algorithm (SOA) [3], the Grasshopper Optimization Algorithm (GOA) [4], the Firefly Optimization Algorithm (FOA) [5], the Gray Wolf Algorithm (GWA) [6], the Ant Lion Optimizer (ALO) [7], the Manta-Ray Foraging Optimization (MRFO) [8], the Artificial Bee Colony (ABC) [9], Differential Evolution (DE) [10], the Salp Swarm Algorithm (SSA) [11], and Particle Swarm Optimization (PSO) [12]. For instance, in PSO, each particle within the swarm represents a potential solution. In PSO, each swarm particle represents a solution. The bird's behavior was simulated, swarming in order to update the population until it exceeded the stopping criterion and else. Algorithms based on the international system usually need fewer parameters and operators. Additionally, information in the search field is preserved for communication across squadron members [13]. As a result, it greatly reduces the algorithm's computational scope and complexity, as well as being superior to evolution-based algorithms and phenomenon-based physics algorithms such as Black Hole (BH) [14], Thermal Exchange Optimization (TEO) [14], Chemical Reaction Optimization (CRO) [15], etc. The optimization of the algorithms also helps in choosing the features of the ideal classifier by selecting the most relevant data set. It is a data mining technique that eliminates redundant and unnecessary data from data sets to make them easier to analyze. With the difficulty of mining and translating a large amount of data into usable data, feature selection (FS) is becoming increasingly important [16]. Feature-selection methods can be largely divided into three categories: filters, wrappers, and embedded [17]. Despite the good advantages of metaheuristic algorithms for dealing with optimization problems, their main problem was reaching optimal local solutions. Due to the poor balance between the exploitation and exploration processes, which reduces the speed of convergence in the search process [18], the design of new optimization models is a challenge for many complex practical applications.

The sine cosine algorithm (SCA) was a brand-new algorithm that Mirjalili unveiled in 2016 [19]. Researchers have been attracted to the SCA algorithm because it is straightforward and requires fewer parameters. The cosine method, like other metaheuristic algorithms, has major drawbacks, such as a slow convergence time and the fact that it is not sufficient to provide accuracy for every solution. In addition, only a few possibilities have been explored. In recent years, a variety of methods and algorithms have been developed to enhance the original SCA and address its drawbacks by taking advantage of the advantages of hybridization methods. A revolutionary hybrid feature selection method known as SCAGA, which combines the Sine Cosine Algorithm (SCA) and genetic algorithm (GA), by researchers under the supervision of Laith Abu Alika [20]. Also, a new approach known as ARSCA has been proposed as an alternative to SCA, aiming to correct the limitations of the original algorithm [21], in addition to a number of other works. In order to advance SCA and encourage a better balance between the exploitation and exploration phases, this work offers a new locally weighted (LW) method. The SCA can enhance its convergence frequency and update the positions of its followers through mutation by using the LW technique. This is the order in which we tackled the project: Below are some of the features that our algorithm offers: We discussed the conversion factor used to enhance the accuracy and potential for convergence of the algorithm. Furthermore, to enhance efficiency and avoid being trapped in local optima, the SCA technique includes a novel approach known as local weighted (LW). These methods help achieve a balance between exploitation and exploration in the SCA algorithm. The suggested approach was recently tested to determine its effectiveness and how it compares to other well-known metaheuristics. This was done through the Evolutionary Computational Computing Competition (CEC) benchmarking functions, which included CEC 2017.

2. Principles of SCA

This section provides a comprehensive explanation of the SCA optimization algorithm.

2.1. Sine Cosine Algorithm

An optimization method that acts at the population level is the SCA algorithm. It starts by producing a series of random answers and works its way towards the best one through mathematical calculations involving sine and cosine functions. The algorithm adheres to a straightforward mathematical paradigm, like other meta-optimization methods. A collection of solutions, often referred to as "search agents," are placed at random in the search space to start the search process. The results of these evaluations are then compared with an objective function. The algorithm continues to update the solutions in order to generate new ones while keeping track of the best solution discovered thus far, which serves as its location. The sine and cosine functions, as shown in equations 1 and 2, are frequently used as part of the SCA algorithm's optimization phase. Up until a predetermined maximum number of repetitions, this process is repeated. The following is a description of the mathematical model used to update the positions in the SCA algorithm [19]:

$$X_i^{t+1} = X_i^t + r_1 \times \sin(r_2) \times |r_3 P_i^t - X_i^t| \quad \ldots(1)$$

$$X_i^{t+1} = X_i^t + r_1 \times \cos(r_2) \times |r_3 P_i^t - X_i^t| \quad \ldots(2)$$

Where X_i^t is the position of the current solution in $\mathrm{i}-$ th dimension at $\mathrm{t}-$ th iteration, r_1 , r_2 , r_3 are random numbers, P_i is position of the destination point in $\mathrm{i}-$ th dimension. These two equations can be combined and utilized in the following manner.

$$X_i^{t+1} = \begin{Bmatrix} X_i^t + r_1 \times sin(r_2) \times |r_3 P_i^t - X_i^t|, & r_4 < 0.5 \\ X_i^t + r_1 \times cos(r_2) \times |r_3 P_i^t - X_i^t|, & r_4 \geq 0.5 \end{Bmatrix} \quad \ldots(3)$$

where r_4 is a random number in [0,1].

Let r_1 , r_2 , r_3, and r_4 be variables representing key parameters in a stochastic control algorithm (SCA). The parameter r_2 determines the degree of motion required for the solution to approach or move away from the destination. Meanwhile, the parameter r_2 introduces a random weight for the destination, allowing for stochastic emphasis (when $r_3 > 1$) or de-emphasis (when $r_3 < 1$) of the destination's impact on determining the distance. The variable r_4 is employed to randomly alternate between sine and cosine functions in Equation (3), with its value ranging [0,1].

The equations provided above demonstrate that the parameter r_1 plays a crucial role in determining the size of the subsequent solution. The solution area can either rotate around the current solution area (during the exploration stage) or remain within it (during the exploitation stage). In order to maintain a balance between the abilities to explore and exploit, equations (1)-(3) are dynamically adjusted. As per the following equation:

$$r_1 = b - t(b/T) \quad \ldots(4)$$

In the given context, where t represents the current iteration, T represents the maximum number of iterations, and b is a constant that gradually reduces the range of sine and cosine functions from b to 0 (typically b = 2) throughout the iterations, the following paraphrase can be provided: During the sequence of iterations, the behavior of the Sine Cosine Algorithm (SCA) can be categorized into two modes based on the ranges of the sine and cosine functions. When these ranges fall within the intervals [1, 2] and [-2, -1] respectively, the SCA is said to be in the global space exploration mode. Conversely, when the ranges are limited to the interval [-1, 1], the SCA operates in the local search space, emphasizing exploitation. The pseudocode for the SCA mechanism is displays in **Algorithm 1** [19].

Algorithm 1: The Sine Cosine Algorithm.
1. *Begin by creating a group of random agents.* 2. *Evaluate the cost function for all agent.* 3. *Identify the agent with the best performance based on the cost function.* 4. *Adjust the values of r1, r2, r3, and r4.* 5. *Revise the location of the search agents by utilizing equation (3).* 6. *If the current iteration (t) is less than the maximum allowed iterations (T), return to step 2.* 7. *Provide the best solution found thus far as the global optimum.*

3. Proposed Algorithm

In this section, we present the introduction of two approaches: the locally weighted approach (LW) and the proposed approach.

