IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 3, 2020, accepted November 18, 2020, date of publication November 24, 2020, date of current version December 8, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.3039882*

# Steady-State Sine Cosine Genetic Algorithm Based Chaotic Search for Nonlinear Programming and Engineering Applications

**A. A. MOUSA[1,3], M. A. EL-SHORBAGY[2,3], AND M. A. FARAG[3]**

[1]Department of Mathematics and Statistics, College of Science, Taif University, Taif 21944, Saudi Arabia
[2]Department of Mathematics, College of Science and Humanities in Al-Kharj, Prince Sattam bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia
[3]Department of Basic Engineering Science, Faculty of Engineering, Menoufia University, Shebin El-Kom 32511, Egypt

Corresponding author: M. A. El-Shorbagy (mohammed_shorbagy@yahoo.com)

This work was supported by Taif University Researchers Supporting Project Number (TURSP-2020/48), Taif University, Taif, Saudi Arabia.

**ABSTRACT** This paper proposes a newly meta-heuristic approach, steady-state sine cosine genetic algorithm-based chaotic search, for solving nonlinear programming and engineering applications. It is a combination of sine cosine approach (SCA), steady-state genetic algorithm (SSGA), and chaotic search (CS), and named as chaos-enhanced SCA with SSGA. The proposed approach integrates SSGA's exploitation ability and SCA's exploration ability and local search capability of CS. The performance of the new approach works in two different stages. Firstly, SCA and SSGA start together to increase exploration capability and exploitation tendencies. Secondly, CS used to improve the approximate solution obtained from the first stage and reach the global solution. Hence, the proposed new approach will be more robust as it avoids trapping into local minima in addition to the speed of the search process and rapid convergence towards the global solution. The efficiency of the proposed approach is verified by using it to solve 32 well-known benchmark problems and different engineering design problems. Simulation results show that the proposed approach is competitive and better in most cases as a comparison to others.

**INDEX TERMS** Chaos search, hybrid approach, nonlinear programming problems, engineering applications, optimization.

## I. INTRODUCTION

The optimization problem is deemed as the procedure of obtaining the best element from a suite of available choices to find the minimum/maximum value of an objective function while satisfying some constraints. One of the important optimization problems is the nonlinear programming problem (NPP) which is stated as follows [1]:

$$\begin{aligned}
&\text{Min } F(z)\\
&\text{Subject to: } z \in S;\\
&\qquad S = \left\{ \begin{array}{l} g_t(z) \le 0, t = 1, 2, \ldots, p \\ h_j(z) = 0, j = 1, 2, \ldots, q, \end{array} \right\}\\
&\qquad z_i^l \le z_i \le z_i^u \quad i = 1, 2, \ldots, dim; \qquad (1)
\end{aligned}$$

where $F(z)$ is the objective function, $z = [z_1, z_2, \ldots, z_{\dim}]^T$ is the decision vector, $g_t(z)$ are the inequality constraint functions, $h_j(z)$ are equality constraint functions and $S$ refers to the feasible search space that has all the possible solutions which satisfy all the constraints. Every decision variable $z_i$ is limited by its upper and lower boundaries $[z_i^l, z_i^u]$.

Recently, optimization problems have emerged in numerous fields such as information theory [2], economics [3], computer science [4], and statistical physics [5]. Therefore, the development of optimization algorithms is an interesting point for researchers.

Optimization methods are techniques that are employed to obtain the optimal solutions of optimization problems in different fields. Generally, the optimization techniques are usually classified into deterministic and stochastic techniques [6]. Deterministic algorithms have an enormous advantage, such as the fact that they can find global optima and that they converge to a solution much faster with higher

The associate editor coordinating the review of this manuscript and approving it for publication was Chao-Yang Chen.

accuracy while fulfilling the local search task compared to stochastic methods. However, they are very CPU-intensive and useless on intractable NP-hard problems, non-convex optimization problems, and complex problems [7]. Therefore, many stochastic optimization techniques/meta-heuristic algorithms have been developed to solve such difficult optimization problems.

Meta-heuristic algorithms tackle optimization problems as black boxes. This means that the derivation of the mathematical models is unneeded. In other words, it is trying to improve its inputs for maximizing or minimizing its outputs only by evaluating the objective function and check the feasibility of constraint functions [8]. The basic concept of most meta-heuristic algorithms is the inspiration from nature, animal behaviors, or physical phenomena. The popular meta-heuristic algorithms that solving optimization problems are genetic algorithm (GA) [9], evolutionary strategy (ES) [10], harmony search algorithm [11], [12], differential evolution (DE) [13], particle swarm optimization (PSO) [14], imperialistic competitive algorithm [15], simulated annealing (SA) [16], whale optimization algorithm (WOA) [17], salp swarm algorithm (SSA) [18], grasshopper optimization algorithm (GOA) [19], firefly algorithm (FA) [20], monkey algorithm (MA) [21], melody search algorithm [22], shuffled frog leaping algorithm (SFLA) [23], artificial bee colony (ABC) [24], social spider optimization (SSO) [25], grey wolf optimization (GWO) [26], ant colony optimization (ACO) [27], SCA [28], algorithms that are inspired by fish schools [29], bats [30], and cuckoo birds [31], etc. The meta-heuristic algorithms solve optimization problems randomly. Also, they are characterized by a gradient-free mechanism and high flexibility which means that they are applicable readily to complex real-life problems in various fields [32].

Although meta-heuristic algorithms perform well compared to deterministic methods, they may face difficulties such as being stuck in a local optimum, insufficient diversity of solutions, and an imbalance between exploitation and exploration trends in some complex problems. To overcome these weaknesses, most of the researchers proposed hybridization strategies between meta-heuristic algorithms to improve the solution quality, benefit from their advantages, and to overcome any deficiencies. In addition, these hybrid strategies have many characteristics as gift robust algorithms with faster performance and treatment of large tasks. For example, Goel and Maini [33] proposed the algorithm of the FA and ACO to be hybrid, HAFA. FA was used to scan for the unexplored space for solutions, while in ACO, the method of shaking pheromones was used to prevent pheromone stagnation in the exploited areas. In [34], Nasr *et al.* presented a hybrid optimization algorithm between one of the intelligence techniques (genetic algorithm) and chaos theory. The chaos theory was introduced as a local search to improve the solution quality and find the optimal solution. In [35], Jadon *et al.* suggested a hybrid ABC algorithm with DE for creating a better convergence algorithm and improved the balance between exploitation and exploration capabilities. While Turanoğlu and Akkaya proposed a new hybrid heuristic algorithm called SA based on bacterial foraging optimization (BFO) algorithm in [36]. In [37] the authors provided a hybrid variation of the classic cat swarm optimization (CSO) algorithm; where it is an easy to use, efficient, and fast algorithm. In addition, in [38], Abualigah *et al.* proposed a novel combination of krill herd (KH) algorithm with harmony search (HS) algorithm, namely, H-KHA, to increase the global search capacity (diversification). The enhancement involves adding a global search operator of the HS algorithm to the KH algorithm for enhancing the exploration search capacity thereby moving krill individuals toward the best global solution. Furthermore, in [39], Shaheen *et al.* proposed a hybrid algorithm between the GWO and PSO method to reach the optimal solution; where GWO was hybridized with the PSO method to improve the progress of the GWO. Finally, In [40] a modern, reliable and efficient hybrid approach, based on the combination of a firefly algorithm (FA) with the adaptive particularly tunable fuzzy PSO (APT-FPSO) is proposed. In the internal configuration of the FA in order to prevent premature convergence of the originating FA, the APT-FPSO uses an enhanced version of the fuzzy-based PSO, by improved exploitation and exploration procedures.

Genetic algorithm (GA) is one of the meta-heuristic algorithms that introduced as a proficient global technique for solving complex optimization problems based on the technicalities of natural selection, evolution, and genetics [9]. GA is well suitable for solving nonlinear optimization problems, where it uses a set of points for the evaluation process in the search domain. GA has still gained an increasing interest in the field of different industrial applications [41]–[43]. However, GA when dealing with complex and large systems has many disadvantages such as extreme slowness and the difficulty in ensuring access to the global optimum solution as it requires an increase in the number of iterations (i.e. long search time) [44], [45]. To avoid these defects, some of the researchers have added a new genetic operator and improved the control parameters and the structure of GA [46], [47]. There are many dissimilar models for GA such as steady-state genetic algorithm (SSGA) and generational genetic algorithm (GGA) [48]. GGA and SSGA both follow the general scheme of GA. The difference between GGA and SSGA is that SSGA produces two new chromosomes or a small number of chromosomes in every iteration; while GGA produces a large number of chromosomes in each iteration and thus the population changes significantly [49]. So, the SSGA is currently a better choice when fitness evaluations and feasibility of the chromosomes are computationally expensive. Due to these features, it is used in the proposed approach.

On the other hand, SCA is a recent meta-heuristic algorithm that was introduced by Mirjalili [28] for solving optimization problems. It is a variant of GWO. SCA is looking for the best solutions via a mathematical model that is based on trigonometric sine and cosine functions. Mirjalili [28] proved that SCA has efficient performance compared to other recent meta-heuristic algorithms. Nevertheless, as with any meta-heuristic algorithm, SCA depends on adaptive and random variables; thus, an acceptable solution cannot always be obtained. In addition, there is no internal memory for SCA that keeping track of the previously found solutions. So, SCA loses all information about the solutions that exceed the global best and it never maintains the possible solutions set which may have a chance to reach the global solution [50]. Accordingly, the SCA exploitability is weak, which sometimes leads to solutions falling into local traps.

While, chaos theory has been implemented in several fields such as economics, sociology, philosophy, physics, engineering, biology, and meteorology. It was initially introduced by Lorenz in 1963 and described as the so-called "butterfly effect" [51]. Chaos is a bounded dynamic unstable behavior that is highly sensitive to its initial conditions; where small changes in the starting values or the parameters tend to affect the immensely various future behaviors. There are so-called chaotic maps that have three main characteristics which are ergodicity, randomness, and sensitivity to the initial condition [52]. Recently, the mathematics of chaos theory has been used to improve various optimization methods. So, most researchers incorporate chaotic maps with the optimization techniques in order to get the global solutions of complex problems [53]–[55] and accelerate the convergence to the optimal solution such as adaptive firefly algorithm with chaos [56], Chaos-enhanced cuckoo search optimization algorithm [57], chaos game optimization [58], chaos grasshopper optimization algorithm [59], chaotic fruit fly optimization algorithm [60] and chaos-enhanced bat algorithm [61].

So, in this paper a newly meta-heuristic hybrid approach, steady-state sine cosine genetic algorithm based chaotic search, for solving nonlinear programming and engineering applications. It is a combination of sine cosine approach (SCA), steady-state genetic algorithm (SSGA), and chaotic search (CS), and named as chaos-enhanced SCA with SSGA. The motivation for proposing this algorithm that it integrates SSGA's exploitation ability and SCA's exploration ability and local search capability of CS. The proposed hybrid approach is tested on a suite of unconstrained and constrained optimization problems [62]–[65] with different degrees of difficulty and problems related to engineering constrained design [66], [67].

The main contributions of this paper are:

1. A new meta-heuristic hybrid approach, a combination between SCA, SSGA, and CS, for solving nonlinear programming and engineering applications is presented and evaluated.
2. Integrating SSGA's exploitation ability, SCA's exploration ability.
3. CS strategy, to enhance the solution quality and access to the optimal solution, is applied.
4. According to the convergence curves and statistical results, comparisons between SCA, SSGA, and the proposed approach demonstrated the relevance of the combination between SCA and SSGA and the proposed approach's ability to reach better solutions than SCA and SSGA.
5. The proposed meta-heuristic hybrid approach is effectively applied for nonlinear programming and as well engineering applications; where good results were achieved, compared to earlier studies.

The remainder of this article is presented as follows: first of all, a brief discussion of the related works about the proposed algorithm is presented in Section 2 which briefly describes SSGA, SCA, and chaos theory. Section 3 includes the explanation about a chaos-based hybrid SCA-SSGA algorithm, while Section 4 includes reporting of as well as analyzing the experimental results. Finally, Section 5 presents the conclusion of this article.

## II. RELATED WORKS

In this section, we give a brief description of SSGA, SCA, and chaos theory.

### A. STEADY-STATE GENETIC ALGORITHM

During the seventies of the twentieth century, J. Holland put forward the GA as an optimization algorithm for obtaining a global or near-global optimal solution [9]. GA mimics natural evolution: genetic inheritance and the Darwinian theory of biological evolution [68]. GA commences with a population of chromosomes (solutions) that are evolved towards better solutions by an iterative process. GA has many varying models for solving any problem such as GGA and SSGA [48]. The SSGA starts with an initial population of a specific size referred to by $pop_{size}$. SSGA obtains an offspring population, with size $new_{pop}$, created via crossover as well as mutation of certain individuals that are selected from the base population. In fact, the offspring that is newly created is assessed and after that, it is combined with the base population. Each chromosome of the new population is repaired and after that, all chromosomes are ranked based on the value of their objective functions. Following the ranking process, $new_{pop}$ worst chromosomes in the ranking are replaced in order to bring the population back to its base size $pop_{size}$. Consequently, after replacement, the new population contains the best chromosomes. Chromosomes number $new_{pop}$, which is to be removed, is decided by the user. After producing the new population, the termination condition gets checked, and if it is met, then the approach stops; otherwise, the evolution proceeds to complete the new population as described previously [69]. The scheme of SSGA is shown in Figure 1.

