Received March 11, 2022, accepted March 23, 2022, date of publication March 28, 2022, date of current version April 12, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3162853

Golden Search Optimization Algorithm

MOHAMMAD NOROOZI[1], HAMED MOHAMMADI[2], EMAD EFATINASAB[3], ALI LASHGARI[4], MAHDIYEH ESLAMI[5], (Member, IEEE), AND BASEEM KHAN[6], (Senior Member, IEEE)

[1]Department of Industrial and Management Systems Engineering, University of South Florida, Tampa, FL 33620, USA
[2]Department of Industrial Engineering and Management Systems, Amirkabir University of Technology, Tehran 1591634311, Iran
[3]Department of Mathematics, University of Padova, 35122 Padova, Italy
[4]Department of Economics, Kansas State University, Manhattan, KS 66506, USA
[5]Department of Electrical Engineering, Kerman Branch, Islamic Azad University, Kerman 7635168111, Iran
[6]Department of Electrical and Computer Engineering, Hawassa University, Hawassa 05, Ethiopia

Corresponding authors: Mahdiyeh Eslami (mahdiyeh_eslami@yahoo.com) and Baseem Khan (baseem.khan04@gmail.com)

ABSTRACT This study introduces an effective population-based optimization algorithm, namely the Golden Search Optimization Algorithm (GSO), for numerical function optimization. The new algorithm has a simple but effective strategy for solving complex problems. GSO starts with random possible solutions called objects, which interact with each other based on a simple mathematical model to reach the global optimum. To provide a fine balance between the explorative and exploitative behavior of a search, the proposed method utilizes a transfer operator in the adaptive step size adjustment scheme. The proposed algorithm is benchmarked with 23 unimodal, multimodal, and fixed dimensional functions and the results are verified by a comparative study with the well-known Gravitational Search Algorithm (GSA), Sine-Cosine Algorithm (SCA), Grey Wolf Optimization (GWO), and Tunicate Swarm Algorithm (TSA). In addition, the nonparametric Wilcoxon's rank sum test is performed to measure the pair-wise statistical performance of the GSO and provide a valid judgment about the performance of the algorithm. The simulation results demonstrate that GSO is superior and could generate better optimal solutions when compared with other competitive algorithms.

INDEX TERMS Global optimization, golden search, metaheuristic, population-based, benchmark function.

I. INTRODUCTION

Many real world design problems can be considered as optimization problems, and an appropriate optimization method is required for the solution. On the other hand, design problems have become harder when discontinuities, incomplete information, dynamicity, and uncertainties are involved. In such a case, classical optimization algorithms based on mathematical principles demand exponential time or may not find the optimal solution at all. One of these classical algorithms is the gradient-based methods, which utilize the gradient of the objective function for the configuration of the optimization problem. To overcome the mentioned problem, during the last few decades, introducing new efficient metaheuristic optimization algorithms to deal with the drawbacks of classical techniques has been of great concern. The privileges of these algorithms include derivation-free mechanisms, simple concepts and structure, local optima avoidance, and effectiveness for discrete and continuous functions. Accordingly, there is an increasing interest in presenting new metaheuristic algorithms that have high efficiency, great accuracy, and an increased speed rate in dealing with difficult optimization problems. Generally, metaheuristic algorithms have two types: single-solution based methods (also known as trajectory methods) and population-based algorithms.

As the name indicates, in the former type, only one solution is generated (usually at random) and processed during the optimization phase until a stopping criterion is satisfied.

Some of these methods are Simulated Annealing (SA) [1], Tabu Search (TS) [2], Iterated Local Search (ILS) [3], and Vortex Search Algorithm (VSA) [4]. In the latter type, a set of solutions (i.e., a population) is generated randomly and updated iteratively in each iteration of the optimization process until satisfying a stopping criteria. Some well-known examples of these algorithms are the Genetic Algorithm [5], Ant Colony Optimization [6], Particle Swarm Optimization [7], Firefly Algorithm [8] and Harris hawks optimization [9]. Regardless of the wide variety of recently developed metaheuristic algorithms, they have two main

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

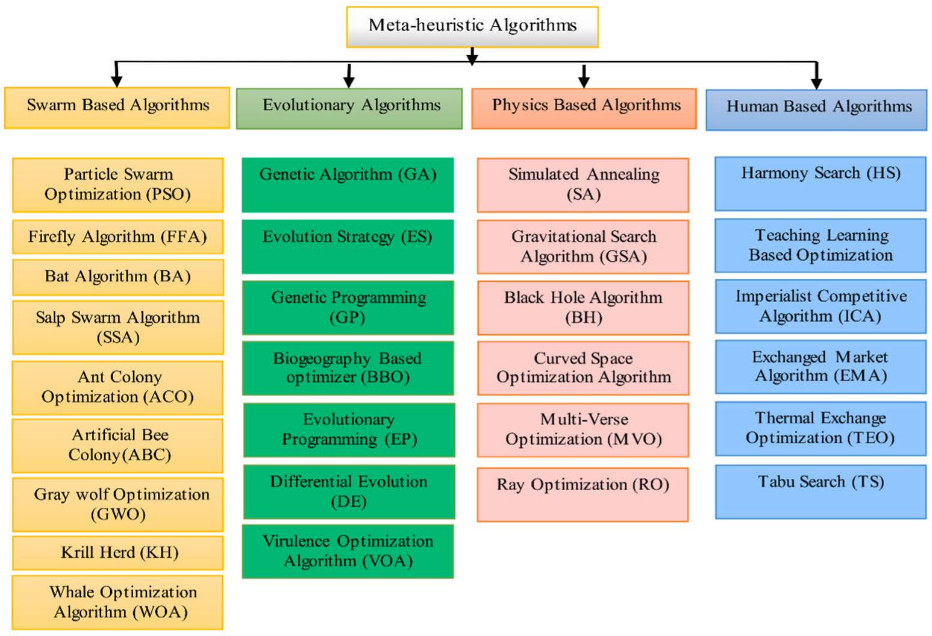
Meta-heuristic Algorithms
Swarm Based Algorithms
Evolutionary Algorithms
Physics Based Algorithms
Human Based Algorithms
Particle Swarm Optimization (PSO)
Firefly Algorithm (FFA)
Bat Algorithm (BA)
Salp Swarm Algorithm (SSA)
Ant Colony Optimization (ACO)
Artificial Bee Colony(ABC)
Gray wolf Optimization (GWO)
Krill Herd (KH)
Whale Optimization Algorithm (WOA)
Genetic Algorithm (GA)
Evolution Strategy (ES)
Genetic Programming (GP)
Biogeography Based optimizer (BBO)
Evolutionary Programming (EP)
Differential Evolution (DE)
Virulence Optimization Algorithm (VOA)
Simulated Annealing (SA)
Gravitational Search Algorithm (GSA)
Black Hole Algorithm (BH)
Curved Space Optimization Algorithm
Multi-Verse Optimization (MVO)
Ray Optimization (RO)
Harmony Search (HS)
Teaching Learning Based Optimization
Imperialist Competitive Algorithm (ICA)
Exchanged Market Algorithm (EMA)
Thermal Exchange Optimization (TEO)
Tabu Search (TS)

FIGURE 1. Categorization of meta-heuristic algorithms.

phases include investigating the various promising regions in a search space (exploration) and the local search around the obtained promising regions (exploitation). The basic differences between metaheuristic methods is the special procedure that they applied for well balance among these two phases. Based on the nature of the metaheuristics, population-based algorithms are more exploration-oriented, while single-solution algorithms are more exploitation-oriented [10].

Although all population-based metaheuristics could obtain relatively acceptable results, there is no metaheuristic method that provides superior performance in solving all optimizing problems [11]. Hence, proposing new high performance metaheuristic algorithms is welcome. In addition, as shown by Mirjalali [12], a meta-heuristic does not necessarily need actual inspiration, and simple mathematical functions can also be used to design optimization algorithms.

Therefore, the motivation of this study is to propose a simple and effective global optimization technique for complex problems. The proposed algorithm utilizes a new adaptive step size adjustment scheme, which has a simple concept and significantly improves the performance of the search process. It is worth mentioning that the presented method could provide a well balance between exploration and exploitation. According to the capability of the proposed algorithm in finding a (near) global solution in a reasonable time, it is called a ''Golden Search Optimizer'' (GSO).

The remainder of this paper is organized as follows: In section II, a survey on metaheuristic algorithms is presented. Section III presents a description and step-by-step framework of the proposed GSO algorithm. In section IV, the empirical evaluation of GSO in an extensive test environment is carried out, and the simulation results are compared with the selected metaheuristic algorithms. Finally, the summary and conclusions of the study are presented in Section V.

II. RELATED WORKS

In recent years, optimization has become a popular research field and an economical way to find an optimal solution to complex problems. According to Fig. 1, we have divided the optimization algorithms into four categories based on the type of inspiration. The first group is swarm-based algorithms, which consist of a population of simple agents interacting locally with one another and with their environment. These algorithms employ artificial swarms of autonomous agents that follow simple rules (relative to the system's complexity) to update their state through time. In the optimization framework, a swarm consists of a number of search agents whose state represents a candidate solution to the optimization problem at hand. The agents adhere to basic rules that promote their cooperation during the search for better solutions. The search is typically defined as an iterative, highly distributed procedure where the agents explore the given search space for the problem while communicating their findings to their mates. Information sharing is a key issue for the efficiency of such computational schemes [13]. Some examples of swarm based algorithms include Particle Swarm Optimization (PSO) [14], Firefly Algorithm (FFA) [15], Ant Colony Optimization (ACO) [16], Artificial Bee Colony (ABC) [17], Krill Herd (KH) [18], Whale Optimization Algorithm (WOA), Crow Search Algorithm (CSA) [19], Rat Swarm Optimizer [20], Sperm Swarm Optimization [21], Chameleon Swarm Algorithm [22]. The second group is evolutionary algorithms, which are efficient heuristic search methods based on Darwinian evolution with powerful characteristics of robustness and flexibility to capture global solutions to complex optimization problems. Genetic Algorithm (GA), Evolution Strategy (ES), Genetic Programming (GP), Biogeography Based Optimizer (BBO), Evolutionary Programming (EP) [23], Differential Evolution (DE) [24], Virulence Optimization Algorithm (VOA) [25] are some of these algorithms. The third category is physics-based algorithms inspired by natural phenomena and imitating physical and biological processes of nature, such as Simulated Annealing (SA) [1], Black Hole Algorithm (BH) [26], Curved Space Optimization Algorithm (CSO) [27], Ray Optimization (RO) [28], Gravitational Search Algorithm (GSA) [29]. The fourth category is human-based algorithms that allow humans to contribute solution suggestions to the evolutionary process, such as Harmony Search (HS) [30], Teaching Learning Based Optimization (TLBO) [31], Imperialist Competitive Algorithm (ICA) [32], Exchanged Market Algorithm (EMA) [33], Thermal Exchange Optimization (TEO) [34] and Tabu Search (TS) [35]. There are other algorithms that are a combination or modification of algorithms in these four categories. Some of them are: modified particle swarm optimization [36], [37], modified harmony search algorithm [38], modified gravitational search algorithm [39], [40], modified ant colony optimization [41], modified sine cosine algorithm [42], [43], modified wild horse optimization [44], modified slime mould algorithm [45], hybrid genetic algorithm and particle swarm optimization [46], hybrid firefly algorithm [47], hybrid sperm swarm optimization and gravitational search algorithm [48], hybrid tunicate swarm algorithm and pattern search [49], hybrid arithmetic optimization algorithm and sine cosine algorithm [50].

