IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received February 5, 2022, accepted February 17, 2022, date of publication February 21, 2022, date of current version March 7, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3153038

Grasshopper Optimization Algorithm With Crossover Operators for Feature Selection and Solving Engineering Problems

AHMED A. EWEES[1], (Senior Member, IEEE), MARWA A. GAHEEN[1], ZAHER MUNDHER YASEEN[2,3,4], AND RANIA M. GHONIEM[5,6]

[1]Department of Computer, Damietta University, Damietta 34511, Egypt
[2]Adjunct Research Fellow, USQ's Advanced Data Analytics Research Group, School of Mathematics Physics and Computing, University of Southern Queensland, Darling Heights, QLD 4350, Australia
[3]New Era and Development in Civil Engineering Research Group, Scientific Research Center, Al-Ayen University, Thi-Qar 64001, Iraq
[4]Institute for Big Data Analytics and Artificial Intelligence (IBDAAI), Kompleks Al-Khawarizmi, Universiti Teknologi MARA, Shah Alam, Selangor 40450, Malaysia
[5]Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh 11671, Saudi Arabia
[6]Department of Computer, Mansoura University, Mansoura 35516, Egypt

Corresponding author: Ahmed A. Ewees (ewees@du.edu.eg)

ABSTRACT Feature selection (FS) is an irreplaceable phase that makes data mining more efficient. It effectively enhances the implementation and decreases the computational problem of learning models. The comprehensive and greedy algorithms are not suitable for the present growing number of features when detecting the optimal subset. Thus, swarm intelligence algorithms (SI) are becoming more common in dealing with FS problems. The grasshopper optimizer algorithm (GOA) represents a new SI; it showed good performance in different fields. Another promising nature-inspired algorithm is a salp swarm algorithm, denoted as SSA, an SI used to tackle optimization issues. In this paper, two phases are applied to propose a new method using crossover-salp swarm with grasshopper optimization algorithm (cSG). In this method, the crossover operators are used to maintain the population of the SSA then the improved SSA is used as a local search to boost the exploration phase of the GOA. Subsequently, this improvement prevents the cSG from premature convergence, high computation time, and being trapped in local minimum. To confirm the effectiveness of proposed cSG method, it is evaluated in different optimizations problems. Eventually, the obtained results are compared to a number of well-known algorithms over global optimization, feature selection datasets, and six real-engineering problems. Experimental results point out that the cSG is superior in solving different optimization problems due to the integration of crossover operators and SSA which enhances its performance and flexibility.

INDEX TERMS Grasshopper optimization algorithm, crossover operator, salp swarm algorithm, optimization problems, feature selection, engineering problems.

I. INTRODUCTION

Recently, feature selection (FS) has acquired much attention from researchers working in machine learning and data mining domain. However, the increase in data size and dimensions causes different problems, such as the appearance of noisy, inconvenient and redundant data. Hence, it is hard to find optimal group of features and remove redundant ones. Dealing with datasets, some features are insignificant in the presence of irrelevancy and redundancy. Therefore, considering such features is not valuable and usually affects the classification accuracy [1]. FS attempts to enhance classification performance through selecting from the original enormous range of features just a small subset of suitable features [2]. The extraction of redundant and irrelevant features will, thus, minimize the data dimensionality, enhance the learning process by simplifying model learning and enhancing performance [3], [4]. Other benefits of FS are that: reducing overfitting minimizes redundant data, decreases chances for noise-based rulemaking, enhances precision and minimizes

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

misleading data which means enhancing the precision of modeling. Furthermore, decreases training time, minimizes data points, reduces complexity of the algorithms, and accelerates the algorithm's training [5], [6].

During the past few years, meta-heuristic algorithms have become very popular. They have been widely applied to find solutions for different complex problems in computer science and engineering. They are essentially utilized to obtain the optimal solution by getting different optimal values for producing a candidate value to completely solve the underlying issue [7], [8]. Generally, meta-heuristic optimization methods consider the optimal value by decreasing or increasing the objective function to get the optimal decision [9]. Most of them are inspired by evolution laws and intelligent behaviors of the natural animals [10]. For example, particle swarm optimizer (PSO) [11], monarch butterfly optimizer (MBO) [12], whale optimization (WO) [13], Grasshopper Optimization (GO) [14], salp swarm algorithm (SSA) [15], moth search (MS) [16], Harris hawks optimizer (HHO) [17], slime mould algorithm (SMA) [18], Gradient-based optimization (GBO) [19], Heap-based optimization (HBO) [20], hunger games search algorithm (HGS) [21], RUNge Kutta optimization (RUN) [22], as well as Colony Predation (CP) Algorithm [23].