### B. SINE COSINE ALGORITHM (SCA)

SCA is a meta-heuristic algorithm that is based on the forms of sine and cosine functions and was first put forward in the year 2016 [28]. SCA starts by creating random solutions. These solutions are evolved repeatedly towards better solutions during the optimization process. In addition, the search domain is controlled by modifying the range of the sine and

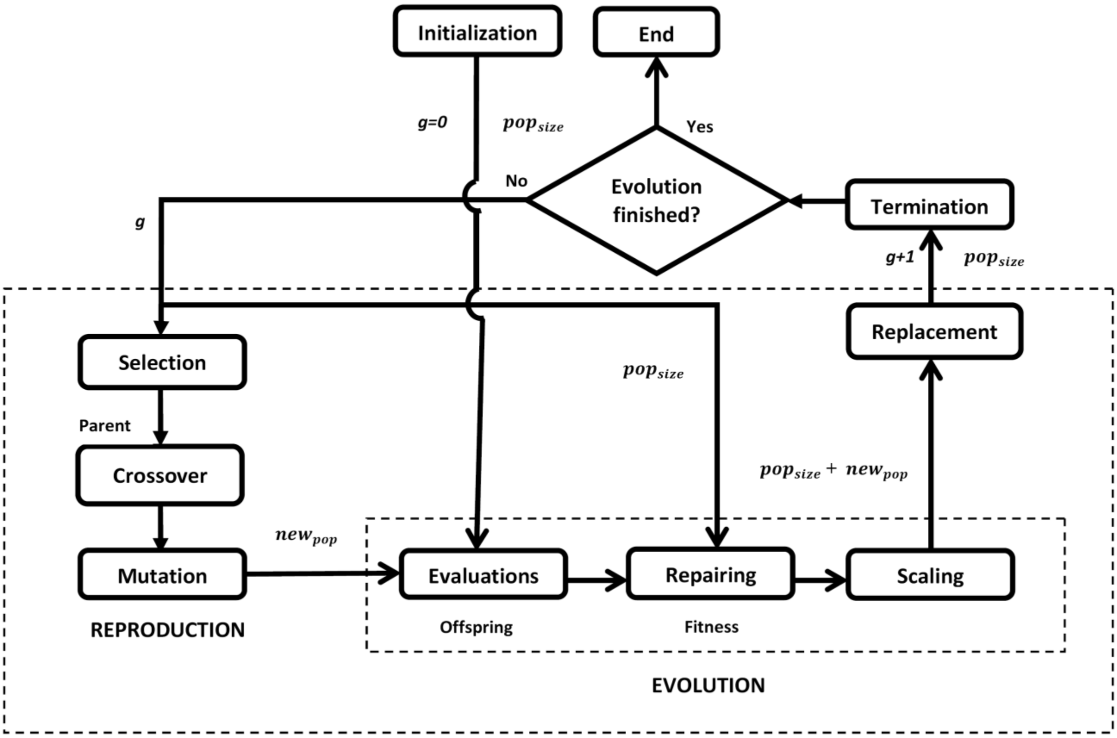


**FIGURE 1.** The scheme of steady-state genetic algorithm.

cosine. These steps are repeated until meeting the termination criterion. SCA's mathematical formulation is:

$$x_i^{t+1} = \begin{cases} x_i^t + r_1 \times \sin(r_2) \times \left| r_3 p_i^t - x_i^t \right|, & r_4 < 0.5 \\ x_i^t + r_1 \times \cos(r_2) \times \left| r_3 p_i^t - x_i^t \right|, & r_4 \geq 0.5; \end{cases} \tag{2}$$

where $x_i^t$ is the position of current solution in $i^{th}$ dimension at $t^{th}$ iteration, $r_1, r_2, r_3, r_4$ are the random numbers, and $p_i$ is the best solution of the destination points in $i^{th}$ dimension. As shown in (2), SCA uses 4 variables ($r_1, r_2, r_3, r_4$) to tune. $r_1$ decides direction of the movement of the search agent which either could be in the area between the destination and solution or could be outside it by the following expression:

$$r_1 = c\left(1 - \frac{t}{T_{\max}}\right); \tag{3}$$

where $c$ is constant, $t$ is the current iteration, and $T_{\max}$ is the iterations maximum number. Meanwhile, $r_2$ is a random number belongs to $[0, 2\pi]$, which shows how far the movement might be towards or outwards the destination. $r_3$ assigns a random weight for the destination by stochastically emphasizing when $r_3 < 1$ but when $r_3 > 1$ deemphasizing the impact of destination in defining the distance. In the end, $r_4 \in [0, 1]$ decided whether sine or cosine formula will be utilized. Figure 2 (taken from [28]) gives a description of how the above equations define space between the two solutions in the search domain, and the pseudo code of SCA [70] is shown in Figure 3.

### C. CHAOS THEORY AND THEIR MAPS

Chaos indicates a random situation in the deterministic system. Such a state is the development of nonlinear systems by deterministic policies and it is a long-term behavior with no fixed period [71]. Chaos theory is concerned with the initial conditions, where chaotic motion can traverse all the states in accordance with itself within a certain range without repeating these states. In this way, the chaotic motion is ergodic. Therefore, in optimization techniques, it is more advantageous to use chaos search compared to disorderly random searches. The chaotic map is an evolution function that shows some kind of chaotic behavior. There are several chaotic maps that have been presented in the literature [72]–[74] such as Sinusoidal map, Tent map, Sine map, Gauss map, Piecewise map, Logistic map. Table 1 presents a summary of the mathematical formulas of various chaotic maps.

## III. CHAOS-ENHANCED SCA WITH SSGA

Chaos-enhanced SCA with SSGA combines the advantages of SSGA, SCA, and chaos strategy so as not to be trapped into local optima, quicken the seeking process and accelerate the convergence to a global solution. The proposed approach, as in Figure 4, functions in two stages: during the first stage, the hybridization between SSGA and SCA is implemented as the global optimization method to obtain the approximate solution of the optimization problem. After that, the chaotic search is applied to speed up the convergence and enhance the solution quality and access to the best solution. The steps

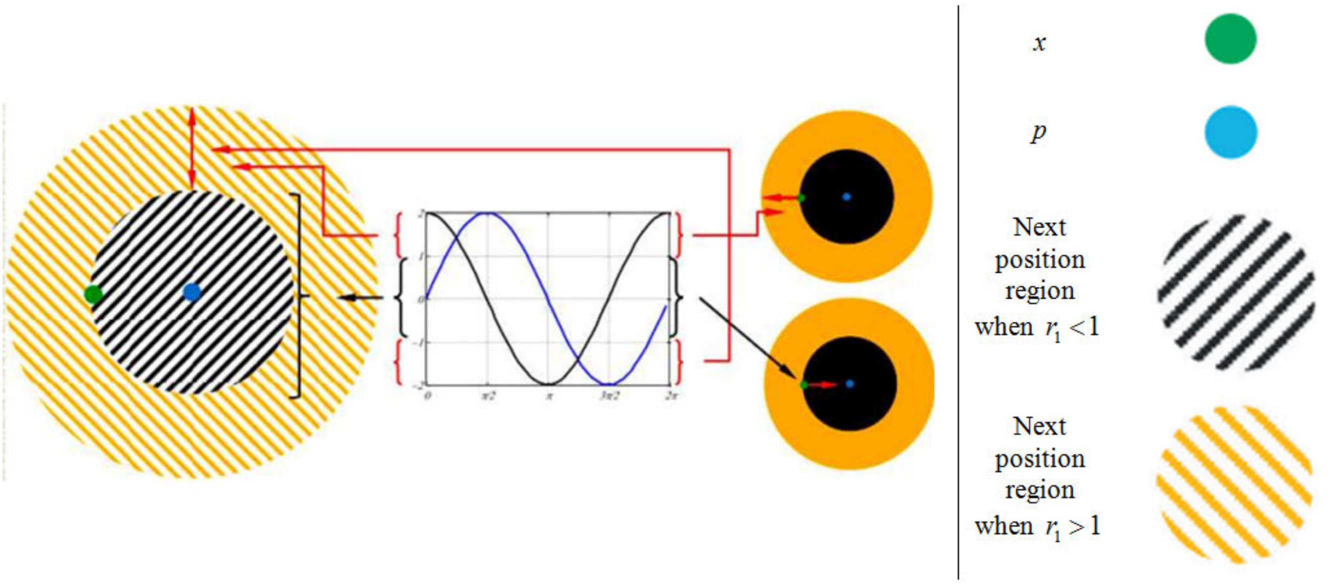


**FIGURE 2.** The effect of Sine and Cosine in Eq. 2 on the next position.

**Begin**
  **Initialize** a set of solution
  **While** $t < T_{\max}$ **do**
    Evaluate each of solutions by its fitness function
    Update the best solution $(p = x^*)$
    Update $r_1, r_2, r_3, r_4$
    Update the position of each solution as in (2)
  **End while**
  **Return** the best Solution
**End**

**FIGURE 3.** The pseudo code of SCA.

of the Chaos-enhanced SCA with SSGA can be illustrated as follows:

**Stage 1: SSGA and SCA**

**Step 1. Initialization.** A population of size $pop_{size}$ is generated.

**Step 2. Evaluation.** The fitness of each individual in this population is evaluated.

**Step 3. Ranking.** On the basis of fitness values, the population is ranked.

**Step 4. SSGA method.** Operators of SSGA (crossover and mutation) are applied to the top 20% of individuals to create another 20% of individuals.

**Step 5. SCA method.** Apply SCA equations for updating the 80% of individuals with the worst fitness.

**Step 6. Termination criterion.** Repeated the steps from 2 to 5 until the termination criterion is reached and obtained the best solution.

**Stage 2: Chaotic search**

Optimizing the objective function by using SSGA and SCA may yield an approximated solution $y^* = \left(y_1^*, y_2^*, \ldots, y_{\text{dim}}^*\right)$. By chaotic search, we can perturb $y^*$ to explore its local region and get the optimal solution as shown below [75]:

**Step 1.** The boundary range $[a_i, b_i] \forall i = 1, 2, \ldots, \dim$ of chaotic search is obtained by $y_i^* - \varepsilon < a_i, y_i^* + \varepsilon > b_i$,

**TABLE 1. The mathematical formulas of various chaotic maps.**

| Map name | Formula |
|---|---|
| Sine Map | $x_{k+1} = \frac{c}{4}\sin(\pi x_k);\ 0 < c \le 4$ |
| Sinusoidal Map | $x_{k+1} = a x_k^2 \sin(\pi x_k);$ for $a = 2.3$ and $x_0 = 0.7$ it has the following simplified form $x_{k+1} = \sin(\pi x_k)$ |
| Singer Map | $x_{k+1} = \mu(-13.302875x_k^4 + 28.75x_k^3 - 23.31x_k^2 + 7.86x_k);\ \mu \in [0.9, 1.08]$ |
| Chebyshev Map | $x_{k+1} = \cos(k \cos^{-1}(x_k))$ |
| Tent Map | $x_{k+1} = \begin{cases} x_k / 0.7 & x_k < 0.7 \\ \frac{10}{3}(1 - x_k) & x_k \ge 0.7 \end{cases}$ |
| Piecewise Map | $x_{k+1} = \begin{cases} x_k / q & 0 < x_k < q \\ (x_k - q)/(0.5 - q) & q \le x_k < 0.5 \\ (1 - q - x_k)/(0.5 - q) & 0.5 \le x_k < 1 - q \\ (1 - x_k)/q & 1 - q < x_k < 1 \end{cases};\quad q \in [0, 0.5]$ |
| Gauss Map | $x_{k+1} = \exp(-\alpha x_k^2) + \beta;$ $\alpha$ and $\beta$ are real parameters |
| Logistic Map | $x_{k+1} = a x_k (1 - x_k);$ *a is parameter* |
| Liebovitch Map | $x_{k+1} = \begin{cases} \alpha x_k & 0 < x_k < v_1 \\ \frac{v_2 - x_k}{v_2 - v_1} & v_1 < x_k < v_2 \\ 1 - \beta(1 - x_k) & v_2 < x_k < 1 \end{cases}$ *where* $\alpha = \frac{v_2}{v_1}(1 - (v_2 - v_1))$, and $\beta = \frac{1}{v_2 - 1}((v_2 - 1) - v_1(v_2 - v_1))$ |
| Intermittency Map | $x_{k+1} = \begin{cases} \varepsilon + x_k + a x_k^m & 0 < x_k < v \\ \frac{x_k - v}{1 - v} & v < x_k < 1 \end{cases}$ *where* $a = \frac{1 - \varepsilon - v}{v^m}$ and $\varepsilon \ll v$ |

where $\varepsilon$ is a specified radius of the chaotic search region.

**Step 2.** A chaotic number $z^k \forall k = 1, \ldots, k_{\max}$ is obtained by various chaotic maps.

**Step 3.** For all $z^k \forall k = 1, \ldots, k_{\max}$ the chaotic number $z^k$ is mapped into the boundary range $[a_i, b_i]$ by [76]:

$$y_i^k = a_i + z^k (b_i - a_i) \tag{4}$$

which leads to the following:

$$y_i^k = y_i - \varepsilon + 2\varepsilon z^k \quad \forall i = 1, 2, \ldots.\dim. \tag{5}$$

**Step 4.** If $f(y^k) < f(y^*)$, the best solution $y^*$ is updated by setting $y^* = y^k$.

**Step 5.** If $f(y^*)$ is not enhanced for all iterations ($k_{\max}$), the chaotic search process is stopped and $y^*$ is put out as the optimal solution.

## IV. RESULTS OF NUMERICAL EXPERIMENTS

This section illustrates the results about the performance of the chaos-enhanced SCA with SSGA, where the efficiency of the proposed approach is examined by 25 unconstrained benchmark functions of 10 dimensions provided by Suganthan *et al.* [62] during the CEC 2005 special session on real parameter optimization. Such benchmark functions consist of 5 unimodal functions and 20 multimodal functions. Features of these functions are listed in [62]. In addition, the proposed algorithm solved 7 constrained benchmarks problems (available in Appendix A) taken from the literature [63]–[65]. Moreover, six engineering design problems (available in Appendix B)—common in the field of mechanical engineering, chemical engineering, and electrical engineering [50], [51]—are solved to verify the reliability and validity of the proposed approach for engineering applications. Mechanical applications are the optimum design of the speed reducer, the pressure vessel, the disk brake, and the three-bar truss. Meanwhile, the problem of the cost minimization of a transformer design is chosen to be solved as the electrical application. Finally, the problem of the optimal design of chemical reactors is selected to be solved as an application of chemical engineering. The mathematical formulations of all constrained test functions and different engineering applications are available in the Appendix.