Some of the most famous of these optimization algorithms are described below. The Genetic Algorithm (GA) is the most popular evolutionary inspiration technique that mimics the principles of Charles Darwin's Compatibility Survival Theory. This method involves the selection process, the crossover, and the mutation process to replace the worst solution in each generation. In this algorithm, solutions are improved according to the best solutions obtained by each particle so far and the best solution found for the overall swarm. The ACO algorithm imitates the collective behavior of ants in finding the shortest path from the nest to the food source. One of the most important and most interesting behaviors of ants is their behavior when finding food, and in particular, how to find the shortest path between food and nest. This kind of behavior by the ants has a kind of swarm intelligence that has recently attracted the attention of scientists. In the natural world, ants of some species (initially) wander randomly, and upon finding food, return to their colony while laying down pheromone trails. If other ants find such a path, they are likely not to keep travelling at random, but instead to follow the trail, returning and reinforcing it if they eventually find food. The DE is presented to overcome the main defect of the genetic algorithm, namely the lack of local search in this algorithm.

The main difference between GA and DE is in the selection operators. In the ABC model, the colony consists of three groups of bees: employed bees, onlookers, and scouts. It is assumed that there is only one artificially employed bee for each food source. In other words, the number of employed bees in the colony is equal to the number of food sources around the hive. Employed bees go to their food source and come back to the hive to dance in this area. The employed bee whose food source has been abandoned becomes a scout and starts to search for a new food source. Onlookers watch the dances of employed bees and choose food sources depending on the dances. The FA is a meta-heuristic algorithm, inspired by the flashing behavior of fireflies. The primary purpose of a firefly's flash is to act as a signal to attract other fireflies. The Firefly Algorithm is based on the life cycle of firefly worms. In the firefly algorithm, worms tend to be attracted to high-attraction lights. The worm with less light can be absorbed by the worm with more light. In this situation, the swarm moves like a PSO. Of course, the degree of motion is also taken randomly to prevent the early convergence of the FA.

Optimization algorithms have advantages and disadvantages. For example, the PSO algorithm solves problems with small and simple dimensions well, but has poor performance in high-dimensional, complex, and hybrid problems and suffers from premature convergence [51].

The Gravitational search algorithm (GSA) is a novel meta-heuristic stochastic optimization algorithm inspired by the law of gravity and mass interactions [29]. In GSA, individuals, called agents, are a collection of masses that interact with each other based on Newtonian gravity and the laws of motion. The agents share information using the gravitational force to guide the search towards the best location in the search space. The high performance and the global search ability of GSA in solving various nonlinear functions are inferred from the results of experiments undertaken previously [29], [52]. In GSA, all agents move to a new place, the direction and distance are determined by their velocities. By changing the velocities over time, the agents are likely to move towards the global optima.

The sine cosine algorithm (SCA) is a relatively new meta-heuristic optimization approach developed by Mirjalili in 2016 [12]. Compared with other meta-heuristics, the SCA has a simple concept and structure and does not have complicated mathematical functions. In the SCA, the formulas for updating the population rely solely on sine and cosine functions. SCA is better than other competitive methods at finding optimal solutions and is suitable for tackling real-world optimization problems [53].

The Tunicate Swarm Algorithm (TSA) is a recently developed swarm-based algorithm [54]. Tunicates use swarm intelligence and jet propulsion at sea to choose the optimal state for seeking food in their surroundings. TSA outperforms other competitor approaches when it comes to identifying optimal solutions and is well-suited to real-world optimization challenges.

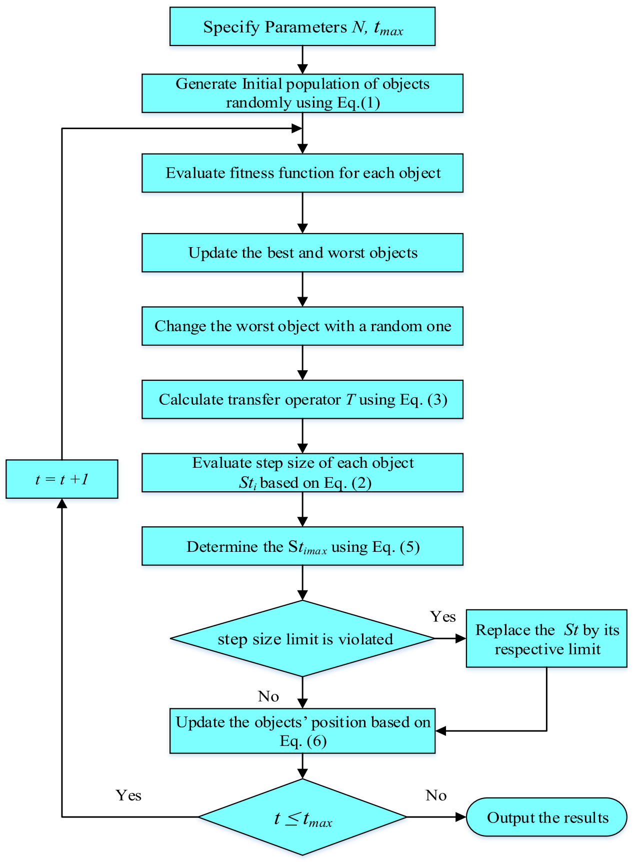
Specify Parameters N, t_{max}
Generate Initial population of objects randomly using Eq.(1)
Evaluate fitness function for each object
Update the best and worst objects
Change the worst object with a random one
Calculate transfer operator T using Eq. (3)
Evaluate step size of each object St_i based on Eq. (2)
Determine the St_{imax} using Eq. (5)
step size limit is violated
Yes
Replace the St by its respective limit
No
Update the objects' position based on Eq. (6)
$t = t + 1$
Yes
$t \le t_{max}$
No
Output the results

FIGURE 2. **The GSO flowchart.**

The Grey Wolf Optimizer (GWO) is a novel heuristic swarm intelligent optimization algorithm proposed by Seyedali Mirjalili *et al.* in 2014 [55]. The wolf, as a top predator in the food chain, has a strong ability to capture prey. Wolves generally like social life, and in the interior of the wolves, there exists a rigid social hierarchy. The GWO algorithm mimics the leadership hierarchy and hunting mechanism of grey wolves in nature. Four types of grey wolves, such as alpha, beta, delta, and omega, are employed in simulating the leadership hierarchy [55]. In addition, the three main steps of hunting: searching for prey, encircling prey, and attacking prey are implemented. The GWO is able to provide very competitive results compared to other well-known metaheuristics [55].

III. GOLDEN SEARCH OPTIMIZATION (GSO) ALGORITHM

Despite the wide variety of population-based algorithms in the field of stochastic optimization, they are almost all the same in the process of finding the optimum. These algorithms commonly start the search process with a randomly generated initial population (potential solutions). These random solutions are evaluated during iterations using a fitness function and improved by a set of formulas, which is the core of an optimization technique, until satisfying some termination condition.

In this study, according to the principles and commonly used procedures of metaheuristic algorithms, a very simple but effective optimization method called the Golden Search Optimizer (GSO) is proposed. In fact, the new method combines some major advantages of previously presented algorithms like particle swarm optimization (PSO) and sine cosine algorithms (SCA) to provide a fine balance between global exploration and local exploitation and to avoid premature convergence. In the GSO algorithm, the objects update their position using a step size parameter, which is almost the same as the velocity in the PSO algorithm [7]. However, instead of simple random values, the GSO utilizes sine and cosine functions. The oscillation behavior of sine and cosine functions allows one object to be re-positioned around another one, and it can guarantee exploitation of the space defined between two solutions. In addition, the exploration capability of the algorithm will be improved by increasing the range of sine and cosine functions, which allow a solution to update its position outside the space between itself and another solution [12]. Apart from its simple structure, the algorithm outperforms other metaheuristics in terms of convergence to the global best solution.

As a population-based metaheuristic optimization technique, GSO also commences the search process with an initial random population of objects (candidate solutions). In every iteration, the algorithm updates the position of the objects using a step size parameter until satisfying some termination criteria. The following is the detailed mathematical expression of the GSO steps.

A. ALGORITHMIC STEPS

Theoretically, as a global optimization algorithm, GSO encompasses both exploration and exploitation phases and could provide a well balance between these two contradictory abilities.

The algorithm has three main parts; including population initialization, population evaluation, and updating the current population. The step-by-step procedure of the proposed GSO is detailed as follows.

Step 1— population initialization

GSO starts the search process with a set of randomly generated objects (possible solutions) in the search space according to the following equation:

$$O_i = lb_i + rand \times (ub_i - lb_i)\,; \quad i = 1, 2, \ldots, N \qquad (1)$$

where O_i presents the location of ith objects in the search space. Moreover, ub_i and lb_i are the lower and upper bounds of the object, respectively.

Step 2- population evaluation

In this step, initial population will be evaluated based on objective function and the object with the best fitness value selected as $Ogbest_i$

Step3- golden change

In the third step, the objects will be sorted according to their fitness and the object with the worst fitness will be changed by a random solution.

Step4- step size evaluation

In each iteration of the optimization process, the objects are moved toward the best solution using the step size operator (St_i). St equation consists of three parts. The first part represents the previous value of the step size that is multiple by the transform operator (T) which decreases iteratively to balance global and local search of the algorithm. The second part presents the distance between current position of the ith object and its personal best position obtained so far by the cosine of a random value in the range of 0 and 1.

The final part represents the distance between the ith object's current position and the best position obtained thus far among all objects, multiplied by the sine of a random value between 0 and 1. In the first iteration of optimization process, St_i will be generated randomly and updated using the following equation during the iterations:

$$St_i(t+1) = T.St_i(t) + C_1.\cos(r_1).(Obest_i - x_i(t)) + C_2.\sin(r_2).\left(Ogbest_i - x_i(t)\right) \quad (2)$$

where C_1 and C_2 are a random numbers between 0 and 2, r_1 and r_2 are random numbers in the range of (0,1), $Obest_i$ is the best previous position obtained by the ith object so far and T is transfer operator which transforms search from exploration to exploitation to improve the search performance and controlling the balance between global search in early iterations and local search in late iterations. Actually, T is a decreasing function and evaluated using Eq. (3)

$$T = 100 \times \exp(-20 \times \frac{t}{t_{max}}) \quad (3)$$

where. t_{max} is the maximum number of iterations.