These strong meta-heuristic algorithmic methods have been suggested, improved, or hybridized to solve different problems in several fields, such as, Mozaffari *et al.* [24] introduced a new inclined planes system optimization (IPO) algorithm to address optimization problems. Mozaffari & Lee [25] applied PSO to detect the optimal multilevel thresholds in image segmentation. Faris *et al.* [26] proposed an enhanced MBO with the aim of unconstrained global search as well as neural network training. Alweshah *et al.* [27] applied MBO with k-nearest neighbor (KNN) classifier for tackling feature selection problems. For image segmentation issues the MBO [28] was applied at multiple threshold values. Strumberger *et al.* [29] merged MSA with the algorithm of artificial bee colony, abbreviated as ABC, to handle constrained optimization issues. Elgamal *et al.*, [2] improved HHO by the chaotic maps and simulated annealing (SA) algorithm and apply it to address feature selection issues in the medical field. Kundu & Garg [30] improved HHO by applying enhanced teaching–learning-based optimization to address engineering design and numerical optimization issues. Premkumar *et al.* [31] introduced GBO to address the multiobjective optimization problems. Helmi *et al.* [32] merged GBO with GWO for human activity recognition. To boost the optimization performance of SMA the authors of [33] hybridized SMA with inertia weight parameter, as well as reverse learning strategy for the best operation of cascade hydropower stations. Moreover, in [34] the SMA was hybridized with WOA adopting thresholding technique to address the image segmentation problem. Abdel-Basset *et al.* [35] improved HBO algorithms for parameters estimation of proton exchange membrane fuel cells model. Onay & Aydemir [36] proposed a chaotic HGS for handling real engineering problems and global optimizations. Devi *et al.* [37] proposed a binary HGSO algorithm relied on V- and S-shaped transfer functions to handle feature selection problems. Ahmadianfar *et al.* [22] proposed RUN optimizer to handle global optimization problems. Shi *et al.* [38] applied CPA with kernel extreme learning machine (KELM) for detecting COVID-19 from view of biochemical indexes. According to the good results of the previous studies in solving different types of problems, the improved and hybrid algorithms can effectively find the best solution compared to the ordinary algorithm. Consequently, in this work, the crossover operators are combined with SSA to enhance the search capability for the original GOA.

The grasshopper optimization algorithm (GOA) is a new algorithmic method inspired from lives of grasshoppers proposed in [14]. The search method of nature-inspired optimization algorithms contains two phases that are: exploration and exploitation, which appear during the search for food [39], [40]. GOA has many merits: simplicity, robustness, few parameters, ease in implementation, and strong potential exploratory ability [41]. It has been utilized for handling many optimization problems in various applications as feature selection [42], global optimization [43], power system stability [44], numerical optimization [45], and skin color segmentation [46]. Despite the advantages of GOA, it suffers from some shortcomings in exploiting the search space, premature convergence in a number of complex optimization techniques, and can get stuck into local optima. To overcome that, many improvements were proposed. Ewees *et al.* [47] introduced an opposition-based GOA for engineering problems and benchmark optimization functions. Bala *et al.* [48] presented an enhanced GOA for fault prediction in airplane engines. Huang *et al.* [49] proposed an improved GOA for enhancing the parameters of power filter (HAPF). Li *et al.* [50] integrated GOA with differential evolution (DE) to detect color differences of dyed fabrics. Motlagh & Foroud [51] proposed a hybrid GOA with SVM to recognize power quality disturbances.