The proposed method is encrypted on the Intel Core machine i5, 1.80 GHz, and 4 GB of RAM in the MATLAB (R2016b). As with any meta-heuristic algorithm, the proposed hybrid algorithm contains a set of parameters that affect the algorithm's performance. Table 2 defines the parameters used to execute the proposed method.

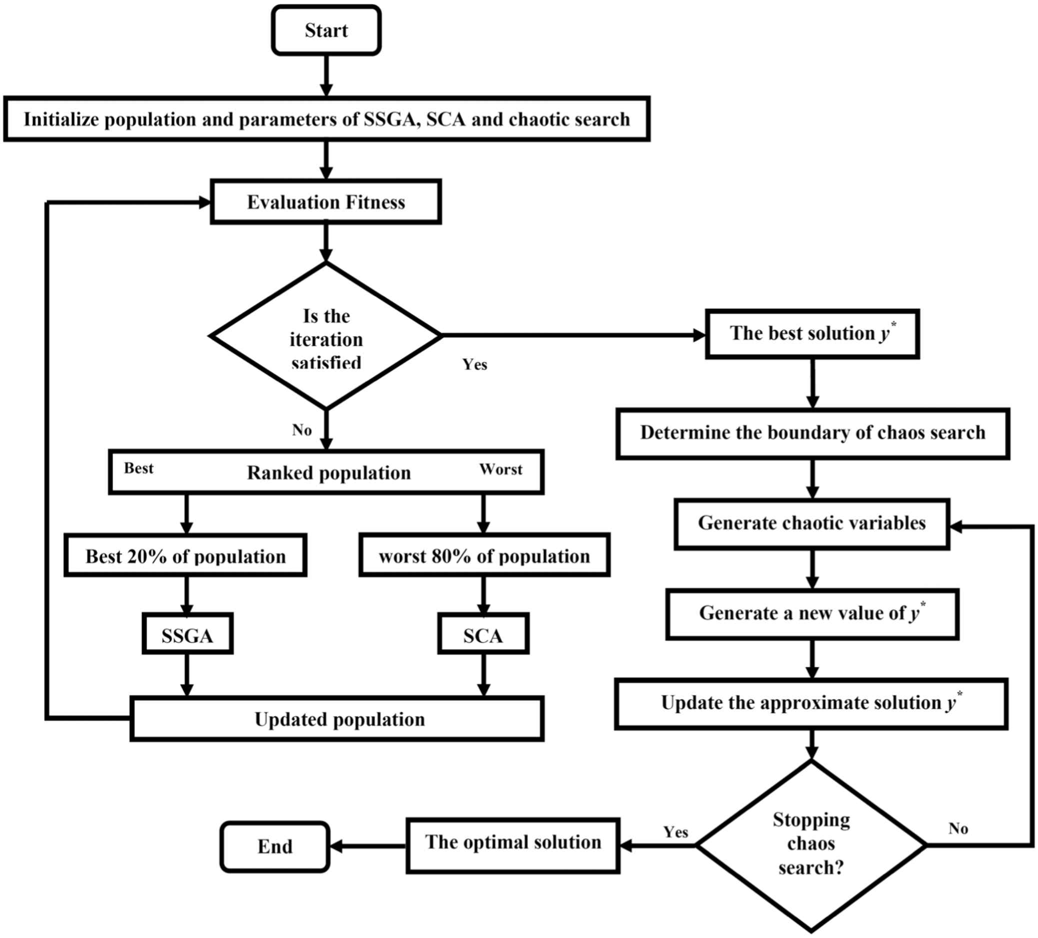


**FIGURE 4.** The flow chart of the proposed approach.

**TABLE 2.** The description of the parameters set.

| | |
|---|---|
| Generation of hybrid SCA with SSGA | 50-1000 |
| Generation gap | 0.90 |
| Mutation operator | Real-value |
| Mutation rate | 0.1 |
| Crossover operator | Single point |
| Crossover rate | 0.95 |
| Chaotic search iteration $k_{max}$ | 1000 |
| Specified neighborhood radius | 1E-4 |
| Initial chaotic random numbers $z^0$ | 1E-3 |

### A. PERFORMANCE EVALUATION OF CHAOTIC HYBRID APPROACH WITH DIFFERENT CHAOTIC MAPS

To evaluate the different chaotic maps, 6 unconstrained problems (chosen from the 25 CEC 2005 benchmark functions) are solved by chaos-enhanced SCA with SSGA algorithm. These problems are F1 from unimodal function and F9, F11, F13, F14, and F15 from multimodal functions. For each chaotic map, every problem is solved 30 times by the proposed algorithm. Table 3 presents the best solutions obtained by chaos-enhanced SCA with SSGA for each problem with different chaotic maps. Figure 5 gives a comparison between the 10 chaotic maps in accordance with the rank of the best results obtained by the proposed algorithm with every map. Table 3 and Figure 5 show that the quality of the solutions obtained by the chaos-enhanced SCA with SSGA with the sine map is better than the solutions obtained with other maps (i.e., the sine map obtains the best performance, where the sine chaotic map with the hybridization of SSGA and SCA gives the best results).

According to these results, the sine map is chosen to be used for the chaotic search in the proposed algorithm. The equation of the sine map [62] is:

$$z^{k+1} = \frac{c}{4} \sin\left(\pi z^k\right); \quad 0 < c \leq 4, \; k = 1, 2, \ldots \tag{6}$$

Figure 6 shows the sequence of 1000 chaotic random numbers generated by a sine map with $z^0 = 10^{-3}$ and $c = 4$.

### B. THE SIMULATION RESULTS OF UNCONSTRAINED PROBLEMS (25 CEC'2005 BENCHMARK PROBLEMS)

In this subsection, the chaos-enhanced SCA with SSGA is tested on 25 CEC 2005 benchmark functions of

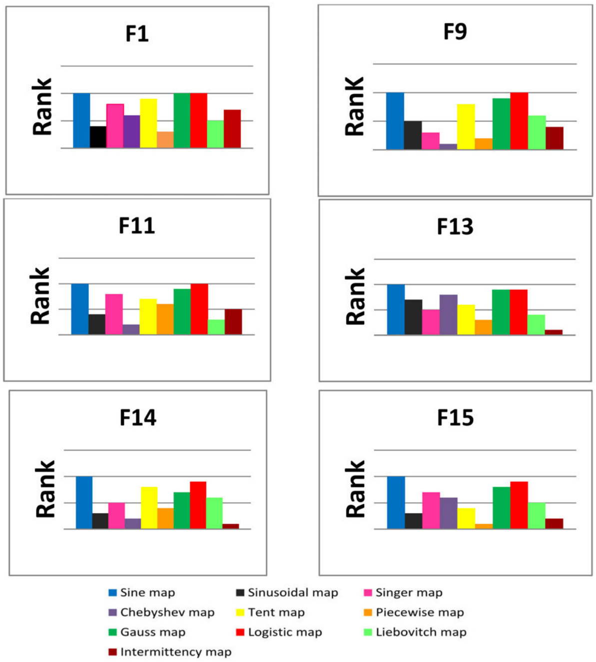


**FIGURE 5.** Comparison between the 10 chaotic maps in accordance with the rank of the best result gotten by each map.

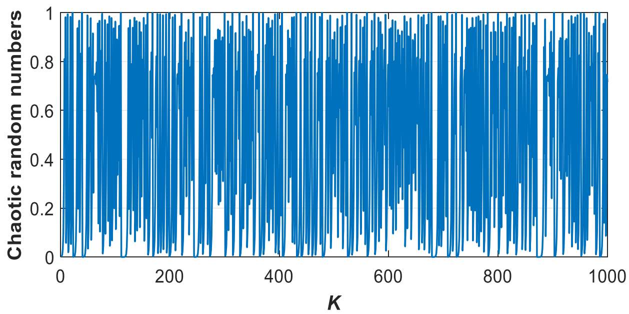


**FIGURE 6.** The chaotic random numbers generated by sine map.

dimension 10 [62]. The approach has been run 50 times for each SSGA, SCA, and the proposed algorithm based on a sine map. The termination condition is set to 100000 function evaluations. Table 4 gives a comparison between the global solutions and solutions obtained by SSGA, SCA, and the proposed algorithm of the 25 CEC2005 benchmark functions. Also, the percentage of improvement in the results obtained by the chaos-enhanced SCA with SSGA is shown in Table 4.

Moreover, a set of well-known swarm intelligence and evolutionary algorithms has been utilized for evaluating the performance of the proposed approach in comparison with them. This set of approaches was used before in [30], [71], [77]. Derrac *et al.* in [77] used CEC'2005 functions to illustrate the use of the set of nonparametric statistical procedures, conducting an analysis of the results of this set of well-known swarm intelligence and evolutionary algorithms in solving CEC2005 functions. Meanwhile, El-Shorbagy *et al.* [71] used this set of algorithms in evaluating the performance of CGA in solving the CEC2005 benchmark suite. Chakri *et al.* [30] used this set of evolutionary and

**TABLE 3.** The best values for each problem obtained by the proposed approach with different chaotic maps.

| Function Number | Optimal value | Chaos map name | Best value |
|---|---|---|---|
| F1 | -450.0000 | Sine map | -450.0000 |
| | | Sinusoidal map | -449.9970 |
| | | Singer map | -449.9988 |
| | | Chebyshev map | -449.9983 |
| | | Tent map | -449.9998 |
| | | Piecewise map | -449.9956 |
| | | Gauss map | -450.0000 |
| | | Logistic map | -450.0000 |
| | | Liebovitch map | -449.9973 |
| | | Intermittency map | -449.9986 |
| F9 | -330.0000 | Sine map | -330.0000 |
| | | Sinusoidal map | -329.9929 |
| | | Singer map | -329.8693 |
| | | Chebyshev map | -329.4527 |
| | | Tent map | -329.9998 |
| | | Piecewise map | -329.8693 |
| | | Gauss map | -329.9999 |
| | | Logistic map | -330.0000 |
| | | Liebovitch map | -329.9931 |
| | | Intermittency map | -329.9857 |
| F11 | 90.0000 | Sine map | 91.1704 |
| | | Sinusoidal map | 92.3247 |
| | | Singer map | 91.5900 |
| | | Chebyshev map | 93.9487 |
| | | Tent map | 91.8637 |
| | | Piecewise map | 92.0844 |
| | | Gauss map | 91.3237 |
| | | Logistic map | 91.1704 |
| | | Liebovitch map | 92.7274 |
| | | Intermittency map | 92.0844 |
| F13 | -130.00000 | Sine map | -129.8126 |
| | | Sinusoidal map | -129.6811 |
| | | Singer map | -129.5929 |
| | | Chebyshev map | -129.7687 |
| | | Tent map | -129.6330 |
| | | Piecewise map | -129.4615 |
| | | Gauss map | -129.7811 |
| | | Logistic map | -129.7845 |
| | | Liebovitch map | -129.5655 |
| | | Intermittency map | -129.2990 |
| F14 | -300.0000 | Sine map | -298.4206 |
| | | Sinusoidal map | -297.4316 |
| | | Singer map | -297.5384 |
| | | Chebyshev map | -297.4022 |
| | | Tent map | -297.9341 |
| | | Piecewise map | -297.5085 |
| | | Gauss map | -297.8617 |
| | | Logistic map | -298.0607 |
| | | Liebovitch map | -297.6681 |
| | | Intermittency map | -297.4022 |
| F15 | 120.00000 | Sine map | 120.0000 |
| | | Sinusoidal map | 120.0362 |
| | | Singer map | 120.0111 |
| | | Chebyshev map | 120.0111 |
| | | Tent map | 120.0256 |
| | | Piecewise map | 120.3592 |
| | | Gauss map | 120.0041 |
| | | Logistic map | 120.0002 |
| | | Liebovitch map | 120.0127 |
| | | Intermittency map | 120.3382 |

**TABLE 4.** Comparison between the results obtained by SSGA, SCA and proposed algorithm and the global solution of 25 CEC2005 benchmark functions with improvement percentage.

| Function | Global solution | SSGA | SCA | The proposed algorithm | Improvement percentage of the proposed algorithm over SSGA (%) | Improvement percentage of the proposed algorithm over SCA (%) |
|---|---|---|---|---|---|---|
| 1 | 450.000 | -449.8876 | -449.9955 | 450.000 | 0.02498 | 0.001 |
| 2 | 450.000 | -449.5177 | -449.9205 | -450.0000 | 0.107 | 0.017667 |
| 3 | 450.000 | 426.4662 | 426.4662 | 427.3796 | 0.213721 | 0.213721 |
| 4 | 450.000 | -448.3935 | -449.7211 | -450.0000 | 0.3582791 | 0.062016 |
| 5 | 310.000 | -308.5684 | -309.4667 | -309.9491 | 0.44745347 | 0.1556385 |
| 6 | 390.000 | 392.9308 | 392.1299 | 390.3624 | 0.65365199 | 0.44732125 |
| 7 | 180.000 | 176.3826 | 176.3826 | 177.3475 | 0.5470494 | 0.5440731 |
| 8 | 140.000 | -119.4877 | -119.8724 | -119.990 | 0.420378 | 0.1 |
| 9 | 330.000 | -329.0030 | -329.0248 | -330.0000 | 0.303036 | 0.296391 |
| 10 | 330.000 | -324.0303 | -324.0303 | -326.0202 | 0.6163528 | 0.6163528 |
| 11 | 90.000 | 94.4454 | 94.2294 | 91.1704 | 2.67642 | 3.246333 |
| 12 | 460.000 | -451.6924 | -456.5120 | -457.7845 | 1.348727 | 0.27874404 |
| 13 | 130.000 | -127.6223 | -128.5227 | -129.8126 | 1.71623 | 1.003557 |
| 14 | 300.00 | -296.504 | -297.163 | -298.4206 | 0.6463994 | 0.4232021 |
| 15 | 120.000 | -127.6223 | 123.2967 | 120.0000 | 5.972546 | 2.673794 |
| 16 | 120.000 | 241.8480 | 230.8657 | 226.5979 | 6.30565479 | 1.849907 |
| 17 | 120.000 | 226.9454 | 226.6490 | 223.1319 | 1.68036 | 1.551783 |
| 18 | 10.0000 | 310.3357 | 310.1975 | 310.0000 | 0.1081732 | 0.06366911 |
| 19 | 10.0000 | 310.3956 | 310.6151 | 310.0000 | 0.1275469 | 0.1980265 |
| 20 | 10.0000 | 310.0759 | 310.0217 | 310.0007 | 0.0244778 | 0.0102 |
| 21 | 360.000 | 660.6811 | 660.6811 | 660.0024 | 0.1027 | 0.10272 |
| 22 | 360.000 | 661.0894 | 660.8124 | 660.0015 | 0.1647886 | 0.1229396 |
| 23 | 360.000 | 463.3158 | 462.2815 | 460.0002 | 0.71567 | 0.4935305 |
| 24 | 260.000 | 464.0742 | 462.0442 | 460.0000 | 0.877023 | 0.4423819 |
| 25 | 260.000 | 467.3184 | 465.2815 | 460.0000 | 1.56604 | 101348 |

swarm intelligence algorithms to compare the performance of the directional bat algorithm with them in solving the CEC2005 benchmark suite.