Step5- step size limitation

At each iteration, the algorithm proceeds by adjusting the distance that each object moves in every dimension of the problem hyperspace. Equation (2) shows that the step size is a stochastic variable and may allow the objects follows wider cycles in the problem space. In order to control these oscillations and to avoid explosion and divergence, a reasonable interval is introduced to clamp the object's movement according to:

$$-St_{imax} \leq St_i \leq St_{imax} \quad (4)$$

where St_{imax} is a designated maximum movement allowed, which defines the maximum change one object can undergo in its positional coordinates during an iteration based on the following equation:

$$St_{imax} = 0.1 \times (ub_i - lb_i) \quad (5)$$

Step 6- update position (generate new population)

In this stage, the objects move toward the global optimum in the search space according to the following equation:

$$O_i(t+1) = O_i(t) + St_i(t+1) \quad (6)$$

The flowchart of the proposed GSO algorithm is presented in Fig. 2.

B. COMPARATIVE TIME COMPLEXITY ANALYSIS

Computational time complexity analysis can be conducted in order to evaluate the overall performance of a new optimization algorithm from different points of view. In computer sciences, the "Big O notation" is a mathematical notation that represents the required running time of an algorithm by considering the growth rate when dealing with different inputs [13].

The time complexity analysis of most algorithms involves analyses of three components. Likewise, the time complexity analysis of the proposed GSA also requires analyses of these three components:

1. Time complexity of initialization of the population, generally calculated by O($N \times D$) where N denotes the population size and D denotes the dimensions of the problem.

2. Time complexity of initial fitness evaluation, generally evaluated by O ($N \times F(\boldsymbol{X})$), where $F(\boldsymbol{X})$ represents the objective function.

3. Time complexity of the main loop, generally calculated by O ($t_{max} \times (N \times D + N \times F(\boldsymbol{X}))$), where t_{max} is the maximum number of iterations.

Hence, the total time complexity of GSO algorithm is O(t_{max} ($N \times D + N \times F(\boldsymbol{X})$)).

IV. COMPARATIVE ANALYSIS OF THE GSO

Because of the stochastic nature of metaheuristic algorithms, several test cases should be employed to confirm the effectiveness of an algorithm. In this study, the performance of GSO is evaluated on a well-studied and diverse set of benchmark functions from the literature [56], [57] against a good combination of some well-known state-of-the-art algorithms. All of these functions are minimization problems, which are useful for evaluating the search efficiency and convergence rate of optimization algorithms. The mathematical formulation and characteristics of these test functions are available in Tables 1, 2 and 3. Figures 3, 4 and 5 show three-dimensional drawings of these benchmark functions. Moreover, the cost functions along with the dimensions, ranges, and minimum inputs related to the single exponential benchmark functions are shown in Table 1-3.

This benchmark set covers three main groups: unimodal functions with a unique global best for testing the convergence speed and exploitation ability of the algorithms; multimodal functions with multiple local solutions and a global optimum for testing local optima avoidance and exploration capability of an algorithm; and finally multimodal functions with a fixed dimension. The proposed algorithm is coded in MATLAB R2020b programming software.

The MATLAB code will be released after the acceptance of the paper. In order to verify the success of a newly proposed computational intelligence algorithm, it is recommended that the performance of the new method be compared with the other algorithms that are widely accepted in the field. In this paper, the results and performance of the proposed GSO are compared with other well-established optimization algorithms such as the Gravitational Search

TABLE 1. Description of unimodal benchmark functions.

Function	Range	f_{min}	n (*Dim*)
$F_1(X)=\sum_{i=1}^{n} x_i^2$	$[-100,100]^n$	0	30
$F_2(X)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	$[-10,10]^n$	0	30
$F_3(X)=\sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j\right)^2$	$[-100,100]^n$	0	30
$F_4(X)=\max_i\{\lvert x_i\rvert,1\le i\le n\}$	$[-100,100]^n$	0	30
$F_5(X)=\sum_{i=1}^{n-1}\left[100\left(x_{i+1}-x_i^2\right)^2+(x_i-1)^2\right]$	$[-30,30]^n$	0	30
$F_6(X)=\sum_{i=1}^{n}([x_i+0.5])^2$	$[-100,100]^n$	0	30
$F_7(X)=\sum_{i=1}^{n} ix_i^4+random[0,1)$	$[-1.28,1.28]^n$	0	30

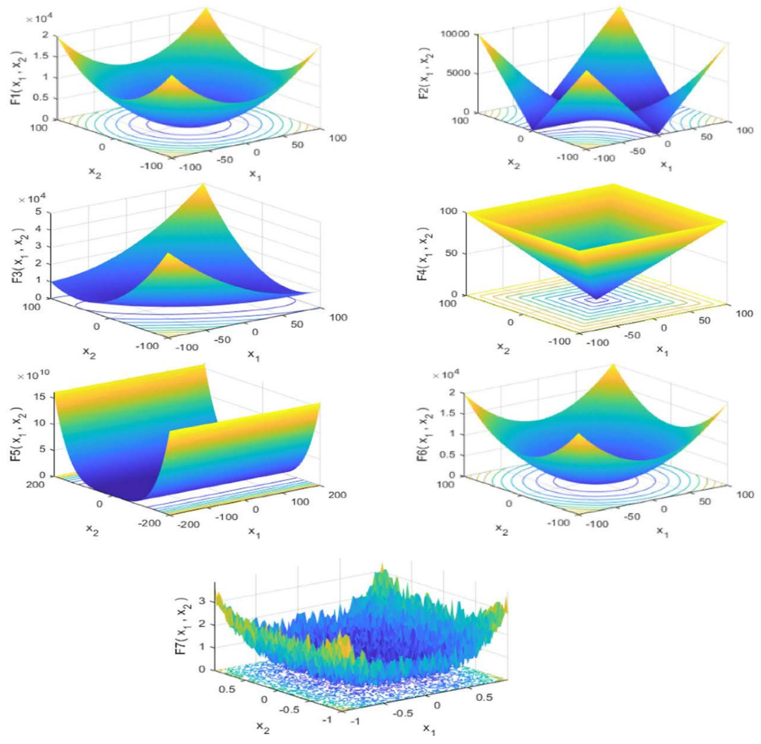

FIGURE 3. 3-D versions of unimodal benchmark functions.

Algorithm (GSA) [29], Sine-Cosine Algorithm (SCA) [12], Tunicate Swarm Algorithm (TSA) [54], and Grey Wolf Optimizer (GWO) [55]. These algorithms have proved their effectiveness and robustness when compared with other well-established methods like PSO, GA, FA, and so on [12], [29], [54], [55]. It should be noted that the performance and convergence of these metaheuristic methods completely depend on the internal parameters of the algorithms. GSO has a simple structure and needs only two main parameters, N (number of objects) and *MaxIter* (maximum number of iterations).

It is found through experiments that a lower value of N results in premature convergence and a higher value improves exploration but increases elapsed time significantly. The appropriate value of N is considered to be 30, and the maximum number of iterations is equal to 1000. In Table 4, the key parameters of the selected methods are presented.

These values have been determined using the reference-based parameter identification process according to those recommended by their authors in the original papers. The references for each method are presented in the third column of Table 4.

Because metaheuristic methods are stochastic, the results of a single run may be unreliable, and the algorithms may obtain better or worse solutions than the previous one. Therefore, statistical analysis should be applied to have a fair comparison and effective evaluation of the algorithms. Regarding this issue, for the selected algorithms,

TABLE 2. Description of multimodal benchmark functions.

Function	Range	f_{min}	*n* (*Dim*)
$F_8(X) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	$[-500, 500]^n$	428.9829×*n*	30
$F_9(X) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^n$	0	30
$F_{10}(X) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]^n$	0	30
$F_{11}(X) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^n$	0	30
$F_{12}(X) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$	$[-50, 50]^n$	0	30
$y_i = 1 + \frac{x_{i+4}}{4} \quad u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & a < \ x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$			
$F_{13}(X) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	$[-50, 50]^n$	0	30

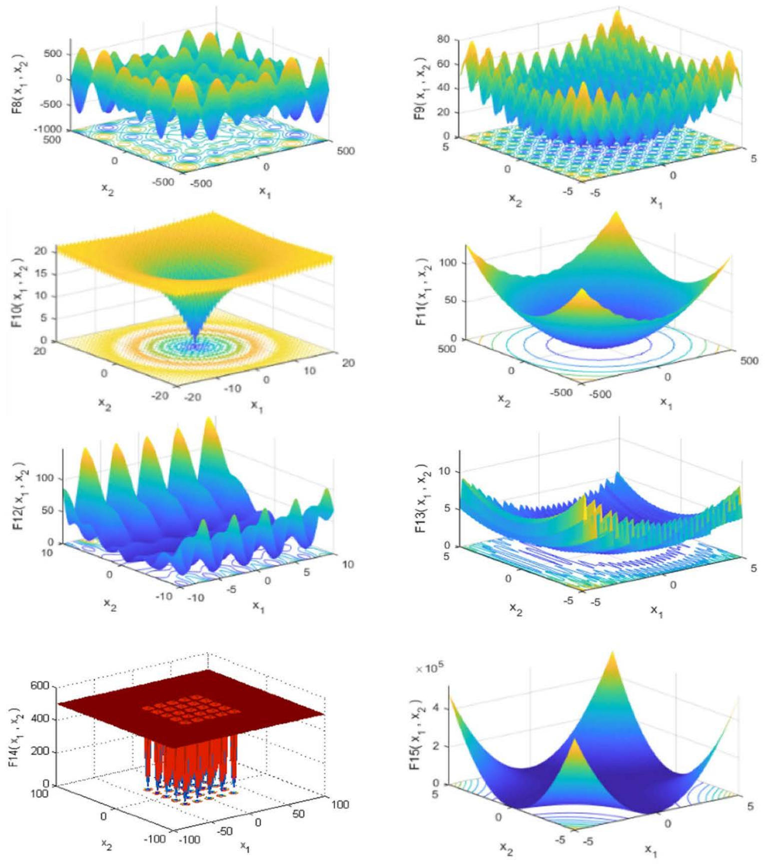

FIGURE 4. 3-D versions of multimodal benchmark functions.

TABLE 3. Description of fixed-dimension multimodal benchmark functions.