3.1. Locally weighted technique (LW)

The locally weighted method, commonly known as "local search" is a heuristic method used to resolve challenging optimization issues. Using an adjacent solution from the search space, it continuously modifies the existing solution. The secret to a good local search comes in strategically selecting the right neighbors because there are frequently an endless number of potential neighbors for a solution. At the conclusion of each iteration in the optimization process, the local search algorithm (LW) is a suggested algorithm that makes use of this local search strategy to improve the present solution, known as the "position of solution." The LW local search method's steps are outlined in great depth in Algorithm 2 [22].

At the start of each iteration (denoted as t), the proposed algorithm optimizes a population (pop^t) consisting of Npop positions. Each position is represented by a solution ($x_i^t = (x_{i,1}^t, x_{i,2}^t, \ldots x_{i,dim}^t)$), which is refined using the (Sine Cosine Algorithm) based on the algorithm's guidelines. The optimized solution for a SCA is denoted as xm_i^t , Subsequently, this position is further improved using the LW approach to generate a new solution ($xnew_i^t$) using the following formula:

$$weight[j] = {}^{1}\!/_{(1 + \exp\,(xnew_i^t - x[i,j]))} \quad \ldots(5)$$

$$xnew_i^t = xnew_i^t + Z[j] * (weight[j] * (x[r_1,:] - x[r_2,:]))$$
(6)

$$Z = 0.01 \times {}^{a \times \delta}\!/_{b^{\frac{1}{\gamma}}}, \delta = \left(\frac{\mu(1+\gamma) \times sin\left(\frac{\pi\gamma}{2}\right)}{\mu\left(\frac{1+\gamma}{2}\right) \times \gamma \times 2^{\left(\frac{\gamma-1}{2}\right)}}\right)^{\frac{1}{\gamma}} \quad \ldots(7)$$

Where $x[r_1,:], x[r_2,:]$ are two random locations within the population that differ from location $x[i,:]$. Z is levy flight distribution [48 from Harris], γ is constant set equal to 1.5, a & b random value in (0,1), $x[r_3,:]$ and $x[r_4,:]$ different random locations from $x[i,:]$.

3.2. Update the positions

The SCA algorithm uses an iterative process to carry out a randomized exploration of the solution space, just as other swarm intelligence optimization techniques do. Although it cannot guarantee finding the best answer in one step, adjusting the initial values and iteration settings can enhance the likelihood of obtaining the optimal solution. SCA can effectively balance the ability to explore globally and exploit locally because of its quick convergence speed and simple structure. The process typically begins by generating an initial set of random solutions through a population-based approach. These initial, seemingly nonsensical answers will be improved once the optimization problem's objective function is evaluated. It will default to the local optimal value and ignore the global ideal value if the distribution of the agents in the search space is overly concentrated, which will decrease the algorithm's ability to explore the entire world. On the other hand, if the distribution of the agents is excessively dispersed, the algorithm's local exploitation will be lessened since the local optimal value will be disregarded. Therefore, balancing local exploitation and global exploration is a crucial aspect of optimization algorithms. The sine-cosine algorithm employs the cosine search method for exploitation and the sine search method for exploration in order to perform this function.

SCA is a numerical optimization algorithm that draws inspiration from the sine cosine wave-based algorithm and is utilized for problem-solving purposes. Although there are some advantages to using this algorithm, it's essential to be aware of its limitations. The sine-cosine algorithm, in particular, has a few common drawbacks that should be taken into consideration, including:

- The algorithm heavily depends on generating random numbers to select potential solutions and determine jumps, which leads to a significant reliance on chance. As a result, this can lead to unstable outcomes and unexpected performance.
- Having limited exploration capability can hinder finding the best possible solutions. SCA may not be able to explore the problem space thoroughly, which could lead to missing out on reasonable solutions or settling for suboptimal ones. It's essential to explore the problem space to achieve optimal results thoroughly.
- One issue with the SCA is its reliance on updating solutions through random jumps, which can be inefficient. This approach may lead to the overlooking of good solutions or arriving at unsuitable points in the search space, ultimately wasting time and resources.
- The SCA has a limitation in its adaptability to nonlinear problems. Although it is effective in solving linear or similar problems, it may face difficulties when dealing with nonlinear problems with complex relationships between variables. The SCA may not be efficient enough to handle these complexities.

These limitations highlight the need for careful consideration and evaluation when applying the Sine Cosine Algorithm to different problem domains.

In this work, a new method was proposed that combines local search with SCA to find the best solutions and avoid getting stuck in local minima. As part of this method, the updated sites and solutions were modified (see Figure 1). To begin with, every individual in the population undergoes optimization (referred to as a "position"). This can be achieved using the SCA method, specifically the cosine equation, or by utilizing a suggested approach known as the "proposed strategy for updating follower positions." Subsequently, the LW technique is employed to enhance the outcomes and select the optimum position for the given "Position". Within the population, two groups exist. The suggested technique is used with SCA to achieve the best possible outcome. By combining these methodologies, the goal is to avoid local minima and streamline the search process to uncover optimal solutions. Algorithm 3 outlines the specific steps of the suggested technique.

$$xnew_i^t = X_i^t + r_1 \times \cos(r_2) \times |r_3 P_i^t - X_i^t| \quad \ldots(8)$$

The exclusion method we use to update the current site ensures that the update is always for the better and steers clear of poor solutions. If the new site obtained is worse than the current site, it is ignored and the current one is kept, otherwise it is approved and the current site is updated with it, as shown in Algorithm 3.

Algorithm 2: The developed Locally weighted technique.
For *i = 1 :Npop*
If random < 0.5 :
Randomly choose two particles x_{rand1}^t and x_{rand2}^t from pop^t
Calculate $weight$ for particle *i* by eq. (5)
Calculate $the\ random\ value\ z\ based\ on$ Levy flight for particle *i* by eq. (7)
Calculate the new position $xnew_i^t$ of particle *i* by eq. (6)
End For

3.3 Scenario for LWSCA optimization

The algorithm depicted in Figure 2.1 combines local search and the sine cosine algorithm to efficiently discover optimal solutions without getting trapped in local minima. The algorithm consists of two primary stages. In the initial phase, the "Proposed Strategy for Updating Follower Locations" is utilized to enhance the positioning of every individual within the group. The most advantageous location is selected once the solutions have been optimized using the LW approach. The optimal site is then determined through a weight equation, and the followers are selected using our proposed method that divides the population into two groups. We aim to expedite the search process, discover optimal solutions, and avoid getting stuck in local minima by combining strategies. The algorithm clearly outlines the proposed technique. Figure 2.1 presents a visual representation of the suggested approach. There are two steps to the process. First, each member of the population undergoes optimization using either SCA or our proposed approach for updating positions. Afterward, to improve the solutions and determine the optimal position locations, the local search method is employed.

There are two sets in the population, and the SCA algorithm is used to evolve the best position. The suggested technique for positions is also employed to expedite the search process and prevent getting stuck in local minima. Algorithm 2.3 outlines the phases of the suggested technique in detail.