The algorithms considered are as follows: steady-state genetic algorithm (SSGA) [77], [78], the CHC algorithm [79], two instances of the classic scatter search model SS-Arit and SS-BLX [80], [81], restart covariant matrix evolutionary strategy with an increasing population (IPOP-CMA-ES) [82], the PSO algorithm [77], [83], and self-adaptive differential evolution (SaDE) [84], the classical differential evolution with two crossover strategies Rand/1/exp (DE-Exp) and Rand/1/bin (DE-Bin) [85], chaotic genetic algorithm (CGA) [71], and finally directional bat algorithm (dBA) [30]. The parameter settings of these algorithms used in the comparison are given in [77] and [30]. All the above approaches had been run 50 times for each benchmark problem. Each approach terminated either when the maximal number of evaluations reached 100000 or when the error obtained is less than $10e^{-8}$. The results of the average errors (the differences between the best-obtained results and the true global optima) of these algorithms provided in [30], [71], [77] are used to perform the comparison between the proposed approach performance and the other techniques on CEC2005 benchmark functions. The comparison between the average errors obtained for 25 unconstrained benchmark problems with the proposed algorithm and the other optimization approaches is given in Table 5. Moreover, the different approaches used in the comparison are ranked based on the average error values and the results are given in Figure 7.

#### 1) PERFORMANCE ANALYSIS ON SOLVING UNCONSTRAINED OPTIMIZATION PROBLEMS

The simulation result in Table 4 shows the chaos-enhanced SCA with SSGA can improve the quality of the solutions of both SSGA and SCA. Moreover, it can be noted that the proposed algorithm improves the quality of the solutions in all functions, where the proposed algorithm obtains solutions better than those obtained by SSGA or SCA. In addition, the chaos-enhanced SCA with SSGA converges to the optimal value, escaping from local optima. This means that introducing a hybrid between SCA and SSGA with chaotic search accelerates the seeking operation of the global optimal solution. Moreover, the comparison between the average error obtained by the proposed approach and 11 continuous optimization algorithms given in Table 5 proves that the proposed approach obtains better solutions than those of all 11 optimization algorithms on average. Moreover, as per the results are given in Figure 7, the proposed approach has the first rank 20 times and the second rank twice, which means that the proposed approach is better than other swarms intelligent and evolutionary algorithms. Thus, it can be stated that the proposed hybrid algorithm works well and converges rapidly towards the optimal solution to unconstrained benchmark problems.

### *C. THE SIMULATION RESULTS OF CONSTRAINT PROBLEMS*

For the constrained problems, every constrained problem is independently solved by the proposed approach 30 times

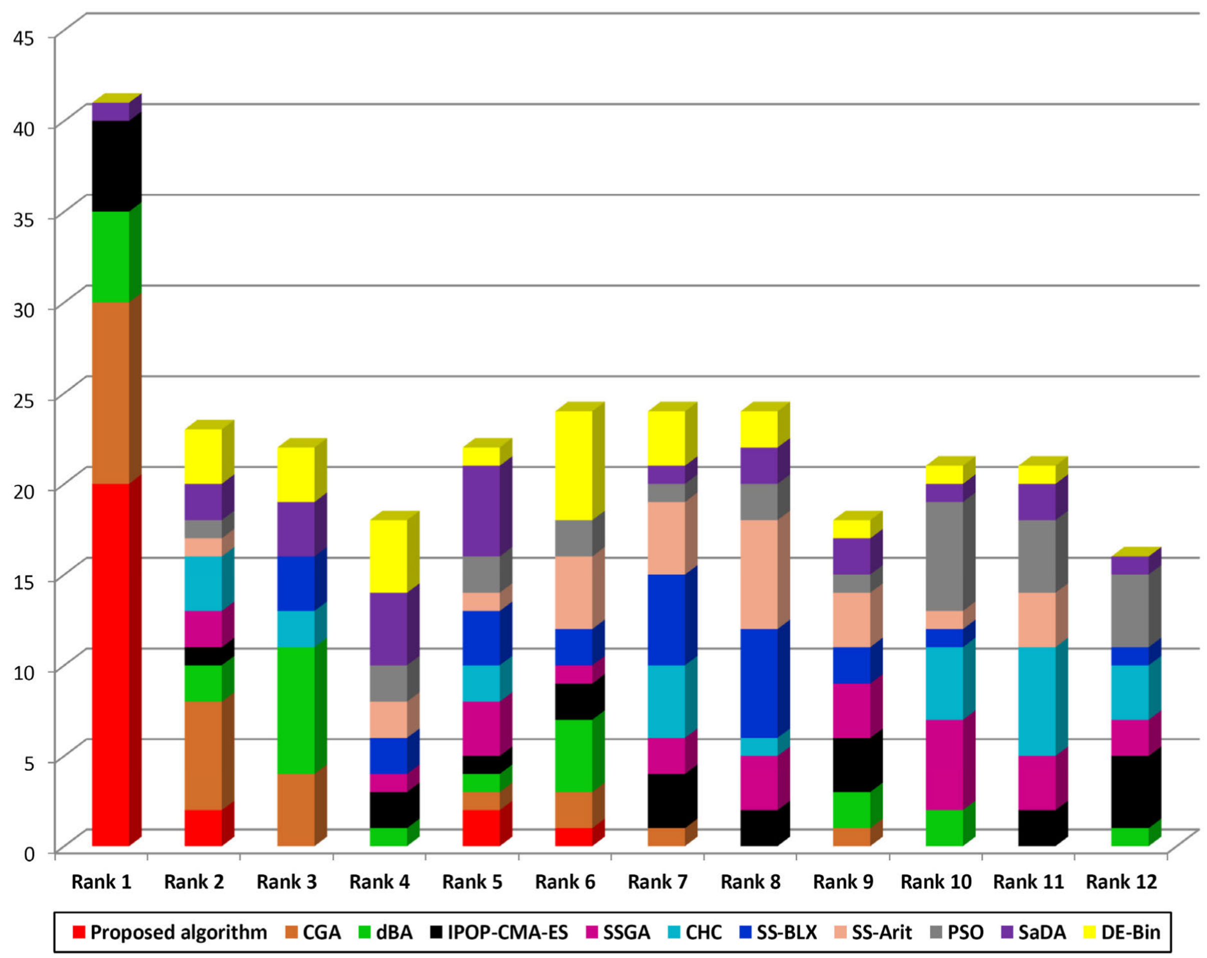


**FIGURE 7.** The comparison of the different algorithms for unconstrained functions based on their ranks.

(similar to the algorithms which are used in the comparison [71]) with the same number of iterations. The best values, mean values, worst values, and standard deviation are used to evaluate the proposed algorithm convergence speed, as well as the solution accuracy and stability compared to the SSGA and SCA.

The results are illustrated in Table 6, while Figure 8 gives the convergence curves of the best run of the proposed algorithm, SCA, and SSGA for all constrained problems. Table 7 gives the comparison between the optimal solution and the best values of the 7 benchmark functions obtained by Augmented Lagrange particle swarm optimization (ALPSO) [63], chaotic genetic algorithm (CGA) [71], self-adaptive velocity particle swarm optimization (SAVPSO) [65], A hybridization of GA with an artificial immune system (GA-AIS) [86], Adaptive Penalty Method (APM) [86], and the proposed algorithm. The parameter settings of CGA and ALPSO were described in [71], where the parameter settings of GA-AIS and APM were given in [86]. In addition, The ALPSO, CGA, APM, SAVPSO, GA-AIS, and the proposed algorithm are ranked according to the error values(the difference between the mean values of the result obtained by algorithms over 30 runs and the global solution) and the results are shown in Table 8.

#### 1) PERFORMANCE ANALYSIS ON SOLVING CONSTRAINED OPTIMIZATION PROBLEMS

As a result of Table 6 and Figure 8, the proposed algorithm obtains better results than those obtained by both SSGA and SCA and converges to the global solution faster than both SSGA and SCA. This means that the hybridization of SCA and chaotic search with SSGA speed up the optimum seeking operation and lead to avoiding falling in local optima. In addition, Tables 7 and 8 prove that the proposed approach is superior to ALPSO, APM, GA-AIS, SAVPSO, and CGA with regard to most test functions. For test functions C1, C3, C5, C6, and C7, the proposed algorithm gives an accurate global optimizing capability, where it obtained the solution equal to the global solutions. For test functions C2 and C4, the proposed algorithm exhibits an excellent optimization capability, where it obtained global solutions. Therefore, the results indicate that the proposed algorithm is considerably superior

**TABLE 5.** Average error obtained by proposed algorithm and 11 continuous optimization algorithms for the 25 CEC'2005 benchmark functions.

| Function | SSGA [78] | CHC [79] | SS-BLX [80] | SS-Arit [81] | IPOP-CMA-ES [82] | PSO [83] | SaDE [84] | DE-Bin [85] | DE-Exp [85] | CGA [71] | dBA [30] | Proposed algorithm |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F1 | 8.420E-9 | 2.464 | 34.02 | 1.064 | 0.000 | 0.00012 | 8.416E-9 | 7.716E-9 | 8.260E-9 | 0.000 | 0.000 | 0.00 |
| F2 | 8.719E-5 | 0.01.180 | 1.730 | 5.282 | 0.000 | 0.02.595 | 8.208E-9 | 8.342E-9 | 8.181E-9 | 0.000 | 0.00 | 0.00 |
| F3 | 79480.0 | 269900.0 | 184400.0 | 253500.0 | 0.000 | 51740.0 | 6560.0 | 42.33 | 99.35 | 22.6204 | 2.356E+05 | 22.6204 |
| F4 | 0.002585 | 91.90 | 6.228 | 5.755 | 2932.0 | 2.488 | 8.087E-9 | 7.686 E-9 | 8.350E-9 | 0.000 | 0.001215 | 0.00 |
| F5 | 134.3 | 264.1 | 2.185 | 14.43 | 8.104E-1 | 409.5 | 8.640E-9 | 8.608E-9 | 8.514E-9 | 1.6E 3 | 0 | 0.0509 |
| F6 | 6.171 | 1.416E+6 | 114.5 | 494.5 | 0.000 | 731.0 | 0.01612 | 7.956E-9 | 8.391E-9 | 1.724 | 35.38 | 0.3624 |
| F7 | 1271.0 | 1269.0 | 1966.0 | 1908.0 | 1267.0 | 26.78 | 1263.0 | 1266.0 | 1265.0 | 2.653 | .43414 | 2.6525 |
| F8 | 20.37 | 20.34 | 20.35 | 20.36 | 20.01 | 20.43 | 20.32 | 20.33 | 20.38 | 20.08 | 20.35 | 20.01 |
| F9 | 7.286E-9 | 5.886 | 4.195 | 5.960 | 28.41 | 14.38 | 8.330E-9 | 4.546 | 8.151E-9 | 0.000 | 8.216 | 0.000 |
| F10 | 17.12 | 7.123 | 12.39 | 21.79 | 23.27 | 14.04 | 15.48 | 12.28 | 11.18 | 3.980 | 10.49 | 3.9798 |
| F11 | 3.255 | 1.599 | 2.929 | 2.858 | 1.343 | 5.590 | 6.796 | 2.434 | 2.067 | 3.276 | 3.758 | 1.1704 |
| F12 | 279.4 | 706.2 | 150.6 | 241.1 | 212.7 | 636.2 | 56.34 | 106.1 | 63.09 | 2.640 | 188.5 | 2.2155 |
| F13 | 67.13 | 82.97 | 32.45 | 54.79 | 1.134 | 1.503 | 70.70 | 1.573 | 64.03 | 0.363 | 1.045 | 0.1874 |
| F14 | 2.264 | 2.073 | 2.796 | 2.970 | 3.775 | 3.304 | 3.415 | 3.073 | 3.158 | 3.052 | 2.965 | 1.5794 |
| F15 | 292.0 | 275.1 | 113.6 | 128.8 | 193.4 | 339.8 | 84.23 | 372.2 | 294.0 | 0.000 | 216.6 | 0.000 |
| F16 | 105.3 | 97.29 | 104.1 | 113.4 | 117.0 | 133.3 | 122.7 | 111.7 | 112.5 | 92.745 | 118.2 | 106.5979 |
| F17 | 118.5 | 104.5 | 118.3 | 127.9 | 338.9 | 149.7 | 138.7 | 142.1 | 131.2 | 109.6 | 126.6 | 103.1319 |
| F18 | 806.3 | 879.9 | 766.8 | 657.8 | 557.0 | 851.2 | 532.0 | 509.7 | 448.2 | 300.0 | 447.1 | 300.00 |
| F19 | 889.9 | 879.8 | 755.5 | 701.0 | 529.2 | 849.7 | 519.5 | 501.2 | 434.1 | 300.0 | 449.9 | 300 |
| F20 | 889.3 | 896.0 | 746.3 | 641.1 | 526.4 | 850.9 | 476.7 | 492.8 | 418.8 | 362.8 | 394.6 | 300.0007 |
| F21 | 852.2 | 815.8 | 485.16 | 500.5 | 442.0 | 913.8 | 514.0 | 524.0 | 542.0 | 300.0 | 413.5 | 300.00 |
| F22 | 751.9 | 774.2 | 682.8 | 694.1 | 764.7 | 807.1 | 765.5 | 771.5 | 772.0 | 752.4 | 588.9 | 300.0015 |
| F23 | 1004.0 | 1075 | 574.0 | 582.8 | 853.9 | 102.8 | 650.9 | 633.7 | 582.4 | 559.5 | 559.5 | 100.0002 |
| F24 | 236.0 | 295.9 | 251.3 | 201.1 | 610.1 | 412.0 | 200.0 | 206.0 | 202.0 | 200.0 | 200.00 | 200 |
| F25 | 1747.0 | 1764.0 | 1794.0 | 1804.0 | 1818.0 | 509.9 | 1738.0 | 1744.0 | 1742.0 | 430.77 | 318.4 | 200 |