Function	Range	f_{min}	n (*Dim*)
$F_{14}(X)=\left(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+(x_i-a_{ij})^6}\right)^{-1}$	$[-65.53, 65.53]^2$	1	2
$F_{15}(X)=\sum_{i=1}^{11}\left[a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}\right]^2$	$[-5,5]^4$	0.00030	4
$F_{16}(X)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	$[-5,5]^2$	-1.0316	2
$F_{17}(X)=\left(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos x_1+10$	$[-5,5]^2$	0.398	2
$F_{18}(X)=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]\times[30+(2x_1-3x_2)^2\times(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$	$[-2,2]^2$	3	2
$F_{19}(X)=-\sum_{i=1}^{4}c_i\,exp\left(-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})\right)^2$	$[1,3]^3$	-3.86	3
$F_{20}(X)=-\sum_{i=1}^{4}c_i\,exp\left(-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})\right)^2$	$[0,1]^6$	-3.32	6
$F_{21}(X)=-\sum_{i=1}^{5}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	$[0,10]^n$	-10.1532	4
$F_{22}(X)=-\sum_{i=1}^{7}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	$[0,10]^n$	-10.4028	4
$F_{23}(X)=-\sum_{i=1}^{10}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	$[0,10]^n$	-10.5363	4

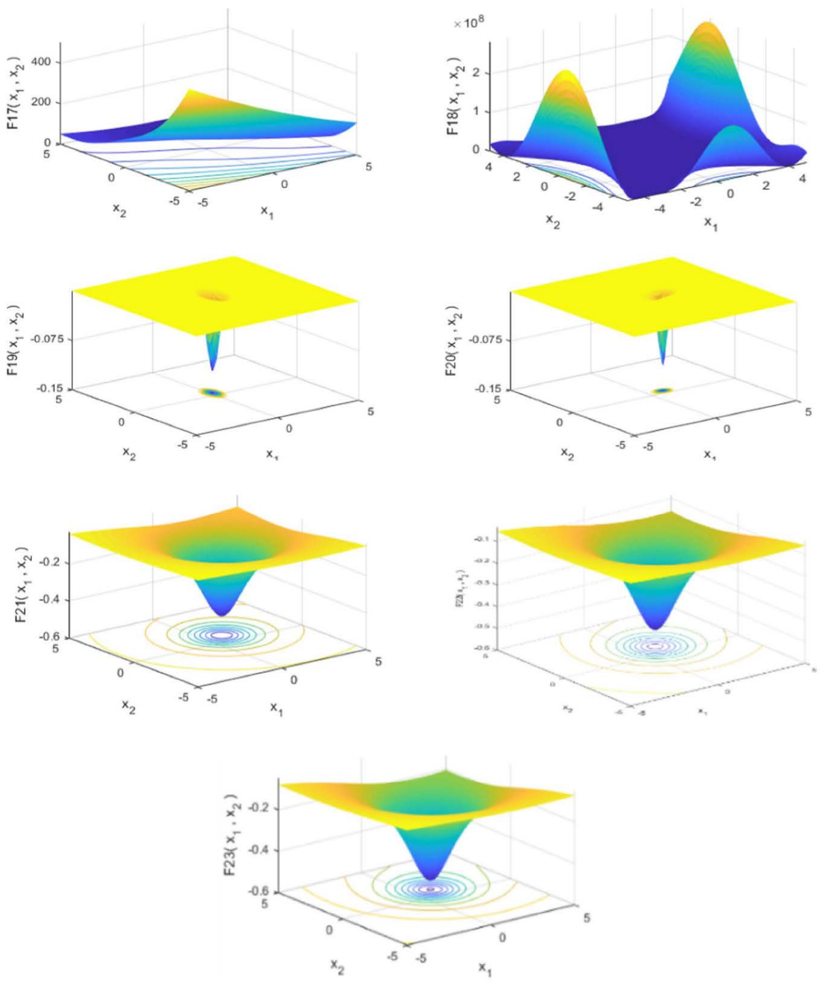

FIGURE 5. 3-D versions of fixed-dimension multimodal benchmark functions.

TABLE 4. Parameter setting of the selected algorithms.

year	Algorithm	Refernces	Parameter	Specifications
2022	Golden Search Optimization (GSO)	Current study	Number of objects	30
			Maximum iteration	1000
2009	Gravitational Search Algorithm (GSA)	Rashedi and Nezamabadi-pour [29]	Search agents	50
			Gravitational constant	100
			Alpha coefficient	20
			Number of generations	1000
2014	Grey Wolf Optimizer (GWO)	Mirjalili et al. [55]	Search agents	80
			Control parameter ($\vec{a}$)	[2,0]
			Number of generations	1000
2016	Sine Cosine Algorithm (SCA)	Mirjalili [12]	Search agents	80
			Number of elites	2
			Number of generations	1000
2020	Tunicate Swarm Algorithm (TSA)	Kaur et al. [54]	Search agents	80
			Parameter $Pmin$	1
			Parameter $Pmax$	4
			Number of generations	1000

TABLE 5. Comparison of different methods in solving unimodal test functions in Table 1.

Function	Statistics	GSO	GSA	SCA	TSA	GWO
F_1	Best	**0.00**	1.0013E-17	1.5523E-07	5.1458E-61	2.4915E-61
	Worst	**0.00**	3.1868E-17	0.0043	1.1586E-54	3.8647E-58
	Mean	**0.00**	2.1148E-17	2.3458E-04	8.3155E-56	4.9162E-59
	Median	**0.00**	2.0077E-17	1.9737E-05	7.1012E-58	1.0534E-59
	Std.	**0.00**	5.8150E-18	7.9295E-04	2.4905E-55	1.0230E-58
F_2	Best	**0.00**	1.5282E-08	1.5005E-09	1.1196E-35	8.3612E-36
	Worst	**0.00**	3.3313E-08	9.8446E-06	3.2814E-32	5.3488E-34
	Mean	**0.00**	2.3935E-08	1.6882E-06	2.1532E-33	8.3658E-35
	Median	**0.00**	2.3469E-08	5.4000E-07	3.1044E-34	5.9294E-35
	Std.	**0.00**	4.0025E-09	2.4046E-06	6.0237E-33	9.8594E-35
F_3	Best	**0.00**	102.9550	70.8285	2.5684E-32	1.2533E-19
	Worst	**0.00**	468.6160	2.6762E+03	2.4492E-17	3.5572E-13
	Mean	**0.00**	245.4694	789.1620	8.1741E-19	1.5096E-14
	Median	**0.00**	221.1150	619.4506	1.8696E-24	2.0740E-17
	Std.	**0.00**	100.1024	746.2287	4.4714E-18	6.5547E-14
F_4	Best	**0.00**	2.2498E-09	1.2610	3.2458E-08	9.8174E-16
	Worst	**0.00**	5.0857E-09	35.6743	6.3429E-05	2.4431E-13
	Mean	**0.00**	3.3030E-09	9.3080	1.0102E-05	1.9487E-14
	Median	**0.00**	3.2020E-09	6.9806	2.0270E-06	6.3817E-15
	Std.	**0.00**	7.4424E-10	8.0720	1.6927E-05	4.4955E-14
F_5	Best	**3.5924E-04**	25.7459	27.3230	25.6273	25.2273
	Worst	**3.5924E-04**	220.9110	49.5110	29.5430	28.7294
	Mean	**3.5924E-04**	42.2647	29.9106	28.4422	26.9256
	Median	**3.5924E-04**	26.1443	29.0097	28.8115	27.1173
	Std.	**1.6541E-19**	45.4674	4.1508	0.7616	0.8418
F_6	Best	1.9836E-07	**9.711E-18**	3.4070	2.0585	0.2466
	Worst	0.0220	**8.645E-16**	4.4435	4.7791	1.2619
	Mean	0.0021	**3.097E-17**	4.0360	3.6724	0.6376
	Median	1.9836E-07	**2.953E-17**	4.0572	3.5615	0.7452
	Std.	0.0056	**6.165E-18**	0.2954	0.6918	0.3353
F_7	Best	**3.3002E-10**	0.0061	0.0015	6.7114E-04	1.5222E-04
	Worst	**1.2212E-04**	0.0462	0.0431	0.0036	0.0022
	Mean	**7.2800E-06**	0.0237	0.0116	0.0018	7.9976E-04
	Median	**3.3002E-10**	0.0222	0.0078	0.0018	7.0098E-04
	Std.	**2.4887E-05**	0.0098	0.0101	7.7268E-04	4.6287E-04

TABLE 6. Comparison of different methods in solving multimodal test functions in Table 2.

Function	Statistics	GSO	GSA	SCA	TSA	GWO
F_8	Best	**-1.2050E+04**	-3.6279E+03	-5.2993E+03	-7.8992E+03	-8.8178E+03
	Worst	**-1.1096E+04**	-2.0033E+03	-3.5321E+03	-5.2761E+03	-4.9742E+03
	Mean	**-1.2005E+04**	-2.7826E+03	-4.0769E+03	-6.6126E+03	-6.2524E+03
	Median	**-1.2050E+04**	-2.7464E+03	-3.9720E+03	-6.6131E+03	-6.2270E+03
	Std.	**186.4737**	365.4671	336.8249	599.2609	852.4634
F_9	Best	**0.00**	8.9546	1.0560E-06	77.7761	0.00
	Worst	**0.00**	21.8891	51.4451	254.9883	10.0548
	Mean	**0.00**	15.6209	5.9694	151.4539	0.8853
	Median	**0.00**	15.9193	9.3391E-04	149.6596	0.00
	Std.	**0.00**	3.1043	12.2476	35.8717	2.4438
F_{10}	Best	**8.8818E-16**	2.5288E-09	1.5579E-05	1.5099E-14	1.1546E-14
	Worst	**8.8818E-16**	4.4823E-09	20.2198	4.3125	2.2204E-14
	Mean	**8.8818E-16**	3.4912E-09	14.3622	2.4095	1.5928E-14
	Median	**8.8818E-16**	3.4766E-09	20.1275	2.9381	1.5099E-14
	Std.	**0.00**	5.1530E-10	8.9778	1.3920	2.5861E-15
F_{11}	Best	**0.00**	1.6952	4.8381E-07	0.00	0.00
	Worst	**0.00**	10.6642	0.7703	0.0159	0.0140
	Mean	**0.00**	4.2510	0.1368	0.0077	0.0014
	Median	**0.00**	3.5667	0.0032	0.0082	0.00
	Std.	**0.00**	2.0234	0.2218	0.0057	0.0041
F_{12}	Best	3.9317E-08	**8.2033E-20**	0.2631	0.2738	0.0121
	Worst	1.5374E-04	**0.1037**	5.6300	13.8088	0.0920
	Mean	**7.0132E-06**	**0.0198**	0.9568	6.3735	0.0364
	Median	4.0116E-07	**1.3512E-19**	0.4964	6.7411	0.0329
	Std.	**2.7947E-05**	**0.0400**	1.1497	3.4586	0.0201
F_{13}	Best	2.6099E-05	**1.3407E-18**	1.8452	1.7796	0.1006
	Worst	9.1819E-04	**0.0110**	22.5849	4.1077	1.0416
	Mean	**5.0093E-04**	7.3249E-04	3.4211	2.8976	0.5280
	Median	5.1139E-04	**2.0686E-18**	2.3552	2.8914	0.5238
	Std.	**4.2513E-04**	0.0028	3.9911	0.6436	0.2359

30 independent runs are performed and statistical results are collected and reported in Tables 5-7. In the first stage of experiments, all the functions are optimized with the dimension presented in the last column of Tables 1-3. The results of Tables 5-7 show the best (minimum), worst (maximum), mean (average), median, and standard deviation (*Std*) of the solutions obtained from experiments using the selected optimization algorithms. The best results among the five algorithms are shown in bold. According to the results of these tables in the following subsections, the exploration, exploitation, convergence rate, and scalability of the new method are investigated using a comparative performance comparison of GSO against four selected algorithms.

