IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 10 June 2024, accepted 7 July 2024, date of publication 15 July 2024, date of current version 24 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3427632

RESEARCH ARTICLE

In Search of Excellence: SHOA as a Competitive Shrike Optimization Algorithm for Multimodal Problems

HANAN K. ABDULKARIM[1] AND TARIK A. RASHID[2], (Member, IEEE)
[1]Software Engineering Department, Engineering College, Salahaddin University-Erbil, Erbil 44002, Iraq
[2]Computer Science and Engineering Department, University of Kurdistan Hewlêr, Erbil 00964, Iraq

Corresponding author: Hanan K. AbdulKarim (hanan.abdulkarim@su.edu.krd)

ABSTRACT This paper proposes the Shrike Optimization Algorithm (SHOA) as a swarm intelligence optimization algorithm. Many creatures, who live in groups and survive for the next generation, randomly search for food; they follow the best one in the swarm, a phenomenon known as swarm intelligence. While swarm-based algorithms mimic the behaviors of creatures, they struggle to find optimal solutions in multi-modal problem competitions. The swarming behaviors of shrike birds in nature serve as the main inspiration for the proposed algorithm. The shrike birds migrate from their territory in order to survive. However, the SHOA replicates the survival strategies of shrike birds to facilitate their living, adaptation, and breeding. Two parts of optimization exploration and exploitation are designed by modeling shrike breeding and searching for foods to feed nestlings until they get ready to fly and live independently. This paper is a mathematical model for the SHOA to perform optimization. The SHOA benchmarked 19 well-known mathematical test functions, 10 from CEC-2019 and 12 from CEC-2022's most recent test functions, for a total of 41 competitive mathematical test functions and four real-world engineering problems with different conditions, both constrained and unconstrained. The statistical results obtained from the Wilcoxon ranking sum and Fridman test show that SHOA has a significant statistical superiority in handling the test benchmarks compared to competitor algorithms in multi-modal problems. The results for engineering optimization problems show the SHOA outperforms other nature-inspired algorithms in many cases.

INDEX TERMS Shrike, optimization, constrained optimization, swarm intelligence, multi-modal, meta-heuristic, population-based optimization, engineering problem.

I. INTRODUCTION

Optimization techniques have become important in the last few decades. Optimization is finding the best optimal or semi-optimal solution by achieving a specific objective without violating constraints. In some cases, no objective functions exist, but a feasible solution depending on constraints is an optimal solution, called a feasibility problem. Many complex and rough-solvable problems in engineering, science, medicine, statistics, and computer science have been solved by optimization algorithms within a short time. Mathematical calculation and programs have been used to solve such a problem, but recently, for solving complex problems, some meta-heuristic optimization algorithms have been used to find acceptable solutions. Many optimization algorithms are nature-inspired algorithms designed by mimicking creatures from nature; many of those algorithms depend on swarms' social behavior and are called swarm-based algorithms.

Optimization algorithms have been classified as single-based and population-based. Single-based optimization searches for an optimum solution using a single solution like simulated Annealing (SA) [1], Hill Climbing (HC) [2], Variable Neighborhood Search (VNS) [3], and Tabu Search (TS) [4], while the population-based optimization algorithms use a group of solutions as a population and search around the number of the neighbors of the solutions in the search space, it should have good exploration and exploitation

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

techniques to not trap in the local optima, population-based like Genetic Algorithm (GA) [5], Differential Evaluation (DE) [6], Genetic Programming (GP) [7], and swarm-based algorithms.

Swarm-based algorithms are stochastic because they work on the Swarm Intelligence (SI) of the creature's behavior. Ant Colony Optimization (ACO) [8] is an old algorithm that studies the collective behavior of ants searching for food sources. It simply translates the fact that every ant has its own decision for foraging on a specific path, each ant signs the path by pheromone when it transits to the food source, and it will add pheromone again when returning to the nest, so other ants will take a path with higher pheromone and leave their path, then the shortest path will be accomplished by leaving the low pheromone path and use the higher-level pheromones path. Ant System (AS) [9],and [10] applied to solve various combinatorial optimization problems. The application of AS includes the Traveling Salesman Problem (TSP), the Quadratic Assignment Problem (QAP), and the Job-shop Scheduling Problem (JSP), it shows the ability to solve those problems, also applied in the classification field [11], and cloud computing [12]. Recently Ant Nesting Algorithm was proposed [13] searching to build nests and deposit foods.

Particle Swarm Optimization (PSO) [14] mimics the inspiration of SI of birds, and fish while the author considered birds, simple techniques were used that birds follow the flock fly direction, the best food source obtained so far, and the best food sources that the swarm found, simply it uses rules to find the best solution in the search space, and it is applied in many fields of design [15], image processing [16] which successfully improves solutions. The Society and Civilization algorithm [17] is the adaption of societies simulated for optimization problem-solving. Artificial Bee Colony (ABC) works on honey bees finding food sources in [18] and [19], it works on how explored bees find food sources and share information with employed bees, the onlooker bees exploit food sources more to find better sources and keep the best food source, ABC outperforms many optimization algorithms for some optimization problems of global optimization [20] and [21], feature selection [22], neural network fields [23], vehicle routing [24]. Fitness Dependent Optimizer (FDO) also working on bees foraging behavior is proposed for optimization problems [25]. Bacterial Foraging behavior (BFO) the bacterial foraging behavior has been a source for development [26], applied for electrical power filter problems [27], and designed fuzzy control for the system [28]. In Firefly Algorithm (FA), the flashing light and attractiveness of fireflies were formulated as FA algorithm, used to solve multi-modal problems [29], design structure [30], and many other applications. The moth-flame Optimization (MFO) Algorithm [31] was developed by studying moths' navigation in nature and how they move around lights. Solving problems with clustering suffers from exploration the MFO is added to handle the clustering problem [32]. Recently some new population-based metaheuristic algorithms have been proposed by researchers like new artificial protozoa optimizer (APO) inspired by biology [33], the Horned Lizard Optimization Algorithm (HLOA) used the defensive strategies of the horned lizard reptile [34], the Black Winged Kite birds' skills in the fields of hunting and migrating are modeled as Black Winged Kite Algorithm (BKA) [35], the Hiking Optimization Algorithm (HOA), which hikers travel uphill, and HOA seeks to find the local or global optimal solution to optimization issues [36] In the exploration, the authors mention trigonometric functions that are used to search space globally to cover all space and avoid local optima. Although search space coverage uses sin and cosine to guarantee searching for all solutions, while in the exploitation, refinement and local search on founded solutions will be done to improve accuracy and convergence to the global optimum [37].

Nature-inspired algorithms have demonstrated exceptional performance in optimization problems, particularly multi-modal problems. Soccer inspired metaheuristic-based sports concepts, they became popular because they developed methods and concepts [38]. The Squirrel Search Algorithm (SSA) is an optimizer that emulates the dynamic hunting behavior of southern flying squirrels [39], the Marine Predator Algorithm (MPA) is a metaheuristic algorithm designed to emulate the hunting behavior exhibited by marine predators [40]. A new algorithm is proposed in [41] as a peacock algorithm that mimics the mating and hunting behaviors of peacock birds. The Cheetah (C) algorithm is inspired by the cheetah's foraging strategy [42]. Mountain Gazelle Optimizer (MGO) is an algorithm that takes inspiration from the social life of mountain gazelles [43]. Since the reviewed methods initially were proposed, the researchers have worked to review, enhance or implement them in many domains and for various challenges [44], [45], [46], [47], [48], [49], [50], [51], and [52].

This paper proposes a swarm-based SHOA refers to a bird-inspired class, to increase the number of solved multi-modal and complex problems because none of the optimization algorithms can solve all problems. Depending on the nature of the problem, a specific algorithm must be applied to find the best solution. Intensification and diversification are the essential components of meta-heuristic algorithms. The main contributions of this study are:

1. The proposed SHOA is designed for multi-modal problems by finding many local optima and keeping them to produce global optima because multi-modal problems have many local optima and many optimization algorithms lack the ability to find the global optima.
2. In the SHOA mathematical proposal produced, depending on the shrike bird's physical simulations of a parent bird's dominance in a specific life stage, the roles of female and male birds were separated depending on reality and lifestyle.
3. The SHOA, applying randomization will diverge the algorithm from the current solution, which is considered a local optimum, and redirect the algorithm to search the space globally to increase diversity, while finding a solution during the local search by choosing the best solution

so far will converge the algorithm to an optimal solution, increasing convergence.

The remainder of this article is structured as follows: (Part II) represents the literature viewed, and important points are discussed, (Part III) presents inspiration from shrike birds and a mathematical model for the proposed SHOA (Part IV) results and discussion on comparative benchmarks and some competitive functions, and (Part V) SHOA applied real-world cases, studies as engineering problems and the performance compared with other optimization algorithms, finally (Part VI) conclude the work of this study and show direction for coming studies.

A. LITERATURE REVIEW

Metaheuristic optimization algorithms are nature-inspired algorithms, they study techniques and rules of the creature's inspiration and convert them to algorithm steps to solve problems. Recently developed algorithms are studied and classified, in this study some of the classes are prepared and noted to classify algorithms rather than categories mentioned by researchers. Nature-inspired algorithms are refined to classes with some examples shown in Figure 1. Some of the recently proposed algorithms with classes are:

1. Animal-inspired classes are sub-grouped into the bird, mammal, fish, insect, inspired, some examples are; Eurasian Oystercatcher Optimizer (EOO) [53], White Shark Optimizer (WSO) [54], Fox Optimizer (FOX) [55], Orca Optimization Algorithm (OOA) [56], Walrus Optimization Algorithm (WaOA) [57], Aphid-Ant Mutualism (AAM) [58], Fire Hawk Optimizer (FHO) [59], Honey Badger Algorithm (HBA) [60], Tunicate Swarm Algorithm (TSA) [61], Pufferfish Optimization Algorithm (POA) [62], Marine Predator Algorithm (MPA), Horned Lizard Optimization Algorithm (HLOA).
2. Plant, Microorganism, Physics, Human Activity, Mathematics, Algorithm-Specific and miscellaneous classes, all classes are shown in Figure 1, some examples of recently proposed algorithms are:
   a) Water Wheel Plant Algorithm (WWPA) [63]
   b) Artificial Protozoa Optimizer (APO)
   c) Black Hole Mechanics Optimization (BHMO) [64]
   d) Chef-Based Optimization Algorithm (CBOA) [65]
   e) Gradient-Based Optimizer (GBO) [66]
   f) One-to-One-Based Optimizer [67]
   g) PID-based Search Algorithm (PSA) [68]
   h) Ali Baba and the Forty Thieves Optimization (AFT) [69]

B. MULTIMODAL OPTIMIZATION AND PARAMETER TUNING

Multimodal optimization problems (MMOPs) necessitate the simultaneous search for several optimum solutions. Assert that the algorithm needs to broaden its population diversity to identify more global optima, and enhance its refinement capabilities to boost the accuracy of the discovered solutions [70]. The group of multimodal approaches using the methods of speciation and crowding-niching. Whereas speciation separates the population into individuals of the same species, crowding-niching splits the population into niches occupied by various species. Several multimodal techniques, including the Fitness Sharing (FS) technique, consider sharing models and similarity functions [71].

The researchers have implemented several niching strategies to divide the population into distinct niches, each responsible for conducting searches on one or more peaks [70]. Real-world design challenges known as constrained numerical optimization problems (CNOPs) require a feasible, ultimate optimized solution, and restrictions act as roadblocks to potential solutions. When it comes to addressing conventional unconstrained numerical optimization problems (UNOPs), nature-inspired meta-heuristics are popular. Effective algorithms require population diversity to accomplish design space exploration, but they reduce diversity through optimization to leverage the space's global optimum [72]. Handling the niche centre distinction (NCD) problem as an optimization problem. Performance measures assess success, accuracy, feasibility [72].Many contexts, including data mining, power systems, pattern recognition, and vehicle routing issues, have used MMO algorithms. Several researchers have proposed multi-objective evolutionary optimization strategies for solving MMOPs using bi-objective problems. In addition to the steps used in Evolutionary Algorithms (EAs), MMO algorithms employ additional strategies to converge on numerous solutions. Authors suggested EAs use one of two prevalent niching techniques: species-based DE (SDE) or crowding-based DE (CDE) [70]. Researchers have used a stable mutation approach to create new individuals, the SoftMax function to determine individual probabilities, and an archive technique to retain stagnant individuals [73].

In [74], the authors solve MMOPs using Distributed Individuals for Multiple Peaks (DIMP) used with DE, by applying age to each individual, DIMP allows every individual to function as a dispersed unit to monitor a peak, avoiding the challenges associated with population division and preserve enough variation to find new peaks throughout their lifetime.

Differential evolution (DE) is an efficient yet straightforward approach extensively researched for both MMOPs and single optimum optimization problems [74].

The authors in [75], applied the two-phase stream clustering algorithm based on fitness proportionate sharing to produce data for MMOP. The authors then developed a novel dynamic clustering algorithm to extract the cluster structure automatically from scratch and approximate the density distribution of the data stream using a recursive lower bound of the Gaussian kernel function. Applying the Cluster-Chaotic-Optimization (CCO) approach for a specific optimization issue, extending its capabilities to discover, register, and retain many optima effectively [71].