Another promising nature-inspired algorithm is called the salp swarm algorithm, abbreviated as SSA, which is a new swarm intelligence algorithm proposed by [52]. It is based on gathering behavior, sailing the intellect of salp swarms [53]. SSA is simple to implement [54]. It confirmed its ability to settle large-and-small-scale issues [52]. SSA is characterized by its strength and flexible stochastic nature [15]. In recent years SSA has attracted attention of several researchers. One of the leading works on SSA was presented in [15] in which the authors applied a crossover operator with transfer functions to enhance SSA. In [55] the extreme learning machine (ELM) was hybridized with SSA to handle complex engineering problems. Another work on feature selection was conducted in [56], Sayed *et al.* suggested a new hybridization method relied on SSA and chaos theory to address feature selection issues. The outcomes of the suggested method showed that the CSSA is a good optimizer compared to several former methods. SSA was conducted together with k-NN in [57] to select sub-features and obtain higher classification reliability in optimizing chemical synthesis activities. Ismael *et al.* [58], applied SSA to investigate the issue of choosing the optimal conductor in a radiate distribution framework in Egypt. The acquired results showed the impact of the method in satisfying the thematic function and constraints. Moreover, SSA has been utilized to define the optimal controller parameters for the AVR system [59]. The simplicity of the SSA method achieved a high-quality set of optimal controller parameters. Xing & Jia [60] proposed an amended SSA to optimize GLCM multi-threshold picture segmentation algorithm. The findings showed that the suggested method achieved better segmentation results and can handle complicated picture segmentation functions. Hegazy *et al.* [54] suggested chaotic-SSA to handle feature selection issues in the wrapper mode for data classification.

The proposed method was validated on 27 datasets. The findings proved that the suggested approach successfully balanced the primary two aims of any meta-heuristic tactic: exploration and exploitation.

According to the characteristics of GOA and SSA, a hybrid salp swarm with grasshopper optimization algorithm (cSG) is suggested for improving the potential of GOA. Whereas, the GOA suffers from premature convergence, high computation time, and getting trapped in a local minimum. This paper combines GOA with SSA to benefit from the strong potential exploratory capability of GOA besides the strength, flexible stochastic nature, and fast convergence ability for SSA to effectively accelerate the search capability of cSG. Moreover, the crossover operators are used to maintain the population of the SSA and enhance the exploration phase. Subsequently, the improved SSA acts as a local search for the GOA.

The contribution of this paper can be summarized in the subsequent points:

- Integrate the crossover operators with SSA to improve the search ability of the original GOA. This version is called cSG method.
- Extensive experiments are carried out for validating the performance of GOA using CEC2017 functions and eleven well-known benchmarks for feature selection, then comparing its performance with eleven well-known optimizers to confirm its superiority.
- A comparative study is applied depending on other optimization algorithms for solving six popular engineering problems.

The forthcoming of this paper is arranged as follows: Section 2 shows a brief description of the used methods. Section 3 sheds light on the proposed cSG method. Section 4 provides the experiments and results. Whereas the last section concludes the paper and lists some of the feature works.

II. MATERIAL AND METHODS

This section gives details on the different stages of the SSA, the GOA algorithm and the crossover operator:

A. SALP SWARM ALGORITHM (SSA)

SSA is considered an optimization method. It was suggested by the authors of [52]. It imitates the existing salp chains which are: a swarm utilized to move and forage to get to the food source. Salp is distributed under the category of the Salpidae family. Also, the swarming salps are highly noticeable as they can construct cooperative chains through foraging actions in deep oceans. Such attitude makes salps gain more kinetic ability throughout tracking the food source [61]. The SSA method is based on the mechanism of swarming salps out of producing the salp chain, which assists SSA in relieving inertia to the native optima to several extents. However, SSA cannot constantly carry out an appropriate balance among exploration and exploitation stages. Subsequently, the premier method sometimes fails to acquire a high-quality, comprehensive optimum in several real-world issues [62]. The Salp chain is subdivided into two sets of salps: leader and followers. Accordingly, the first class represents the leader of a salp, whereas the second class is the followers. Leader salp leads the direction and movement of the swarm, whilst the followers benefit from other peers [63].