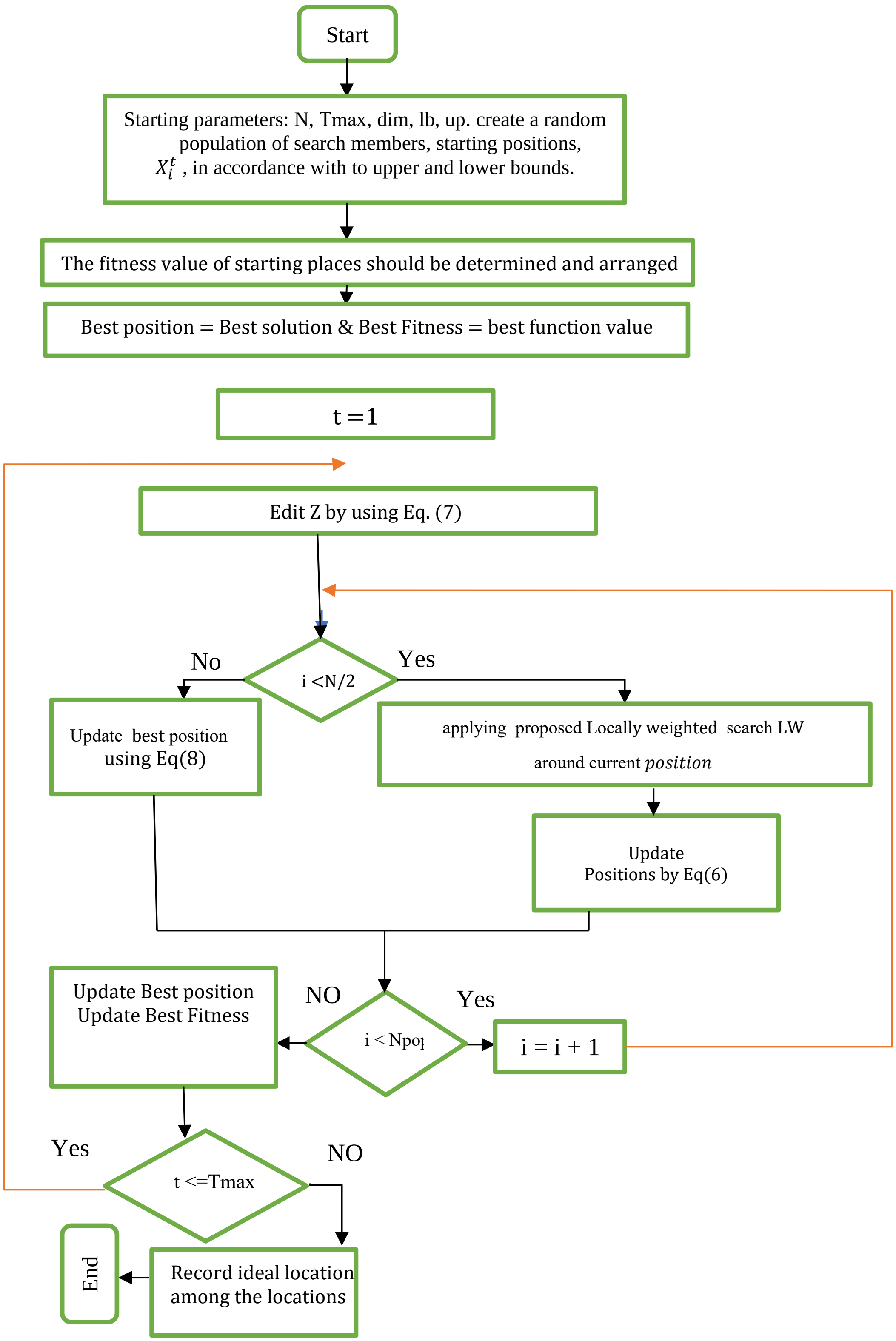


Fig. 1. Flow chart of the proposed LWSCA.

Algorithm 3: LWSCA Pseudocode.

Initialize the population matrix (location) $X_i, i = 1,2,..,N$
Dimensions and population size include upper and lower bounds.
Compute the fitness value of each location as $Fitness_i, , i = 1,2,..,N$
Compute the Best location.
and food location as the ideal location.
while (stopping condition is not hold)
  evaluate r1 by Eq. (4)
  for (all Location))
    if ($i \leq N/2$) then
      Apply (LW) technique as Algorithm (2)
    else
      Appply eq. (8) to compute the new position $xnew_i^t$

  Compute the value of the fitness of the new location as New_Fitness.
  If (New_Fitness < $Fitness_i$)
    $X_i = xnew_i^t$
  End
  Ubdate the Best Position and Best Fitness
end

4. Experimental result and analysis

In this section, we assess and compare the performance of the proposed method with other similar methods by conducting experiments on various feature selection datasets. All experiments are carried out under identical conditions.

4.1. Benchmark Examination for the IEEE CEC 2017

There are 29 distinct functions in the IEEE CEC 2017 benchmark collection. The range [-100, 100] is used to determine the search zone for each variable and function in each dimension. Each function is fully described in the technical article that goes along with it [30]. A summary of all 29 functions is also included in the paper as an illustration.

Several algorithms, including CSO, PSO, WOA, BAT, HHO, and SCA, are in competition in this test. Table 1 lists the parameters that have been chosen for each participant.

Table 1: Parameter settings.

Algorithm	Parameter setting
CSO	Phi , range: 0
PSO	Vmax = 6, wMin = 0.2, wMax = 0.9, c1 = c2 =2
WOA	Convergence constant (a) [0, 2] Coefficient (b) 1, range:2
BAT	Loudness, Pulse rate, range:0.5, 0.5 Frequency minimum, range:0 Frequency maximum, range:2
HHO	Beta, range:1.5
SCA	Convergence constant r_1, range: [0, 2]

For these optimization strategies, the population size and total number of members exceed 30 and 2500, respectively. The Friedman rank test was used to find the best option [23], and it was strongly felt that the LWSCA algorithm was superior to the other algorithms. A Wilcoxon site ranking test was also conducted at a significant level of 5% to consider any statistical discrepancies between the results achieved by the LWSCA and those of its competitors. The p-value obtained from the Wilcoxon signed-rank test is shown in Table 6. The symbol “<0.05” in Table 6 indicates whether LWSCA performed significantly better, significantly worse, or nearly as good as its peers. The average rating in Table 7 shows that LWSCA ranked highest, while BAT and CSO ranked last. The LW-SCA method outperforms the standard SCA method and other techniques. This shows how LWSCA stands out in comparison. In addition, as shown in Figures 2,3,4,5, LWSCA's new local search function enables it to identify the global solution surrounding getting trapped or trapped in the local Optima.

The properties of F1 and F2, two monomorphic functions employed in this test to evaluate the algorithm's efficiency in utilizing these scalar functions and to examine the applicability of the SCA approach, are shown in Table 2. Furthermore, Figure 2 shows that by including its innovative local search capabilities, LWSCA successfully finds the global optimal solution and avoids becoming stuck in local optima.

Table 2. Results from 2500 iterations of the LWSCA vs some Algorithms variants on IEEE CEC2017 F1and F2.