Because parameter tuning is a hyper-optimization problem, it is particularly difficult when tweaking optimization techniques. Currently, it is unclear how to tune parameters

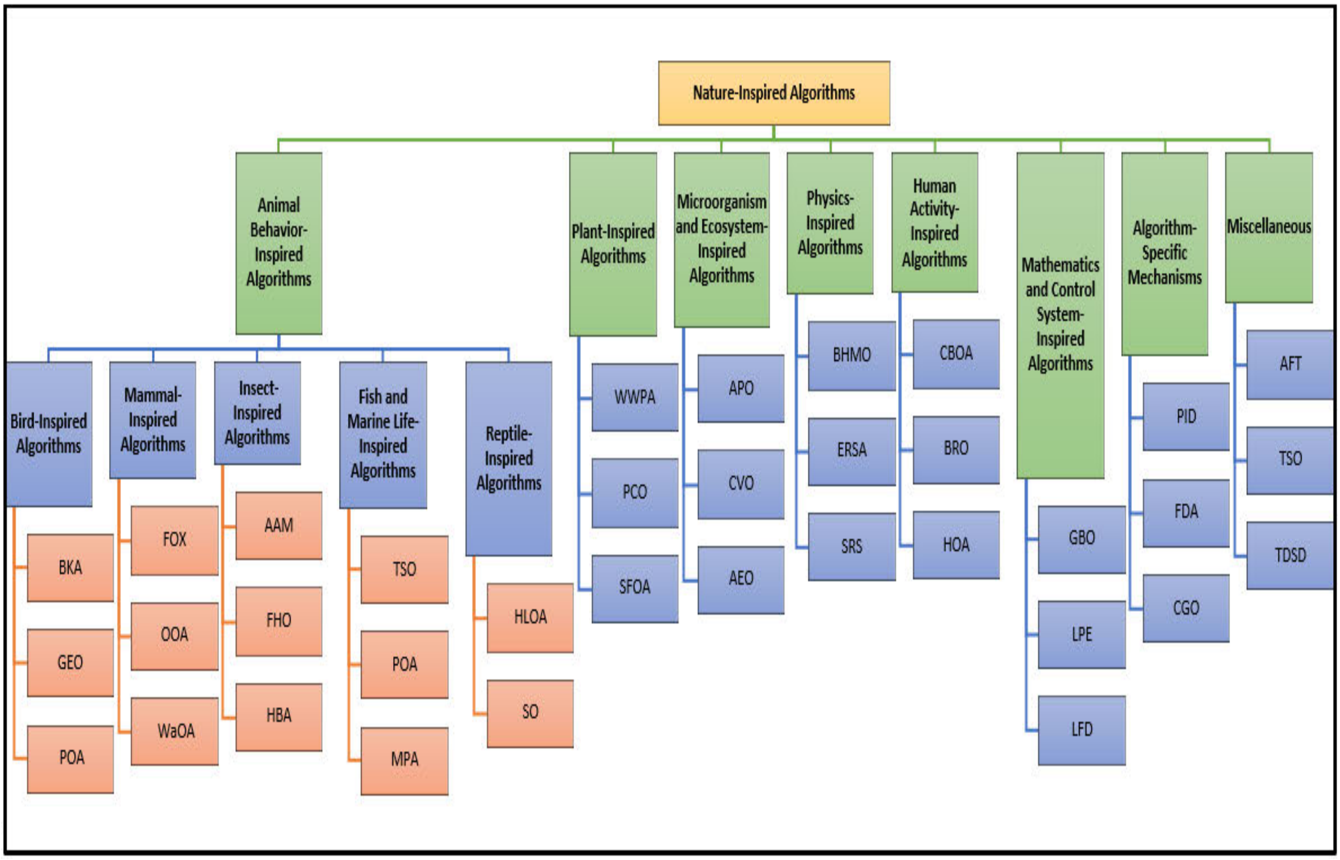

FIGURE 1. Classification of metaheuristic optimization algorithm.

effectively and how to control parameters properly for any given algorithm, and a given set of problems. Although a good tuning tool apply, the tuned parameters may not perform well for other problems, or for different types with unknown optimality [76]. Parameter tuning may be done in a variety of ways, such as hyper-optimization, which applies the ideal parameter setting n* after the algorithm has been tuned for the given issue M. A parallel or loop structure for tuning and problem-solving will be used repeatedly as an additional method for parameter tuning [76]. The approach known as self-adaptive multi-population (SAMP) involves the dynamic addition and delete of populations according to their variety. The free population, the initial population in this approach, is a single randomly initiated population. After evolution, the authors declared a solution to have converged if the gap between them decreases below a specific threshold. If all current populations have converged, new populations will be randomly introduced. SAMP maintains at least one free population at all times to prevent the algorithm from becoming stuck in local optima [77]. The researchers also researched random partitioning, a potential population partitioning approach that divides a single population into several smaller sub-populations at random using seed-based partitioning with fixed seeds and random partitioning with a primary population [77]. The significance of initialization is crucial for the accuracy and rate of convergence of certain algorithms. Researchers should utilize various initialization techniques for different situations, as starting solutions may impact the effectiveness of the optimization algorithm. Uniform distributions are not the optimal initialization strategy for all functions [78]. Robust Optimization Over Time (ROOT) is a discipline that focuses on investigating and advancing algorithms, incorporating the principles of both adaptive and robust optimization [79]. The No Free Lunch (NFL) theorem states that there are no metaheuristic algorithms or optimization strategies available to handle the issue optimally. While metaheuristic optimization approaches can be helpful in solving some issues, they can also be ineffective in solving other difficulties. There is still room for improvement in the field of metaheuristic optimization algorithms, and several academics are working to build new metaheuristic algorithms [80]. Applying diversity measures improves the understanding and efficiency of algorithms. Researchers suggest replacement and exclusion operator strategies. By randomly reinitializing the population with a less-fit solution, the inclusion operator in multi-population swarm algorithms preserves population variety. There is a replacement operator that creates new, and randomly generated solutions to replace the ones that already exist. The majority of diversity-increasing methods help an algorithm's fundamental structure to be modified [81]. Researchers have found the use of the enhanced search method with various swarm algorithms that use Cauchy, Levy, and uniform distributions [82].

The effective application of swarm-based algorithms by the scientific and business communities has demonstrated the worth of these methods in practice. The benefits of SI-based algorithms are the reasoning success of the algorithms mentioned before. Swarm-based optimization methods work with groups as a population and have some randomness during searching for a solution. Despite all optimization algorithms, there is no universal algorithm used to solve all optimization problems. Still, some algorithms outperform others in many types of optimization problems. The researchers are working to find an algorithm that outperforms other algorithms for most of the problems or find new algorithms that can solve unsolved problems.

II. SHRIKE BIRDS

A. SOURCES OF INSPIRATION FOR SHRIKES

The Laniidae family of passerine birds includes shrikes, which are distinguished by their propensity to impale their flesh on thorns after capturing insects, small birds, or animals. The shrikes are two genera with 34 species distributed throughout the world. In North America, there is a member of the Shrike family called Loggerhead Shrikes. Loggerhead shrikes, also called butcherbirds and migrating shrikes, reach a weight of roughly 48 grams [83], [84], and [85]. Within the Laniidae family, this remarkable bird is rather huge, and its large head may have contributed to its unique name. Males and females have similar appearances; it is difficult to distinguish between them. They have black, white, and grey markings on their bodies and a black mask that covers their eyes [86].

Over its range, the loggerhead's appearance varies slightly by region. Loggerheads eat mainly small vertebrates and small mammals. They live, migrate, eat in population, and use cooperative breeding [84]. Make nests on the trees; the female will deposit between four and seven eggs in a clutch, which she will then incubate for roughly sixteen days [86]. For a period of seventeen to twenty days, both parents are responsible for taking care of the nestlings. After leaving the nest, the young birds remain close to their parents for three weeks, during which time they get food from both parents, develop their flight, and at night return to be warmed by the parents. For more information, return to reference [85]. The population of the shrike bird life cycle is simulated in the Figure 2. There are three nests: A, P, and Q are the population of birds' nests; the nest AA parent will brood eggs at the nest (AB); the nestling will grow up and become adults ready to fly and later depend on themselves, the breeding and surviving of the birds of nest A shown from (A to C).

B. SHRIKE OPTIMIZATION ALGORITHM

Depending on the nesting and reproductive behavior of the shrike birds explained in the previous section, the shrikes live in a population out of the urban area; the population has many nests, and each nest starts with two birds as parents. The breeding and surviving behaviors of the shrikes were modeled by the Shrike Optimization Algorithm (SHOA). In Figures (3 and 4) the pseudo-code SHOA is presented. The pseudo-code clearly and simply describes the SHOA's execution process.

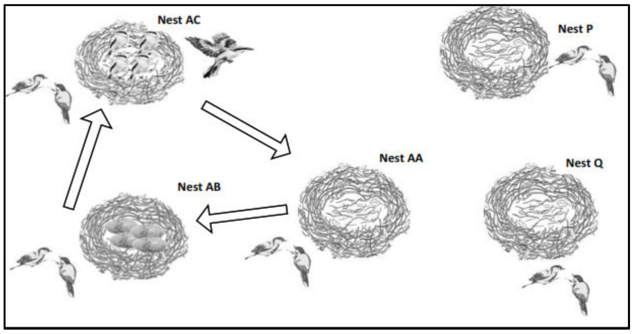

FIGURE 2. Shrike bird life cycle.

The SHOA start by initializing parameters, where N is the size of nests in the population, B is the number of eggs considered nestlings in each nest, and α is constant considered a natural factor affecting the bird. The search space of SHOA starts with a population of N nests, where each nest starts with two parent birds generated randomly. After the population is generated and nests are ready, and B number of nestlings will be generated. Population is represented as equation (1).

$$Population(N) = \begin{pmatrix} \begin{bmatrix} p_{im} & p_{if} \\ n_{ij} & n_{ij} \end{bmatrix} & \cdots & \begin{bmatrix} p_{im} & p_{if} \\ n_{ij} & n_{ij} \end{bmatrix} \\ \vdots & \ddots & \vdots \\ \begin{bmatrix} p_{im} & p_{if} \\ n_{ij} & n_{ij} \end{bmatrix} & \cdots & \begin{bmatrix} p_{im} & p_{if} \\ n_{ij} & n_{ij} \end{bmatrix} \end{pmatrix} \tag{1}$$

where a population like a pool has N nests, each element in the population represents a nest, each $nest_i$ has many solutions parent and nestling considered as a solution of the algorithm, where i = (1,2, …N), and parents are randomly generated using equation (2).

$$p_i = LB + rand(UB - LB) \tag{2}$$

In the initialization process, after two birds are generated as parents for each nest, the fittest will be selected as dominance M_{parent} is male parent, and other remains F_{parent} as a female parent. In the breading phase, every nest generates B nestling using equations (3) and (4). Where Δegg_j generate from both parents, and r is a random value in [-1, 1], then Δegg_j used to generate $nestling_j$, where i= (1,2, …B).

$$\Delta egg_j = (F_{parent} - M_{parent}) + r \tag{3}$$

$$nestling_j = F_{parent} + \Delta egg_j \tag{4}$$

The nestlings will depend on their parents; the male parent is dominant, which feeds the nestling, and the female, but the male feeds by itself only; the female also feeds by itself, and

```
Initialize parameters N , Max_Iteration , B, α , k
Initialize population nest_i = (i=1,2,..., N)
for each nest_i
        Generate P_j ( j=1,2) using equation (2)
while (! Max_Iteration)
        for each nest_i
          // every k generation check
        if (nest_i has only 2 birds)
                generate egg_j ( j=1,2,...,B) using Algorithm Fig(4)
        else
                choose best 2 birds and remove others
                 generate egg_j ( j=1,2,...,B) using Algorithm Fig(4)
        get M_parent , F_parent ( nest_i )
        for each bird_j in nest_i feed
                generate random r using equation (5)
                if (bird_j is Parent )
                        calculate Δfood_j using equation (6)
                else // phase 1 exploitation
                        calculate Δfood_j using equation (7)
                        calculate bird_j^(t+1) using equation (9)
                        if( Fit_j^(t+1) better than Fit_j^(t) ) keep bird_j^(t+1)
                        else
                                calculate Δfood_j using equation (8)
                                calculate bird_j^(t+1) using equation (9)
                        if( Fit_j^(t+1) better than Fit_j^(t) ) keep bird_j^(t+1)
                        else // phase 2 exploration
                            explore solution using equation (10)
        keep local best of nest_i
        end
        keep global best from all nest
endwhile
```

FIGURE 3. **Pseudo-code SHOA Algorithm.**

```
Initialize F_Parent , M_Parent , B = nestling size
For each nestling_j
        generate random r ∈ [-1,1]
        calculate Δfood_j using equation (3)
        calculate nestling_j using equation (4)
        calculate Fit_j
        add to nest
```

FIGURE 4. **Pseudo-code generates nestling steps.**

will feed nestling if they don't get food from the male parent. The idea of feeding nestling by a dominant parent leads the search to exploit solutions and converge to the optimum solution. Each nest has only two dominant solutions as a parent, they consider the first optimum and second optimum solution for the current nest, and each nestling gets feeds from the parent this is the solution exploit phase. In SHOA, after initialization nests and parameter parents should be specified depending on their objective function, and then r will be generated for each dimension using equation (5).

$$r = e^{-2xt/T_{max}} \tag{5}$$

The r parameter represents a natural factor in feeding, and the reason for the calculation of such a factor is to increase exploration. Where x is the dimension variable for bird_j, t is the current iteration, and T_{max} is the maximum iteration allowed for running SHOA. Then using equation (6) each parent bird will feed itself.

$$\Delta food_j = bird_j \times r \tag{6}$$

But for feeding nestlings, the Δ food is generated using formula (7), which $bird_j$ is the current bird state with a M_{parent} is male parent bringing food.

$$\Delta \text{food}_j = \text{r} \times (bird_j - M_{parent}) + M_{parent} \tag{7}$$

whereas the nestlings didn't survive by food from the male parent, then they tried to survive through the female parent using formula (8), the same as formula (7), but r $\in$ [-1, 1], and sin(α), where α is used as a constant factor.

$$\Delta \text{food}_j = \text{r} \times (bird_j - F_{parent}) + sin(\alpha) \tag{8}$$

After generating food, the birds' next status will be calculated using formula (9), which is the current state of birds getting food.

$$\text{bird}_j^{t+1} = \text{bird}_j^t + \Delta \text{food}_j \tag{9}$$

Calculate the fitness for each bird is better than the current state, then the current bird bird_j^t, will be updated with the new state as bird_j^{t+1}, the fittest one will survive for the next generation, not all birds get food at the same time. If any bird_j does not get food from its parent it will survive using equation (10) to generate Δfood_j, where r is randomly generated between [-1, 1] and another variable parameter α = rand [0, dimension], α used as random variable to increase randomization, the sine of the variable will change over time depending on different values, this step is exploring the space by finding new solutions far from the current state, and randomly searching other possibilities, in this phase, the current solution will diverge from local best to generate new solution far away from parent.

$$\text{bird}_j^{t+1} = \text{bird}_j^t + (\text{r} \times bird_j + sin(\alpha)) \tag{10}$$

The SHO algorithm keeps the best of each nest as the local best, then the population keeps the best from all local best as global. The idea of multi-modality can be solved using a group of solutions; where each nest has many birds, each k iteration the nest will regenerate after the old nestling finishes their nestling period time. The algorithm will keep just the two best birds as parents and remove all other birds

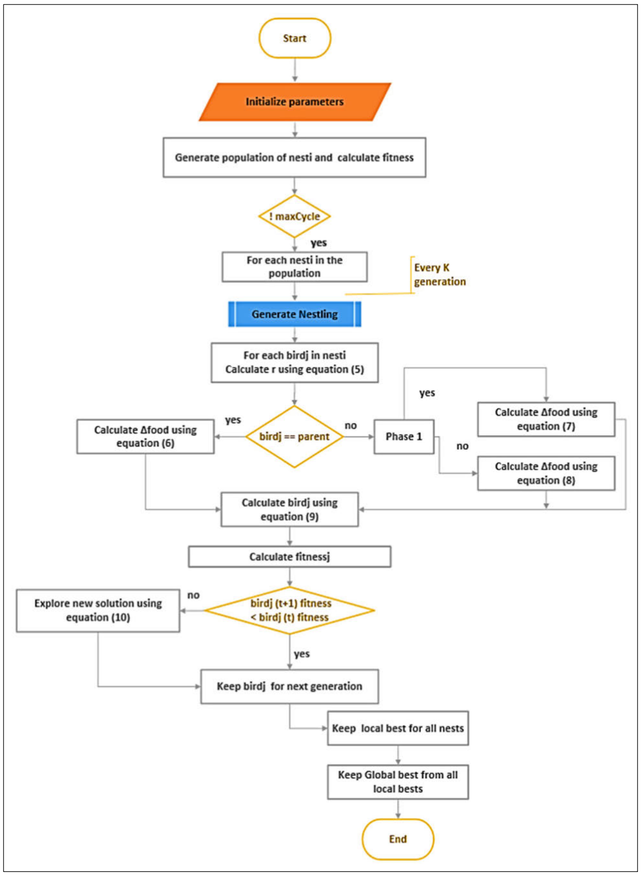

FIGURE 5. Flow chart of SHOA.