A. EXPLOITATION CAPABILITY

Unimodal test functions can be considered to investigate the exploitation capability of an optimization algorithm [55], [58]. In this study, to evaluate the ability of GSO to exploit the promising regions, seven unimodal benchmark functions (*F1* to *F7*) are solved, and the results are compared with four selected optimization methods in Table 5. The results of this table show that, for all unimodal functions except F6, GSO could provide a better solution. In addition, for four functions (*F1-F4*), GSO reached the global optima. It means that the new algorithm has a large potential search space compared with the other optimization algorithms. It should be noted that the good exploitative capability of GSO is due to the effective position updating strategy.

B. EXPLORATION VERIFICATION

In order to evaluate the capability of an optimization algorithm to effectively explore the search space, multimodal benchmark functions that have many local optima are usually considered [55], [58]. Based on the presented procedure, 16 multimodal functions (F8 to F23) are minimized. According to the results of Tables 6-7, it can be observed that the best and mean values reached by GSO for most of the functions (except F12, F13, F17-F19) are significantly better than the other methods. However, the results for F17-F19 functions are also comparable to the other algorithms. In addition, for F12 and F13, mean values of GSO are smaller than the robust GSA method, and the results are much better than those obtained by SCA, TSA, and GWO. From the standard deviation point of view, which indicates the stability of the algorithm, the results show that GSO is a more stable method when compared with the other techniques.

TABLE 7. **Comparison of different methods in solving multimodal test functions in Table 3.**

Function	Statistics	GSO	GSA	SCA	TSA	GWO
F_{14}	Best	**0.9980**	**0.9980**	**0.9980**	**0.9980**	**0.9980**
	Worst	**0.9980**	8.0858	2.9821	12.6705	12.6705
	Mean	**0.9980**	3.6212	1.1964	7.6657	4.1312
	Median	**0.9980**	3.0452	0.9980	10.7632	2.9821
	Std.	**1.4772E-11**	2.1942	0.6054	4.8845	4.1443
F_{15}	Best	**3.1381E-04**	0.0012	3.4063E-04	3.751E-04	3.1749E-04
	Worst	**3.9684E-04**	0.0118	0.0014	0.0566	0.0204
	Mean	**3.3641E-04**	0.0025	8.5975E-04	0.0043	0.0044
	Median	**3.2323E-04**	0.0021	7.3095E-04	4.5390E-04	3.0754E-04
	Std.	**2.4589E-05**	0.0019	3.8089E-04	0.0116	0.0081
F_{16}	Best	**-1.0316**	**-1.0316**	**-1.0316**	-1.0316	**-1.0316**
	Worst	**-1.0316**	**-1.0316**	**-1.0316**	-1.0000	**-1.0316**
	Mean	**-1.0316**	**-1.0316**	**-1.0316**	-1.0306	**-1.0316**
	Median	**-1.0316**	**-1.0316**	**-1.0316**	-1.0316	**-1.0316**
	Std.	1.8597E-06	5.6082E-05	1.0395E-05	0.0058	**4.7385E-09**
F_{17}	Best	**0.3979**	**0.3979**	**0.3979**	**0.3979**	**0.3979**
	Worst	**0.3979**	**0.3979**	0.3992	0.3980	**0.3979**
	Mean	**0.3979**	**0.3979**	0.3982	0.3979	**0.3979**
	Median	**0.3979**	**0.3979**	0.3981	0.3979	**0.3979**
	Std.	**0**	**0**	3.4881E-04	1.3711E-05	1.1055E-06
F_{18}	Best	**3.000**	**3.000**	**3.000**	**3.000**	**3.0000**
	Worst	3.005	**3.000**	**3.000**	84.00	**3.0000**
	Mean	3.002	**3.000**	**3.000**	5.700	**3.0000**
	Median	3.002	**3.000**	**3.000**	3.000	**3.0000**
	Std.	1.7022E-05	**1.5927E-15**	5.3498E-06	14.7885	9.5052E-06
F_{19}	Best	**-3.8628**	**-3.8628**	-3.8625	**-3.8628**	**-3.8628**
	Worst	**-3.8628**	**-3.8628**	-3.8539	-3.8549	-3.8549
	Mean	**-3.8628**	**-3.8628**	-3.8560	-3.8625	-3.8620
	Median	**-3.8628**	**-3.8628**	-3.8548	-3.8628	-3.8628
	Std.	**1.3625E-16**	2.4795E-05	0.0029	0.0014	0.0022
F_{20}	Best	-3.0987	**-3.3220**	-3.1918	-3.3216	**-3.3220**
	Worst	-1.8554	**-3.3220**	-2.0488	-3.0885	-3.0292
	Mean	-2.9535	**-3.3220**	-3.0155	-3.2538	-3.2493
	Median	-2.9874	**-3.3220**	-3.0139	-3.2028	-3.2625
	Std.	0.2446	**1.3550E-15**	0.1974	0.0671	0.0821
F_{21}	Best	**-10.1532**	**-10.1532**	-8.1370	-10.1361	-10.1531
	Worst	**-10.1532**	-2.6829	-0.8802	-2.6669	-5.0999
	Mean	**-10.1532**	-6.3969	-4.3187	-7.2879	-9.4790
	Median	**-10.1532**	-3.9547	-4.9053	-7.4198	-10.1526
	Std.	**2.4990E-07**	3.5901	2.0785	2.8594	1.7469
F_{22}	Best	**-10.4028**	-10.4009	-9.0513	-10.3812	-10.4029
	Worst	**-10.4028**	-10.4029	-0.9074	-2.7427	-5.0877
	Mean	**-10.4028**	-10.4029	-5.4154	-7.8325	-10.2253
	Median	**-10.4028**	-10.4028	-5.0380	-10.2554	-10.4025
	Std.	**5.4202E-15**	4.6649E-06	1.7315	3.1843	0.9703
F_{23}	Best	**-10.5363**	-10.5364	-9.3851	-10.5193	**-10.5363**
	Worst	**-10.5362**	-10.5364	-3.2531	-1.6752	-10.5354
	Mean	**-10.5363**	-10.5364	-5.2925	-7.6730	-10.5359
	Median	**-10.5363**	-10.5364	-5.0398	-10.4114	-10.5360
	Std.	**2.4855E-05**	1.8366E-15	1.0982	3.7585	2.5855E-04

From the analysis, it can be concluded that GSO either outperforms the other algorithms or performs almost equivalently. The consistent performance of the new method for such a comprehensive suite of multimodal benchmark functions verifies its superior capabilities of exploration.

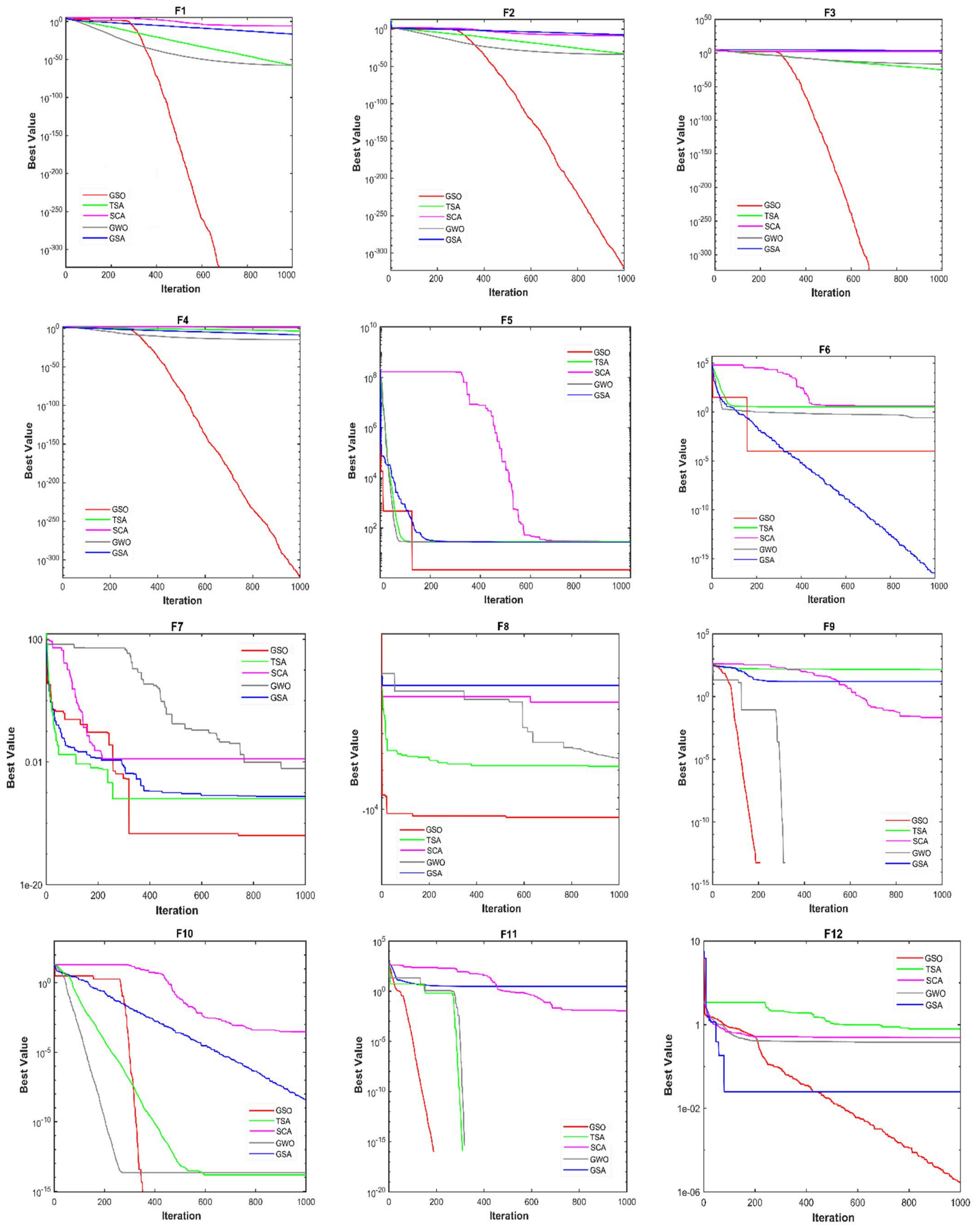

FIGURE 6. Comparison of convergence curves of GSO and selected algorithms for *F1-F23*.