F	Cr.	CSO	PSO	WOA	BAT	HHO	SCA	LWSCA
F1	Avg	4.620E+12	1.130E+12	2.260E+11	3.490E+12	2.070E+12	1.870E+12	1.244E+09
	Std	2.192E+11	4.952E+10	4.472E+09	1.749E+11	7.922E+10	5.719E+10	6.786E+09
	Med	1.779E+12	1.549E+11	1.010E+10	1.325E+12	7.318E+11	5.219E+11	1.717E+05
F2	Avg	9.529E+05	4.689E+05	8.409E+05	5.765E+06	3.530E+05	4.085E+05	3.941E+04
	Std	1.585E+06	2.967E+04	6.001E+04	8.978E+06	2.757E+04	2.341E+04	1.278E+04
	Med	3.806E+05	1.616E+05	1.825E+05	5.638E+05	2.003E+05	1.420E+05	3.919E+04
rank	W/T/L	0/0/2	0/0/2	0/0/2	0/0/2	0/0/2	0/0/2	2/0/0

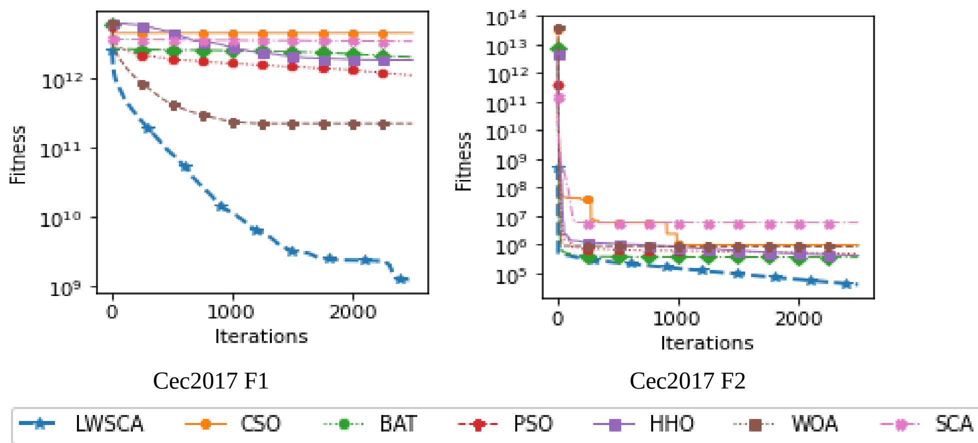


Figure 2: Cec2017 F1 and F2.

According to table3, the algorithm's ability to explore is demonstrated by the fact that it successfully finds and acquires intriguing solutions. The functions F3 ,F4, F6, F7, F8 and F9 used in this investigation have many local maxima that increase exponentially with the number of dimensions. SCA can solve these reference functions in dimensions of 50 and 100. As a result, with each step up in dimension, local maxima in functions F3, F4, F6, F7, F8 and F9 significantly expand. Table 3 shows how well SCA works in solving these reference functions in 50 and 100 dimensions, and shows how well exploration and exploitation capabilities have improved thanks to the use of competitive learning technology based on the LWSCA method. Additionally, as shown in Figure 3, LWSCA's new local search function enables it to locate the global optimal solution and prevent being trapped or gated in local optima.

Table 3. Results from 2500 iterations of the LWSCA vs some Algorithms variants on IEEE CEC2017 F3 to F9.

F	**Cr.**	**CSO**	**PSO**	**WOA**	**BAT**	**HHO**	**SCA**	**LWSCA**
F3	Avg	1.942E+05	2.536E+04	4.469E+03	1.381E+05	6.554E+04	3.857E+04	**5.389E+02**
	Std	1.402E+04	1.296E+03	1.460E+02	1.662E+04	4.230E+03	1.756E+03	5.687E+01
	Med	6.602E+04	4.930E+03	1.123E+03	5.310E+04	2.077E+04	8.607E+03	5.435E+02
F4	Avg	2.677E+03	1.793E+03	1.732E+03	2.151E+03	1.694E+03	1.970E+03	**9.299E+02**
	Std	6.610E+01	6.156E+01	7.472E+01	8.061E+01	3.603E+01	3.351E+01	7.480E+01
	Med	1.486E+03	9.729E+02	9.589E+02	1.180E+03	9.484E+02	1.095E+03	9.015E+02
F5	Avg	7.678E+02	7.041E+02	7.110E+02	7.114E+02	**6.937E+02**	7.138E+02	7.065E+02
	Std	1.194E+01	8.197E+00	1.568E+01	1.110E+01	6.522E+00	5.907E+00	1.277E+01
	Med	7.734E+02	6.856E+02	7.190E+02	7.118E+02	7.025E+02	6.981E+02	7.092E+02
F6	Avg	9.061E+03	3.181E+03	3.568E+03	6.129E+03	3.336E+03	3.715E+03	**1.714E+03**
	Std	2.826E+02	7.016E+01	9.950E+01	4.615E+02	5.799E+01	7.866E+01	9.964E+01
	Med	4.111E+03	1.496E+03	1.780E+03	2.992E+03	1.773E+03	1.714E+03	1.728E+03
F7	Avg	3.134E+03	2.134E+03	2.094E+03	3.010E+03	2.017E+03	2.339E+03	**1.220E+03**
	Std	7.072E+01	5.276E+01	5.448E+01	8.518E+01	3.710E+01	3.010E+01	7.824E+01
	Med	1.774E+03	1.246E+03	1.255E+03	1.752E+03	1.200E+03	1.411E+03	1.208E+03
F8	Avg	1.631E+05	6.715E+04	5.839E+04	3.398E+04	3.209E+04	8.236E+04	**1.876E+04**
	Std	1.002E+04	3.964E+03	8.005E+03	4.088E+03	1.171E+03	4.033E+03	5.462E+03
	Med	7.268E+04	2.026E+04	2.641E+04	1.621E+04	1.463E+04	2.542E+04	1.754E+04
F9	Avg	3.342E+04	3.214E+04	2.595E+04	2.674E+04	2.582E+04	3.221E+04	**8.513E+03**
	Std	5.936E+02	7.082E+02	1.400E+03	1.217E+03	1.303E+03	3.663E+02	7.438E+02
	Med	1.578E+04	1.469E+04	1.196E+04	1.153E+04	1.111E+04	1.506E+04	8.472E+03
ran k	W/T/ L	0/0/7	0/0/7	0/0/7	0/0/7	1/0/6	0/0/2	6/0/1