**TABLE 6.** The Simulation results of the proposed approach, SSGA and SCA for constrained problems [63]–[65].

| Benchmark function | Method | Best values | Mean values | Worst values | Standard of deviation |
|---|---|---|---|---|---|
| | SSGA | 13.000 | 13.0683 | 13.4247 | 0.1494 |
| C1 | SCA | 13.000 | 13.2320 | 13.9686 | 0.4187 |
| | Proposed algorithm | 13.000 | 13.000 | 13.000 | 0 |
| | SSGA | 0.01721 | 0.017827 | 0.0186 | 7.3259e-04 |
| C2 | SCA | 0.0172 | 0.0471 | 0.1461 | 0.0508 |
| | Proposed algorithm | 0.017177 | 0.017187 | 0.017187 | 0 |
| | SSGA | -0.0958248 | -0.09582447 | -0.095824 | 3.8347e-07 |
| C3 | SCA | -0.0958251 | -0.09582467 | -0.09582411 | 4.2223e-07 |
| | Proposed algorithm | -0.095831 | -0.0958275 | -0.095825 | 2.6352e-06 |
| | SSGA | -6.9607e+03 | -6.9440e+03 | -6.9109e+03 | 18.9555 |
| C4 | SCA | -6.96100 e+03 | -6.9603e+03 | -6.9595e+03 | 0.7051 |
| | Proposed algorithm | -6.9618138e+03 | -6.96154605e+03 | -6.96066e+03 | 0.4731 |
| | SSGA | 0. 8007 | 0.8610 | 0.9667 | 0.0749 |
| C5 | SCA | 0.7594 | 0.8271 | 0.9687 | 0.0800 |
| | Proposed algorithm | 0.7500 | 0.7535 | 0.7654 | 0.0063 |
| | SSGA | -0.8286 | -0.6209 | -0.5460 | 0.4250 |
| C6 | SCA | -0.9485 | -0.8076 | -0.5371 | 0.1574 |
| | Proposed algorithm | -1.0001 | -0.9461 | -0.7570 | 0.0810 |
| | SSGA | -3.0643e+04 | -3.0384e+04 | -2.09868e+04 | 260.4626 |
| C7 | SCA | -3.0651 e+04 | -3.0596e+04 | -3.0507e+04 | 48.9616 |
| | Proposed algorithm | -3.06655e+04 | -3.0659e+04 | -3.0646e+04 | 7.2360 |

to other approaches with regard to the accuracy of the optimization and capabilities of global optimizing.

### *D. THE SIMULATION RESULTS OF ENGINEERING DESIGN PROBLEMS*

To check the efficiency of chaos-enhanced SCA with SSGA, each problem of the six applications is independently run ten times with the same number of iterations. Table 9 gives the worst values, best values, average values, and the standard deviation by our proposed approach, SCA, and the SSGA to assess the convergence speed, in addition to the stability and solution accuracy of the algorithm. Furthermore, Figure 9 presents the convergence curves of the best run of the proposed approach, SCA, and SSGA for all applications. In addition, Table 10 gives a comparison between the best solution obtained by our approach and different optimization algorithms such as nonlinear optimization software (CONOPT, KNITRO, MINOS, and SNOPT) [66], hybrid glowworm swarm optimization (HGSO) [87], and multiobjective tabu/scatter search (MITS) [67].

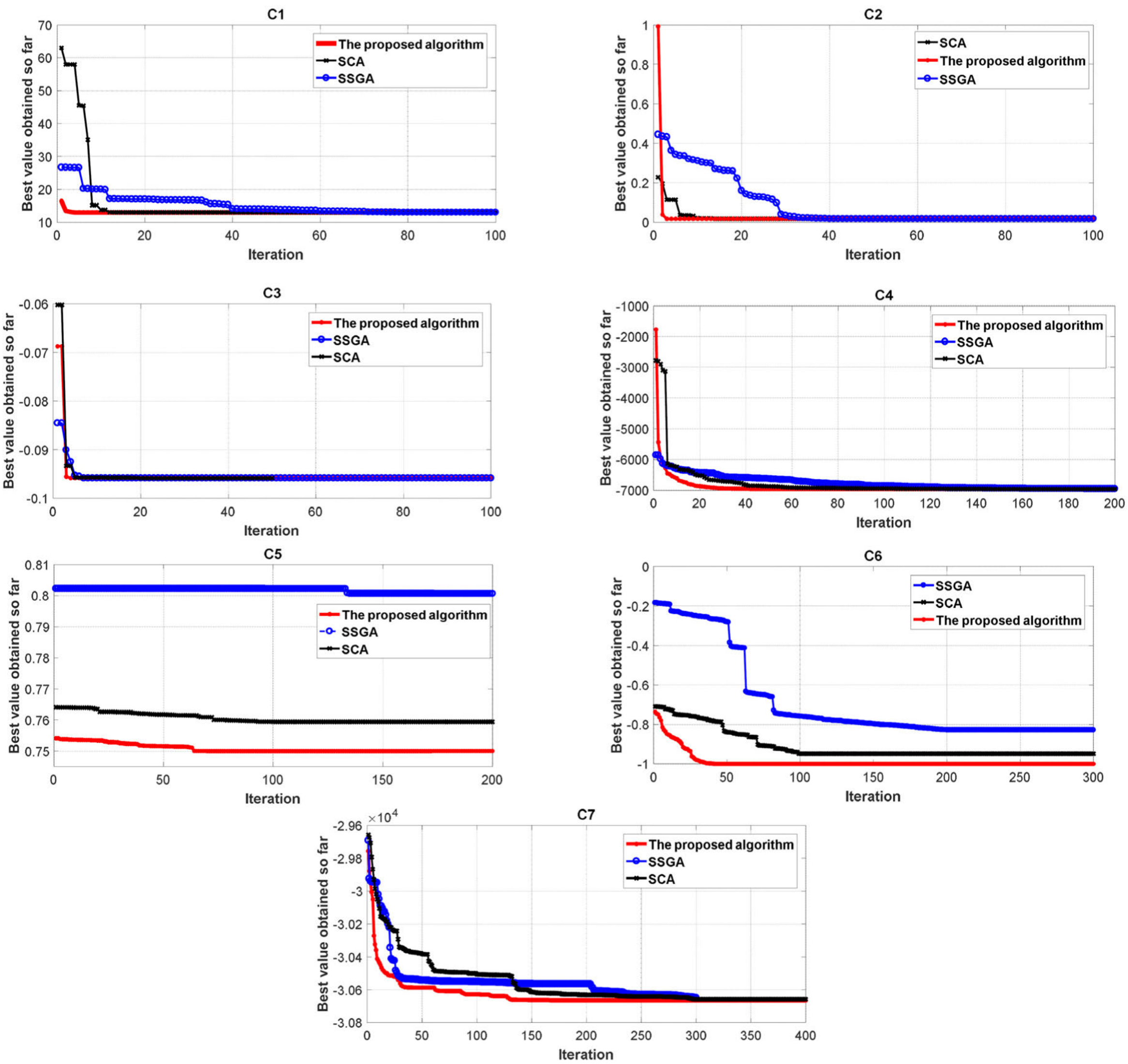


**FIGURE 8.** The convergence curves of proposed algorithm and SSGA for constrained optimization problems.

**TABLE 7.** The comparison of the best solutions obtained by ALPSO [63], CGA [71], SAVPSO [65], APM [86], GA-AIS [86], and the proposed algorithm and the optimal solutions of the 7 constrained benchmark function.

| Function | Optimal solution | ALPSO | APM | GA-AIS | SAVPSO | CGA | Proposed algorithm |
|---|---|---|---|---|---|---|---|
| C1 | 13.0000 | 12.9995 | NA | NA | NA | 13.0000 | 13.0000 |
| C2 | 0.01721 | 0.01719 | NA | NA | NA | 0.01721 | 0.017187 |
| C3 | 0.09583 | 0.09583 | 0.0958250 | 0.0958250 | 0.095825 | 0.09583 | -0.095831 |
| C4 | 6961.81 | 6963.57 | 6961.7961 | 6961.7961 | 6961.813875 | 6961.804 | -6961.814 |
| C5 | 0.75 | 0.75 | .7499568 | 0.75002 | 0.74900 | 0.75 | 0.74900 |
| C6 | -1.000 | NA | -1.0004896 | -.88799 | 1.004814 | NA | -1.0001 |
| C7 | 30665.5 | 30665.5 | 30566.3455 | 30665.3889 | 30665.538672 | -30665.8 | -30665.5 |

NA'' means that the result is not available

### 1) PERFORMANCE ANALYSIS ON SOLVING ENGINEERING APPLICATIONS

As shown in Figure 9 and Table 9, the results of the chaos-enhanced SCA with SSGA were better compared to the results obtained by both SSGA and SCA. Furthermore, the proposed approach converges to the best solution faster than SSGA and SCA which means that our hybrid approach speeds up the optimum seeking process and converges to prove that the proposed approach outperforms other approaches at most applications. Firstly, our algorithm shows its superiority over other approaches in three applications out of six (P1, P5, and P6). But, for the application P2,

**TABLE 8.** Ranking of the average error values of the constrained benchmark functions for ALPSO, CGA, and the proposed algorithm.

| Function | ALPSO | CGA | APM | GA-AIS | SAVPSO | The proposed Algorithm |
|---|---|---|---|---|---|---|
| C1 | Rank 2 | Rank 1 | NA | NA | NA | Rank 1 |
| C2 | Rank 2 | Rank 3 | NA | NA | NA | Rank 1 |
| C3 | Rank 1 | Rank 1 | Rank 2 | Rank 2 | Rank 2 | Rank 1 |
| C4 | Rank 3 | Rank 4 | Rank 5 | Rank 5 | Rank 2 | Rank 1 |
| C5 | Rank 3 | Rank 3 | Rank 2 | Rank 4 | Rank 2 | Rank 1 |
| C6 | NA | NA | Rank 1 | Rank 4 | Rank 2 | Rank 4 |
| C7 | Rank 3 | Rank 1 | Rank 5 | Rank 4 | Rank 2 | Rank 3 |