From this view, this behavior is transformed into a mathematical form to produce SSA. In SSA, the population is subdivided into two sets; the first set is the leader the second is the followers. The leader is set before the followers. To change the status of a set, the leader updates his status that can be denoted by,

$$x_j^i = \begin{cases} F_j + c_1(\big(ub_j - l??_j\big) \times c_2 + lb_j), & c_3 \leq 0 \\ F_j - c_1(\big(ub_j - lb_j\big) \times c_2 + lb_j), & c_3 > 0 \end{cases} \tag{1}$$

where the position can be defined by x_j^1, the boundaries of the upper and lower search area j-th represented by ub_j and lb_j, the target is denoted by F_j, c_2 and random parameters [0, 1] is defined by c_3 where c_1 value is determined as,

$$c_1 = 2e^{-(\frac{4t}{t_{max}})^2} \tag{2}$$

where, the highest loop number is represented by t_{max} and the latest loop is indicated by t.

The followers' status is updated relied on Eq. 9.

$$x_j^i = \frac{1}{2}(x_j^i + x_j^{i-1}) \tag{3}$$

where $i > 1$, and the i-th follower status indicated by x_j^i.

B. GRASSHOPPER OPTIMIZATION ALGORITHM (GOA)

GOA is a well-known optimization method in the last years. It was improved by the authors of [14]. It imitates the grasshopper insect's nature. It is a harmful roach that eats crops, which affects the output of agriculture. The life cycle of grasshoppers contains three periods: firstly, the egg period; then comes the larval period; and at last, the adult period. The larval period is described as a slow movement besides small steps, while in the adulthood period, the grasshopper can move abruptly and rapidly. GOA relies on the following tactic: grasshoppers can form swarms in the larval and adulthood periods together [64]. These swarms are permanently searching for a food source. In general, in the first stage, the grasshoppers' positions alter suddenly, while they are encouraged to proceed locally during the second stage. Grasshoppers essentially perform these two stages over the nymph and adulthood periods together, making grasshoppers the appropriate choice for mathematically resolving their swarming behavior as a robust optimization algorithm [39], [40]. And so, this attitude can be performed mathematically as,

$$x_i = S_i + G_i + A_i, \quad i = 1, 2, \ldots, N \tag{4}$$

where, the grasshopper status in *i-th* dimension indicated by x_i. The social interaction indicated by S_i, and can be determined as,

$$S_i = \sum_{\substack{j=1\\ i\neq j}}^{N} s(d_{ij})\hat{d}_{ij}, d_{ij} = \left|x_i - x_j\right|, \quad \hat{d}_{ij} = \frac{x_i - x_j}{d_{ij}} \tag{5}$$

where the distance among grasshoppers is denoted by d_{ij} and the unit vector among grasshoppers is defined by $\hat{d}_{ij}$. The parameter s can be determined as.

$$s\left(y\right) = fe^{\frac{-y}{l}} - e^{-y} \tag{6}$$

Here, the attractive length scale is represented as l and the attraction intensity is represented as$f$.

Besides, wind and gravity affect the small movements of the grasshoppers which can be determined as,

$$\begin{aligned}\text{Wind advection} &= A_i = u\hat{e}_w,\\ \text{Gravity force} &= G_i = -g\hat{e}_g\end{aligned} \tag{7}$$

where, a drift constant is represented by u and the unit vector of wind direction represented by $\hat{e}_w$, while g and $\hat{e}_g$ denote the constant of the gravity and the unit vector across earth's center, respectively.

Consequently, the status of the grasshoppers is reformed applying the next equation.

$$x_i^d = c\left(\sum_{\substack{j=1\\ i\neq j}}^{N} c\frac{u_d - l_d}{2}s(|x_j^d - x_i^d|)\frac{x_j - x_i}{d_{ij}}\right) + \hat{T}_d, \tag{8}$$

where the boundaries of the upper and lower search area are represented by u and l, and the optimal solution value is represented by $\hat{T}_d$. The problem dimension is denoted by D while the population size is denoted by N, and the c parameter is determined as,

$$c = c_{max} - t\frac{c_{max} - c_{min}}{t_{max}} \tag{9}$$

where c_{max} and c_{min} equal 1 and 0.0001, respectively, the highest loop number is defined by t_{max}, while, the actual loop is denoted by t.