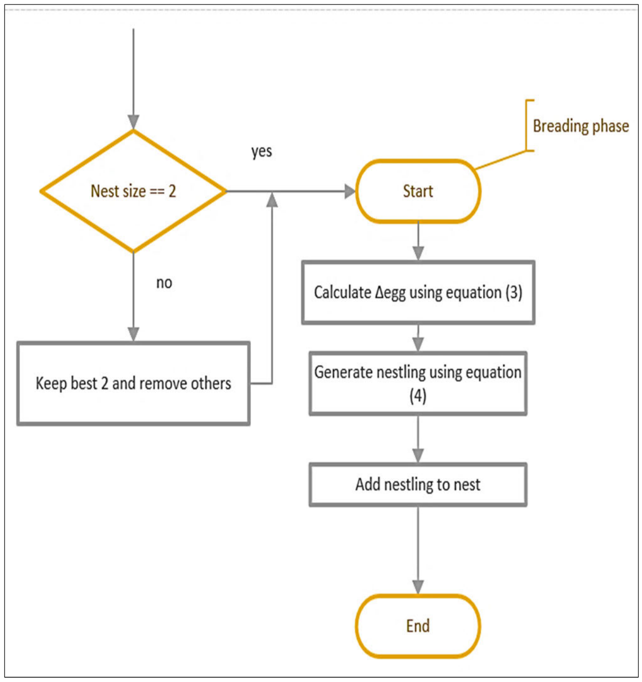

FIGURE 6. Flow chart of generate nestling.

as they die or fly far away from nests as they get ready to live independently. Every parent will generate a new nestling again and a new generation will update the current nest solution, the same algorithm execution process will continue till the stop condition.

Generation after generation of searches conducted using a randomly generated population. The Flow chart of SHOA specifies the process and SHOA's follow steps are presented in Figures (5, 6).

C. TIME AND SPACE COMPLEXITY

The computational time complexity of the SHOA encompasses the time and space complexity is taken into account. The time complexity of SHOA is affected by the initialization process and population updating as follows:

The algorithm initialization process requires O(N × B) time, where, as mentioned, N is the number of nest members in the population and each nest has B birds.

Updating and calculating each nested element as a solution of the algorithms requires M iterations to complete algorithm evaluation O ((N× B) × M), where M is the maximum iteration of an algorithm. For every dimension, the updating of members requires an O ((N× B)× M × d) time, where d is the dimension, for every k iteration SHOA had to choose two bests to regenerate nestling, the time considered as O (N log N). Overall time complexity:

O (N×B (1 + M (1 + d))) + O (N log N).

The space complexity of SHO depends mainly on the population size (N), the number of solutions in each nest (B), and the number of dimensions to be solved (d).

III. IMPLEMENTATION AND RESULTS

Number of global optimization test functions to show the performance of SHOA and the results compared with some well-developed optimization algorithms studied in the literature. Three groups of test functions are selected as uni-modal, multi-modal (simple, complex) [87], [88], [89], [90], and [91], 100-digit Challenge test functions [92], and highly complex benchmark of CEC 2022 (CEC22) [93] as a single objective-constrained bounded numerical optimization benchmark, each having a specific characteristic. The test functions were shifted and rotated by the values shown in Tables (19-23) in Appendix A to increase the complexity of the problems. The results compared with many optimization algorithms such as ANA [13], PSO [15], FDO [25], MFO [31], BKA [35], Fox [55], OOBO [67], and GA [94].

Despite the increasing complexity of the tested functions by rotating and shifting, all uni-modal test functions have a single optimum solution, while increasing the dimension will increase the problem difficulty and computation time to reach a globally optimum solution. The test functions F1-F7 are shown in the uni-modal function considered, while the multi-modal function has many local optima, which increases the difficulty of the algorithm to find an optimum solution because of trapping in local optima. The test functions

F8-13 multi-modal test functions are considered multi-modal problems for comparison, and the specification of functions, rotation, and shift values of the problems are specified in the table in Appendix Table 20. Composition functions are compounds of many functions with rotation, shifting, and add function bias. These functions are important as case studies because the properties of multi-functions are mixed like real-world problems, and they will show the performance of the algorithm in the exploration and exploitation search capability. The test functions F14-F19 mentioned in Table 21 composite functions had $f_{min} = 0$ shown, where σ is used to coverage range control of each f(x), and λ used for compress and stretch the function, all these are tested with composite functions in the current study. Recently, many competition functions have been provided by high-impact conferences to be used as comparison studies for competing for the performance of optimization algorithms. The 100-Digit functions challenge has 10 hard-solved problems as compound functions from the Society for Industrial and Applied Mathematics (SIAM), the purpose of solving such problem in this paper is to find the optimum solution within a specific time because, in the original paper, there is no time limitation for solve problems [92]. The problems are shown in Table 22 the Hundred-Digit Challenge basics with the range of x values, and dimensions of the test problems.

The CEC22 is also used to show the performance of SHOA. Four groups of test functions included as uni-modal just one function F1, basic multi-modal four functions F2-F5, hybrid multi-modal has only three functions F6-F9, and Composition multi-modal has four F9-F12, they are challenge test functions each having a specific characteristic. Test functions have shift, shuffle, and rotate with a matrix downloaded from a special session of conference link. The change of x values with rotation, shift and shuffling values, for hybrid functions will increase the problem complexity.

Numerical examples with dimensions specified in the tables, each test instance run 30 times. The SHOA runs with a population size set to 15, each nest starts with two solutions as a parent of the nest, parents breeding B eggs, nestling birds feeding by the parent during k generations of algorithm cycles, then best 2 keep for next generation, other birds removed from nest, 500 iterations specified for each turn the SHOA and selected optimization algorithm parameters shown in the Table 1. The statical results "Mean "represents the mean value and "Std" is the standard deviation over 30 rounds, the algorithm's extra parameters and specifications are summarized in the Table 1.

The proposed algorithm applies groups and subgroups, the concept designed for multi-modality, but uni-modal test functions are also benchmarked to show the performance of SHOA. The comparison F1, F2, and F7 are median when compared with other algorithms, but in all multi-modal functions have good performance in some show superiority over others. The comparative results mean and standard deviation are shown in Table 2. SHOA outperforms other powerful algorithms in hybrid multi-modal functions.

TABLE 1. Algorithm specifications.

All algorithms run under the same conditions: max iteration = 500 Number of agents in population = 30 Number of rounds = 30 Stop condition = maximum iteration Extra parameters are specified below:	
GA	Selection Roulette Wheel Linear combination Crossover arithmetic = 0.8 Mutation rate = 0.05
PSO	C1,C2 =(2,2) Inertia weight = 0.6 r ∈[-1,1]
MFO	Flame size = moth size All dimensions have the same bound a linearly decreases from -1 to -2
FDO	Initial Weight Factor = 0.0 r levy flight
ANA	r ∈[-1,1]
SHOA	B = 7 k = 50

In Table 3 comparison results for Hundred-Digit challenge problems of SHOA and other algorithms are shown, C01 and C06 are highly complex problems that need more execution time to solve, some algorithms have slow convergence to optimal like ANA, FDO, and PSO takes a lot of time because it has exploited every solution, furthermore PSO fails for C01 and C06 and ANA fails for C06, this proves that ANA and PSO's slow convergence rate during execution. Indeed, results in many test functions like (C01, and C06) show the novel SHOA is more powerful than other algorithms not only at the average value of 30 runs but at other statistical Std values also.

Furthermore, Table 4 shows the comparison result of CEC22 for all problems, SHOA's performance is low compared with other algorithms in Ce01, while in all others it has good performance. Once again, the Wilcoxon rank sum and Fridman test shown in Tables (5-11) demonstrated the statistical performance of SHOA in solving all test problem functions. Two algorithms are compared using the Wilcoxon rank sum test, the Wilcoxon rank sum test a nonparametric statistical test to determine the significant difference between the average of two data samples, is applied. In the Wilcoxon rank sum test, using an index called a p-value of 5%, it is determined whether the superiority of SHOA against any other algorithms is significant from a statistical point of view. While more than one algorithm is used to test for Fridman, any value that is not applicable has a sign (-) in Table (2, 3, or 4) Fridman and Wilcoxon tests are not considered for not applicable results.

The Wilcoxon rank sum is reported in Table 5-7. Based on these results, in cases where the p-value is less than 0.05,

TABLE 2. Comparison result of SHOA on 19 Test Instances (Uni, Multi, Composition) -modals with Algorithms MFO, FDO, FDO, FOX, ANA, PSO, GA, BKA, OOBO.

F	SHOA		MFO		FDO		FOX		ANA		PSO		GA		BKA		OOBO	
	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
F1	1.42E-01	7.85E-02	3.07E-11	3.68E-11	**1.50E-12**	7.89E-07	5.33E+02	5.33E+02	1.02E-03	2.03E-03	7.04E+03	1.85E+03	3.97E+03	1.04E+03	5.01E+02	1.77E+02	5.65E+03	1.29E+03
F2	4.38E-01	5.18E+00	**1.01E-07**	1.08E-07	1.04E-02	1.28E-03	9.71E+00	1.72E+00	3.49E-05	2.70E-05	5.45E+01	4.34E+01	1.85E+01	3.54E+00	4.71E+00	1.21E+00	1.77E+01	2.95E+00
F3	1.06E+01	9.47E-06	**4.59E-02**	6.36E-02	4.14E+00	1.06E-01	1.31E+03	4.94E+02	1.60E+00	1.23E+00	1.88E+04	1.13E+04	6.06E+03	1.28E+03	8.32E+02	2.36E+02	4.82E+03	8.42E+02
F4	8.51E-06	9.54E+00	**0.00E+00**	**0.00E+00**	9.14E-09	7.99E-03	9.66E-01	1.21E+00	1.15E-08	6.21E-08	1.06E+00	1.97E+00	2.91E-03	1.57E-02	8.75E-08	2.78E-07	7.68E-01	6.38E-01
F5	2.42E+01	1.80E-01	**6.58E+00**	1.97E+00	6.19E+01	4.47E+01	2.43E+05	1.55E+05	7.63E+01	2.23E+02	4.46E+07	0.00E+00	1.07E+07	4.52E+06	6.44E+04	4.10E+04	4.17E+06	3.21E+06
F6	**3.33E-02**	2.91E-01	4.22E+06	7.55E+01	4.22E+06	4.22E+06	4.81E+06	9.89E+04	4.22E+06	3.36E+02	5.15E+06	1.23E+05	5.03E+06	7.20E+04	4.38E+06	4.00E+04	4.75E+06	8.97E+04
F7	7.78E-01	2.33E+02	-	-	7.77E-01	6.22E-01	7.22E-01	3.18E-01	7.47E-01	**2.77E-01**	8.44E-01	3.66E-01	-	-	1.01E+00	4.07E-01	1.11E+00	4.62E-01
F8	-5.37E+03	3.28E+00	-5.36E+03	2.14E+02	-5.01E+03	-5.55E+03	-3.46E+03	2.69E+02	**-2.79E+06**	2.67E+05	-2.56E+03	2.62E+02	-2.69E+03	5.66E+02	-4.45E+03	3.05E+02	-3.01E+03	2.86E+02
F9	1.56E+01	1.01E-01	**8.60E+00**	5.66E+00	2.06E+01	1.19E+01	3.66E+01	4.26E+00	2.62E+01	3.51E+00	1.10E+02	1.90E+01	2.70E+01	2.82E+00	3.89E+01	6.46E+00	7.07E+01	9.00E+00
F10	3.67E-01	7.46E-02	2.70E-06	1.99E-06	7.55E-15	7.55E-15	2.45E+00	8.20E-01	5.63E-14	2.72E-13	1.76E+01	1.09E+00	**4.44E-16**	0.00E+00	7.52E+00	9.23E-01	1.61E+01	1.12E+00
F11	**2.36E-01**	5.74E-01	5.13E-01	6.93E-02	5.03E-01	5.88E-01	5.48E-01	1.42E-01	4.19E-01	6.72E-02	8.89E-01	8.26E-02	4.96E-01	8.22E-02	4.77E-01	9.74E-02	4.99E-01	8.74E-02
F12	8.39E-01	1.26E-02	**9.64E-02**	4.41E-01	5.89E+01	6.73E+00	3.76E+04	5.80E+04	3.94E+00	4.78E+00	3.32E+08	2.22E+08	3.34E+07	1.36E+07	2.09E+02	1.44E+02	1.26E+07	1.62E+07
F13	**4.09E-02**	2.94E-05	4.10E+09	7.62E+05	4.10E+09	4.10E+09	1.38E+10	4.37E+09	4.12E+09	8.31E+06	5.87E+10	1.40E+10	4.29E+10	7.63E+09	7.80E+09	9.76E+08	2.40E+10	6.11E+09
F14	8.32E-05	5.01E-06	**4.56E-16**	6.69E-16	2.36E-15	1.19E-14	2.64E-02	5.56E-03	2.73E-09	1.46E-08	4.33E-01	3.90E-01	9.73E-03	1.93E-03	3.04E-03	1.26E-03	2.80E-02	6.10E-03
F15	1.54E-05	1.69E-03	7.47E-10	5.59E-10	**8.88E-16**	7.77E-16	2.11E-01	2.27E-02	1.72E-08	1.49E-08	8.20E-02	2.10E-02	1.20E-01	1.70E-02	5.41E-02	1.18E-02	1.94E-01	1.55E-02
F16	5.35E-03	3.36E-01	2.35E-08	1.58E-08	**4.33E-15**	1.67E-15	1.02E+00	8.63E-03	4.20E-06	2.85E-06	1.06E+00	1.66E-02	9.49E-01	3.40E-02	7.62E-01	8.73E-02	1.03E+00	8.46E-03
F17	**4.53E+00**	8.94E-02	2.38E+01	7.94E-02	2.40E+01	2.37E+01	2.40E+01	1.88E-01	2.38E+01	5.36E-02	3.23E+01	6.90E+00	2.44E+01	4.90E-01	2.37E+01	6.19E-02	2.39E+01	1.38E-01
F18	**3.21E+00**	6.30E-02	2.24E+02	3.93E-03	2.24E+02	2.24E+02	2.24E+02	1.66E-02	2.24E+02	6.62E-03	2.24E+02	6.03E-02	2.24E+02	2.14E-02	2.24E+02	4.69E-03	2.24E+02	9.34E-03
F19	**3.80E+00**	0.00E+00	3.15E+01	6.51E-03	3.15E+01	3.15E+01	3.20E+01	2.04E-01	3.15E+01	2.19E-02	4.72E+01	1.48E+01	3.35E+01	8.66E-01	3.18E+01	9.61E-02	3.19E+01	1.58E-01

TABLE 3. Comparison result of SHOA on Hundred- Digit CEC 2019 test instances with algorithms MFO, FDO, FDO, FOX, ANA, PSO, GA, BKA, OOBO.