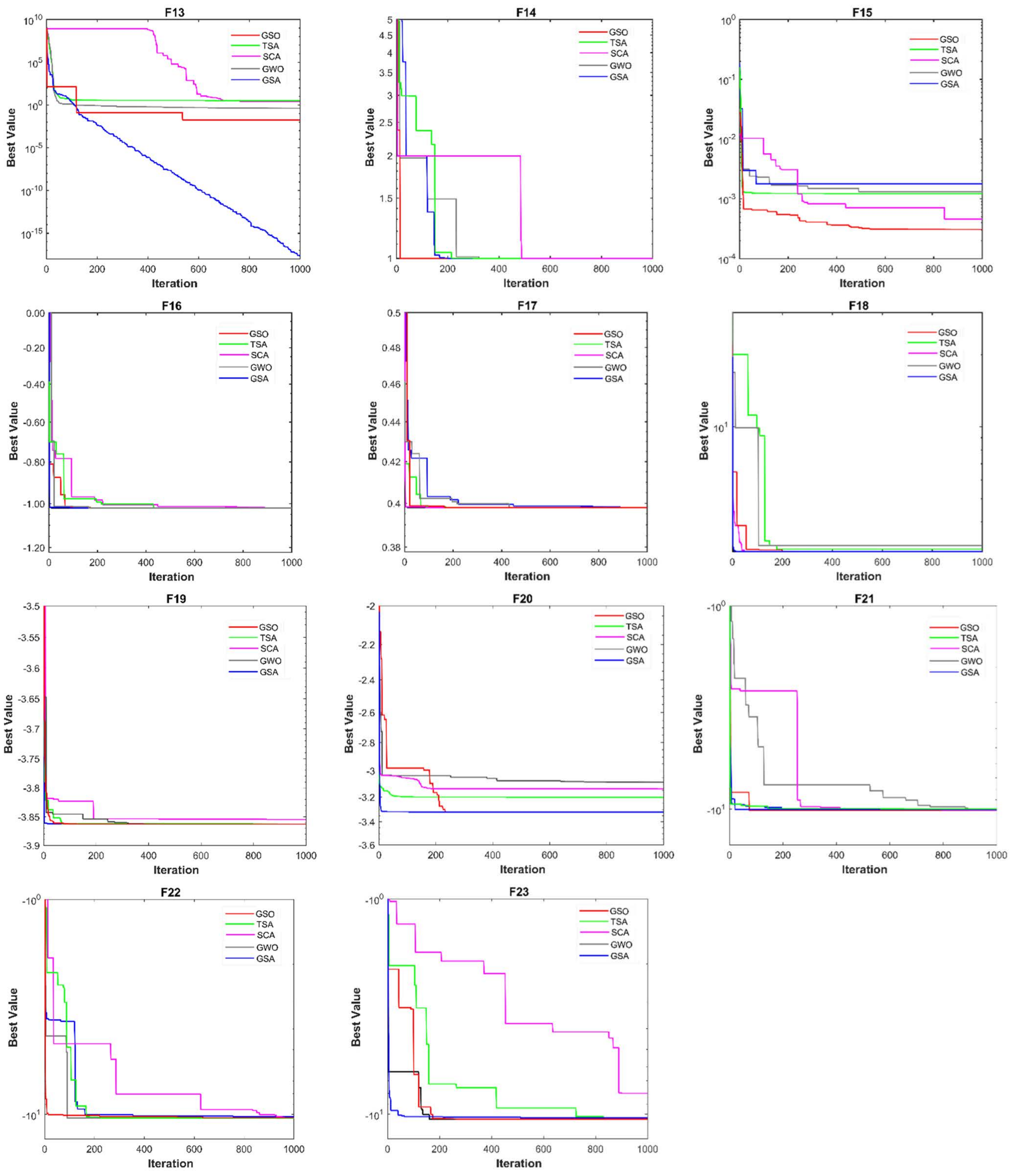

FIGURE 6. ***(Continued.)*** **Comparison of convergence curves of GSO and selected algorithms for *F1-F23*.**

C. CONVERGENCE CAPABILITY

An effective optimization algorithm should converge to a global optimum, not prematurely converge to some local optimum. The convergence progress curves of GSO for some benchmark test functions are compared with GSA, SCA, TSA, and GWO in Fig. 6. The curves are plotted against the number of iterations, which is in the hundreds.

The figure shows that GSO outperforms the other algorithms in most cases. The curves of test functions show that GSO is capable of exploring the search space extensively and identifying the most promising region in less iteration because of its position-updating mechanism and utilization of a random solution instead of worst object during the iterations.

TABLE 8. Comparison of different methods in solving F1- F13 with *Dim*=100.

Function	Statistics	GSO	GSA	SCA	TSA	GWO
F_1	Best	**0.00**	9.1449	109.4939	1.0005e-32	1.9970e-31
	Worst	**0.00**	358.7965	8.7187e+03	2.4209e-28	1.0011e-28
	Mean	**0.00**	99.6956	2.3752e+03	1.4063e-29	1.9615e-29
	Median	**0.00**	76.6121	1.9529e+03	1.4617e-30	9.8708e-30
	Std.	**0.00**	85.6953	1.9898e+03	4.4135e-29	2.4406e-29
F_2	Best	**0.00**	0.0659	0.0164	6.2526e-20	2.2242e-18
	Worst	**0.00**	4.1552	9.1679	4.1657e-18	1.5550e-17
	Mean	**0.00**	1.1819	1.1396	6.7191e-19	5.4811e-18
	Median	**0.00**	1.0639	0.4189	4.7107e-19	5.0501e-18
	Std.	**0.00**	0.8565	1.8451	8.4920e-19	2.8567e-18
F_3	Best	**0.00**	2.8983e+03	6.9139e+04	5.9815	0.0013
	Worst	**0.00**	8.6796e+03	2.2845e+05	923.7997	90.5065
	Mean	**0.00**	4.5394e+03	1.6080e+05	181.9688	11.7494
	Median	**0.00**	4.2716e+03	1.6508e+05	99.9221	1.9712
	Std.	**0.00**	1.0803e+03	3.4812e+04	216.9327	23.3218
F_4	Best	**0.00**	8.2264	76.6566	7.4159	5.7100e-05
	Worst	**0.00**	13.4815	88.1797	40.9080	0.0332
	Mean	**0.00**	10.5279	83.4811	20.3113	0.0043
	Median	**0.00**	10.3173	84.1916	18.3850	0.0024
	Std.	**0.00**	1.2062	2.6972	9.0404	0.0064
F_5	Best	**0.0039**	561.4479	6.5155e+05	96.2634	95.5724
	Worst	**97.1730**	3.4908e+03	1.2263e+08	98.6328	98.4996
	Mean	**7.7575**	1.5527e+03	3.4909e+07	98.2623	97.7396
	Median	**0.0039**	1.3119e+03	2.7435e+07	98.4449	98.0088
	Std.	**24.2253**	785.2370	2.8434e+07	0.5146	0.7823
F_6	Best	**0.0023**	153	96.3543	11.2208	7.4877
	Worst	**0.1249**	714	1.1185e+04	15.3391	10.7297
	Mean	**0.0203**	427.8000	3.1954e+03	13.2997	9.1784
	Median	**0.0023**	403.5000	2.3750e+03	13.4577	9.1352
	Std.	**0.0395**	148.2227	2.6730e+03	1.3311	0.7363
F_7	Best	**1.0856e-06**	0.2445	4.4690	0.0034	9.7286e-04
	Worst	**4.7358e-05**	0.7975	129.7873	0.0179	0.0059
	Mean	**7.5899e-06**	0.5315	40.7898	0.0094	0.0028
	Median	**3.8095e-06**	0.5244	28.3883	0.0094	0.0026
	Std.	**9.2543e-06**	0.1540	34.8030	0.0040	0.0012
F_8	Best	**-4.1142e+04**	-7.9232e+03	-8.3003e+03	-1.7994e+04	-1.9780e+04
	Worst	**-3.6383e+04**	-3.8808e+03	-6.7026e+03	-1.3409e+04	-6.7310e+03
	Mean	**-4.0516e+04**	-5.3879e+03	-7.5852e+03	-1.4830e+04	-1.5971e+04
	Median	**-4.1142e+04**	-5.2638e+03	-7.5823e+03	-1.4646e+04	-1.6186e+04
	Std.	**1.6248e+03**	904.2650	443.0405	1.0864e+03	2.3405e+03
F_9	Best	**0.00**	61.6874	15.1946	677.5464	2.2737e-13
	Worst	**0.00**	101.5179	532.6513	1.0857e+03	3.8250
	Mean	**0.00**	78.7815	194.0863	877.3163	0.3228
	Median	**0.00**	77.7492	187.3070	842.3914	4.5475e-13
	Std.	**0.00**	10.6616	88.5358	119.6674	1.0117
F_{10}	Best	**8.8818e-16**	0.0295	7.0593	3.2863e-14	9.3259e-14
	Worst	**8.8818e-16**	1.8382	20.5997	3.2382	1.3589e-13
	Mean	**8.8818e-16**	1.1835	20.0866	0.1899	1.1031e-13
	Median	**8.8818e-16**	1.1248	20.5310	3.9968e-14	1.1102e-13
	Std.	**0**	0.4295	2.4607	0.7300	9.2078e-15
F_{11}	Best	**0.00**	39.7079	3.3963	0	0
	Worst	**0.00**	81.9494	90.5554	0.0220	0.0185
	Mean	**0.00**	57.8396	29.5224	0.0049	0.0028
	Median	**0.00**	56.4497	22.8226	0	0
	Std.	**0.00**	9.5189	23.7075	0.0071	0.0058
F_{12}	Best	**5.1220e-06**	0.7711	1.1566e+07	2.0521	0.1778
	Worst	**0.0025**	2.8600	3.1018e+08	19.3780	0.4243
	Mean	**2.0671e-04**	1.9197	1.0006e+08	10.5160	0.2692
	Median	**5.1220e-06**	2.0369	8.5138e+07	10.5861	0.2510
	Std.	**5.5042e-04**	0.5249	7.3013e+07	4.7096	0.0747
F_{13}	Best	**4.4115e-07**	41.1755	3.8238e+07	9.9353	5.5985
	Worst	**0.3930**	95.8439	4.0425e+08	13.2658	7.0013
	Mean	**0.0152**	67.4076	2.0184e+08	11.5413	6.2006
	Median	**9.5286e-05**	64.4100	1.8720e+08	11.5733	6.1879
	Std.	**0.0721**	15.4201	1.0263e+08	0.9947	0.3481

TABLE 9. Results of Wilcoxon's rank sum test.