C. CROSSOVER OPERATOR

Crossover is a genetic operator applied to change the chromosome or chromosome programming from one generation to another [65]. Crossover is sexual reproduction. Two chains are selected from the mating pool randomly to crossover for producing optimal offspring. The selected method relied on the Encoding Method. The crossover operator aims to select a middle to solve in a binary search space to imitate detecting a solution among two solutions. It transforms between two input vectors that have the same probability.

It can be categorized into three groups: the first is a single-point crossover in which a crossover point is chosen on the parent organism chain. All data behind that point in the organism chain, is exchanged between the two parent organisms. Positional Bias is used to characterize the strings. The second one is a two-point crossover which is a particular condition of an N-point Crossover tactic. Two points are selected randomly on the independent chromosomes, and the genetic substance is swapped at these points. The third one is a uniform crossover: every gene (bit) is chosen randomly from one of the identical chromosomes' genes of the parent [65].

The crossover between two superior decisions may not consistently yield an optimum or a good decision. The better the parents are, the more expectation; that the child will be good. If children are not good (poor decision) during selection, it will be rejected in the upcoming iteration.

III. PROPOSED METHOD

The proposed cSG method is detailed in this section. It stands for crossover SSA and GOA. In the proposed cSG the crossover operators are used to maintain the population of the SSA and enhance the exploration phase. Subsequently, the improved SSA is applied as a local search to the GOA. This improvement prevents the standard GOA version from premature convergence, the high computation time, besides trapping in local minima.

In the cSG the local search of the standard version of the GOA is supported by the enhanced SSA algorithm. The population of the SSA is improved by adding a new step to maintain the search domain; this step utilizes the crossover operator with the population of the SSA. Therefore, it selects two solutions to start the crossover operators; then it produces two new solutions, which helps maintain the search domain and avoid trapping in a local point.

The proposed cSG begins by providing a new random population (X) with (N) length and (D) dimension. This population contains the initial solutions of the problem. Each solution is evaluated using a fitness function. Eq. (10) shows the fitness function implemented in this study.

$$f_i = \gamma er_i + (1-\gamma)\left(\frac{ld}{d}\right) \tag{10}$$

where er_i defines the error produced by the classification step, (this work applies the KNN classifier); whereas, the second part of this equation refers to the selected features number. ld and d is selected features, and the whole number of all

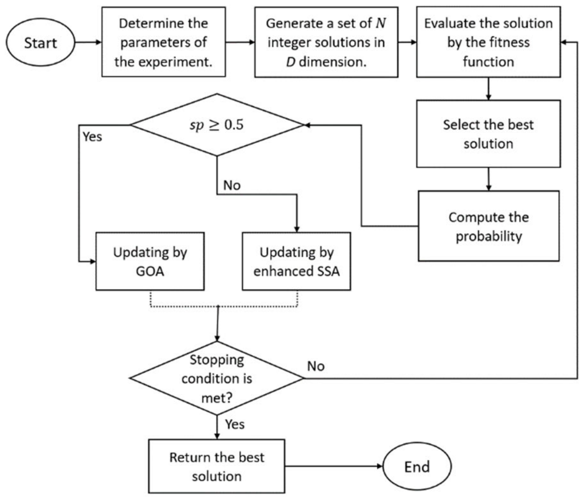

Start
Determine the parameters of the experiment.
Generate a set of N integer solutions in D dimension.
Evaluate the solution by the fitness function
Select the best solution
Compute the probability
sp ≥ 0.5
Yes
No
Updating by GOA
Updating by enhanced SSA
Stopping condition is met?
No
Yes
Return the best solution
End

FIGURE 1. **Phases of the proposed cSG method.**

features, respectively. The parameter ($\gamma \in [0, 1]$) is applied in order to balancing the selected features number and the classification error.

Furthermore, each solution is updated. This update is applied by both the SSA and GOA based on a variable generated by using a probability formula (sp) as described in the following equation:

$$sp_i = \frac{f_i}{\sum_{i=1}^{n} f_i} \tag{11}$$

Here, f_i denotes the last obtained fitness value produced by Eq. 10. The sp_i value determines the updating algorithm by checking its value; therefore, if $sp_i < 0.5$, the solution will be updated using GOA by Eqs. 8-9, otherwise the enhanced SSA will be used to update the solution using Eqs. 1-3. Later, each solution's quality is evaluated by using the fitness function (Eq. 10), and the best one is saved. This sequence is repeated until reaching the stop condition, which in this study set to 2500 fitness evaluations. The main phases of the proposed method are illustrated in Figure 1 and Algorithm 1.