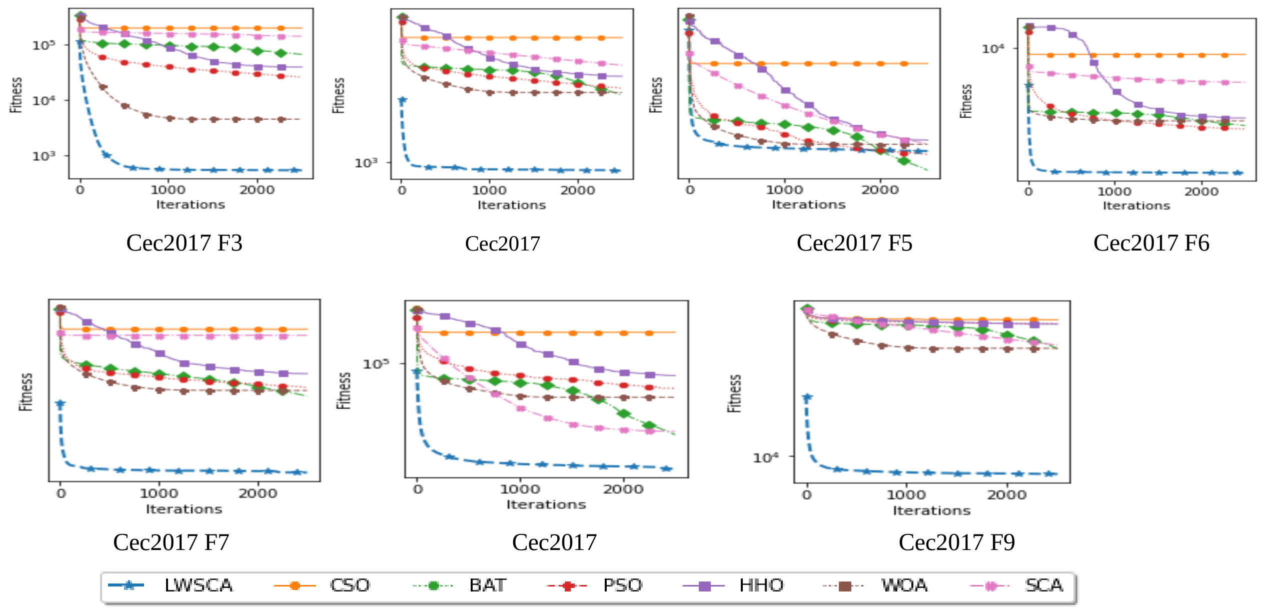


Fig. 3. Cec2017 F3 toF9.

Incorporating hybrid and compound functions is essential to striking the right balance between exploration and exploitation and avoiding being caught in local optima. The results shown in Table 4 show that the suggested LWSCA performs better than other methods while handling hybrid functions F10, F11, F12, F13, F14, F15 and F17. Additionally, Figure 4's results show that the LWSCA technique outperforms rival algorithms in handling the challenging optimization challenges provided by functions F10, F11, F12, F13, F14, F15 and F17.

Table 4. Results from 2500 iterations of the LWSCA vs some Algorithms variants on IEEE CEC2017 F10 to F19.

F	Cr.	CSO	PSO	WOA	BAT	HHO	SCA	LWSCA
F10	Avg	7.992E+05	1.168E+05	1.521E+05	2.214E+06	2.649E+05	1.200E+05	**1.272E+03**
	Std	1.426E+04	1.191E+03	6.232E+02	8.269E+04	2.679E+03	1.956E+03	3.658E+01
	Med	4.817E+04	4.120E+03	2.677E+03	6.500E+04	1.719E+04	8.579E+03	1.271E+03
F11	Avg	2.330E+12	2.430E+11	3.298E+10	2.050E+12	1.160E+12	6.430E+11	**1.251E+07**
	Std	2.132E+11	1.675E+10	3.649E+09	1.911E+11	1.112E+11	2.931E+10	1.291E+07
	Med	8.373E+11	3.448E+10	5.755E+09	6.943E+11	3.967E+11	1.243E+11	7.662E+06
F12	Avg	7.050E+11	4.485E+10	5.507E+08	5.810E+11	2.980E+11	1.280E+11	**1.400E+05**
	Std	2.047E+11	3.459E+09	2.499E+08	1.759E+11	1.173E+11	1.563E+10	1.062E+05
	Med	5.895E+11	4.518E+09	1.125E+08	4.848E+11	1.409E+11	3.750E+10	9.491E+04
F13	Avg	2.689E+08	1.276E+07	1.029E+07	2.436E+08	3.207E+07	3.700E+07	**1.230E+05**
	Std	9.520E+07	1.201E+06	1.707E+06	1.280E+08	3.166E+07	3.373E+06	8.959E+04
	Med	1.040E+08	6.766E+05	2.127E+06	1.088E+08	1.999E+07	3.737E+06	9.055E+04
F14	Avg	3.140E+11	2.040E+09	9.488E+07	2.980E+11	1.250E+11	4.045E+10	**5.559E+04**
	Std	6.306E+10	9.247E+07	4.638E+07	5.835E+10	1.468E+10	2.984E+09	3.883E+04
	Med	1.464E+11	6.167E+07	6.974E+06	9.817E+10	1.708E+10	5.575E+09	4.027E+04
F15	Avg	3.115E+04	1.185E+04	1.382E+04	2.548E+04	1.856E+04	1.377E+04	**4.019E+03**
	Std	1.748E+03	6.527E+02	8.221E+02	2.683E+03	1.753E+03	4.389E+02	4.802E+02
	Med	1.087E+04	4.574E+03	5.421E+03	9.522E+03	7.033E+03	5.916E+03	3.977E+03
F16	Avg	3.547E+07	7.671E+03	9.070E+03	2.913E+07	1.246E+06	3.178E+04	**3.712E+03**
	Std	1.634E+05	3.704E+02	4.726E+02	3.256E+05	1.085E+03	2.911E+02	4.431E+02
	Med	7.325E+04	3.375E+03	4.113E+03	3.652E+04	4.952E+03	4.716E+03	3.657E+03
F17	Avg	6.341E+08	1.277E+07	6.782E+06	5.800E+08	4.460E+07	6.887E+07	**8.429E+05**
	Std	2.342E+08	5.373E+06	1.274E+07	3.567E+08	4.117E+07	1.397E+07	4.307E+05
	Med	3.408E+08	6.203E+06	1.167E+07	3.288E+08	5.392E+07	2.351E+07	8.304E+05
F18	Avg	3.440E+11	6.097E+09	1.340E+08	2.910E+11	1.260E+11	3.739E+10	**3.717E+04**
	Std	2.695E+10	2.194E+08	6.049E+07	3.705E+03	8.203E+09	1.666E+09	3.547E+04
	Med	7.234E+10	6.868E+07	1.058E+07	1.752E+04	6.083E+09	3.407E+09	3.194E+04
F19	Avg	8.150E+03	7.362E+03	6.475E+03	6.527E+03	6.129E+03	7.480E+03	**3.526E+03**
	Std	2.392E+02	3.241E+02	3.512E+02	3.959E+02	2.806E+02	1.611E+02	3.236E+02
	Med	4.499E+03	3.884E+03	3.813E+03	4.256E+03	3.596E+03	4.007E+03	3.555E+03
rank	W/T/L	0/0/10	0/0/10	0/0/10	0/0/10	0/0/10	0/0/10	10/0/0