F	SHOA		MFO		FDO		FOX		ANA		PSO		GA		BKA		OOBO	
	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
C1	**2.72E-01**	5.29E-06	2.04E+05	3.02E+05	4.59E+03	2.07E+04	2.71E+05	3.57E+05	2.15E+06	2.08E+06	-	-	8.05E+03	3.97E+03	2.85E+06	2.38E+06	1.11E+07	6.99E+06
C2	**3.00E+00**	2.41E-01	4.00E+00	0.00E+00	4.00E+00	2.58E-05	4.78E+00	1.16E+00	4.00E+00	3.98E-12	3.89E+02	2.98E+02	4.68E+00	3.70E-01	5.35E+00	1.33E+00	4.00E+00	3.74E-04
C3	**3.13E+00**	6.86E+00	1.37E+01	4.88E-13	1.37E+01	9.89E-08	1.37E+01	2.45E-05	1.37E+01	1.76E-10	1.37E+01	1.29E-03	1.37E+01	4.52E-04	1.37E+01	4.67E-06	1.37E+01	1.66E-04
C4	4.36E+01	3.79E-02	2.94E+01	1.11E+01	**1.17E+00**	8.47E-02	8.01E+03	2.71E+03	4.25E+01	9.90E+00	2.47E+04	8.39E+03	1.29E+04	3.38E+03	6.56E+02	2.30E+02	6.33E+03	1.93E+03
C5	**1.33E-01**	7.06E-01	1.12E+00	1.31E-01	2.14E+00	8.72E-02	3.62E+00	4.29E-01	1.20E+00	8.45E-02	6.68E+00	0.00E+00	4.30E+00	4.98E-01	2.16E+00	6.94E-02	3.49E+00	5.10E-01
C6	**7.97E+00**	4.15E+00	1.21E+01	8.36E-01	1.21E+01	6.10E-01	1.21E+01	6.12E-01	-	-	-	-	1.24E+01	7.97E-01	1.18E+01	8.18E-01	1.22E+01	5.68E-01
C7	1.19E+02	3.73E-01	**1.06E+02**	3.48E+00	1.22E+02	1.43E+01	4.52E+02	7.22E+01	1.16E+02	8.17E+00	6.01E+02	6.04E+01	4.75E+02	3.52E+01	2.36E+02	2.70E+01	3.72E+02	3.68E+01
C8	**2.52E+00**	6.71E-03	4.96E+00	5.77E-01	5.14E+00	9.00E-01	5.85E+00	4.76E-01	5.53E+00	5.12E-01	8.25E+00	5.59E-01	6.33E+00	4.19E-01	6.07E+00	3.98E-01	6.28E+00	2.74E-01
C9	**1.02E+00**	8.88E-16	2.00E+00	9.11E-12	2.00E+00	1.75E-06	2.95E+02	1.43E+02	2.00E+00	5.52E-04	8.00E+02	2.08E+02	2.52E+02	5.89E+01	2.74E+01	1.04E+01	2.41E+02	6.19E+01
C10	**1.72E+00**	0.00E+00	2.72E+00	4.44E-16	2.72E+00	4.44E-16	2.72E+00	4.44E-16	2.72E+00	4.44E-16	2.72E+00	4.44E-16	2.72E+00	4.44E-16	2.72E+00	4.44E-16	2.72E+00	4.44E-16

TABLE 4. Comparison result of SHOA on CEC 2022 test instances with algorithms MFO, FDO, FDO, FOX, ANA, PSO, GA, BKA, OOBO.

F	SHOA		MFO		FDO		FOX		ANA		PSO		GA		BKA		OOBO	
	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Ce1	1.84E+03	8.95E+01	3.44E+04	6.54E+03	1.83E+03	2.44E+03	2.45E+04	3.34E+03	**3.78E+01**	2.50E+01	2.01E+06	7.61E+06	5.57E+04	1.18E+04	1.40E+04	2.54E+03	3.62E+04	5.14E+03
Ce2	**4.51E+02**	1.42E-02	4.07E+03	6.86E+02	5.67E+00	3.75E+00	2.61E+03	3.97E+02	3.25E+01	1.86E+01	6.35E+03	1.02E+03	4.32E+03	4.20E+02	1.27E+03	2.10E+02	4.38E+03	9.01E+02
Ce3	5.39E-02	1.21E+01	1.59E+00	1.70E-01	1.99E-05	6.34E-05	1.02E+00	1.30E-01	**4.33E-06**	4.99E-06	2.14E+00	3.09E-01	1.80E+00	1.53E-01	3.54E-01	7.92E-02	1.74E+00	2.85E-01
Ce4	9.97E+01	4.27E-01	2.11E+02	1.41E+01	**6.71E+01**	2.40E+01	1.73E+02	1.82E+01	1.32E+02	1.16E+01	2.31E+02	1.86E+01	1.91E+02	1.29E+01	1.62E+02	8.32E+00	2.26E+02	1.63E+01
Ce5	1.87E+00	1.11E+01	1.35E+01	2.21E+00	1.45E+00	9.24E-01	8.07E+00	1.85E+00	1.88E+00	7.55E-01	1.78E+01	0.00E+00	8.53E+00	1.26E+00	4.98E+00	1.16E+00	1.56E+01	2.16E+00
Ce6	**7.13E+01**	1.52E+00	9.38E+08	3.74E+08	1.48E+04	5.08E+04	1.06E+09	1.25E+08	8.12E+02	2.90E+03	2.27E+03	8.67E+02	1.09E+03	3.04E+02	2.21E+02	5.19E+01	1.02E+03	3.22E+02
Ce7	2.74E+01	3.18E+00	5.02E+02	1.72E+02	4.06E+01	1.32E+01	1.47E+03	6.82E+02	3.43E+01	3.52E+00	7.19E+01	1.43E+01	4.77E+01	4.69E+00	3.06E+01	2.77E+00	4.58E+01	4.72E+00
Ce8	**2.39E+01**	3.72E+00	3.36E+02	1.11E+02	2.81E+01	5.04E+00	4.04E+02	1.71E+02	2.94E+01	1.39E+00	6.38E+01	1.32E+01	4.32E+01	3.41E+00	2.63E+01	1.70E+00	3.62E+01	2.01E+00
Ce9	**1.91E+02**	9.48E-02	3.70E+02	3.97E+01	**1.77E+02**	1.54E+00	7.67E+02	3.16E+02	1.78E+02	1.79E+00	1.31E+03	4.94E+02	7.81E+02	1.45E+02	2.47E+02	1.68E+01	4.10E+02	5.11E+01
Ce10	1.01E+02	3.03E+01	7.41E+02	6.39E+02	2.06E+01	4.37E+01	2.00E+03	8.83E+02	**4.19E+01**	5.07E+01	5.48E+03	3.03E+02	1.59E+03	8.85E+02	5.14E+02	2.68E+02	3.44E+02	3.26E+02
Ce11	**1.33E+02**	2.57E+00	2.86E+03	5.99E+02	5.48E+02	5.61E+02	5.86E+03	9.92E+02	4.60E+02	4.23E+02	7.85E+03	7.27E+02	4.28E+03	6.19E+02	1.54E+03	2.88E+02	3.04E+03	5.92E+02
Ce12	**2.26E+02**	0.00E+00	4.15E+02	2.37E+01	2.79E+02	5.04E+00	5.72E+02	6.44E+01	2.87E+02	5.65E+00	7.08E+02	7.19E+01	3.43E+02	1.46E+02	3.52E+02	1.11E+01	4.38E+02	2.70E+01

SHOA has a significant statistical superiority compared to the corresponding algorithm.

In Table 7 the test results show the statistical difference between SHOA with correspondence algorithm specified in, F2, and F7 has p-value $> 0.05\%$ while all others are < 0.05, furthers more in Table 6, C4 with ANA, and C7 with FDO have p-value > 0.05, in Table 7 the functions Ce05, and Ce06 with ANA also have p-vales > 0.05. The Fridman statistical test is reported in Tables (8-10). In Table 8 Fridman's results of unimodal functions are presented, MFO takes first place the lowest is the best, and SHOA takes fourth place as it runs median, while MFO is inapplicable to find a solution for F7 under the same conditions which other algorithms found. In Table 9 as all test functions are multi-modal mean ranking of Fridman shows that SHOA takes first place followed by ANA. MFO, FDO, BKA, GA, FOX, OOBO, AND the last PSO.

Hundred-Digit CEC19 test case Fridman results show SHOA taking first ranking place followed by MFO, FDO, ANA, BKA, FOX, OOBO, GA, and PSO. Comparison of Fridman for the last test instances presented in Tables 10 and 11 for both CEC19 and CEC22 respectively, in the overall mean ranking has the lowest value, but in all cases, it performs well even in medium and good performance for all cases.

Furthermore, Figure 7 reports the best solution found during the search space of SHOA with all competitive algorithms selected for Function (F1-F19), the algorithms convergence curve shows smoothly converges to the best solution.

In Figure 8, the 100-digit problems' convergence curve is presented to show the searching space and converge to the optimal solution, some algorithms converge to the optimal solution from the first 25 iterations, while others need more. MFO has fast convergence in C01, C02. SHOA also

TABLE 5. Wilcoxon ranking sum test SHOA on 19 different modal test instances against algorithms SHOA vs (MFO, FDO, FDO, FOX, ANA, PSO, GA, BKA, OOBO).

F	MFO	FDO	FOX	ANA	PSO	GA	BKA	OOBO
F1	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
F2	~ 0.001	**0.673**	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
F3	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
F4	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
F5	~ 0.001	~ 0.01	~ 0.001	0.013	~ 0.001	~ 0.001	~ 0.001	~ 0.001
F6	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
F7	**-**	**0.797**	**0.861**	**0.491**	0.01	**-**	**0.861**	0.003
F8	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
F9	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
F10	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
F11	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
F12	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
F13	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
F14	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
F15	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
F16	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
F17	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
F18	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
F19	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001

TABLE 6. Wilcoxon ranking sum test SHOA on Hundred Digit CEC 2019 test instances against Algorithms SHOA vs (MFO, FDO, FDO, FOX, ANA, PSO, GA, BKA, OOBO).

F	MFO	FDO	FOX	ANA	PSO	GA	BKA	OOBO
C1	~ 0.001	~ 0.001	~ 0.001	~ 0.001	**-**	~ 0.001	~ 0.001	~ 0.001
C2	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
C3	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
C4	~ 0.001	~ 0.001	~ 0.001	**0.629**	~ 0.001	~ 0.001	~ 0.001	~ 0.001
C5	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
C6	~ 0.001	~ 0.001	~ 0.001	~ 0.001	**-**	~ 0.001	~ 0.001	~ 0.001
C7	~ 0.001	0.797	~ 0.001	0.014	~ 0.001	~ 0.001	~ 0.001	~ 0.001
C8	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
C9	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
C10	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001

TABLE 7. Wilcoxon ranking sum test SHOA on CEC 2022 test instances against Algorithms SHOA vs (MFO, FDO, FDO, FOX, ANA, PSO, GA, BKA, OOBO).

F	MFO	FDO	FOX	ANA	PSO	GA	BKA	OOBO
Ce1	~ 0.001	**0.094**	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
Ce2	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
Ce3	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
Ce4	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
Ce5	~ 0.001	0.011	~ 0.001	**0.975**	~ 0.001	~ 0.001	~ 0.001	~ 0.001
Ce6	~ 0.001	~ 0.001	~ 0.001	**0.558**	~ 0.001	~ 0.001	~ 0.001	~ 0.001
Ce7	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
Ce8	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
Ce9	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
Ce10	~ 0.001	~ 0.001	~ 0.001	0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
Ce11	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001
Ce12	~ 0.001	~ 0.001	~ 0.001	~ 0.001	~ 0.001	0.021	~ 0.001	~ 0.001

TABLE 8. Fridman mean ranking SHOA on 7 single modal test instances against Algorithms MFO, FDO, FDO, FOX, ANA, PSO, GA, BKA, OOBO.

F	SHOA	MFO	FDO	FOX	ANA	PSO	GA	BKA	OOBO
F1	4.00	1.17	2.00	5.60	2.83	8.67	7.27	5.40	8.07
F2	3.73	1.00	3.30	5.97	2.13	9.00	7.57	4.87	7.43
F3	3.93	1.23	2.00	5.83	2.83	8.93	7.90	5.17	7.17
F4	5.23	1.87	5.73	8.07	2.25	7.70	2.05	4.03	8.07
F5	3.03	1.20	3.57	5.93	2.20	9.00	7.80	5.07	7.20
F6	1.00	3.00	3.00	6.70	3.00	8.83	8.10	5.00	6.37
F7	3.67	-	3.70	3.53	3.80	4.80	-	3.57	4.93
Mean Rank	24.6	**9.47**	23.3	41.63	**19.05**	56.93	40.68	33.10	49.23
	4	**1**	3	7	**2**	9	6	5	8

TABLE 9. Fridman mean ranking SHOA on 14 different multi-modal test instances against algorithms MFO, FDO, FDO, FOX, ANA, PSO, GA, BKA, OOBO.

F	SHOA	MFO	FDO	FOX	ANA	PSO	GA	BKA	OOBO
F8	2.62	2.73	4.02	6.13	1.00	8.47	8.03	4.63	7.37
F9	2.23	1.30	2.80	6.37	4.43	8.93	4.50	6.37	8.07
F10	5.00	4.00	2.57	6.00	2.43	8.90	1.00	7.00	8.10
F11	1.03	5.20	6.73	5.73	3.03	8.90	4.90	4.57	4.90
F12	2.10	1.03	4.00	5.90	2.90	9.00	7.77	5.10	7.20
F13	1.00	2.50	2.50	6.03	4.00	8.77	8.13	5.10	6.97
F14	4.00	1.13	2.10	7.50	2.77	8.90	6.00	5.00	7.60
F15	4.00	2.03	1.03	8.70	2.93	5.83	6.97	5.20	8.30
F16	4.00	2.00	1.00	7.23	3.00	8.97	5.97	5.03	7.80
F17	1.00	3.70	4.50	6.67	3.55	8.97	7.57	3.32	5.73
F18	1.00	7.00	7.00	4.00	2.00	7.00	7.00	7.00	3.00
F19	1.00	2.95	2.95	6.50	3.10	9.00	8.00	5.57	5.93
Mean Rank	**28.98**	35.58	41.20	76.77	35.15	101.63	75.83	63.88	80.97
Rank	**1**	3	4	7	2	9	6	5	8

TABLE 10. Fridman mean ranking SHOA on Hundred CEC 2019 test instances against Algorithms MFO, FDO, FDO, FOX, ANA, PSO, GA, BKA, OOBO.

F	SHOA	MFO	FDO	FOX	ANA	PSO	GA	BKA	OOBO
C01	1.03	4.60	2.10	4.23	6.53	#	3.20	6.47	7.83
C02	1.00	3.00	3.00	6.53	3.00	9.00	7.07	7.40	5.00
C03	1.00	4.00	4.00	7.00	4.00	9.00	4.00	4.00	8.00
C04	3.45	2.37	1.00	6.77	3.18	8.90	7.90	5.00	6.43
C05	1.00	2.25	4.37	6.77	2.75	8.90	7.73	4.63	6.60
C06	2.00	5.62	5.58	5.37	1.00	8.93	6.32	4.52	5.67
C07	3.17	1.22	2.95	7.30	2.67	8.87	7.63	5.00	6.20
C08	1.00	2.95	3.57	5.00	4.02	9.00	6.90	5.97	6.60
C09	1.00	3.00	3.00	7.47	3.00	8.93	6.80	5.03	6.77
C10	1.00	7.00	7.00	3.00	7.00	3.00	7.00	7.00	3.00
Mean Rank	**15.65**	36.00	36.57	59.43	37.15	74.53	64.55	55.02	62.10
Rank	**1**	2	3	6	4	9	8	5	7

TABLE 11. Fridman mean ranking SHOA on CEC22 test instances against Algorithms MFO, FDO, FDO, FOX, ANA, PSO, GA, BKA, OOBO.