NA'' means that the result is not available

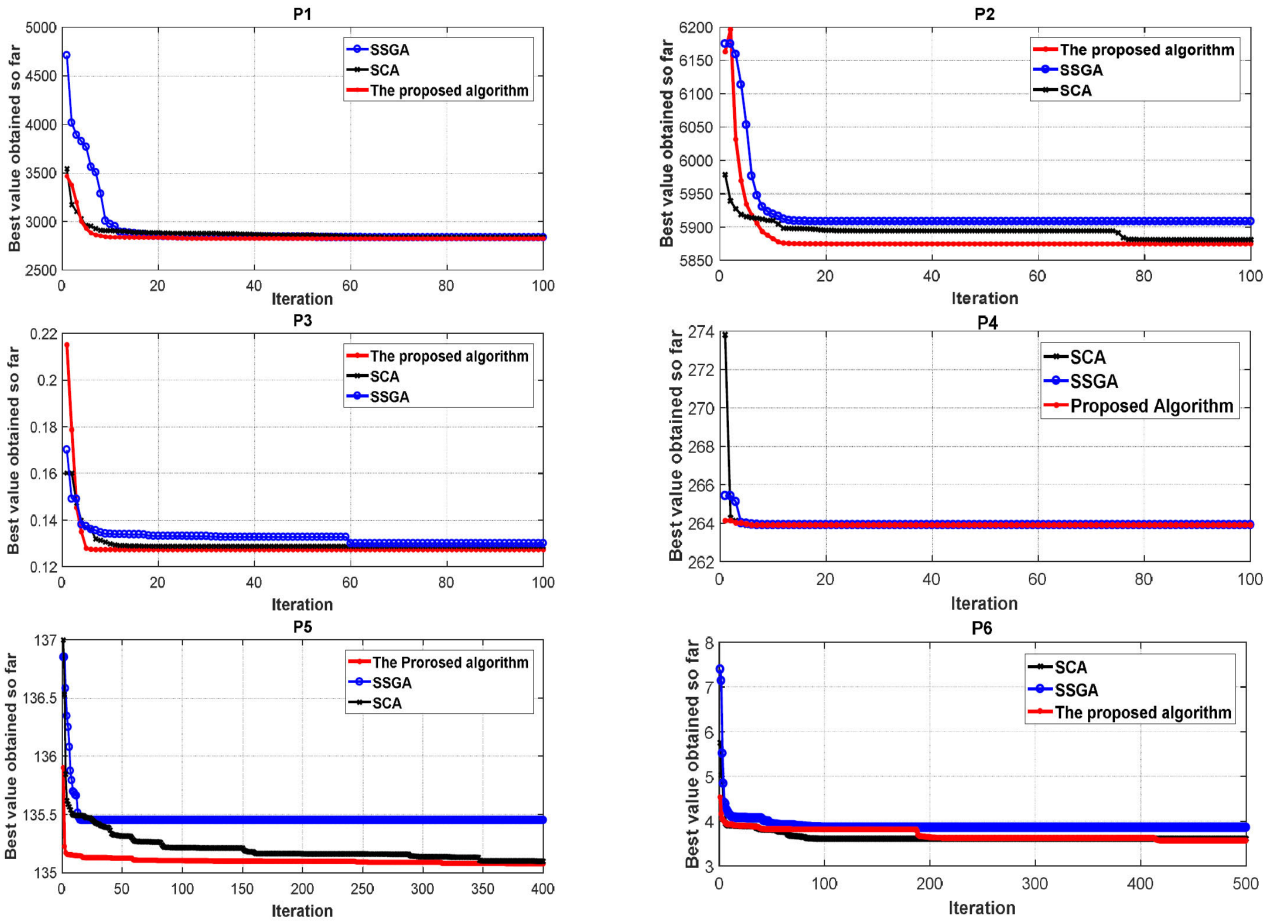


**FIGURE 9.** The convergence curves of the best run of the proposed approach, SCA and SSGA for the engineering applications.

the proposed algorithm outperforms CONOPT, KNITRO, HGSO, and MITS, but it is equal in performance with SNOPT and MINOS. Also, for application P4 the, proposed algorithm outperforms MITS, but it is equal in performance with the others. For application P3, the proposed algorithm's performance is equal to the other methods. Finally, we can say that the results of various applications demonstrate that our proposed algorithm is considerably superior to other approaches with respect to global optimizing capabilities and accuracy of optimization.

A comparative study was carried out in this section to evaluate the proposed meta-heuristic hybrid approach. Firstly, meta-heuristic algorithms suffer from the solution quality. Therefore the suggested method was introduced to improve the efficiency of the solution by integrating the merits of two meta-heuristic algorithms. In addition, a chaotic search (CS) was used to ensure that the optimal solution was found. On the other hand, unlike deterministic algorithms, our method is searching by using a population of points, not a single point. It can thus give an optimal solution. Furthermore, our

**TABLE 9. The Simulation results of the proposed approach, SSGA and SCA for engineering design problems.**

| | Application | Method | Best values | Mean values | Worst values | Standard deviation |
|---|---|---|---|---|---|---|
| Mechanical application | p1 | SSGA | 2.8240459e+03 | 2.8291317e+03 | 2.8351097e+03 | 4.57428 |
| | | SCA | 2.8272e+03 | 2.8326e+03 | 2.8373e+03 | 3.9639 |
| | | Proposed algorithm | 2.823663279e+03 | 2.82367294e+03 | 2.823672949e+03 | 0.004668 |
| | p2 | SSGA | 5.9084614e+03 | 5.9333307e+03 | 5.95473599e+03 | 1.7404914 |
| | | SCA | 5.8762049e+03 | 5.8973e+03 | 5.93742238e+03 | 2.0693 |
| | | Proposed algorithm | 5.87487202e+03 | 5.8862885e+03 | 5.90846135e+03 | 0.0320 |
| | p3 | SSGA | 0.129917 | 0.1399 | 0.1566 | 0.0101 |
| | | SCA | 0.1288 | 0.1441 | 0.15160 | 0.0113 |
| | | Proposed algorithm | 0.1274000 | 0.1274000 | 0.1274000 | 0 |
| | p4 | SSGA | 263.8967 | 263.9107 | 263.9376 | 0.0166 |
| | | SCA | 263.8960 | 263.8963 | 263.8964 | 1.8291e-04 |
| | | Proposed algorithm | 263.895843 | 263.895846 | 263.895853 | 4.2302e-06 |
| Electrical application | p5 | SSGA | 135.4517 | 135.9875 | 136.8686 | 0.5821 |
| | | SCA | 135.1005 | 135.1454 | 135.2102 | 0.0499 |
| | | Proposed algorithm | 135.0759 | 135.0790 | 135.0899 | 0.0044 |
| Chemical application | P6 | SSGA | 3.8572 | 3.9483 | 4.2147 | 0.1197 |
| | | SCA | 3.6098 | 3.6809 | 3.7439 | 0.0606 |
| | | Proposed algorithm | 3.5545 | 3.6013 | 3.6233 | 0.032 |

**TABLE 10. The comparison between the results of the proposed algorithm and various continuous optimization approaches in engineering design problems.**

| | Application | CONOPT [66] | KNITRO [66] | MINOS [66] | SNOPT [66] | HGSO [87] | MITS [67] | Proposed algorithm |
|---|---|---|---|---|---|---|---|---|
| Mechanical applications | P1 | 2823.6724 | 2823.6722 | 2823.6721 | 2823.6718 | 2994.47107 | 2996.39519 | 2823.663279 |
| | P2 | 5874.8789 | 5874.8790 | 5874.872 | 5874.872 | 6059.71434 | 6090.539376934 | 5874.87202 |
| | P3 | 0.12740 | 0.12740 | 0.12740 | 0.12740 | NA | NA | 0.12740 |
| | P4 | 263.8958433 | 263.8958434 | 263.895843 | 263.895843 | 263.89584 | 263.9077 | 263.895843 |
| Electrical application | P5 | 135.075962 | 135.075960 | NA | 135.075962 | NA | NA | 135.07590 |
| Chemical application | P6 | 3.9511634 | 3.9511632 | 4.286758 | 3.951163 | NA | NA | 3.6098 |

NA'' means that the result is not available

approach uses only details on the objective function, not other properties. It can thus solve the optimization problems of non-smooth, non-continuous, and non-differentiable that currently occur in real-life applications. Another encouraging point is that the results of the simulation indicate the superiority of the proposed meta-heuristic hybrid approach to those stated in the literature, as it is better than both SCA, SSGA, and other approaches. The reason for this is due to the integration between SCA (exploration ability) and SSGA (exploitation ability) and CS (local search capability). Finally, due to the simplicity of the new meta-heuristic hybrid approach, it can address complex problems of realistic dimensions. But in general, as with all meta-heuristic approaches, the proposed meta-heuristic hybrid approach can usually make improvements in terms of either computational speed or accuracy. In other words, a guarantee of improvement in the computational speed or accuracy is not guaranteed when solving any optimization problem. However, it can be said that the proposed approach is competitive and able to solve nonlinear programming and engineering applications efficiently.

Computational complexity (CC) is concerned with evaluating the amount of work required to solve a specific problem. Evaluating the CC of the proposed algorithm includes looking for a very wide area; where the CC for these issues is NP. The suggested method can be used to arrive at an appropriate solution with an acceptable running time and give an appropriate answer that may be the best answer, or close to it. The key explanation for the lower CC of running the proposed algorithm is that our method prevents a systematic search of the problem area. Finally, with an acceptable choice of SCA parameters, SSGA operators, and CS procedures, the proposed hybrid method can be used to achieve optimum computational complexity.

## V. CONCLUSION

This article proposed newly meta-heuristic approach, steady-state sine cosine genetic algorithm-based chaotic search, to solve nonlinear programming and engineering applications. It integrates the merits of the sine cosine algorithm (SCA), steady-state genetic algorithm (SSGA), and chaotic search (CS) and named as chaos-enhanced SCA with SSGA. It was tested by different benchmark problems and engineering applications to demonstrate its superior to find the global optimal solution. The proposed algorithm showed several advantages, which we mention as follows:

1) The proposed algorithm merges SSGA's exploitation ability, SCA's exploration ability, and local search

capability of CS which leads to an increase in its capacity to reach the global solution.
2) Incorporating the SSGA with SCA and CS avoid the trapping into local minima, quicken the seeking process, and accelerate the convergence to the global solution.
3) Generating the solutions in a new generation by SSGA operations and mechanisms of SCA preserves the diversity of the individuals and prevents them from being similar to each other.
4) The results proved that the proposed algorithm improved the solution quality and the convergence rate of both SCA and SSGA.
5) The results of numerical experiments proven that the proposed algorithm is superior to those stated in the literature, as it is significantly better than other methods of comparison.
6) It can be used to solve complex practical optimization problems due to its simplicity.

In future works, other large-scale and more complex engineering problems can be considered such as resource allocation problem, economic load dispatch problem, unit commitment problem, optimization of wind turbines sitting in a wind farm, real-time applications, etc. Also, the proposed approach can be modified to solve other optimization problems such as nonlinear bilevel programming problems, interval quadratic programming problems, data clustering problems, etc. Finally, the proposed algorithm can be developed in order that it can solve multiobjective optimization problems.

## APPENDIX

The mathematical formulations of constrained benchmark problems and engineering design problems are presented below:

### APPENDIX I. CONSTRAINED BENCHMARK PROBLEMS

C1: Constrained problem 1

$$\begin{aligned}
&\text{Min } x_1^2 + x_2^2\\
&\text{Subject to: } x_1 - 3 = 0\\
&\qquad -x_2 + 2 \le 0\\
&\qquad -10 \le x_t \le 10, \quad t = 1, 2
\end{aligned}$$

C2: Constrained problem 2

$$\begin{aligned}
&\text{Min } \frac{1}{4000}(x_1^2 + x_2^2) - \cos(\frac{x_1}{\sqrt{1}})\cos(\frac{x_2}{\sqrt{2}}) + 1\\
&\text{Subject to: } x_1 - 3 = 0\\
&\qquad -x_2 + 2 \le 0\\
&\qquad -10 \le x_t \le 10, \quad t = 1, 2
\end{aligned}$$

C3: Constrained problem 3

$$\begin{aligned}
&\text{Min } \frac{-\sin(2\pi x_1)^3 \sin(2\pi x_1)}{x_1^3(x_2 + x_1)}\\
&\text{Subject to: } -x_1 + (x_2 - 4)^2 + 1 \le 0\\
&\qquad x_1^2 - x_2 + 1 \le 0\\
&\qquad 0.1 \le x_1 \le 10, \quad 0 \le x_2 \le 10
\end{aligned}$$

C4: Constrained problem 4

$$\begin{aligned}
&\text{Min } (x_1 - 10)^3 + (x_2 - 20)^3\\
&\text{Subject to: } -(x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \le 0\\
&\qquad -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \le 0\\
&\qquad 10 \le x_1 \le 13,\\
&\qquad 0 \le x_2 \le 100
\end{aligned}$$

C5: Constrained problem 5

$$\begin{aligned}
&\text{Min } x_1^2 + (x_2 - 1)^2\\
&\text{Subject to: } -x_1^2 + x_2 = 0\\
&\qquad -1 \le x_t \le 1, \quad t = 1, 2
\end{aligned}$$

C6: Constrained problem 6

$$\begin{aligned}
&\text{Min } \left(\sqrt{\text{dim}}\right)^{\text{dim}} \prod_{t=1}^{\text{dim}} x_t\\
&\text{Subject to: } \sum_{t=1}^{\text{dim}} x_t^2 - 1 = 0\\
&\qquad 0 \le x_t \le 1, \quad t = 1, 2, \ldots, 4
\end{aligned}$$

C7: Constrained problem 7

$$\begin{aligned}
&\text{Min } 5.357857x_3^2 + 0.8356891x_1x_5 + 37.293239x_1\\
&\qquad - 40792.141\\
&\text{Subject to:}\\
&\begin{bmatrix} -85.334407 - 0.0006262x_1x_4\\ -0.0056858x_2x_5 + 0.0022053x_3x_5 \end{bmatrix} \le 0\\
&\begin{bmatrix} 85.334407 + 0.0006262x_1x_4\\ +0.0056858x_2x_5 - 0.0022053x_3x_5 - 92 \end{bmatrix} \le 0\\
&\begin{bmatrix} -80.51249 - 0.0029955x_1x_2\\ -0.0071317x_2x_5 - 0.0021813x_3^2 + 90 \end{bmatrix} \le 0\\
&\begin{bmatrix} 80.51249 + 0.0029955x_1x_2 + 0.0071317x_2x_5\\ +0.0021813x_3^2 - 110 \end{bmatrix} \le 0\\
&\begin{bmatrix} -9.300961 - 0.0012547x_1x_3 - 0.0047026x_3x_5\\ -0.0019085x_3x_4 + 20 \end{bmatrix} \le 0\\
&\begin{bmatrix} 9.300961 + 0.0012547x_1x_3 + 0.0047026x_3x_5\\ +0.0019085x_3x_4 - 25 \end{bmatrix} \le 0\\
&78 \le x_1 \le 100, 33 \le x_2 \le 45,\\
&27 \le x_3 \le 45, 27 \le x_4 \le 45,\\
&27 \le x_5 \le 45
\end{aligned}$$