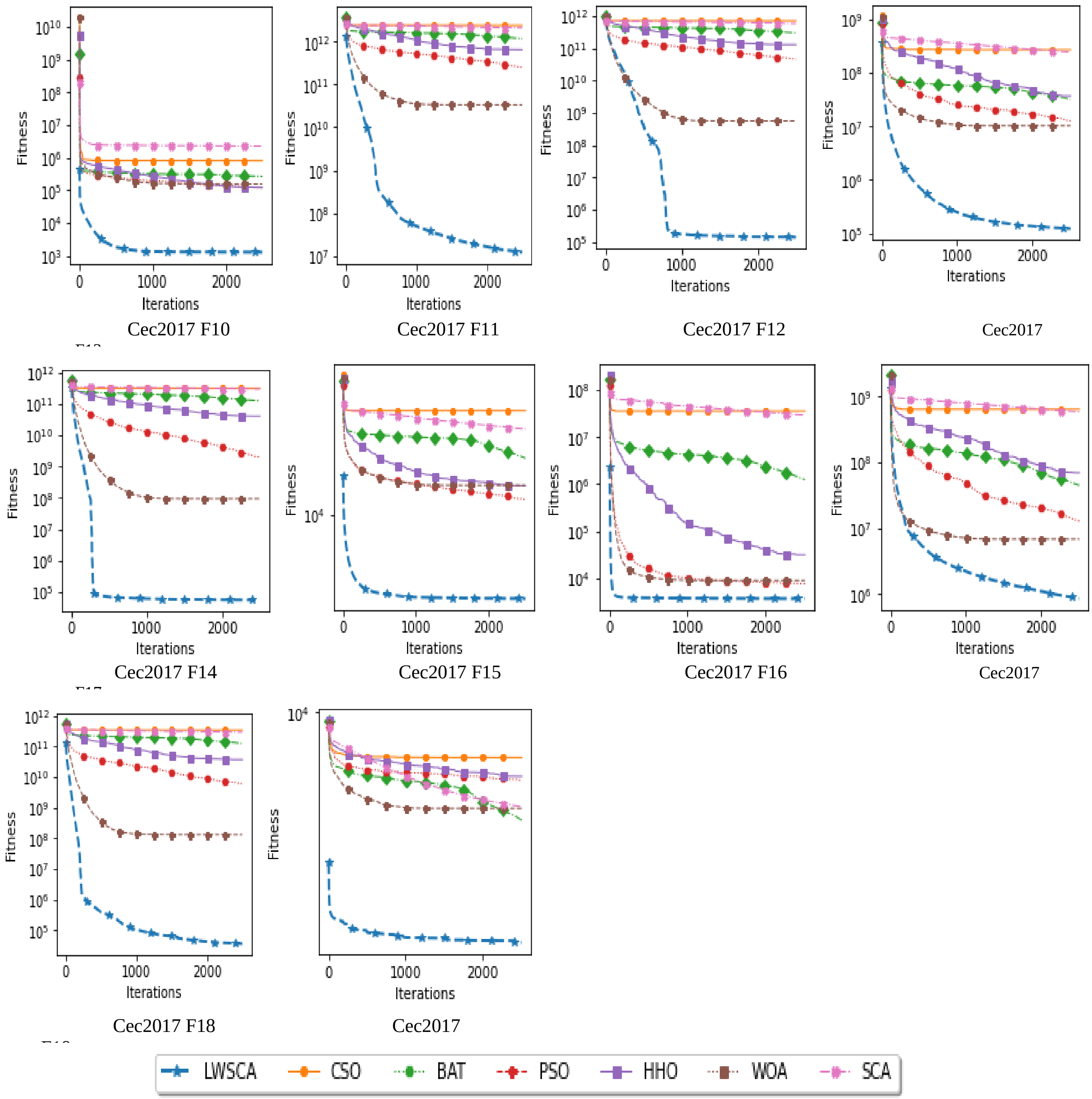
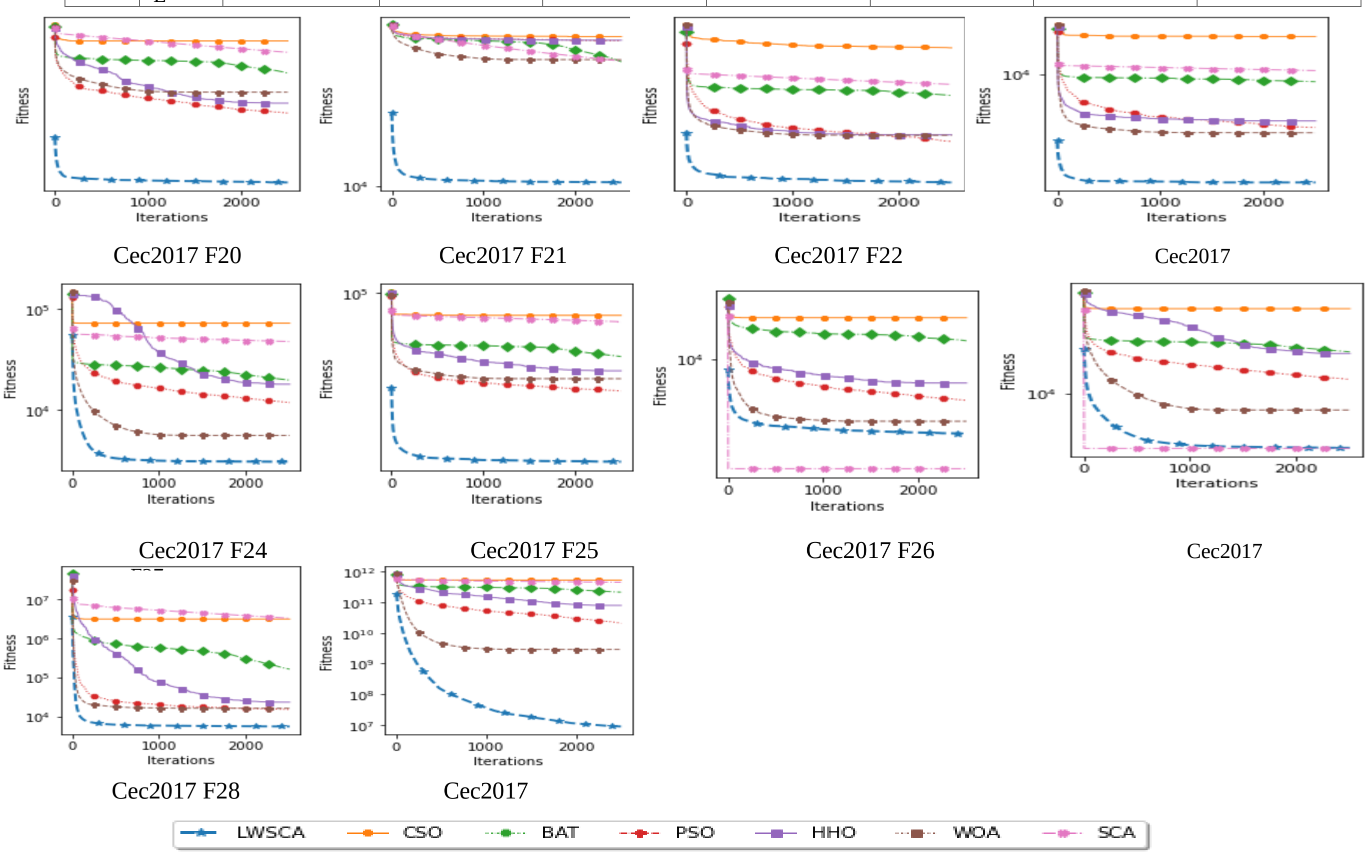


Fig. 4. Cec2017 F10 toF19.

The enclosed Table 5 demonstrates the utilization of the technique on compound functions, specifically F20 to F29. The outcomes reveal that the proposed LWSCA algorithm outperforms other algorithms in these functions F20, F21, F22, F23, F24, F25, F28, and F29. Additionally, the results depicted in Figure 5 provide further evidence that the SCA approach is more effective than alternative methods in resolving challenging optimization problems posed by the F20, F21, F22, F23, F24, F25, F28, and F29 functions.