F	SHOA	MFO	FDO	FOX	ANA	PSO	GA	BKA	OOBO
Ce01	2.73	6.40	2.27	5.13	1.00	9.00	7.87	4.00	6.60
Ce02	3.00	6.77	1.00	5.07	2.00	8.97	7.07	4.00	7.13
Ce03	3.00	6.57	1.13	5.03	1.87	8.60	7.47	4.00	7.33
Ce04	1.90	7.30	1.17	4.90	3.03	8.40	5.95	4.32	8.03
Ce05	2.23	7.33	1.73	5.33	2.07	8.67	5.65	4.02	7.97
Ce06	1.87	8.37	4.57	8.63	2.00	6.40	5.13	3.20	4.83
Ce07	1.23	8.13	3.98	8.87	3.05	6.93	5.37	2.30	5.13
Ce08	1.40	8.43	2.87	8.57	3.53	7.00	5.93	2.30	4.97
Ce09	3.00	5.30	1.37	7.37	1.63	8.70	7.77	4.00	5.87
Ce10	2.63	5.67	1.42	7.13	1.95	9.00	6.97	5.37	4.87
Ce11	1.47	5.37	2.40	7.87	2.30	8.97	7.00	3.90	5.73
Ce12	1.50	5.77	2.62	8.02	3.38	8.92	4.00	4.50	6.30
Mean Rank	**25.97**	81.40	26.52	81.92	27.82	99.55	76.17	45.90	74.77
Rank	**1**	7	2	8	3	9	6	4	5

converges to optimal at the first quarter of iterations. Figure 9 presents convergence curve of CEC22 test instance, some lines appear in some functions meaning multiple algorithms had same value or near value, the line at the top hides others.

The authors of JDE100 [95] run 50 runs for each function with a different initial population but only the best 25 are selected for the final result shown in Table 12, while in SHOA 30 consecutive runs were implemented with different initial populations and all used in the resulted table, and SHOA run on 5e+02 maximum function evaluation, but JDE100 maximum evaluation is 1e+12. In all function comparisons, SHOA performs less mean and std than JDE100. The results are shown in Table 12. The Table 13 shows the performance of

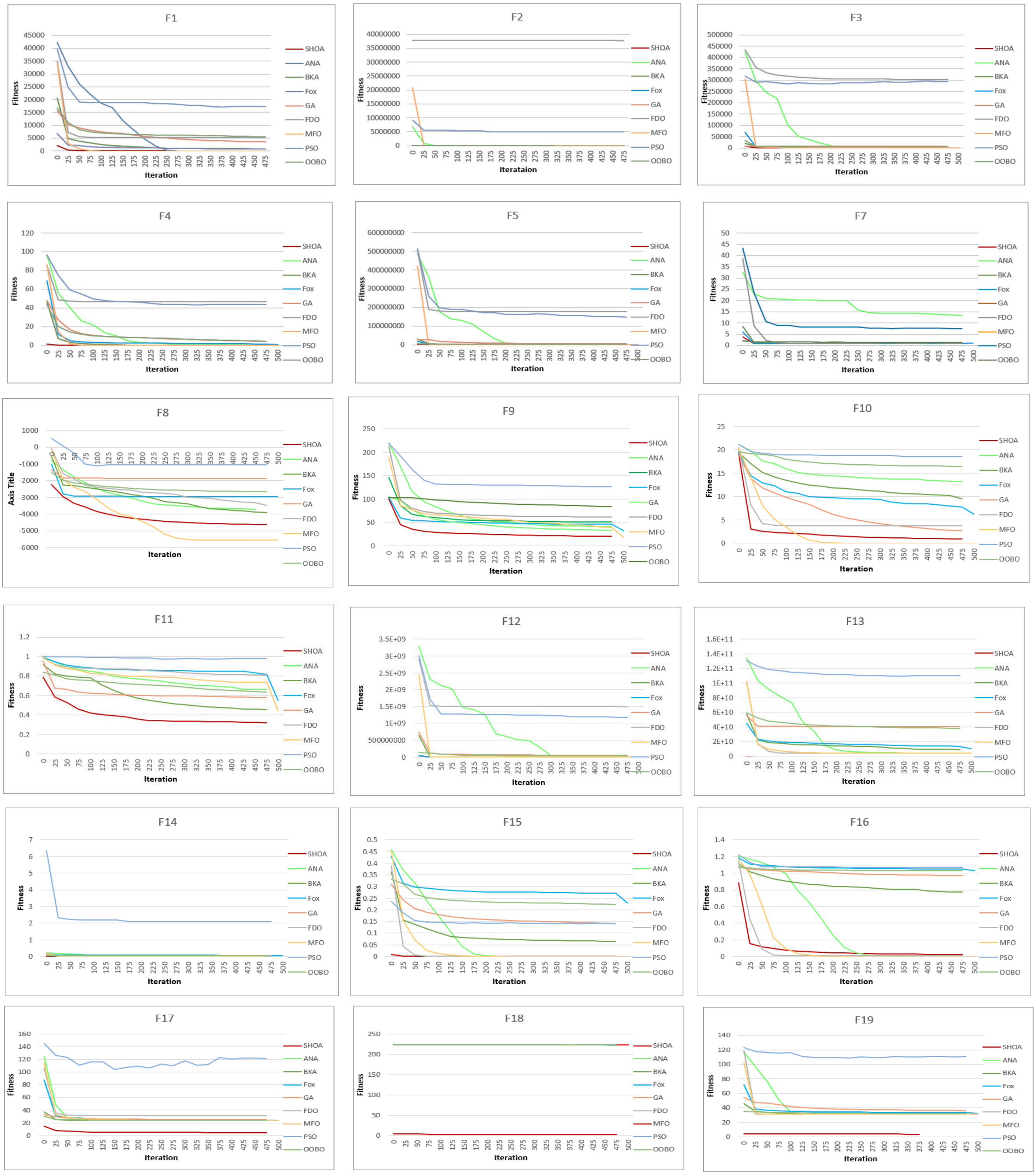

FIGURE 7. Convergence curve F1-F19.

SHOA for CEC22 benchmarks for each (2, 10, 20) dimension, where D = 20 has already been studied but the Table above shows a comparison, there are no application results for F6, F7, F8 in two dimensions because of hybrid functions need more dimensions. The performance of SHOA wouldn't be different with high dimensions when comparing between 10 and 20 dimensions, but overall algorithm performance will be high in small dimensions like D = 2.

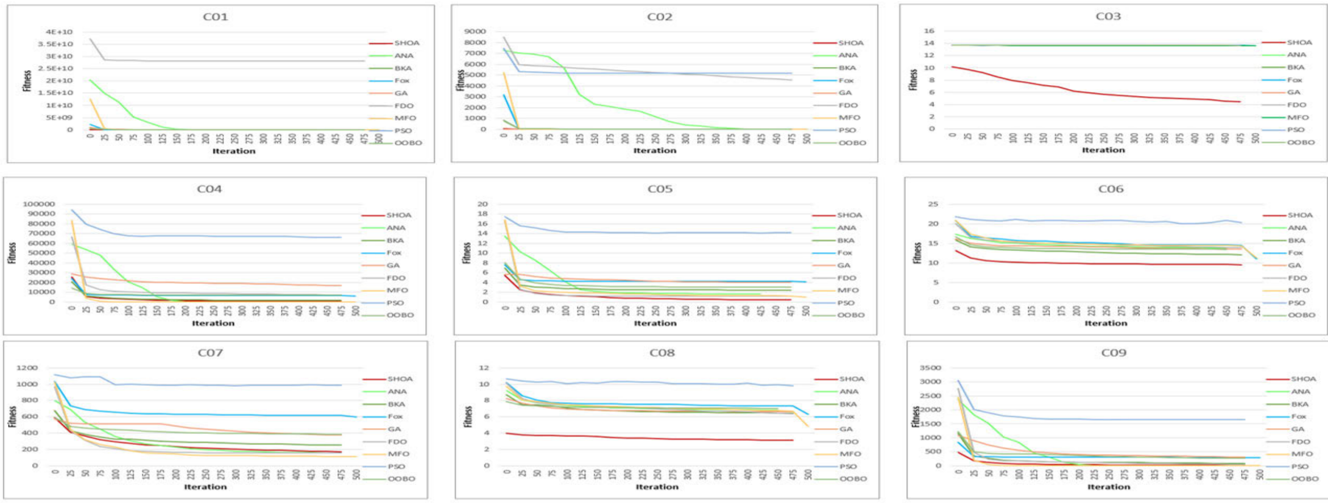

FIGURE 8. Convergence Curve CEC 2019.

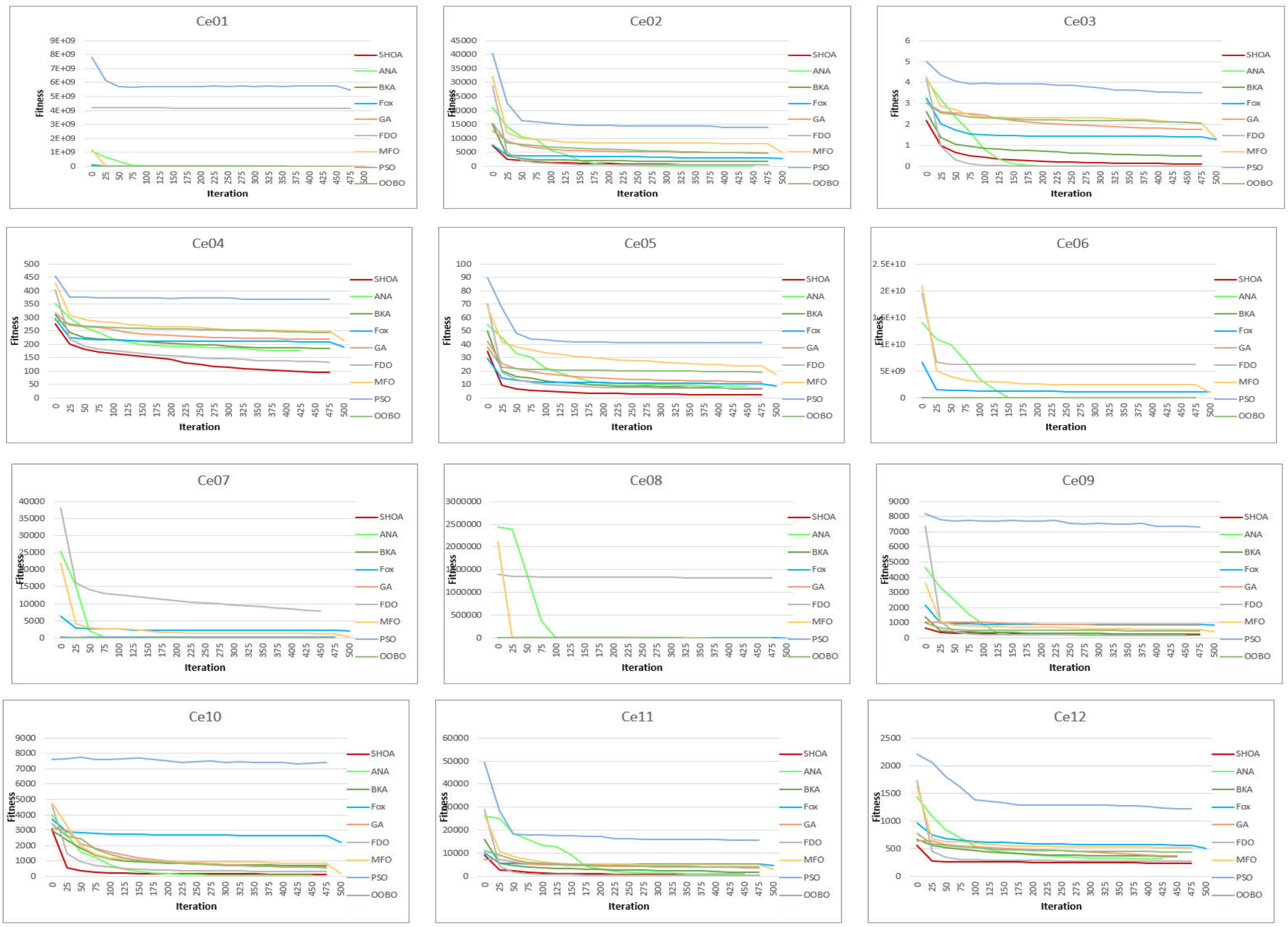

FIGURE 9. Convergence Curve CEC22 test instances.

Table 14 presents the comparison results of different nesting sizes (4 to 7) and selects samples from the CEC22 instance test cases for D = 10, due to the challenge of achieving an optimal or semi-optimal solution through multiple

TABLE 12. Comparison of SHOA with winner CEC 2019.

Alg.	SHOA		JDE100	
F	Mean	Std	Mean	Std
C01	2.18E-01	2.18E-01	1.59E+05	1.597E+05
C02	3.00E+00	3.00E+00	2.385E+06	2.719E+04
C03	3.13E+00	3.13E+00	1.31E+06	8.519E+05
C04	4.36E+01	4.36E+01	3.475E+05	1.149E+05
C05	1.33E-01	1.33E-01	1.673E+05	8.426E+04
C06	7.97E+00	7.97E+00	3.841E+04	2.063E+03
C07	1.19E+02	1.19E+02	9.105E+06	4.528E+06
C08	2.52E+00	2.52E+00	1.219E+09	4.388E+08
C09	1.02E+00	1.02E+00	9.207E+08	1.131E+08
C10	1.72E+00	1.72E+00	1.541E+06	7.46E+05

TABLE 13. Compare values Of CEC 2022 for different dimension.

Alg.	D=2	D=10	D=20
F	Mean	Mean	Mean
Ce1	1.09E-07	1.63E+00	1.84E+03
Ce2	1.09E-07	2.45E-01	4.51E+02
Ce3	1.07E-10	1.18E-05	5.39E-02
Ce4	4.01E-05	1.21E+01	9.97E+01
Ce5	4.18E-09	6.97E-03	1.87E+00
Ce6	-	6.22E-01	7.13E+01
Ce7	-	4.16E+00	2.74E+01
Ce8	-	4.16E+00	2.39E+01
Ce9	9.28E-03	1.87E+02	1.91E+02
Ce10	-1.14E+01	1.01E+02	1.01E+02
Ce11	1.57E-02	1.69E+01	1.33E+02
Ce12	8.91E-01	1.92E+02	2.26E+02

TABLE 14. Comparison results of CEC22 with different nestling for D=10.

Nest size	metric	B=4	B=5	B=6	B=7
Ce1	Mean	3.06E+00	2.21E+00	2.06E+00	1.63E+00
	std	1.70E+00	1.08E+00	9.97E-01	6.54E-01
Ce6	Mean	7.12E-01	9.08E-01	8.39E-01	6.22E-01
	std	5.04E-01	5.13E-01	4.54E-01	4.28E-01
Ce12	Mean	1.94E+02	1.94E+02	1.93E+02	1.92E+02
	std	1.46E+00	1.94E+02	1.45E+00	1.30E+00

algorithms at the appropriate time. The results show that with increasing nest size, mean and std with a single model were improved, while multi-modal was complex and difficult to solve; no significant difference was found.

IV. ENGINEERING PROBLEMS SOLVING

In this study, four constrained engineering problems, namely three-bar truss design, gear train design, antenna array design, and frequency-modulated sound wave design, are considered to investigate the applicability of SHOA. The problems have equality and inequality constraints, the SHOA should be equipped with the constrained solutions. Although, in constraint problem solving there will be feasible and infeasible solutions, to investigate infeasible solutions, some algorithms use penalty functions [96], in this study the death penalty is used, and the infeasible solutions are discarded and not investigated with a penalty to speed up the algorithm process. It is worth noting that the population size is set to 15, and iterations set to 500, for 30 rounds for all the problems in this section.

A. GEAR TRAIN DESIGN PROBLEM SOLVING

The gear train design is a mechanical engineering problem, the main objective is to minimize the desired ratio with the current ratio [97], the objective function was formulated as follows:

$$f(\vec{x}) = \left(\frac{1}{6.931} - \frac{G_a G_b}{G_c G_d}\right)^2 \tag{11}$$

where $\frac{1}{6.931}$ desired ratio, Ga, Gb, Gc, Gd teeth of gears A, B, C, and D respectively, with the ratio is:

$$\text{Gear Ratio} = \frac{G_a G_b}{G_c G_d} \tag{12}$$

subject to: $\forall \{G_i,\ 12 \le G_i \le 60\}$, where G_i is teeth of Ga, Gb, Gc, Gd.