The complexity of the cSG consists of the complexity of GOA, SSA and crossover operators as follows:

$O(cSG) = O(GOA) + (O(SSA) + O(Crossover))$ where, O(GOA) $= O(t(N^2 \times D + N \times C))$, $O(SSA) = O(t(D \times N + C \times N))$, and O(Crossover) $= O(t \times N)$; here, (t) is the iterations number, (N) is the solutions number, and (D) dimension. (C) denotes the cost of fitness function.

IV. EXPERIMENTS

The performance of suggested cSG method is assessed in this section by using four experiments. Accordingly, the first one evaluates the components of the proposed method. Whereas, the second one tests the performance of cSG using CEC2017 functions. The third experiment aims to test the cSG in solving the general feature selection problems. Whereas, the fourth one tests the performance of the cSG in solving six common engineering problems design. The implementation of the experiments was applied using "MATLAB 2014b" and "Windows 10" over "CPU Core i5" with "8GB of RAM". The compared methods and the parameter settings are listed in the next subsection.

Algorithm 1 Pseudocode of the Proposed Method cSG

```
Determine the number of dimension ($d$), solutions ($N$), and number of fitness function evaluation ( $stopCondition$)
Define the parameters for both SSA and GOA.
Generate the initial population $X$.
While ($t <= stopCondition$)
Compute the fitness value for each solution $X$ using Eq. (10).
Save the best solution based on the fitness value.
FOR ($i = 1$ to $N$)
Calculate the probability $sp$ for each solution using Eq. (11).
If ($sp >= 0.50$)
    Normalize the distance between the solutions in $X$ in the interval [1,4].
    Update $X$ using operators of GOA (as in Eqs. 8-9).
Else
If ($rand() < 0.25$)
    Select randomly values from a solution.
    Apply crossover to update the current solution.
End if
    Update $X$ using SSA equations as in Eqs. (1-3)
Enf if
End for
End while
Output the result.
```

A. PARAMETER SETTINGS

The proposed method is compared with eleven algorithms, namely salp swarm algorithm (SSA) [52], grasshopper optimization algorithm (GOA) [14], particle swarm optimization (PSO) [11], genetic algorithm (GA) [66], multi-verse optimizer (MVO) [67], opposition-based learning GOA [47], Harris hawks optimizer (HHO) [17], Gradient-based optimizer (GBO) [19], slime mould optimization algorithm (SMA) [18], RUNge Kutta optimizer (RUN) [22], hunger games search algorithm (HGS) [21], Differential evolution based upon learning the covariance matrix and setting the bimodal distribution parameter (CoBiDE) [68], enhanced SSA based upon particle swarm optimizer (SSAPSO) [53], and LSHADE with semi-parameter adaptation hybrid with CMA-ES (LSHADSP) [69].

Table 1 records the parameter settings of the algorithms for all experiments. In addition, the population size for all algorithms was set to 25 and the fitness value evaluation was set to 2500; each experiment was implemented 30 times for a statistical purpose.

TABLE 1. Parameter settings for all methods.

Algorithm	Settings
cSG	C_2 *and* $C_3 \in [0, 1]$, $C_{max} = 1$, $C_{min} = 0.00004$, pm=0.2
GOA	$C_{max} = 1$, $C_{min} = 0.00004$
SSA	$C_2 \in [0, 1]$, $C_3 \in [0, 1]$
CoBiDE	pb = 0.4, ps = 0.5
SSAPSO	C2 ∈ [0, 1], C3 ∈ [0, 1], wDamp =0.99, w=1, C1=1
LSHADSP	L Rate = 0.80, arc rate = 1.4, pbest rate = 0.11, first calss = 0.50, memory size = 5
OBLGOA	$C_{max} = 1$, $C_{min} = 0.00004$, OBL ratio=0.5
HHO	*E* in [0, 2]
SMA	z=0.03
RUN	r ∈ [1, -1], L ∈ [0, 1], g ∈ [0, 2]
GBO	pr=0.5
HGS	VC2 = 0.03
MVO	$WEP_{min} = 0.2$, $WEP_{max} = 1$
GA	pc=0.8, beta=8 gamma=0.2, mu=0.02, pm=0.2
PSO	C1=1, wDamp =0.99, w=1, C2=2

TABLE 2. Comparison between the proposed method and its components.