Table 5. Results from 2500 iterations of the LWSCA vs some Algorithms variants on IEEE CEC2017 F20 to F29.

F	Cr.	CSO	PSO	WOA	BAT	HHO	SCA	LWSCA
F20	Avg	5.221E+03	3.889E+03	4.235E+03	4.984E+03	4.576E+03	4.048E+03	**2.928E+03**
	Std	1.339E+02	4.596E+01	1.013E+02	1.187E+02	8.516E+01	4.460E+01	1.186E+02
	Med	3.413E+03	2.819E+03	2.953E+03	3.143E+03	3.014E+03	2.903E+03	2.923E+03
F21	Avg	3.570E+04	3.444E+04	2.934E+04	2.901E+04	2.881E+04	3.462E+04	**1.032E+04**
	Std	6.786E+02	1.745E+03	1.324E+03	1.065E+03	1.208E+03	4.319E+02	9.087E+02
	Med	1.746E+04	1.639E+04	1.348E+04	1.387E+04	1.333E+04	1.674E+04	1.021E+04
F22	Avg	7.841E+03	4.883E+03	5.034E+03	6.516E+03	6.155E+03	5.047E+03	**3.965E+03**
	Std	3.582E+02	8.097E+01	1.883E+02	3.379E+02	2.132E+02	7.492E+01	3.106E+02
	Med	4.972E+03	3.451E+03	3.751E+03	4.520E+03	4.207E+03	3.578E+03	3.949E+03
F23	Avg	1.358E+04	6.537E+03	6.260E+03	1.032E+04	9.445E+03	6.884E+03	**4.196E+03**
	Std	5.384E+02	7.984E+01	1.723E+02	3.107E+02	2.405E+02	6.064E+01	2.899E+02
	Med	5.434E+03	3.653E+03	3.773E+03	4.876E+03	4.486E+03	3.768E+03	4.196E+03
F24	Avg	7.226E+04	1.180E+04	5.585E+03	4.748E+04	1.972E+04	1.795E+04	**3.068E+03**
	Std	6.108E+03	6.977E+02	1.335E+02	5.804E+03	9.735E+02	7.762E+02	4.205E+01
	Med	3.259E+04	5.652E+03	3.462E+03	2.438E+04	1.011E+04	7.134E+03	3.077E+03
F25	Avg	7.546E+04	2.923E+04	3.408E+04	6.918E+04	4.500E+04	3.764E+04	**1.202E+04**
	Std	2.820E+03	7.221E+02	1.215E+03	2.631E+03	7.011E+02	4.847E+02	1.019E+03
	Med	2.562E+04	1.081E+04	1.417E+04	2.126E+04	1.479E+04	1.277E+04	1.237E+04
F26	Avg	1.535E+04	6.511E+03	5.236E+03	**3.200E+03**	1.208E+04	7.791E+03	4.610E+03
	Std	1.113E+03	1.867E+02	4.691E+02	6.390E-05	9.094E+02	1.743E+02	5.815E+02
	Med	8.130E+03	4.684E+03	4.151E+03	3.200E+03	6.339E+03	4.608E+03	4.533E+03
F27	Avg	5.591E+04	1.334E+04	7.187E+03	**3.300E+03**	2.330E+04	2.254E+04	3.339E+03
	Std	2.195E+03	5.386E+02	2.741E+02	3.671E-05	8.839E+02	6.693E+02	6.075E+01
	Med	1.847E+04	5.717E+03	4.245E+03	3.300E+03	9.849E+03	7.295E+03	3.320E+03
F28	Avg	3.164E+06	1.537E+04	1.650E+04	3.323E+06	1.641E+05	2.357E+04	**5.639E+03**
	Std	1.062E+06	8.103E+02	9.872E+02	5.647E+05	2.448E+04	8.454E+02	4.884E+02
	Med	2.641E+05	7.157E+03	8.264E+03	1.831E+05	1.969E+04	7.943E+03	5.596E+03
F29	Avg	5.250E+11	2.074E+10	2.898E+09	4.480E+11	2.120E+11	7.870E+10	**8.920E+06**
	Std	4.397E+10	5.753E+08	3.097E+08	4.420E+10	7.920E+09	1.983E+09	6.028E+06
	Med	9.897E+10	7.147E+08	5.299E+08	8.507E+10	9.554E+09	6.234E+09	6.886E+06
rank	W/T/L	0/0/10	0/0/10	0/0/10	2/0/8	0/0/10	0/0/10	8/0/2



Cec2017 F20 Cec2017 F21 Cec2017 F22 Cec2017

Cec2017 F24 Cec2017 F25 Cec2017 F26 Cec2017

Cec2017 F28 Cec2017

Fig. 5. Cec2017 F20 toF29.

Table 6. Wilcoxon rank − sum of the $LWSCA$ vs. another algorithms on CEC2017.

Fun	CSO	PSO	WOA	BAT	HHO	SCA
1	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
2	< 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
4	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
5	< 0.05	< 0.05	< 0.05	0.210614	< 0.05	< 0.05
6	< 0.05	< 0.05	< 0.05	< 0.05	0.163971	0.570291
7	< 0.05	< 0.05	< 0.05	< 0.05	0.894837	< 0.05
8	< 0.05	0.432254	< 0.05	0.110934	< 0.05	< 0.05
9	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
10	< 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
12	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
13	< 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
15	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
16	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
17	< 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
19	< 0.05	< 0.05	< 0.05	< 0.05	0.931838	< 0.05
20	< 0.05	< 0.05	0.488917	< 0.05	< 0.05	0.591599
21	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
22	< 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
24	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
25	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
26	< 0.05	0.956593	< 0.05	< 0.05	< 0.05	0.968987
27	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
28	< 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

Table 7. Friedman test result of the LWSSA vs another algorithms on CEC2017.