Function	Wilcoxon test Parameters	GSO vs GSA	GSO vs SCA	GSO vs TSA	GSO vs GWO
F1	*p-* value	1.7344E-06	1.7344E-06	1.7344E-06	1.7344E-06
	R+	465	465	465	465
	R-	0	0	0	0
	Winner	GSO	GSO	GSO	GSO
F2	*p-* value	1.7344E-06	1.7344E-06	1.7344E-06	1.7344E-06
	R+	465	465	465	465
	R-	0	0	0	0
	Winner	GSO	GSO	GSO	GSO
F3	*p-* value	1.7344E-06	1.7344E-06	1.7344E-06	1.7344E-06
	R+	465	465	465	465
	R-	0	0	0	0
	Winner	GSO	GSO	GSO	GSO
F4	*p-* value	1.7344E-06	1.7344E-06	1.7344E-06	1.7344E-06
	R+	465	465	465	465
	R-	0	0	0	0
	Winner	GSO	GSO	GSO	GSO
F5	*p-* value	1.7344E-06	1.7344E-06	1.7344E-06	1.7344E-06
	R+	465	465	465	465
	R-	0	0	0	0
	Winner	GSO	GSO	GSO	GSO
F6	*p-* value	7.4523E-7	1.7344E-06	1.7344E-06	2.353E-06
	R+	0	465	465	462
	R-	465	0	0	3
	Winner	GSA	GSO	GSO	GSO
F7	*p-* value	1.7344E-06	1.7344E-06	1.7344E-06	1.7344E-06
	R+	465	465	465	465
	R-	0	0	0	0
	Winner	GSO	GSO	GSO	GSO
F8	*p-* value	1.7344E-06	1.7344E-06	1.7344E-06	1.7344E-06
	R+	465	465	465	465
	R-	0	0	0	0
	Winner	GSO	GSO	GSO	GSO
F9	*p-* value	1.7344E-06	1.7344E-06	1.7344E-06	0.02
	R+	465	465	465	21
	R-	0	0	0	0
	Winner	GSO	GSO	GSO	GSO
F10	*p-* value	1.73E-06	1.73E-06	1.73E-06	1.73E-06
	R+	465	465	465	465
	R-	0	0	0	0
	Winner	GSO	GSO	GSO	GSO
F11	*p-* value	1.73E-06	1.73E-06	1.473E-03	0.068
	R+	465	465	91	10
	R-	0	0	0	0
	Winner	GSO	GSO	GSO	N.A
F12	*p-* value	0.041	1.73E-06	1.73E-06	1.73E-06
	R+	140	465	465	465
	R-	325	0	0	0
	Winner	GSA	GSO	GSO	GSO
F13	*p-* value	3.74E-04	1.73E-06	1.73E-06	1.73E-06
	R+	66	465	465	465
	R-	399	0	0	0
	Winner	GSA	GSO	GSO	GSO
F14	*p-* value	1.73E-06	2.56E-06	4.81E-06	3.22E-04
	R+	465	435	403	152
	R-	0	0	3	1
	Winner	GSO	GSO	GSO	GSO
F15	*p-* value	1.73E-06	2.35E-06	0.006	0.393
	R+	465	462	366	274
	R-	0	3	99	191
	Winner	GSO	GSO	GSO	N.A
F16	*p-* value	0.059	0.371	0.132	1.59E-06
	R+	304	276	50	0
	R-	161	189	415	465
	Winner	N.A	N.A	N.A	GWO
F17	*p-* value	1.37E-06	3.18E-06	1.73E-06	1.73E-06
	R+	465	465	465	465
	R-	0	0	0	0
	Winner	GSO	GSO	GSO	GSO

TABLE 9. *(Continued.)* **Results of Wilcoxon's rank sum test.**

F18	*p-* value	1.08E-06	1.73E-06	3.11E-05	1.73E-06
	R+	0	0	30	0
	R-	465	465	435	465
	Winner	GSA	SCA	TSA	GWO
F19	*p-* value	8.38E-08	1.73E-06	1.73E-06	1.73E-06
	R+	465	465	465	465
	R-	0	0	0	0
	Winner	GSO	GSO	GSO	GSO
F20	*p-* value	5.029E-07	0.894	1.73E-06	1.92E-06
	R+	0	239	0	1
	R-	465	226	465	464
	Winner	GSA	N.A	TSA	GWO
F21	*p-* value	0.046	1.73E-06	1.73E-06	1.73E-06
	R+	329	465	465	465
	R-	136	0	0	0
	Winner	GSO	GSO	GSO	GSO
F22	*p-* value	1.44E-07	1.73E-06	1.73E-06	2.13E-06
	R+	465	465	465	463
	R-	0	0	0	2
	Winner	GSO	GSO	GSO	GSO
F23	*p-* value	1.46E-05	1.73E-06	1.73E-06	3.51E-6
	R+	435	465	465	458
	R-	30	0	0	7
	Winner	GSO	GSO	GSO	GSO
Total	Superior /Inferior/N.A	17/5/1	20/1/2	20/2/1	19/2/2

D. IMPACT OF HIGH-DIMENSIONALITY

In order to present the ability of GSO to solve very high-dimensional problems effectively, the dimensions of all the scalable benchmark functions (F1-F13) have been increased to 100. The algorithms' parameters are the same as in Table 4. The results are compared with those obtained by the other methods in Table 8. The results show that all 100-dim functions become more difficult than their 30-dim counterparts.

As it can be seen from Table 8, the GSO outperforms the competitors for all high-dimensional functions, which indicates the scalability and efficiency of the proposed method in terms of the number of variables in the optimization problem.

E. STATISTICAL SIGNIFICANCE ANALYSIS

A non-parametric pairwise statistical analysis should be conducted in order to determine the statistical significance of the comparative results between two or more algorithms. As recommended by Derrac *et al.* [59], to assess meaningful comparison between the proposed and alternative methods, the nonparametric Wilcoxon's rank sum test is performed between the results. In this regard, utilizing the best results obtained from 30 runs of each method, a pair-wise comparison was conducted.

Wilcoxon's rank sum test returns p-value, the sum of positive ranks ($R+$) and the sum of negative ranks ($R-$) [60]. Table 9 presents the results of Wilcoxon's rank sum test of GSO when compared with other methods. The p-value indicates the minimum significance level for detecting differences. In this study, $\alpha = 0.05$ is considered the level of significance If the p-value of the given algorithm is greater than 0.05, then there is no significant difference between the two compared methods. Such a result is indicated with "N.A" in the winner rows of Table 9. On the other hand, if the p-value is less than α, it definitively means that, in each pair-wise comparison, the better result obtained by the best algorithm is statistically significant and it was not gained by chance. In such cases, if the R+ is greater than the R-, GSO performs better than the alternative method; otherwise, GSO performs poorly and the alternative algorithm performs better [61].

According to the results of Wilcoxon's rank sum test in Table 9, the pairwise comparison between GSO and GSA reveals that in the optimization of 23 test functions, the new method has superior performance in 17 cases and has inferior performance in 5 cases. In addition, for F16, both methods are statistically equivalent. Similarly, in the other pairwise comparison, for the majority of the test suite, GSO provides better results. Therefore, the nonparametric statistical analysis proves that GSO generates significantly better solutions and, comparatively, has superior performance over the other algorithms.

V. CONCLUSION

This study develops a novel population-based Golden Search Optimizer based on some principles of metaheuristic algorithms. In the proposed GSO algorithm, during the search process, the objects (candidate solutions) interact with each other and improve their positions based on the best position obtained so far as the reference point. In summary, the main features of GSO are as follows: it has just two internal parameters; it is easy to code; and it is easy to apply. The performance of the new method is tested by utilizing several experiments. First, a set of various unimodal and multi-modal benchmark functions have been considered to investigate the exploration, exploitation, and convergence rate of the proposed algorithm. Moreover, the results were compared with four well-known and recently developed algorithms, including GSA, SCA, TSA, and GWO. As per the results and findings, it was observed and may be concluded that GSO is capable of finding the global solution for most of the unimodal, multi-modal, and composite benchmark functions and outperforming the other algorithms in a statistically significant manner. While, for most of the benchmark test functions, all the competitor methods rarely reach the global optimal solutions. In order to evaluate the capability of the new method for solving high-dimensional problems, 13 scalable benchmark functions are optimized. Optimization results of scalable test functions proved that the performance of GSO remains consistent even if the dimensions of the problems are increased to 100. Finally, to provide a meaningful comparison and valid judgment between the proposed and alternative algorithms, the nonparametric Wilcoxon Rank Sum test has been conducted. According to the statistical experiments, it is evident that GSO outperformed well-established optimization methods, and its superiority is statistically significant.

REFERENCES

[1] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[2] D. de Werra and A. Hertz, "Tabu search techniques," *Oper.-Res.-Spektrum*, vol. 11, no. 3, pp. 131–141, 1989.

[3] H. R. Lourenço, O. C. Martin, and T. Stützle, "Iterated local search: Framework and applications," in *Handbook of Metaheuristics*. Boston, MA, USA: Springer, 2019.

[4] B. Doǧan and T. Ölmez, "A new metaheuristic for numerical function optimization: Vortex search algorithm," *Inf. Sci.*, vol. 293, pp. 125–145, Feb. 2015.

[5] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.

[6] M. Dorigo and G. Di Caro, "Ant colony optimization: A new meta-heuristic," in *Proc. Congr. Evol. Comput. (CEC)*, 1999, pp. 1470–1477.

[7] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, Nov./Dec. 1995, pp. 1942–1948.

[8] X-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *Int. J. Bio-Inspired Comput.*, vol. 2, no. 2, pp. 78–84, 2010.

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

[10] M. S. Kiran, "TSA: Tree-seed algorithm for continuous optimization," *Expert Syst. Appl.*, vol. 42, no. 19, pp. 6686–6698, 2015.

[11] D. H. Wolper and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[12] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016.

[13] E.-G. Talbi, *Metaheuristics: From Design to Implementation*. Hoboken, NJ, USA: Wiley, 2009.

[14] R. Eberhart and J. Kennedy, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, Nov./Dec. 1995, pp. 1942–1948.

[15] X.-S. Yang and X. He, "Firefly algorithm: Recent advances and applications," *Int. J. Swarm Intell.*, vol. 1, no. 1, pp. 36–50, 2013.

[16] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006.

[17] D. Karaboga and B. Basturk, "Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems," *Int. Fuzzy Syst. Assoc. World Congr.*, 2007, pp. 789–798.

[18] A. H. Gandomi and A. H. Alavi, "Krill herd: A new bio-inspired optimization algorithm," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 17, pp. 4831–4845, May 2012.

[19] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm," *Comput. Struct.*, vol. 169, pp. 1–12, Jun. 2016.

[20] G. Dhiman, M. Garg, A. Nagar, V. Kumar, and M. Dehghani, "A novel algorithm for global optimization: Rat swarm optimizer," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 8 pp. 1–26, 2020.