In Table 15 for SHOA with AZOA [98], MFO, Non-Linear (NL) [97], and Cuckoo Search (CS) [99] shown, the table presents gear teeth of A, B, C, and D, optimal, and ratio (x) as comparison parameters, where the ratio must be closer to (1/6.931). In this study, SHOA had high performance over other algorithms, MFO has good optimal error but the (ratio > 1.442) CS also performed well in the second stage, but AZOA had bad performance because their ratio rates as constraints were not satisfied.

TABLE 15. Comparative result gear train design problem.

Algorithm	x_1 (G_a)	x_2(G_b)	x_3 (G_c)	x_4 (G_d)	Optimal Error	Ratio
SHOA	27.47	16.60	57.04	55.41	3.21E-20	0.1442
NL	18	22	45	60	5.70E-04	0.1466
CS	19	16	43	49	2.70E-12	0.1442
AZOA	60	17.52	12	24.29	0	3.606
MFO	43	19	16	49	2.70E-12	1.042

B. THREE-BAR TRUSS DESIGN PROBLEM SOLVING

The three-bar truss problem is a civil engineering design problem whose objective is to achieve the minimum weight subjected to stress, deflection, and buckling constraints and evaluate the optimal cross-sectional area ($A1$, $A2$). Mathematically, to minimize the weight of a three-bar truss construction, according to [100], an objective function and constraints are formulated as follows:

$$\text{Minimize}\, f(x) = \left(2\sqrt{2}\, x_1 + x_2\right) \times l \tag{13}$$

Subject to:

$$C_1(x) = \frac{\sqrt{2}\, x_1 + x_2}{\sqrt{2}\, x^2{}_1 + 2x_1 x_2} P - \sigma \le 0 \tag{14}$$

$$C_2(x) = \frac{x_2}{\sqrt{2}\, x^2{}_1 + 2x_1 x_2} P - \sigma \le 0 \tag{15}$$

$$C_3(x) = \frac{1}{\sqrt{2}\, x_2 + x_1} P - \sigma \le 0 \tag{16}$$

$\forall i, 0 \leq x_i \leq 1$, *where* $i = 1, 2$, the constant parameters are: $l = 100\ cm, P = 2\ KN/cm^2, \sigma = 2\ KN/cm^2$.

The comparative result shown in Table 16 presents the performance of the SHOA compared with AZOA, CS, MFO, and engineering design optimization by (Ray T, and Saini) TSa [100] algorithms, SHOA had tested nearly 20 times in 30 rounds all minimum fitness was between (263.89, and 263.90) and maximum fitness was between (263.94, and 263.97). Table 16 shows the performance of SHOA either better or equal other algorithms.

TABLE 16. Comparative result three-bar truss design problem.

Algorithm	x_1 [A_1]	x_2 [A_2]	Optimal weight
SHOA	0.788235	0.409503	263.8968
TSa	0.795	0.395	264.3
CS	0.788670	0.40902	263.9716
AZOA	0.7885471	0.408610	263.8958
MFO	0.7882447	0.4094669	263.8959

C. ANTENNA SPACED ARRAY PROBLEM SOLVING

Optimization of antenna arrays means reducing the side-lobe level (SLL) of a non-uniformly spaced linear array. The fitness value for the problem has been formulated to the maximum SLL to optimize the non-uniformly spaced array [25], and [101]. Objective function and constraints are formulated as follows:

$$f(\vec{x}) = max[20\ log\ |G(\theta)|] \tag{17}$$

where the:

$$G(\theta) = \sum\nolimits_{i=1}^{n} cos\ [2\pi x_i\,(cos\ \theta - cos\ \theta_s)] + cos\,[2.25 \times 2\pi\ (cos\ \theta - cos\ \theta_s)] \tag{18}$$

$n = 4$ *and* $\theta = 45°, \theta_b = 90°$

Subject to:

$$d = |xi - xj| > 0.25\lambda \tag{19}$$

$$0.125\lambda < min\ x_i \leq 2.0\lambda$$
$$x_i \in (0, 2.25)\ , i = 1, 2, 3, 4. i \neq j \tag{20}$$

To minimize SLL, the element should optimize without violation of above constraints above, where θ *is elevation angle*, *and* θ_b *is beam angle*, xi which is an element of the antenna must be greater than 0.125λ, the distance between elements must be more than 0.25λ.

The comparative assessment in Table 17 shows an optimal value between SHOA and other algorithms that applied antenna array space problem, the SHOA found a minimum optimal out of 30 rounds as shown in Table 17, and the maximum optimal value was (-177.46) with parameters (1.237, 0.789, 1.513, 0.432), the assessment shows the superiority of SHOA in all rounds when compared with FDO and ANA.

TABLE 17. Comparative result antenna spaced array problem.

Algorithm	P_1	P_2	P_3	P_4	Optimal SLL
SHOA	0.584	1.288	1.865	1.599	-280.491
FDO	0.713	1.595	0.433	0.130	-120
ANA	1.5959	0.3081	0.8747	0.6072	-70.720

TABLE 18. Comparative result of frequency-modulated sound wave.

Alg.	a_1	w_1	a_2	w_2	a_3	w_3	Error	avg.
SHOA	1.032	5.013	1.893	4.822	1.80	4.887	2.498	10.3
FDO	0.97	-0.24	-4.31	-0.01	-0.57	4.93	3.22	NA
ANA	−0.12	3.05	−5.21	−4.44	2.56	2.85	53.249	NA
fGA	NA	NA	NA	NA	NA	NA	0.0	8.4

TABLE 19. Uni-modal test functions with dimension = 10.

Function	Range	Shift	f_{min}
$F1(x) = \sum_{i=1}^{n} x_i^2$	[-100, 100]	[-30, -30, -30, ...]	0
$F2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	[-10, 10]	[-3, -3, -3, ...]	0
$F3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	[-100, 100]	[-30, -30, -30, ...]	0
$F4(x) = max_i\{\lvert x_i \rvert, 1 \leq i \leq n\}$	[-100, 100]	[-30, -30, -30, ...]	0
$F5(x) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$	[-30, 30]	[-15, -15, -15, ...]	0
$F6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	[-100, 100]	[-75, -75, -75, ...]	0
$F7(x) = \sum_{i=1}^{n} i x_i^4 + rand[0,1]$	[-1.28, 1.28]	[-0.25, -0.25, -0.25, ...]	0

D. FREQUENCY-MODULATED SOUND WAVE DESIGN PROBLEM SOLVING

The frequency modulation in sound waves is required to find optimal parameters to transfer sounds, it has six parameters to optimize as ($a1, w1, a2, w2, a3, w3$), which is a highly complex problem in the multimodal field, with fitness function is a minimum summation of square error between evaluated and modeled data, the fitness and constraints are formulated as follows:

$$f(\vec{p}) = \sum\nolimits_{i=1}^{100} (y(t) - y_0(t))^2 \tag{21}$$

where:

$$(\vec{p}) = (a1, w\ 1, a2, w\ 2, a3, w3) \tag{22}$$

$$y(t) = a_1 \cdot sin\,(w_1 \cdot t \cdot \theta + a_2 \cdot sin\,(w_2 \cdot t \cdot \theta + a_3 \cdot sin\,(w_3 \cdot t \cdot \theta))) \tag{23}$$

$$y(t) = (1 \cdot 0) \cdot sin((5 \cdot 0) \cdot t \cdot \theta + (1 \cdot 5) \cdot sin((4.8) \cdot t \cdot \theta + (2 \cdot 0) \cdot sin((4 \cdot 9) \cdot t \cdot \theta))) \tag{24}$$

With $\theta = (2\pi/100)$, the range of the parameter is $[-6.4, 6.35]$, and minimum fitness values are the optimal solution for the sound wave problems to transfer sound with the lowest error rate [94], [102], and [103].

TABLE 20. Multi-modal test functions with dimension = 10.

Function	Range	*Shift*	*min*
$F8(x)=\sum_{i=1}^{n}-x_i^2\sin\left(\sqrt{\lvert x_i\rvert}\right)$	[-500, 500]	[-300, -300, -300, ...]	-418,9829
$F9(x)=\sum_{i=1}^{n}\left[x_i^2-\cos(2\pi x_i)+10\right]$	[-5.12, 5.12]	[-2, -2, -2, ...]	0
$F10(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)+20+e$	[-32, 32]		0
$F11(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	[-600, 600]	[-400, -400, -400, ...]	0
$F12(x) = \frac{\pi}{n}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i-1)^2\left[1+10\sin^2(\pi y_{i+1})\right] + (y_n-1)^2\right\} + \sum_{i=1}^{n}u(x_i,10,100,4)$ $y_i = 1+\frac{(x+1)}{4}$ $u(x_i,a,k,m)=\begin{cases} k(x_i-a)^m & x_i>a \\ 0-a< & x_i<a \\ k(-x_i-a)^m & x_i<-a\end{cases}$	[-50, 50]	[-30, -30, -30, ...]	0
$F13(x) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i-1)^2\left[1+\sin^2(3\pi x_i+1)\right] + (x_n-1)\left[1+\sin^2(2\pi x_n)\right]\right\} + \sum_{i=1}^{n}u(x_i,5,100,4)$ $u(x_i,a,k,m)=\begin{cases} k(x_i-a)^m & x_i>a \\ 0-a< & x_i<a \\ k(-x_i-a)^m & x_i<-a\end{cases}$	[-50, 50]	[-10, -10, -10, ...]	0

In Table 18 comparative results of a frequency-modulated wave for the SHOA and FDO, ANA and Fork Genetic Algorithm (fGA) [94] are shown, the table presented six parameters, with the best fitness value out of 30 runs considered as an optimal result, and an average of 30 runs of the SHOA with the mentioned algorithms FDO and fGA, the unknown data is written as NA. The result shows that SHOA has a higher performance than all algorithms. The SHOA finds better fitness than FDO and a better average than fGA out of 30 runs. The FDO average result was NA, an optimal value has been generated depending on the presented parameters from FDO, and the fGA reached the optimal solution $\left(\overrightarrow{\mathbf{p}}\right) = \mathbf{0.0}$, but the parameters had not been presented.

TABLE 21. Composite test functions with dimension = 10, Range [−5,5], f_{min} = 0.

Functions
F14 (CF1) $f1, f2, \ldots\ldots, f10 = \text{sphere function}$ $[\delta_1, \delta_2, \delta_3 \ldots \delta_{10}] = [1,1,1,\ldots,1]$ $[\lambda_1,\lambda_2,\ldots,\lambda_{10}] = \left[\frac{5}{100},\frac{5}{100},\ldots,\frac{5}{100}\right]$
F15 (CF2) $f1, f2, \ldots\ldots, f10 = \text{Griewank's function}$ $[\delta_1, \delta_2, \delta_3 \ldots \delta_{10}] = [1,1,1,\ldots,1]$ $[\lambda_1,\lambda_2,\ldots,\lambda_{10}] = \left[\frac{5}{100},\frac{5}{100},\ldots,\frac{5}{100}\right]$
F16 (CF3) $f1, f2, \ldots\ldots, f10 = \text{Griewank's function}$ $[\delta_1, \delta_2, \delta_3 \ldots \delta_{10}] = [1,1,1,\ldots,1]$ $[\lambda_1,\lambda_2,\ldots,\lambda_{10}] = [1,1,\ldots,1]$
F17 (CF4) $f1, f2 = \text{Ackley'sfunction}$ $f3, f4 = \text{Rastrigin's function}$ $f5, f6 = \text{Weierdtrass function}$ $f7, f8 = \text{Griewank's function}$ $f9, f10 = \text{Sphere function}$ $[\delta_1, \delta_2, \delta_3 \ldots \delta_{10}] = [1,1,1,\ldots,1]$ $[\lambda_1,\lambda_2,\ldots,\lambda_{10}] = \left[\frac{5}{32},\frac{5}{32},1,1,\frac{5}{0.5},\frac{5}{0.5},\frac{5}{100},\frac{5}{100},\frac{5}{100},\frac{5}{100}\right]$
F18 (CF4) $f1, f2 = \text{Rastrigin's function}$ $f3, f4 = \text{Weierdtrass function}$ $f5, f6 = \text{Griewank's function}$ $f7, f8 = \text{Ackley'sfunction}$ $f9, f10 = \text{Sphere function}$ $[\delta_1, \delta_2, \delta_3 \ldots \delta_{10}] = [1,1,1,\ldots,1]$ $[\lambda_1,\lambda_2,\ldots,\lambda_{10}] = \left[\frac{1}{5},\frac{1}{5},\frac{5}{0.5},\frac{5}{0.5},\frac{5}{100},\frac{5}{100},\frac{5}{32},\frac{5}{32},\frac{5}{100},\frac{5}{100}\right]$
F18 (CF4) $f1, f2 = \text{Rastrigin's function}$ $f3, f4 = \text{Weierdtrass function}$ $f5, f6 = \text{Griewank's function}$ $f7, f8 = \text{Ackley'sfunction}$ $f9, f10 = \text{Sphere function}$ $[\delta_1, \delta_2, \delta_3 \ldots \delta_{10}] = [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.9,1]$ $[\lambda_1,\lambda_2,\ldots,\lambda_{10}] = \left[0.1\times\frac{1}{5},\ 0.2\times\frac{1}{5},\ 0.3\times\frac{5}{0.5},0.4\times\frac{5}{0.5},\ 0.5\times\frac{5}{100},\ 0.6\times\frac{5}{100},\ 0.7\times\frac{5}{32},0.8\times\frac{5}{32},0.9\times\frac{5}{100},1\times\frac{5}{100}\right]$

V. CONCLUSION

In this study, the theoretical model for the novel SHO swarm-based algorithm has been provided via concepts of exploration and exploitation. It mimics the bird's breed adaptation and lifestyle in the population.

The SHOA applied to 41 benchmark functions as (unimodal, multi-modal, composite, and 100-Digit Challenge) test functions, CEC22 single objective bounded constrained optimization and engineering problems (constrained, unconstrained) are solved. The performance has been compared with recent and powerful algorithms. The results demonstrated the effectiveness of the newly developed approach SHOA in solving all test functions and a variety of engineering problems and showed that this can provide reliable and accurate solutions in a variety of contexts. Through the SHOA study, the following were concluded:

- Faster convergence rate, the mechanism adapts the best birds for the next generation.

TABLE 22. Summary of basic "The Hundred-Digit Challenge" benchmarks.

Name	Functions	Dim	Range
C01	STRONG CHEBYSHEV POLYNOMIAL FITTING PROBLEM	9	[-8192,8192]
C02	INVERSE HILBERT MATRIX PROBLEM	16	[-16384, 16384]
C03	LENNARD-JONES MINIMUM ENERGY CLUSTER	18	[-4,4]
C04	RASTRIGIN'S FUNCTION	10	[-100, 100]
C05	GRIEWANGK'S FUNCTION	10	[-100, 100]
C06	WEIERSTRASS FUNCTION	10	[-100, 100]
C0	MODIFIED SCHWEFEL'S FUNCTION	10	[-100, 100]
C08	EXPANDED SCHAFFER'S F6 FUNCTION	10	[-100, 100]
C09	HAPPY CAT FUNCTION	10	[-100, 100]
C10	ACKLEY FUNCTION	10	[-100, 100]

TABLE 23. Summary of CEC2022 benchmarks.