## APPENDIX II. ENGINEERING DESIGN PROBLEMS

P1: Optimal design of a small propeller-type aircraft engine speed reducer for by minimizing weight

$$\text{Min}\begin{bmatrix}0.7854x_1x_2^2\left(-43.0934+14.933x_3+3.3333x_3^2\right)\\ +7.477\left(x_6^3+x_7^3\right)-1.508\left(x_6^2+x_7^2\right)\\ +0.7854\left(x_4x_6^2+x_5x_7^2\right)\end{bmatrix}$$

$$\text{Subject to: }\frac{27}{x_1x_2^2x_3}-1\le 0,$$
$$\frac{3947.5}{x_1x_2^2x_3^2}-1\le 0,$$
$$\frac{1.93x_4^3}{x_2x_6^4x_3}-1\le 0,$$
$$\frac{1.93x_5^3}{x_2x_3x_7^4}-1\le 0,$$
$$\frac{\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2+16900000}}{110x_6^3}-1\le 0,$$
$$\frac{\sqrt{\left(\frac{745x_5}{x_2x_3}\right)^2+15750000}}{85x_7^3}-1\le 0,$$
$$\frac{x_1}{12x_2}-1\le 0,\ \frac{x_2x_3}{40}-1\le 0,$$
$$\frac{5x_2}{x_1}-1\le 0,\ \frac{1.9+1.5x_6}{x_4}-1\le 0,$$
$$\frac{1.9+1.1x_7}{x_5}-1\le 0,$$
$$2.6\le x_1\le 3.6,$$
$$0.7\le x_2\le 0.8,$$
$$17\le x_2\le 28,$$
$$7.3\le x_4\le 8.3,$$
$$7.8\le x_5\le 8.3,$$
$$2.9\le x_6\le 3.9,\quad 5\le x_7\le 5.5.$$

P2: Optimal design of a pressure vessel by minimizing overall costs, including material costs, shaping and welding.

$$\text{Min}\left[19.84x_1^2x_3+1.7781x_2x_3^2+3.1661x_1^2x_4+0.6224x_1x_3x_4\right]$$
$$\text{Subject to: }0.0193x_3-x_1\le 0,$$
$$0.00954x_3-x_2\le 0,$$
$$x_4-240\le 0,$$
$$-\pi x_3^2x_4-\frac{4}{3}\pi x_3^3+1296000\le 0,$$
$$0.5\le x_1\le 1,$$
$$0.3\le x_2\le 0.5,$$
$$40\le x_3\le 50,$$
$$170\le x_4\le 240.$$

P3: Optimal design of a disc brake

$$\text{Min}\left[4.9\left(-x_1^2+x_2^2\right)(x_4-1)/100000\right]$$
$$\text{Subject to: }-30+2.5(x_4+1)\le 0$$
$$20-(-x_1+x_2)\le 0$$
$$\frac{x_3}{3.14\left(-x_1^2+x_2^2\right)}-0.4\le 0$$
$$\frac{980000\left(-x_1^2+x_2^2\right)}{x_3x_4\left(-x_1^3+x_2^3\right)}\le 32,$$
$$900-\frac{0.0266x_3x_4\left(-x_1^3+x_2^3\right)}{\left(-x_1^2+x_2^2\right)}-1\le 0$$
$$\frac{0.00222x_3\left(-x_1^3+x_2^3\right)}{\left(-x_1^2+x_2^2\right)^2}-1\le 0$$
$$55\le x_1\le 80,$$
$$75\le x_2\le 110,$$
$$1000\le x_3\le 3000,$$
$$2\le x_4\le 20.$$

P4: Design of three-bar truss

$$\text{Min}\left[2\sqrt{2}\,x_1+x_2\right]L$$
$$\text{Subject to: }\frac{1}{x_1+\sqrt{2}x_2}p-\sigma\le 0$$
$$\frac{x_2}{2x_1x_2+\sqrt{2}x_1^2+}p-\sigma\le 0$$
$$\frac{\sqrt{2}\,x_1+x_2}{2x_1x_2+\sqrt{2}\,x_1^2}p-\sigma\le 0$$
$$0.1\le x_1\le 1,\ 0.1\le x_2\le 1;$$
$$L=100\text{cm},$$
$$\sigma=2\text{kN/cm}^2,$$
$$p=2\text{kN/cm}^2$$

P5: Cost Minimization of a Transformer Design

$$\text{Min}\begin{bmatrix}0.0607x_1x_4x_5^2\left(x_1+x_2+x_3\right)\\ +0.0187x_2x_3\left(x_1+1.57x_2+x_4\right)\\ +0.0204x_1x_4\left(x_1+x_2+x_3\right)\\ +0.0437x_2x_3x_6^2\left(x_1+1.57x_2+x_4\right)\end{bmatrix}$$
$$\text{Subject to: }0.001x_1x_2x_3x_4x_5x_6-2.07\ge 0,$$
$$\left[1-0.00058x_2x_3x_6^2\left(x_1+1.57x_2+x_4\right)-0.00062x_1x_4x_5^2\left(x_1+x_2+x_3\right)\right]\ge 0,$$
$$x_t\ge 0,\quad t=1,\ldots.,6$$

P6: optimal design of chemical reactors as Geometric programming problem

$$\text{Min}\left[-x_1-x_2+0.4x_1^{0.67}x_7^{-0.67}+\frac{0.4x_2^{0.67}}{x_8^{0.67}}+10\right]$$
$$\text{Subject to: }0.1x_1+0.0588x_5x_7\le 1,$$
$$0.1x_1+0.1x_2+0.0588x_6x_8\le 1,$$
$$\frac{4x_3}{x_5}+\frac{2}{x_3^{0.71}x_5}+\frac{0.0588x_7}{x_3^{1.3}}\le 1$$
$$\frac{4x_3}{x_6}+\frac{2}{x_4^{0.71}x_6}+\frac{0.0588x_8}{x_4^{1.3}}\le 1$$
$$0.1\le x_t\le 10,\quad t=1,\ldots 8$$

## ACKNOWLEDGMENT

The authors would like to thank the referees for their helpful feedback and suggestions, which have helped to strengthen the consistency of the arguments and the structure of the article.

## REFERENCES

[1] S. S. Rao, *Engineering Optimization: Theory and Practice*. Hoboken, NJ, USA: Wiley, 2019.

[2] M. W. Girard, G. Gour, and S. Friedland, "On convex optimization problems in quantum information theory," *J. Phys. A, Math. Theor.*, vol. 47, no. 50, Nov. 2014, Art. no. 505302.

[3] A. Ponsich, A. L. Jaimes, and C. A. C. Coello, "A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications," *IEEE Trans. Evol. Comput.*, vol. 17, no. 3, pp. 321–344, Jun. 2013.

[4] P. C. S. Rao, P. K. Jana, and H. Banka, "A particle swarm optimization based energy efficient cluster head selection algorithm for wireless sensor networks," *Wireless Netw.*, vol. 23, no. 7, pp. 2005–2020, Oct. 2017.

[5] A. K. Hartmann and H. Rieger, *Optimization Algorithms in Physics*, vol. 2. Berlin, Germany: Wiley-Vch, 2002.

[6] T. Weise, *Global Optimization Algorithms-Theory and Application*. Kreuth, Germany: Self-Published Thomas Weise, 2009. [Online]. Available: http://www.it-weise.de/

[7] P. Collet and J. P. Rennard, "Stochastic optimization algorithms," in *Intelligent Information Technologies: Concepts, Methodologies, Tools, and Applications*. Hershey, PA, USA: IGI Global, 2008, pp. 1121–1137.

[8] S. Droste, T. Jansen, and I. Wegener, "Upper and lower bounds for randomized search heuristics in black-box optimization," *Theory Comput. Syst.*, vol. 39, no. 4, pp. 525–544, Jul. 2006.

[9] M. A. El-Shorbagy, A. Y. Ayoub, A. A. Mousa, and I. M. El-Desoky, "An enhanced genetic algorithm with new mutation for cluster analysis," *Comput. Statist.*, vol. 34, no. 3, pp. 1355–1392, Sep. 2019.

[10] S. Belaqziz, S. Mangiarotti, M. Le Page, S. Khabba, S. Er-Raki, T. Agouti, L. Drapeau, M. H. Kharrou, M. El Adnani, and L. Jarlan, "Irrigation scheduling of a classical gravity network based on the Covariance Matrix Adaptation–Evolutionary Strategy algorithm," *Comput. Electron. Agricult.*, vol. 102, pp. 64–72, Mar. 2014.

[11] Z. W. Geem and Y. Yoon, "Harmony search optimization of renewable energy charging with energy storage system," *Int. J. Electr. Power Energy Syst.*, vol. 86, pp. 120–126, Mar. 2017.

[12] H. Ouyang, X. Kong, B. Hu, Z. Li, and G. Liu, "Competition harmony search algorithm with dimension selection for continuous optimization probems," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2018, pp. 6032–6037.

[13] A. W. Mohamed, "An efficient modified differential evolution algorithm for solving constrained non-linear integer and mixed-integer global optimization problems," *Int. J. Mach. Learn. Cybern.*, vol. 8, no. 3, pp. 989–1007, Jun. 2017.

[14] M. A. E. Shorbagy and A. A. Mousa, "Chaotic particle swarm optimization for imprecise combined economic and emission dispatch problem," *Rev. Inf. Eng. Appl.*, vol. 4, no. 1, pp. 20–35, 2017.

[15] S. Xu, Y. Wang, and P. Lu, "Improved imperialist competitive algorithm with mutation operator for continuous optimization problems," *Neural Comput. Appl.*, vol. 28, no. 7, pp. 1667–1682, Jul. 2017.

[16] L. Wei, Z. Zhang, D. Zhang, and S. C. H. Leung, "A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints," *Eur. J. Oper. Res.*, vol. 265, no. 3, pp. 843–859, Mar. 2018.

[17] I. Aljarah, H. Faris, and S. Mirjalili, "Optimizing connection weights in neural networks using the whale optimization algorithm," *Soft Comput.*, vol. 22, no. 1, pp. 1–15, Jan. 2018.

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

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

[20] M. A. Elsisy, D. A. Hammad, and M. A. El-Shorbagy, "Solving interval quadratic programming problems by using the numerical method and swarm algorithms," *Complexity*, vol. 2020, Sep. 2020, Art. no. 6105952.

[21] K. Lenin, B. R. Reddy, and M. S. Kalavathi, "Modified monkey optimization algorithm for solving optimal reactive power dispatch problem," *Indonesian J. Electr. Eng. Informat. (IJEEI)*, vol. 3, no. 2, pp. 55–62, Jun. 2015.

[22] M. Kiani-Moghaddam, M. Shivaie, and P. D. Weinsier, "Two-level multi-dimensional enhanced melody search algorithm for dynamic planning of MV open-loop distribution networks," in *Classical and Recent Aspects of Power System Optimization*. New York, NY, USA: Academic, 2018, pp. 285–330.

[23] B. He, L. Che, and C. Liu, "Novel hybrid shuffled frog leaping and differential evolution algorithm," *Jisuanji Gongcheng yu Yingyong(Comput. Eng. Appl.)*, vol. 47, no. 18, pp. 4–8, 2011.

[24] Y. Zhou, J. Wang, S. Gao, X. Yang, and J. Yin, "Improving artificial bee colony algorithm with historical archive," in *Bio-inspired Computing–Theories and Applications*. Singapore: Springer, 2016, pp. 185–190.

[25] P. Abrol and S. Gupta, "Social spider foraging-based optimal resource management approach for future cloud," *J. Supercomput.*, vol. 76, no. 3, pp. 1880–1902, Mar. 2020.

[26] H. Xu, X. Liu, and J. Su, "An improved grey wolf optimizer algorithm integrated with Cuckoo Search," in *Proc. 9th IEEE Int. Conf. Intell. Data Acquisition Adv. Comput. Syst., Technol. Appl. (IDAACS)*, vol. 1, Sep. 2017, pp. 490–493.

[27] K. L. Du and M. N. S. Swamy, "Ant colony optimization," in *Search and Optimization by Metaheuristics*. Cham, Switzerland: Birkhäuser, 2016, pp. 191–199.

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

[29] J. Batista Monteiro Filho, I. Maria Carneiro de Albuquerque, and F. Buarque de Lima Neto, "Fish school search algorithm for constrained optimization," 2017, *arXiv:1707.06169*. [Online]. Available: http://arxiv.org/abs/1707.06169

[30] A. Chakri, R. Khelif, M. Benouaret, and X.-S. Yang, "New directional bat algorithm for continuous optimization problems," *Expert Syst. Appl.*, vol. 69, pp. 159–175, Mar. 2017.

[31] M. Shehab, A. T. Khader, M. Laouchedi, and O. A. Alomari, "Hybridizing cuckoo search algorithm with bat algorithm for global numerical optimization," *J. Supercomput.*, vol. 75, no. 5, pp. 2395–2422, May 2019.

[32] H. H. Hoos and T. Stützle, *Stochastic Local Search: Foundations and Applications*. Amsterdam, The Netherlands: Elsevier, 2004.

[33] R. Goel and R. Maini, "A hybrid of ant colony and firefly algorithms (HAFA) for solving vehicle routing problems," *J. Comput. Sci.*, vol. 25, pp. 28–37, Mar. 2018.

[34] S. Nasr, M. El-Shorbagy, I. El-Desoky, Z. Hendawy, and A. Mousa, "Hybrid genetic algorithm for constrained nonlinear optimization problems," *Brit. J. Math. Comput. Sci.*, vol. 7, no. 6, pp. 466–480, Jan. 2015.

[35] S. S. Jadon, R. Tiwari, H. Sharma, and J. C. Bansal, "Hybrid artificial bee colony algorithm with differential evolution," *Appl. Soft Comput.*, vol. 58, pp. 11–24, Sep. 2017.

[36] B. Turanoğlu and G. Akkaya, "A new hybrid heuristic algorithm based on bacterial foraging optimization for the dynamic facility layout problem," *Expert Syst. Appl.*, vol. 98, pp. 93–104, May 2018.