	Crossover SSA		GOASSA		cSG	
	Fitness	Std	Fitness	Std	Fitness	Std
F1	2.30E+04	4.1E+03	1.66E+04	6.9E+03	**9.61E+03**	**1.2E+03**
F2	1.33E+04	4.3E+03	1.93E+04	3.0E+03	**6.15E+03**	**8.7E+02**
F5	1.59E+04	1.1E+04	7.46E+02	**9.2E+01**	**4.82E+02**	1.2E+02
F6	1.03E+03	8.2E+01	5.34E+02	**3.0E+01**	**2.58E+02**	5.0E+01
F14	2.07E+01	9.1E-02	**2.03E+01**	1.1E-01	2.05E+01	**3.7E-02**
F15	2.81E+01	4.1E+00	2.97E+01	2.1E+00	**2.29E+01**	**1.5E+00**
F16	9.95E+01	1.3E+02	1.12E+02	1.1E+02	**3.76E+01**	**5.3E+00**
F23	2.08E+01	1.0E-01	**2.04E+01**	1.0E-01	2.13E+01	**2.0E-02**
F24	2.96E+01	1.3E+01	2.77E+01	9.0E+00	**2.46E+01**	**3.8E+00**
F25	3.48E+02	2.1E+02	2.90E+02	9.1E+01	**2.40E+02**	**2.1E+00**

B. EXPERIMENT 1: COMPARISON WITH CSG, CROSSOVER SSA, AND GOASSA

This experiment compares the cSG to two other models which formed the proposed methods to show the effectiveness of each model on the cSG in solving global optimization problems. These models are: crossover SSA and GOASSA. The first model is the improved version of the SSA algorithm by using the crossover operator, whereas the second model improves the local search of the GOA with the SSA algorithm without using the crossover operators. Ten different functions: (1) unimodal, (2) multimodal, (3) hybrid, and (4) composition, were selected for this evaluation from the CEC2017 benchmark. All results are recoded in Table 2.

By inspecting these results, we can show that, the cSG outperformed the two other models with regard to average of fitness function values by showing the optimal results over 8 functions which are F1, F2, F5, F6, F15, F16, F24, and F25, whereas the GOASSA reached the best values across only two functions, which are F14 and F23. In this regard, the crossover SSA failed to fulfill the optimal values over all functions. Moreover, the cSG had been noticed as the most stable algorithm since it succeeded in achieving the smallest values over eight functions, whereas the GOASSA showed a smaller standard deviation over two functions, and the crossover SSA was ranked third in all functions in terms of the stability behavior.

From the above results, we can conclude that, the combination of both crossover operators and SSA adds a significant improvement and enhances the searching behavior of the original GOA. Therefore, in the following experiments we evaluate and compare the performance of the cSG over different problems and some recently meta-heuristics algorithms.

C. EXPERIMENT 2: SOLVING GLOBAL OPTIMIZATION PROBLEMS

This experiment evaluates the proposed method over the CEC2017 benchmark [70] and compares the results of the cSG with some recently meta-heuristic algorithms from the literature including: the SSA, HHO, GBO, SMA, RUN, HGS, CoBiDE, OBLGOA, SSAPSO, and LSHADSP (it was named in the following tables as LSHADSP). We reported the results of: (1) the average fitness, (2) the standard deviation, and (3) the computational time across 29 test functions of the CEC2017 benchmark in Tables 3-5. The experiment settings were 200 iterations and 30 population size. The dimension was set to 50 with 9000 fitness function evaluations.

As far as the unimodal functions, denoted as F1-F2, the cSG fulfilled the best values respecting average fitness as demonstrated in Table 3, whereas the other competitors failed to achieve the optimal values. Furthermore, the cSG was ranked first concerning standard deviation, as it was the most stable algorithmic method over the two functions F1 and F2, followed by the RUN and LSHADSP algorithms, whereas the GPO was ranked third in the stability measure as in Table 4.