Fun	CSO	PSO	WOA	BAT	HHO	SCA	LWSCA
1	6.965517	3	1.965517	6.034483	4.965517	4.034483	1.034483
2	6.206897	3.413793	3.827586	6.586207	4.37931	2.586207	1
3	6.62069	3.034483	2	6.37931	5	3.965517	1
4	7	2.931034	2.896552	5.827586	2.482759	4.931034	1.931034
5	7	1.137931	4.862069	4.655172	3.413793	2.862069	4.068966
6	6.931034	1.034483	4.068966	6.068966	3.862069	2.896552	3.137931
7	6.62069	3	2.965517	6.37931	1.827586	4.931034	2.275862
8	7	3.655172	4.965517	2.344828	1.827586	5.068966	3.137931
9	6.724138	5.206897	3.241379	2.931034	2.862069	6	1.034483
10	6.310345	2.931034	2.068966	6.586207	5.034483	4.068966	1
11	6.793103	3	2	6.103448	5.068966	4.034483	1
12	6.517241	3	2	6.37931	5.068966	4.034483	1
13	6.37931	2.310345	3.103448	6.310345	5	3.862069	1.034483
14	6.758621	2.896552	2.103448	6.206897	4.862069	4.172414	1
15	6.586207	1.793103	3.137931	6.103448	5.034483	3.827586	1.517241
16	6.62069	1.482759	3	6.344828	4.310345	4.344828	1.896552
17	6.344828	2.448276	2.931034	6.551724	4.896552	3.793103	1.034483
18	7	3.896552	3.172414	1.172414	5.551724	5.37931	1.827586
19	6.517241	3.586207	3.344828	5.586207	2.172414	4.62069	2.172414
20	6.896552	1.310345	3.448276	5.62069	4.586207	2.827586	3.310345
21	6.827586	5.137931	2.931034	3.344828	2.793103	5.827586	1.137931
22	6.62069	1.172414	2.931034	6.103448	4.896552	2.310345	3.965517
23	6.793103	1.448276	2.310345	5.931034	4.931034	2.482759	4.103448
24	6.793103	3.068966	2	6.206897	5	3.931034	1
25	6.931034	1.172414	3.827586	6.068966	4.793103	3	2.206897
26	6.793103	3.689655	2.827586	1	6.137931	3.62069	3.931034
27	7	4	3	1.103448	6	5	1.896552
28	6.448276	2.206897	3.517241	6.37931	5.172414	3.241379	1.034483
29	6.62069	2.689655	2.310345	6.37931	4.793103	4.206897	1
Avg.	6.711058	2.74673	2.991677	5.265161	4.369798	3.995244	1.920333
Rank	7	2	3	6	5	4	1

4.2 LWSCA Computational Complexity

We shall step-by-step investigate the computational complexity of LWSCA with N as the population size, D as the issue dimension, and L as the maximum number of iterations. The worst-case computation time can be calculated as follows. During the initialization stage, LWSCA undergoes two processes. The first is random initialization, which has the same computational complexity as the original SCA, O(N × D). The second is obtaining the optimal solution, which also has a computational complexity of O(N), similar to the original SSA. Therefore, in the initialization stage, the computational complexity of LWSCA is O(N × D + N). In each iteration stage, the agent's position is first updated based on Equation (1) or Equation (9), resulting in a computational complexity of O(N × D). Then, the location of each agent is further updated using the LW strategy, which has a computational complexity of O(N). Hence, in each iteration, the computational complexity of LWSCA is O(N × D + N). Consequently, the overall computational complexity of the original LWSCA is O(L × N × D + L × N), which can be simplified as O(L × N × N × (D + 1)).

5. Conclusion and future works

In this paper, we introduce a novel algorithm that combines the Sine Cosine Algorithm (SCA) with a locally weighted (LW) strategy to improve the utilization of conventional algorithms. We assess the performance of this hybrid algorithm by comparing it to five other feature selection methods: CSO, PSO, WOA, BAT, and HHO. We define various parameters, including the maximum value of the objective function, deviation, best performance, average performance, subjective assessment, and aesthetic time. By examining how the algorithm converges, we conclude that it excels at maintaining a balance between exploitation and exploration. Empirical numerical and statistical results demonstrate that this hybrid approach outperforms its competitors in terms of convergence speed. These findings suggest that it presents viable solutions and could serve as an alternative method for improving the efficiency of real-world tasks, particularly those of a complex nature. To widen its applicability across various domains, future research could explore its use with diverse datasets. For instance, it might find application in analyzing data from pocket turbine tests. Additionally, it could be worthwhile to investigate this optimization algorithm (AOA) alongside other established optimization techniques that have traditionally addressed integral selection problems.

References

[1] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," ACM Comput. Surv., vol. 35, no. 3, pp. 268–308, 2003.

[2] Zbigniew Michalewics, "Michalewicz Z. Genetic Algorithms + Data Structures = Evolution Programs (3ed).PDF."

[3] G. Dhiman and V. Kumar, "Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems," Knowledge-based Syst., vol. 165, pp. 169–196, 2019.

[4] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: theory and application," Adv. Eng. Softw., vol. 105, pp. 30–47, 2017.

[5] F. Qiang, T. Nan, and Z. Yi-ming, "Firefly algorithm based on multi-group learning mechanism.," Appl. Res. Comput. Yingyong Yanjiu, vol. 30, no. 12, 2013.

[6] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," Adv. Eng. Softw., vol. 69, pp. 46–61, 2014.

[7] S. Mirjalili, "The ant lion optimizer," Adv. Eng. Softw., vol. 83, pp. 80–98, 2015.

[8] W. Zhao, Z. Zhang, and L. Wang, "Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications," Eng. Appl. Artif. Intell., vol. 87, p. 103300, 2020.

[9] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical report-tr06, Erciyes university, engineering faculty, computer …, 2005.

[10] K. Price, R. M. Storn, and J. A. Lampinen, Differential evolution: a practical approach to global optimization. Springer Science & Business Media, 2006.

[11] 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, 2017, doi: 10.1016/j.advengsoft.2017.07.002.

[12] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proceedings of ICNN'95 - International Conference on Neural Networks, vol. 4, pp. 1942–1948. doi: 10.1109/ICNN.1995.488968.

[13] M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, "Metaheuristic algorithms: A comprehensive review," Comput. Intell. Multimed. big data cloud with Eng. Appl., pp. 185–231, 2018.

[14] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering," Inf. Sci. (Ny)., vol. 222, pp. 175–184, Feb. 2013, doi: 10.1016/J.INS.2012.08.023.

[15] A. Y. S. Lam and V. O. K. Li, "Chemical-Reaction-Inspired Metaheuristic for Optimization," IEEE Trans. Evol. Comput., vol. 14, no. 3, pp. 381–399, Jun. 2010, doi: 10.1109/TEVC.2009.2033580.

[16] U. M. Khaire and R. Dhanalakshmi, "Stability of feature selection algorithm: A review," J. King Saud Univ. Inf. Sci., vol. 34, no. 4, pp. 1060–1073, 2022.

[17] A. A. El-Fergany and H. M. Hasanien, "Single and multi-objective optimal power flow using grey wolf optimizer and differential evolution algorithms," Electr. Power Components Syst., vol. 43, no. 13, pp. 1548–1559, 2015.

[18] K. R. Harrison, A. P. Engelbrecht, and B. M. Ombuki-Berman, "Optimal parameter regions and the time-dependence of control parameter values for the particle swarm optimization algorithm," Swarm Evol. Comput., vol. 41, pp. 20–35, 2018.

[19] S. Mirjalili, "SCA: A Sine Cosine Algorithm for solving optimization problems," Knowledge-Based Syst., vol. 96, pp. 120–133, 2016, doi: 10.1016/j.knosys.2015.12.022.

[20] L. Abualigah and A. J. Dulaimi, "A novel feature selection method for data mining tasks using hybrid sine cosine algorithm and genetic algorithm," Cluster Comput., vol. 24, pp. 2161–2176, 2021.

[21] X. Yang et al., "An adaptive quadratic interpolation and rounding mechanism sine cosine algorithm with application to constrained engineering optimization problems," Expert Syst. Appl., vol. 213, p. 119041, 2023.

[22] K. Sindhya, A. Sinha, K. Deb, and K. Miettinen, "Local search based evolutionary multi-objective optimization algorithm for constrained and unconstrained problems," in 2009 IEEE Congress on Evolutionary Computation, 2009, pp. 2919–2926.

[23] 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. 2, 2022, doi: 10.1007/s00521-022-06921-2.