[21] H. A. Shehadeh, I. Ahmedy, and M. Y. I. Idris, "Sperm swarm optimization algorithm for optimizing wireless sensor network challenges," in *Proc. 6th Int. Conf. Commun. Broadband Netw. (ICCBN)*, 2018, pp. 53–59.

[22] M. S. Braik, "Chameleon swarm algorithm: A bio-inspired optimizer for solving engineering design problems," *Expert Syst. Appl.*, vol. 174, Jul. 2021, Art. no. 114685.

[23] M. Abd Elaziz, S. Lu, and S. He, "A multi-leader whale optimization algorithm for global optimization and image segmentation," *Expert Syst. Appl.*, vol. 175, Aug. 2021, Art. no. 114841.

[24] R. Storn, "On the usage of differential evolution for function optimization," in *Proc. North Amer. Fuzzy Inf. Process.*, 1996, pp. 519–523.

[25] M. Jaderyan and H. Khotanlou, "Virulence optimization algorithm," *Appl. Soft Comput.*, vol. 43, pp. 596–618, Jun. 2016.

[26] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering," *Inf. Sci.*, vol. 222, pp. 175–184, Feb. 2013.

[27] F. F. Moghaddam, R. F. Moghaddam, and M. Cheriet, "Curved space optimization: A random search based on general relativity theory," 2012, *arXiv:1208.2214*.

[28] A. Kaveh and M. Khayatazad, "A new meta-heuristic method: Ray optimization," *Comput. Struct.*, vol. 112, pp. 283–294, Dec. 2012.

[29] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *J. Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, 2009.

[30] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: Harmony search," *J. Simul.*, vol. 76, no. 2, pp. 60–68, Feb. 2001.

[31] R. V. Rao, *Teaching-Learning-Based Optimization Algorithm*, vol. 45. Cham, Switzerland: Springer, 2016, pp. 9–39.

[32] E. Atashpaz-Gargari and C. Lucas, "Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2007, pp. 4661–4667.

[33] N. Ghorbani and E. Babaei, "Exchange market algorithm," *Appl. Soft Comput.*, vol. 19, pp. 177–187, Jun. 2014.

[34] A. Kaveh and A. Dadras, "A novel meta-heuristic optimization algorithm: Thermal exchange optimization," *Adv. Eng. Softw.*, vol. 110, pp. 69–84, Aug. 2017.

[35] F. Glover and M. Laguna, "Tabu search," in *Handbook of Combinatorial Optimization*. Boston, MA, USA: Springer, 1998, pp. 2093–2229.

[36] M. Eslami, H. Shareef, A. Mohamed, and M. Khajehzadeh, "Optimal location of PSS using improved PSO with chaotic sequence," in *Proc. Int. Conf. Electr., Control Comput. Eng. (InECCE)*, Jun. 2011, pp. 253–258.

[37] M. Eslami, H. Shareef, and A. Mohamed, "Optimization and coordination of damping controls for optimal oscillations damping in multi-machine power system," *Int. Rev. Electr. Eng.*, vol. 6, no. 4, pp. 1984–1993, 2011.

[38] Y. M. Cheng, L. Li, T. Lansivaara, S. C. Chi, and Y. J. Sun, "An improved harmony search minimization algorithm using different slip surface generation methods for slope stability analysis," *Eng. Optim.*, vol. 40, no. 2, pp. 95–115, 2008.

[39] M. Khajehzadeh, M. R. Taha, and M. Eslami, "Multi-objective optimisation of retaining walls using hybrid adaptive gravitational search algorithm," *Civil Eng. Environ. Syst.*, vol. 31, no. 3, pp. 229–242, Jul. 2014.

[40] M. Khajehzadeh, M. R. Taha, A. El-Shafie, and M. Eslami, "Search for critical failure surface in slope stability analysis by gravitational search algorithm," *Int. J. Phys. Sci.*, vol. 6, no. 21, pp. 5012–5021, 2011.

[41] W. Gao, "Modified ant colony optimization with improved tour construction and pheromone updating strategies for traveling salesman problem," *Soft Comput.*, vol. 25, no. 4, pp. 3263–3289, Feb. 2021.

[42] Y. Ji, J. Tu, H. Zhou, W. Gui, G. Liang, H. Chen, and M. Wang, "An adaptive chaotic sine cosine algorithm for constrained and unconstrained optimization," *Complexity*, vol. 2020, pp. 1–36, Oct. 2020.

[43] M. Khajehzadeh, M. R. Taha, S. Keawsawasvong, H. Mirzaei, and M. Jebeli, "An effective artificial intelligence approach for slope stability evaluation," *IEEE Access*, vol. 10, pp. 5660–5671, 2022.

[44] M. H. Ali, S. Kamel, M. H. Hassan, M. Tostado-Véliz, and H. M. Zawbaa, "An improved wild horse optimization algorithm for reliability based optimal DG planning of radial distribution networks," *Energy Rep.*, vol. 8, pp. 582–604, Nov. 2022.

[45] M. Farhat, S. Kamel, A. M. Atallah, M. H. Hassan, and A. M. Agwa, "ESMA-OPF: Enhanced slime mould algorithm for solving optimal power flow problem," *Sustainability*, vol. 14, no. 4, p. 2305, Feb. 2022.

[46] I. Cherki, A. Chaker, Z. Djidar, N. Khalfallah, and F. Benzergua, "A sequential hybridization of genetic algorithm and particle swarm optimization for the optimal reactive power flow," *Sustainability*, vol. 11, no. 14, p. 3862, Jul. 2019.

[47] M. Khajehzadeh, M. R. Taha, and M. Eslami, "A new hybrid firefly algorithm for foundation optimization," *Nat. Acad. Sci. Lett.*, vol. 36, no. 3, pp. 279–288, Jun. 2013.

[48] M. Khajehzadeh, M. R. Taha, and M. Eslami, "Multi-objective optimization of foundation using global-local gravitational search algorithm," *Struct. Eng. Mech.*, vol. 50, no. 3, pp. 257–273, May 2014.

[49] M. Khajehzadeh, S. Keawsawasvong, P. Sarir, and D. K. Khailany, "Seismic analysis of earth slope using a novel sequential hybrid optimization algorithm," *Periodica Polytechnica Civil Eng.*, vol. 66, no. 2, pp. 355–366, 2022.

[50] H. Abdel-Mawgoud, A. Fathy, and S. Kamel, "An effective hybrid approach based on arithmetic optimization algorithm and sine cosine algorithm for integrating battery energy storage system into distribution networks," *J. Energy Storage*, vol. 49, May 2022, Art. no. 104154.

[51] I. Naruei, F. Keynia, and A. S. Molahosseini, "Hunter–Prey optimization: Algorithm and applications," *Soft Comput.*, vol. 26, no. 3, pp. 1279–1314, Feb. 2022.

[52] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "Filter modeling using gravitational search algorithm," *Eng. Appl. Artif. Intell.*, vol. 24, no. 1, pp. 117–122, Feb. 2011.

[53] L. Abualigah and A. Diabat, "Advances in sine cosine algorithm: A comprehensive survey," *Artif. Intell. Rev.*, vol. 54, no. 4, pp. 1–42, 2021.

[54] 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.

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

[56] J. G. Digalakis and K. G. Margaritis, "On benchmarking functions for genetic algorithms," *Int. J. Comput. Math.*, vol. 77, no. 4, pp. 481–506, Sep. 2001.

[57] K. Hussain, M. N. M. Salleh, S. Cheng, and R. Naseem, "Common benchmark functions for metaheuristic evaluation: A review," *Int. J. Informat. Vis.*, vol. 1, nos. 2–4, pp. 218–223, 2017.

[58] Q. Askari, I. Younas, and M. Saeed, "Political optimizer: A novel socio-inspired meta-heuristic for global optimization," *Knowl.-Based Syst.*, vol. 195, May 2020, Art. no. 105709.

[59] 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.

[60] *SPSS-Tutorials Version 23*, New York, NY, USA, 2016.

[61] M. Toz, "Chaos-based vortex search algorithm for solving inverse kinematics problem of serial robot manipulators with offset wrist," *Appl. Soft Comput.*, vol. 89, Apr. 2020, Art. no. 106074.

MOHAMMAD NOROOZI received the Master of Science degree in industrial engineering from the Amirkabir University of Technology, Iran, in 2016. He is currently pursuing the Ph.D. degree in the field of industrial engineering with the University of South Florida. His research interests include machine learning, optimization methods, deep learning, computer vision, and AI.

HAMED MOHAMMADI received the Master of Science degree in industrial engineering from the Amirkabir University of Technology, Iran, in 2016. He is currently pursuing the Ph.D. degree in the field of industrial engineering with the University of Central Florida. His research interests include machine learning, deep learning, supply chain management, business intelligence, health systems, and human factor.

EMAD EFATINASAB received the B.S. degree from the University of Birjand, Iran, in 2021. He is currently pursuing the master's degree with the Department of Computer Science, University of Padova, Italy. His research interests include artificial intelligence, optimization, machine learning, business intelligence, and computer vision.

ALI LASHGARI received the M.Sc. degree in economics from Kansas State University, USA, and the M.Sc. degree in industrial engineering from the Amirkabir University of Technology. He is currently pursuing the Ph.D. degree in economics with Kansas State University. His research interests include financial econometrics, machine learning, and applied statistics.

MAHDIYEH ESLAMI (Member, IEEE) received the B.Sc. degree from the Shahid Bahonar University of Kerman, Iran, in 2000, the M.Sc. degree from the Iran University of Science and Technology (IUST), Tehran, Iran, in 2004, and the Ph.D. degree from the National University of Malaysia (UKM), in 2012, all in electrical engineering. From 2012 to 2013, she was a Postdoctoral Research Fellow with the Power Research Group, UKM. Since her Ph.D. degree, she has published more than 80 articles in various fields related to power system. She is currently an Assistant Professor with the Department of Electrical Engineering, Islamic Azad University, Kerman Branch. Her research interests include power system stability, application of AI techniques in power systems, power system dynamics, and FACTS devices.

BASEEM KHAN (Senior Member, IEEE) received the B.Eng. degree in electrical engineering from Rajiv Gandhi Technological University, Bhopal, India, in 2008, and the M.Tech. and D.Phil. degrees in electrical engineering from the Maulana Azad National Institute of Technology, Bhopal, in 2010 and 2014, respectively. He is currently working as a Faculty Member of Hawassa University, Ethiopia. He has published more than 100 research papers in well reputable research journals, including IEEE Transactions, IEEE Access, *Computer and Electrical Engineering* (Elsevier), *IET GTD*, *IET PRG*, and *IET Power Electronics*. Furthermore, he has published, authored and edited books with Wiley, CRC Press, and Elsevier. His research interests include power system restructuring, power system planning, smart grid technologies, meta-heuristic optimization techniques, reliability analysis of renewable energy systems, power quality analysis, and renewable energy integration.