Types	No	Functions	F_i^*
Unimodal	1	Shifted and full Rotated Zakharov Function	300
Basic Functions	2	Shifted and full Rotated Rosenbrock's Function	400
	3	Shifted and full Rotated Expanded Schaffer's f6 Function	600
	4	Shifted and full Rotated Non-Continuous Rastrigin's Function	800
	5	Shifted and full Rotated Levy Function	900
Hybrid multimodal Functions	6	Hybrid Function 1 (N = 3)	1800
	7	Hybrid Function 2 (N = 6)	2000
	8	Hybrid Function 3 (N = 5)	2200
Composite multimodal Functions	9	Composition Function 1 (N = 5)	2300
	10	Composition Function 2 (N = 4)	2400
	11	Composition Function 3 (N = 5)	2600
	12	Composition Function 4 (N = 6)	2700
Search Range [-100, 100]			

- More stable than compared algorithms, the balance of convergence and divergence leads to the best solution.
- Accurate search, high exploration, and investigation promise a promising area of space within a reasonable amount of time.
- High performance in solving constrained and unconstrained multimodal real optimization problems.
- Highly multi-modal problem optimizer, because each nest is considered a population with an optimal solution, and all are considered a single population that finds the global optimum from local optimums.
- There are fewer control parameters compared with other optimization algorithms, which leads to low computation time.

The proposed SHOA is a single objective and the graphical abstract of SHOA presented in Figure 10 in the Appendix B; for the future, many research works can be conducted in multi-objective, binary, and discrete versions, all of which can be used to solve a variety type of problems.

APPENDIX A

See Tables 19–23.

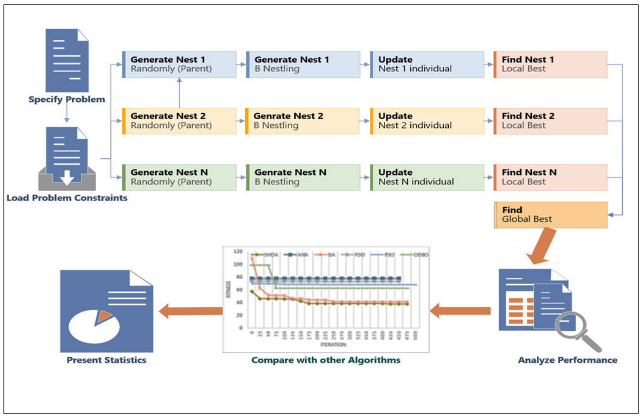

FIGURE 10. Novel SHOA process structure.

APPENDIX B

See Figure 10.

ACKNOWLEDGMENT

The authors would like to sincerely thank the members of the Software Engineering Department from Salahaddin University-Erbil for their invaluable contributions that enabled this study to be completed successfully.

REFERENCES

[1] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983. [Online]. Available: http://science.sciencemag.org/

[2] R. Greiner, "PALO: A probabilistic hill-climbing algorithm," *Artif. Intell.*, vol. 84, nos. 1–2, pp. 177–208, Jul. 1996, doi: 10.1016/0004-3702(95)00040-2.

[3] P. Hansen and N. Mladenović, "Variable neighborhood search," *Comput. Oper. Res.*, vol. 24, pp. 1097–1100, 1997, doi: 10.1016/S0305-0548(97)00031-2.

[4] F. Glover, "Tabu search: A tutorial," *Interfaces*, vol. 20, no. 4, pp. 74–94, Aug. 1990, doi: 10.1287/inte.20.4.74.

[5] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence*, 2nd ed., Cambridge, MA, USA: MIT Press, 1975. [Online]. Available: https://psycnet.apa.org/record/1975-26618-000

[6] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, pp. 341–359, Dec. 1997, doi: 10.1023/A:1008202821328.

[7] J. R. Koza, "Hierarchical genetic algorithms operating on populations of computer programs," in *Proc. IJCAI*, 1989, pp. 768–774.

[8] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006, doi: 10.1109/MCI.2006.329691.

[9] M. Dorigo, "Optimization, learning and natural algorithms," Ph.D. dissertation, Politecnico di Milano, Milano, Italy, 1992.

[10] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern., B*, vol. 26, no. 1, pp. 29–41, Feb. 1996, doi: 10.1109/3477.484436.

[11] D. Martens, M. De Backer, R. Haesen, J. Vanthienen, M. Snoeck, and B. Baesens, "Classification with ant colony optimization," *IEEE Trans. Evol. Comput.*, vol. 11, no. 5, pp. 651–665, Oct. 2007, doi: 10.1109/tevc.2006.890229.

[12] L. Zuo, L. Shu, S. Dong, C. Zhu, and T. Hara, "A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing," *IEEE Access*, vol. 3, pp. 2687–2699, 2015, doi: 10.1109/ACCESS.2015.2508940.

[13] D. N. Hama Rashid, T. A. Rashid, and S. Mirjalili, "ANA: Ant nesting algorithm for optimizing real-world problems," *Mathematics*, vol. 9, no. 23, p. 3111, Dec. 2021, doi: 10.3390/math9233111.

[14] R. Eberhart and J. Kennedy, "New optimizer using particle swarm theory," in *Proc. Int. Symp. Micro Mach. Hum. Sci.*, 1995, pp. 39–43, doi: 10.1109/mhs.1995.494215.

[15] S. M. Mikki and A. A. Kishk, "Quantum particle swarm optimization for electromagnetics," *IEEE Trans. Antennas Propag.*, vol. 54, no. 10, pp. 2764–2775, Oct. 2006, doi: 10.1109/TAP.2006.882165.

[16] M. Braik, A. F. Sheta, and A. Ayesh, "Image enhancement using particle swarm optimization," *Int. J. Innov. Comput. Appl.*, vol. 1, pp. 138–145, Jan. 2007.

[17] T. Ray and K. M. Liew, "Society and civilization: An optimization algorithm based on the simulation of social behavior," *IEEE Trans. Evol. Comput.*, vol. 7, no. 4, pp. 386–396, Aug. 2003, doi: 10.1109/TEVC.2003.814902.

[18] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Fac. Comput. Eng. Dept., Erciyes Univ., Kayseri, Türkiye, Tech. Rep., TR06, 2005.

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

[20] W. Gao, S. Liu, and L. Huang, "A global best artificial bee colony algorithm for global optimization," *J. Comput. Appl. Math.*, vol. 236, no. 11, pp. 2741–2753, May 2012, doi: 10.1016/j.cam.2012.01.013.

[21] M. El-Abd, "Generalized opposition-based artificial bee colony algorithm," in *Proc. IEEE Congr. Evol. Comput.*, 2012, pp. 1–4.

[22] M. S. Uzer, N. Yilmaz, and O. Inan, "Feature selection method based on artificial bee colony algorithm and support vector machines for medical datasets classification," *Sci. World J.*, vol. 2013, no. 1, Jan. 2013, Art. no. 419187, doi: 10.1155/2013/419187.

[23] J. A. Bullinaria and K. AlYahya, "Artificial bee colony training of neural networks: Comparison with back-propagation," *Memetic Comput.*, vol. 6, no. 3, pp. 171–182, Sep. 2014, doi: 10.1007/s12293-014-0137-7.

[24] B. Yao, Q. Yan, M. Zhang, and Y. Yang, "Improved artificial bee colony algorithm for vehicle routing problem with time windows," *PLoS ONE*, vol. 12, no. 9, Sep. 2017, Art. no. e0181275, doi: 10.1371/journal.pone.0181275.

[25] J. M. Abdullah and T. Ahmed, "Fitness dependent optimizer: Inspired by the bee swarming reproductive process," *IEEE Access*, vol. 7, pp. 43473–43486, 2019, doi: 10.1109/ACCESS.2019.2907012.

[26] K. M. Passino, "Bacterial foraging optimization," *Int. J. Swarm Intell. Res.*, vol. 1, no. 1, pp. 1–16, Jan. 2010, doi: 10.4018/jsir.2010010101.

[27] S. S. Patnaik and A. K. Panda, "Particle swarm optimization and bacterial foraging optimization techniques for optimal current harmonic mitigation by employing active power filter," *Appl. Comput. Intell. Soft Comput.*, vol. 2012, pp. 1–10, Jan. 2012, doi: 10.1155/2012/897127.

[28] M. A. Muñoz, S. K. Halgamuge, W. Alfonso, and E. F. Caicedo, "Simplifying the bacteria foraging optimization algorithm," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2010, pp. 1–7, doi: 10.1109/CEC.2010.5586025.

[29] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Stochastic Algorithms: Foundations and Applications* (Lecture Notes in Computer Sciences), vol. 5792. Berlin, Germany: Springer, 2009, pp. 169–178, doi: 10.1007/978-3-642-04944-6_14.

[30] H. S. Gebremedhen, D. E. Woldemichael, and F. M. Hashim, "A firefly algorithm based hybrid method for structural topology optimization," *Adv. Model. Simul. Eng. Sci.*, vol. 7, no. 1, pp. 1–20, Dec. 2020, doi: 10.1186/s40323-020-00183-0.

[31] 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.

[32] S. P. H. Boroujeni and E. Pashaei, "Data clustering using chimp optimization algorithm," in *Proc. 11th Int. Conf. Comput. Eng. Knowl. (ICCKE)*, Oct. 2021, pp. 296–301, doi: 10.1109/iccke54056.2021.9721483.

[33] X. Wang, V. Snásel, S. Mirjalili, J.-S. Pan, L. Kong, and H. A. Shehadeh, "Artificial protozoa optimizer (APO): A novel bio-inspired metaheuristic algorithm for engineering optimization," *Knowl.-Based Syst.*, vol. 295, Jul. 2024, Art. no. 111737, doi: 10.1016/j.knosys.2024.111737.

[34] H. Peraza-Vázquez, A. Peña-Delgado, M. Merino-Treviño, A. B. Morales-Cepeda, and N. Sinha, "A novel metaheuristic inspired by horned lizard defense tactics," *Artif. Intell. Rev.*, vol. 57, no. 3, p. 59, Feb. 2024, doi: 10.1007/s10462-023-10653-7.

[35] J. Wang, W.-C. Wang, X.-X. Hu, L. Qiu, and H.-F. Zang, "Black-winged kite algorithm: A nature-inspired meta-heuristic for solving benchmark functions and engineering problems," *Artif. Intell. Rev.*, vol. 57, no. 4, p. 98, Mar. 2024, doi: 10.1007/s10462-024-10723-4.

[36] S. O. Oladejo, S. O. Ekwe, and S. Mirjalili, "The hiking optimization algorithm: A novel human-based metaheuristic approach," *Knowl.-Based Syst.*, vol. 296, Jul. 2024, Art. no. 111880, doi: 10.1016/j.knosys.2024.111880.

[37] A. Baskar et al., "A novel two-phase trigonometric algorithm for solving global optimization problems," *Ann. Oper. Res.*, Mar. 2024, doi: 10.1007/s10479-024-05837-5.

[38] E. Osaba and X. S. Yang, "Soccer-inspired metaheuristics: Systematic review of recent research and applications," in *Applied Optimization and Swarm Intelligence* (Springer Tracts in Nature-Inspired Computing), E. Osaba and X. S. Yang, Eds., Singapore: Springer, 2021, pp. 81–102, 10.1007/978-981-16-0662-5_5.

[39] M. Jain, V. Singh, and A. Rani, "A novel nature-inspired algorithm for optimization: Squirrel search algorithm," *Swarm Evol. Comput.*, vol. 44, pp. 148–175, Feb. 2019, doi: 10.1016/j.swevo.2018.02.013.

[40] S. Mugemanyi, Z. Qu, F. X. Rugema, Y. Dong, L. Wang, C. Bananeza, A. Nshimiyimana, and E. Mutabazi, "Marine predators algorithm: A comprehensive review," *Mach. Learn. Appl.*, vol. 12, Jun. 2023, Art. no. 100471, doi: 10.1016/j.mlwa.2023.100471.

[41] R. Chaudhary and H. Banati, "Peacock algorithm," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Wellington, New Zealand, 2019, pp. 2331–2338, doi: 10.1109/CEC.2019.8790371.

[42] M. A. Akbari, M. Zare, R. Azizipanah-abarghooee, S. Mirjalili, and M. Deriche, "The cheetah optimizer: A nature-inspired metaheuristic algorithm for large-scale optimization problems," *Sci. Rep.*, vol. 12, no. 1, pp. 1–20, Jun. 2022, doi: 10.1038/s41598-022-14338-z.

[43] B. Abdollahzadeh, F. S. Gharehchopogh, N. Khodadadi, and S. Mirjalili, "Mountain gazelle optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems," *Adv. Eng. Softw.*, vol. 174, Dec. 2022, Art. no. 103282, doi: 10.1016/j.advengsoft.2022.103282.

[44] W. N. E. A. Wan Afandie, T. K. A. Rahman, and Z. Zakaria, "Optimal load shedding using bacterial foraging optimization algorithm," in *Proc. IEEE 4th Control Syst. Graduate Res. Colloq.*, Aug. 2013, pp. 93–97, doi: 10.1109/ICSGRC.2013.6653282.

[45] B. B. Mangaraj, M. R. Jena, and S. K. Mohanty, "Bacteria foraging algorithm in antenna design," *Appl. Comput. Intell. Soft Comput.*, vol. 2016, pp. 1–11, Jan. 2016, doi: 10.1155/2016/5983469.

[46] J. Skackauskas, T. Kalganova, I. Dear, and M. Janakiram, "Dynamic impact for ant colony optimization algorithm," *Swarm Evol. Comput.*, vol. 69, Mar. 2022, Art. no. 100993, doi: 10.1016/j.swevo.2021.100993.

[47] T. Dokeroglu, E. Sevinc, and A. Cosar, "Artificial bee colony optimization for the quadratic assignment problem," *Appl. Soft Comput.*, vol. 76, pp. 595–606, Mar. 2019, doi: 10.1016/j.asoc.2019.01.001.

[48] Z. Xu and C. Feng, "Design of available land space optimization model based on ant colony optimization algorithm," in *Proc. IEEE Int. Conf. Integr. Circuits Commun. Syst. (ICICACS)*, Feb. 2023, pp. 1–6, doi: 10.1109/ICICACS57338.2023.10099535.

[49] B. Qolomany, K. Ahmad, A. Al-Fuqaha, and J. Qadir, "Particle swarm optimized federated learning for industrial IoT and smart city services," in *Proc. GLOBECOM IEEE Global Commun. Conf.*, Dec. 2020, pp. 1–6, doi: 10.1109/GLOBECOM42002.2020.9322464.

[50] H. K. AbdulKarim and T. A. Rashid, "Moth-flame optimization and ant nesting algorithm: A systematic evaluation," in *Proc. 1st Int. Conf. Innov. Inf. Technol. Bus.*, vol. 1, 2023, pp. 139–152, doi: 10.2991/978-94-6463-110-4_11.

[51] T. Djemai, P. Stolf, T. Monteil, and J.-M. Pierson, "A discrete particle swarm optimization approach for energy-efficient IoT services placement over fog infrastructures," in *Proc. 18th Int. Symp. Parallel Distrib. Comput. (ISPDC)*, Jun. 2019, pp. 32–40, doi: 10.1109/ISPDC.2019.00020.

[52] R. Zou, V. Kalivarapu, E. Winer, J. Oliver, and S. Bhattacharya, "Particle swarm optimization-based source seeking," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 3, pp. 865–875, Jul. 2015, doi: 10.1109/TASE.2015.2441746.