[37] V. I. Skoullis, I. X. Tassopoulos, and G. N. Beligiannis, "Solving the high school timetabling problem using a hybrid cat swarm optimization based algorithm," *Appl. Soft Comput.*, vol. 52, pp. 277–289, Mar. 2017.

[38] L. M. Abualigah, A. T. Khader, E. S. Hanandeh, and A. H. Gandomi, "A novel hybridization strategy for krill herd algorithm applied to clustering techniques," *Appl. Soft Comput.*, vol. 60, pp. 423–435, Nov. 2017.

[39] M. A. M. Shaheen, H. M. Hasanien, and A. Alkuhayli, "A novel hybrid GWO-PSO optimization technique for optimal reactive power dispatch problem solution," *Ain Shams Eng. J.*, to be published, doi: 10.1016/j.asej.2020.07.011.

[40] M. N. Gilvaei, H. Jafari, M. J. Ghadi, and L. Li, "A novel hybrid optimization approach for reactive power dispatch problem considering voltage stability index," *Eng. Appl. Artif. Intell.*, vol. 96, Nov. 2020, Art. no. 103963.

[41] A. A. Mousa, M. A. El-Shorbagy, and M. A. Farag, "K-means-Clustering based evolutionary algorithm for multi-objective resource allocation problems," *Appl. Math. Inf. Sci.*, vol. 11, no. 6, pp. 1681–1692, Nov. 2017.

[42] M. A. Farag, M. A. El-Shorbagy, I. M. El-Desoky, A. A. El-Sawy, and A. A. Mousa, ''Binary-real coded genetic algorithm based $k$-means clustering for unit commitment problem,'' *Appl. Math.*, vol. 6, no. 11, pp. 1873–1890, 2015.

[43] M. Hussein, A. EL-Sawy, E.-S. Zaki, and A. Mousa, ''Genetic algorithm and rough sets based hybrid approach for economic environmental dispatch of power systems,'' *Brit. J. Math. Comput. Sci.*, vol. 4, no. 20, pp. 2978–2999, Jan. 2014.

[44] G. Renner and A. Ekárt, ''Genetic algorithms in computer aided design,'' *Comput.-Aided Des.*, vol. 35, no. 8, pp. 709–726, Jul. 2003.

[45] F. Zhao, Z. Yao, J. Luan, and X. Song, ''A novel fused optimization algorithm of genetic algorithm and ant colony optimization,'' *Math. Problems Eng.*, vol. 2016, Aug. 2016, Art. no. 2167413.

[46] S. Iqbal and M. T. Hoque, ''HGRGA: A scalable genetic algorithm using homologous gene schema replacement,'' *Swarm Evol. Comput.*, vol. 34, pp. 33–49, Jun. 2017.

[47] M. Lozano, F. Herrera, and J. Cano, ''Replacement strategies to preserve useful diversity in steady-state genetic algorithms,'' *Inf. Sci.*, vol. 178, no. 23, pp. 4421–4433, Dec. 2008.

[48] F. Altiparmak, M. Gen, L. Lin, and I. Karaoglan, ''A steady-state genetic algorithm for multi-product supply chain network design,'' *Comput. Ind. Eng.*, vol. 56, no. 2, pp. 521–537, Mar. 2009.

[49] N. Verbiest, S. Vluymans, C. Cornelis, N. García-Pedrajas, and Y. Saeys, ''Improving nearest neighbor classification using ensembles of evolutionary generated prototype subsets,'' *Appl. Soft Comput.*, vol. 44, pp. 75–88, Jul. 2016.

[50] H. Nenavath, D. R. Kumar Jatoth, and D. S. Das, ''A synergy of the sine-cosine algorithm and particle swarm optimizer for improved global optimization and object tracking,'' *Swarm Evol. Comput.*, vol. 43, pp. 1–30, Dec. 2018.

[51] E. N. Lorenz, *The Essence of Chaos*. Seattle, WA, USA: Univ. Washington Press, 1995.

[52] J. Mathew, P. Patra, D. K. Pradhan, and A. J. Kuttyamma, ''Eco-friendly computing and communication systems,'' in *Proc. Int. Conf. Eco-Friendly Comput. Commun. Syst. (ICECCS)*, vol. 305. Kochi, India: Springer, 2012,

[53] F. Sorrentino and P. DeLellis, ''Estimation of communication-delays through adaptive synchronization of chaos,'' *Chaos, Solitons Fractals*, vol. 45, no. 1, pp. 35–46, Jan. 2012.

[54] L. Guo, M. Hu, Z. Xu, and A. Hu, ''Synchronization and chaos control by quorum sensing mechanism,'' *Nonlinear Dyn.*, vol. 73, no. 3, pp. 1253–1269, Aug. 2013.

[55] Y. Tang, Z. Wang, W. K. Wong, J. Kurths, and J.-A. Fang, ''Multiobjective synchronization of coupled systems,'' *Chaos: Interdiscipl. J. Nonlinear Sci.*, vol. 21, no. 2, Jun. 2011, Art. no. 025114.

[56] A. Baykasoğlu and F. B. Ozsoydan, ''Adaptive firefly algorithm with chaos for mechanical design optimization problems,'' *Appl. Soft Comput.*, vol. 36, pp. 152–164, Nov. 2015.

[57] L. Huang, S. Ding, S. Yu, J. Wang, and K. Lu, ''Chaos-enhanced cuckoo search optimization algorithms for global optimization,'' *Appl. Math. Model.*, vol. 40, nos. 5–6, pp. 3860–3875, Mar. 2016.

[58] S. Talatahari and M. Azizi, ''Optimization of constrained mathematical and engineering design problems using chaos game optimization,'' *Comput. Ind. Eng.*, vol. 145, Jul. 2020, Art. no. 106560.

[59] S. Dwivedi, M. Vardhan, and S. Tripathi, ''An effect of chaos grasshopper optimization algorithm for protection of network infrastructure,'' *Comput. Netw.*, vol. 176, Jul. 2020, Art. no. 107251.

[60] A. Megahed, A. Shaker, and K. R. Mahmoud, ''Optimal design of planar terahertz yagi-uda antenna using adaptive chaos optimization algorithm,'' *Optik*, vol. 191, pp. 146–151, Aug. 2019.

[61] H. Yu, N. Zhao, P. Wang, H. Chen, and C. Li, ''Chaos-enhanced synchronized bat optimizer,'' *Appl. Math. Model.*, vol. 77, pp. 1201–1215, Jan. 2020.

[62] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger, and S. Tiwari, ''Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization,'' Kanpur Genetic Algorithms Lab. (KanGAL), Indian Inst. Technol. Kanpur, Kanpur, India, Tech. Rep. 2005005, 2005, no. 2005.

[63] K. Sedlaczek and P. Eberhard, ''Constrained particle swarm optimization of mechanical systems,'' in *Proc. 6th WCSMO*, vol. 80. Rio de Janeiro, Brazil, 2005, pp. 1–10. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.60.4525&rep=rep1&type=pdf

[64] T. P. Runarsson and X. Yao, ''Stochastic ranking for constrained evolutionary optimization,'' *IEEE Trans. Evol. Comput.*, vol. 4, no. 3, pp. 284–294, Sep. 2000. [Online]. Available: https://ieeexplore.ieee.org/document/873238

[65] H. Lu and W. Chen, ''Self-adaptive velocity particle swarm optimization for solving constrained optimization problems,'' *J. Global Optim.*, vol. 41, no. 3, pp. 427–445, Jul. 2008. [Online]. Available: https://link.springer.com/article/10.1007/s10898-007-9255-9#citeas

[66] N. Andrei and N. Andrei, *Nonlinear Optimization Applications Using the GAMS Technology*. New York, NY, USA: Springer, 2013. [Online]. Available: https://link.springer.com/book/10.1007/978-1-4614-6797-7

[67] R. P. Beausoleil, ''Solving engineering optimization problems with tabu/scatter search,'' *Revista de Matemática: Teoría y Aplicaciones*, vol. 24, no. 1, pp. 157–188, 2017. [Online]. Available: https://www.scielo.sa.cr/pdf/rmta/v24n1/1409-2433-rmta-24-01-00157.pdf

[68] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Optimization*, vol. 7. Hoboken, NJ, USA: Wiley, 2000.

[69] S. Martorell, S. Carlos, A. Sánchez, and V. Serradell, ''Constrained optimization of test intervals using a steady-state genetic algorithm,'' *Rel. Eng. Syst. Saf.*, vol. 67, no. 3, pp. 215–232, Mar. 2000.

[70] S. Ekiz, P. Erdoğmuş, and B. Özgür, ''Solving constrained optimization problems with sine-cosine algorithm,'' *Periodicals Eng. Natural Sci. (PEN)*, vol. 5, no. 3, pp. 378–386, 2017.

[71] M. A. El-Shorbagy, A. A. Mousa, and S. M. Nasr, ''A chaos-based evolutionary algorithm for general nonlinear programming problems,'' *Chaos, Solitons Fractals*, vol. 85, pp. 8–21, Apr. 2016.

[72] Y. Li, S. Deng, and D. Xiao, ''A novel hash algorithm construction based on chaotic neural network,'' *Neural Comput. Appl.*, vol. 20, no. 1, pp. 133–141, Feb. 2011.

[73] M. S. Tavazoei and M. Haeri, ''Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms,'' *Appl. Math. Comput.*, vol. 187, no. 2, pp. 1076–1085, Apr. 2007.

[74] R. L. Devaney, *An Introduction to Chaotic Dynamical Systems*, 2nd ed. Reading, MA, USA: Addison-Wesley, 1989.

[75] M. M. Salama, A. A. Mousa, and M. A. El-Shorbay, ''A combined genetic algorithms-modified local search scheme for nonlinear programming problems,'' *Global J. Pure Appl. Math.*, vol. 13, no. 9, pp. 4765–4786, 2017.

[76] H.-J. Wang, J.-Q. E, and F.-Q. Deng, ''A novel adaptive mutative scale optimization algorithm based on chaos genetic method and its optimization efficiency evaluation,'' *J. Central South Univ.*, vol. 19, no. 9, pp. 2554–2560, Sep. 2012.

[77] 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.

[78] C. Fernandes and A. Rosa, ''A study on non-random mating and varying population size in genetic algorithms using a royal road function,'' in *Proc. Congr. Evol. Comput.*, May 2001, pp. 60–66.

[79] L. J. Eshelman, ''The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination,'' *Found. Genetic Algorithms*, vol. 1, pp. 265–283, 1991.

[80] F. Herrera, M. Lozano, and D. Molina, ''Continuous scatter search: An analysis of the integration of some combination methods and improvement strategies,'' *Eur. J. Oper. Res.*, vol. 169, no. 2, pp. 450–476, Mar. 2006.

[81] M. Laguna and R. Marti, *Scatter Search: Methodology and Implementations in C*. New York, NY, USA: Springer, 2012.

[82] A. Auger and N. Hansen, ''A restart CMA evolution strategy with increasing population size,'' in *Proc. IEEE Congr. Evol. Comput.*, vol. 2, Sep. 2005, pp. 1769–1776.

[83] J. Kennedy and R. Eberhart, ''Particle swarm optimization,'' in *Proc. Int. Conf. Neural Netw.*, vol. 4, Nov./Dec. 1995, pp. 1942–1948.

[84] A. K. Qin and P. N. Suganthan, ''Self-adaptive differential evolution algorithm for numerical optimization,'' in *Proc. IEEE Congr. Evol. Comput.*, vol. 2, 2005, pp. 1785–1791.

[85] K. Price, R. M. Storn and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Berlin, Germany: Springer-Verlag, 2006.

[86] M. M. Efren, *Constraint-Handling in Evolutionary Optimization*. Berlin, Germany: Springer-Verlag, 2009.

[87] Y. Zhou, G. Zhou, and J. Zhang, ''A hybrid glowworm swarm optimization algorithm for constrained engineering design problems,'' *Appl. Math. Inf. Sci.*, vol. 7, no. 1, pp. 379–388, Jan. 2013.

**A. A. MOUSA** received the M.Sc. and Ph.D. degrees in engineering mathematics from Menoufia University, Egypt, in 2003 and 2006, respectively. He is currently a Full Professor of engineering mathematics with the Department of Basic Engineering Sciences, Faculty of Engineering, Menoufia University, and the Department of Mathematics and statistics, Faculty of Science, Taif University, Saudi Arabia. He is the author of more than 90 scientific articles and textbooks in refereed journals and international conferences. His research interests include theory of optimization and its application, numerical analysis, mathematical programming, metaheuristics, and computational intelligence. He has served as an editorial board member and a referee for many reputed international journals.

**M. A. EL-SHORBAGY** was born in Menoufia, Egypt, in April 1982. He received the B.Sc. degree in electrical engineering, the M.Sc. degree in engineering mathematics, and the Ph.D. degree in engineering mathematics from the Faculty of Engineering, Menofiya University, Shebin El-Kom, Egypt, in 2004, 2010, and 2013, respectively. He is currently an Associate Professor with the Department of Mathematics, College of Science and Humanities in Al-Kharj, Prince Sattam bin Abdulaziz University, Al-Kharj, Saudi Arabia. His current research interests include swarm intelligence, artificial intelligence, evolutionary algorithms, numerical optimization and engineering optimization, cluster analysis, and solving engineering applications.

**M. A. FARAG** was born in Shebin El-Kom, Menoufia, Egypt, in 1990. She received the B.Sc. degree in electrical engineering, the M.Sc. degree in engineering mathematics, and the Ph.D. degree in engineering mathematics from the Faculty of Engineering, Menoufia University, Shebin El-Kom, in 2011, 2016, and 2020, respectively. She is currently an Associate Professor with the Department of Basic Engineering Science, Menofiya University. Her research interests include optimization algorithms, chaotic systems, and solving engineering applications.