[53] A. Salim, W. K. Jummar, F. M. Jasim, and M. Yousif, "Eurasian oystercatcher optimiser: New meta-heuristic algorithm," *J. Intell. Syst.*, vol. 31, no. 1, pp. 332–344, Mar. 2022, doi: 10.1515/jisys-2022-0017.

[54] M. Braik, A. Hammouri, J. Atwan, M. A. Al-Betar, and M. A. Awadallah, "White shark optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems," *Knowl.-Based Syst.*, vol. 243, May 2022, Art. no. 108457, doi: 10.1016/j.knosys.2022.108457.

[55] H. Mohammed and T. Rashid, ''FOX: A FOX-inspired optimization algorithm,'' *Int. J. Speech Technol.*, vol. 53, no. 1, pp. 1030–1050, Jan. 2023, doi: 10.1007/s10489-022-03533-0.
[56] N. A. Golilarz, H. Gao, A. Addeh, and S. Pirasteh, ''ORCA optimization algorithm: A new meta-heuristic tool for complex optimization problems,'' in *Proc. 17th Int. Comput. Conf. Wavelet Act. Media Technol. Inf. Process. (ICCWAMTIP)*, Dec. 2020, pp. 198–204, doi: 10.1109/ICCWAMTIP51612.2020.9317473.
[57] P. Trojovský and M. Dehghani, ''A new bio-inspired metaheuristic algorithm for solving optimization problems based on walruses behavior,'' *Sci. Rep.*, vol. 13, no. 1, p. 8775, May 2023, doi: 10.1038/s41598-023-35863-5.
[58] J. Vela, E. E. Montiel, P. Mora, P. Lorite, and T. Palomeque, ''Aphids and ants, mutualistic species, share a mariner element with an unusual location on aphid chromosomes,'' *Genes*, vol. 12, no. 12, p. 1966, Dec. 2021, doi: 10.3390/genes12121966.
[59] M. Azizi, S. Talatahari, and A. H. Gandomi, ''Fire hawk optimizer: A novel metaheuristic algorithm,'' *Artif. Intell. Rev.*, vol. 56, no. 1, pp. 287–363, Jan. 2023, doi: 10.1007/s10462-022-10173-w.
[60] F. A. Hashim, E. H. Houssein, K. Hussain, M. S. Mabrouk, and W. Al-Atabany, ''Honey badger algorithm: New metaheuristic algorithm for solving optimization problems,'' *Math. Comput. Simul.*, vol. 192, pp. 84–110, Feb. 2022, doi: 10.1016/j.matcom.2021.08.013.
[61] S. Kaur, L. K. Awasthi, A. L. Sangal, and G. Dhiman, ''Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization,'' *Eng. Appl. Artif. Intell.*, vol. 90, Apr. 2020, Art. no. 103541, doi: 10.1016/j.engappai.2020.103541.
[62] O. Al-Baik, S. Alomari, O. Alssayed, S. Gochhait, I. Leonova, U. Dutta, O. P. Malik, Z. Montazeri, and M. Dehghani, ''Pufferfish optimization algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems,'' *Biomimetics*, vol. 9, no. 2, p. 65, Jan. 2024, doi: 10.3390/biomimetics9020065.
[63] A. A. Abdelhamid, S. K. Towfek, N. Khodadadi, A. A. Alhussan, D. S. Khafaga, M. M. Eid, and A. Ibrahim, ''Waterwheel plant algorithm: A novel metaheuristic optimization method,'' *Processes*, vol. 11, no. 5, p. 1502, May 2023, doi: 10.3390/pr11051502.
[64] A. Kaveh, M. R. Seddighian, and E. Ghanadpour, ''Black hole mechanics optimization: A novel meta-heuristic algorithm,'' *Asian J. Civil Eng.*, vol. 21, no. 7, pp. 1129–1149, Nov. 2020, doi: 10.1007/s42107-020-00282-8.
[65] E. Trojovská and M. Dehghani, ''A new human-based metahurestic optimization method based on mimicking cooking training,'' *Sci. Rep.*, vol. 12, no. 1, pp. 1–25, Sep. 2022, doi: 10.1038/s41598-022-19313-2.
[66] M. Said, E. H. Houssein, S. Deb, A. A. Alhussan, and R. M. Ghoniem, ''A novel gradient based optimizer for solving unit commitment problem,'' *IEEE Access*, vol. 10, pp. 18081–18092, 2022, doi: 10.1109/ACCESS.2022.3150857.
[67] M. Dehghani, E. Trojovská, P. Trojovský, and O. P. Malik, ''OOBO: A new metaheuristic algorithm for solving optimization problems,'' *Biomimetics*, vol. 8, no. 6, p. 468, Oct. 2023, doi: 10.3390/biomimetics8060468.
[68] Y. Gao, ''PID-based search algorithm: A novel metaheuristic algorithm based on PID algorithm,'' *Exp. Syst. Appl.*, vol. 232, Dec. 2023, Art. no. 120886, doi: 10.1016/j.eswa.2023.120886.
[69] M. Braik, M. H. Ryalat, and H. Al-Zoubi, ''A novel meta-heuristic algorithm for solving numerical optimization problems: Ali baba and the forty thieves,'' *Neural Comput. Appl.*, vol. 34, no. 1, pp. 409–455, Jan. 2022, doi: 10.1007/s00521-021-06392-x.
[70] Y. Jiang, Z.-H. Zhan, K. C. Tan, and J. Zhang, ''Optimizing niche center for multimodal optimization problems,'' *IEEE Trans. Cybern.*, vol. 53, no. 4, pp. 2544–2557, Apr. 2023, doi: 10.1109/TCYB.2021.3125362.
[71] J. Gálvez, E. Cuevas, and K. Gopal Dhal, ''A competitive memory paradigm for multimodal optimization driven by clustering and chaos,'' *Mathematics*, vol. 8, no. 6, p. 934, Jun. 2020, doi: 10.3390/math8060934.
[72] D. J. Poole and C. B. Allen, ''Constrained niching using differential evolution,'' *Swarm Evol. Comput.*, vol. 44, pp. 74–100, Feb. 2019, doi: 10.1016/j.swevo.2018.11.004.
[73] S. Agrawal, A. Tiwari, P. Naik, and A. Srivastava, ''Improved differential evolution based on multi-armed bandit for multimodal optimization problems,'' *Int. J. Speech Technol.*, vol. 51, no. 11, pp. 7625–7646, Nov. 2021, doi: 10.1007/s10489-021-02261-1.
[74] Z.-G. Chen, Z.-H. Zhan, H. Wang, and J. Zhang, ''Distributed individuals for multiple peaks: A novel differential evolution for multimodal optimization problems,'' *IEEE Trans. Evol. Comput.*, vol. 24, no. 4, pp. 708–719, Aug. 2020, doi: 10.1109/TEVC.2019.2944180.
[75] X. Yan, M. Razeghi-Jahromi, A. Homaifar, B. A. Erol, A. Girma, and E. Tunstel, ''A novel streaming data clustering algorithm based on fitness proportionate sharing,'' *IEEE Access*, vol. 7, pp. 184985–185000, 2019, doi: 10.1109/ACCESS.2019.2922162.
[76] G. Joy, C. Huyck, and X. S. Yang, ''Review of parameter tuning methods for nature-inspired algorithms,'' in *Benchmarks and Hybrid Algorithms in Optimization and Applications* (Springer Tracts in Nature-Inspired Computing), X.-S. Yang, Ed., Singapore: Springer, 2023, pp. 33–47, doi: 10.1007/978-981-99-3970-1_3.
[77] R. Chaudhary and H. Banati, ''Study of population partitioning techniques on efficiency of swarm algorithms,'' *Swarm Evol. Comput.*, vol. 55, Jun. 2020, Art. no. 100672, doi: 10.1016/j.swevo.2020.100672.
[78] Q. Li, S.-Y. Liu, and X.-S. Yang, ''Influence of initialization on the performance of metaheuristic optimizers,'' *Appl. Soft Comput.*, vol. 91, Jun. 2020, Art. no. 106193, doi: 10.1016/j.asoc.2020.106193.
[79] D. Yazdani, M. N. Omidvar, D. Yazdani, J. Branke, T. T. Nguyen, A. H. Gandomi, Y. Jin, and X. Yao, ''Robust optimization over time: A critical review,'' *IEEE Trans. Evol. Comput.*, early access, Aug. 17, 2024, doi: 10.1109/tevc.2023.3306017.
[80] R. Salgotra, P. Sharma, S. Raju, and A. H. Gandomi, ''A contemporary systematic review on meta-heuristic optimization algorithms with their MATLAB and Python code reference,'' *Arch. Comput. Methods Eng.*, vol. 31, no. 3, pp. 1749–1822, Apr. 2024, doi: 10.1007/s11831-023-10030-1.
[81] R. Chaudhary and H. Banati, ''Capitalizing diversity for efficiency enhancement in multi-population swarm algorithms,'' in *Proc. 10th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2019, pp. 1–7, doi: 10.1109/ICCCNT45670.2019.8944872.
[82] R. Chaudhary and H. Banati, ''Improving convergence in swarm algorithms by controlling range of random movement,'' *Natural Comput.*, vol. 20, no. 3, pp. 513–560, Sep. 2021, doi: 10.1007/s11047-020-09826-y.
[83] D. E. Gawlik and K. L. Bildstein, ''Reproductive success and nesting habitat of Loggerhead Shrikes in north-central South Carolina,'' *Wilson Bulletin*, vol. 102, pp. 37–48, Mar. 1990. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/24671971
[84] R. Yosef and T. C. Grubb, ''Territory size influences nutritional condition in nonbreeding loggerhead shrikes (Lanius ludovicianus): A ptilochronology approach,'' *Conservation Biol.*, vol. 6, no. 3, pp. 447–449, Sep. 1992, doi: 10.1046/j.1523-1739.1992.06030447.x.
[85] C. Hill, K. Miles, K. Maddox, and A. Tegeler, ''Are adoptions in a predatory songbird a strategy to aid mate acquisition?'' *J. Field Ornithol.*, vol. 94, no. 3, p. 9, 2023, doi: 10.5751/JFO-00284-940209.
[86] R. Ridgway, *A Manual of North American Birds*. Philadephia, PA, USA: J. B. Lippncott, 1887. [Online]. Available: https://www.biodiversitylibrary.org/page/21816472
[87] X. Yao, Y. Liu, and G. Lin, ''Evolutionary programming made faster,'' *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999, doi: 10.1109/4235.771163.
[88] J. J. Liang, P. N. Suganthan, and K. Deb, ''Novel composition test functions for numerical global optimization,'' in *Proc. IEEE Swarm Intell. Symp.*, Jun. 2005, pp. 68–75, doi: 10.1109/SIS.2005.1501604.
[89] M. Molga and C. Smutnicki, ''Test functions for optimization needs,'' *Test Functions Optim. Needs*, vol. 101, p. 48. Apr. 2005. Accessed: Feb. 11, 2024. [Online]. Available: https://www.scribd.com/doc/46766287/Test-functions-for-optimization-needs
[90] J. Derrac, S. García, D. Molina, and F. Herrera, ''A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms,'' *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, Mar. 2011, doi: 10.1016/j.swevo.2011.02.002.
[91] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. Chen, A. Auger, and S. Tiwari, ''Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization,'' IIT, Kanpur, India, KanGAL Rep. 2005005, May 2005.
[92] K. V. Price, N. H. Awad, M. Z. Ali, and P. N. Suganthan, ''Problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimization,'' Nanyang Technol. Univ., Singapore, Tech. Rep., Nov. 2018. [Online]. Available: https://www3.ntu.edu.sg/home/epnsugan/index_files/CEC2019/CEC2019.htm

[93] A. Kumar, K. V. Price, A. W. Mohamed, A. A. Hadi, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2022 special session and competition on single objective bound constrained numerical optimization," in *Proc. IEEE Congr. Evol. Comput.*, Mar. 2022, pp. 1–20.

[94] S. Tsutsui and Y. Fujimoto, "Forking genetic algorithm with blocking and shrinking modes (fGA)," in *Proc. 5th Int. Conf. Genet Algorithms*, 1993, pp. 206–215.

[95] J. Brest, M. S. Maucec, and B. Boskovic, "The 100-digit challenge: Algorithm jDE100," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2019, pp. 19–26, doi: 10.1109/CEC.2019.8789904.

[96] C. A. Coello Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art," *Comput. Methods Appl. Mech. Eng.*, vol. 191, nos. 11–12, pp. 1245–1287, Jan. 2002, doi: 10.1016/s0045-7825(01)00323-1.

[97] E. Sandgren, "Nonlinear integer and discrete programming in mechanical design optimization," *J. Mech. Design*, vol. 112, no. 2, pp. 223–229, Jun. 1990, doi: 10.1115/1.2912596.

[98] S. Mohapatra and P. Mohapatra, "American zebra optimization algorithm for global optimization problems," *Sci. Rep.*, vol. 13, no. 1, p. 5211, Mar. 2023, doi: 10.1038/s41598-023-31876-2.

[99] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Erratum to: Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems," *Eng. Comput.*, vol. 29, no. 2, p. 245, Apr. 2013, doi: 10.1007/s00366-012-0308-4.

[100] T. Ray and P. Saini, "Engineering design optimization using a swarm with an intelligent information sharing among individuals," *Eng. Optim.*, vol. 33, no. 6, pp. 735–748, Aug. 2001, doi: 10.1080/03052150108940941.

[101] N. Jin and Y. Rahmat-Samii, "Advances in particle swarm optimization for antenna designs: Real-number, binary, single-objective and multiobjective implementations," *IEEE Trans. Antennas Propag.*, vol. 55, no. 3, pp. 556–567, Mar. 2007, doi: 10.1109/TAP.2007.891552.

[102] F. Herrera and M. Lozano, "Gradual distributed real-coded genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 4, no. 1, pp. 43–63, Apr. 2000, doi: 10.1109/4235.843494.

[103] S. Das and P. N. Suganthan, "Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problem," Jadavpur Univ., India Nanyang Technol. Univ., Singapore, Tech. Rep., Dec. 2010, pp. 1–41. [Online]. Available: http://web.mysites.ntu.edu.sg/epnsugan/PublicSite/Shared Documents/CEC 2011-RWP/Tech-Rep.pdf

HANAN K. ABDULKARIM received the B.Sc. (Hons.) and M.Sc. degrees in software engineering from Salahaddin University-Erbil, Erbil, Iraq, in 2008 and 2014, respectively, where she is currently pursuing the Ph.D. degree in software engineering with the College of Engineering, with a focus on swarm intelligence. She worked as the IT Manager and an Instructor with Salahaddin University-Erbil, in 2014. Her areas of research interests include artificial intelligence, nature-inspired algorithms, swarm intelligence, computational intelligence, machine learning, optimization, data, and web mining.

TARIK A. RASHID (Member, IEEE) received the Ph.D. degree in computer science and informatics from the College of Engineering, Mathematical and Physical Sciences, University College Dublin (UCD), Ireland, in 2006. He joined the University of Kurdistan Hewlêr (UKH), Iraq, in 2017. He is currently a Principal Fellow with the Higher Education Academy (PFHEA-UK) and a Professor with the Department of Computer Science and Engineering, UKH. His areas of research interests include artificial intelligence, nature-inspired algorithms, swarm intelligence, computational intelligence, machine learning, and data mining. He is on the prestigious Stanford University list of the World's Top 2% of Scientists for the years 2021, 2022, and 2023.