In the multimodal functions, denoted as F3-F9, the LSHADSP was ranked first by reaching the best average fitness over three functions that are F4, F5 and F9, whereas the cSG, HHO, GPO, RUN, and HGS were ranked second by showing the optimal values on F8 and F9. The SMA and CoBiDE ranked third by performing well across both F9 and F7. Moreover, the LSHADSP was also observed to be the most stable method as it attained the smallest standard deviation across three functions, F5, F7 and F9, which is followed by the cSG, HHO, GPO, RUN, and HGS, respectively.

In the hybrid functions, denoted as F10-F19, the cSG was the best-performing method concerning the average of the fitness function values. It was superior with regard to the best average fitness of the best solutions so-far over four functions, F12, F15, F16, and F19. The RUN was ranked second by showing the optimal fitness values in three out of ten functions (F10, F13, and F17), while the HGS and SMA were ranked third by achieving good performance in only two hybrid functions. The obtained results over the F10 and F14 functions were the best ones with regard to HGS, while the SMA showed the optimal values in both F10 and F18. Additionally, the GBO came in the first rank respecting

TABLE 3. Results of the fitness values of the proposed method and the compared methods over global optimization functions.

Func.	cSG	OBLGOA	GOA	SSA	HHO	GBO	SMA	RUN	HGS	CoBiDE	SSAPSO	LSHADSP
F1	**9.614E+03**	2.425E+04	3.863E+04	4.787E+04	6.589E+04	1.147E+04	3.042E+04	9.783E+03	4.068E+04	2.637E+04	2.337E+04	9.829E+03
F2	**6.146E+03**	2.150E+04	2.856E+04	4.735E+04	5.562E+04	1.248E+04	2.713E+04	1.038E+04	2.787E+04	2.841E+04	5.835E+04	6.201E+03
F3	**7.409E+03**	3.672E+04	3.129E+04	5.728E+04	8.080E+04	1.857E+04	2.835E+04	1.368E+04	3.248E+04	4.323E+04	8.392E+04	7.732E+03
F4	3.078E+02	5.562E+02	8.343E+02	5.821E+02	6.923E+02	3.803E+02	2.666E+02	4.236E+02	2.932E+02	5.094E+02	6.474E+02	**1.250E+02**
F5	4.817E+02	1.013E+05	9.258E+05	1.423E+04	5.158E+04	2.163E+03	3.832E+02	3.538E+02	2.559E+03	2.531E+04	1.427E+05	**3.311E+02**
F6	**2.577E+02**	1.143E+03	2.713E+03	9.382E+02	1.196E+03	6.376E+02	3.331E+02	9.660E+02	5.205E+02	6.707E+02	1.356E+03	2.669E+02
F7	-1.269E+03	-9.319E+02	-8.616E+02	-1.374E+03	-9.182E+02	-1.560E+03	-1.619E+03	-1.629E+03	-1.614E+03	**-8.999E+12**	-9.510E+02	-1.431E+03
F8	-1.241E+01	5.267E+00	3.503E+01	1.592E+01	**-9.037E+01**	**-9.037E+01**	-9.028E+01	**-9.037E+01**	**-9.037E+01**	1.624E+01	5.661E+00	-3.034E+01
F9	-5.917E-02	5.693E+00	8.946E-01	3.094E+00	**-5.93E-02**	**-5.93E-02**	**-5.93E-02**	**-5.93E-02**	**-5.93E-02**	3.828E+00	4.946E+00	**-5.93E-02**
F10	-2.803E+01	2.479E+01	4.343E+01	2.084E+01	**-5.318E+01**	**-5.318E+01**	**-5.318E+01**	**-5.318E+01**	**-5.318E+01**	1.094E+01	3.481E+01	-3.801E+01
F11	-4.941E+03	-3.383E+03	-3.650E+03	-4.279E+03	-4.946E+03	-4.946E+03	-4.942E+03	-4.933E+03	-4.946E+03	**-9.933E+14**	-4.575E+03	-4.946E+03
F12	**5.786E+02**	5.166E+03	4.085E+04	4.418E+03	6.233E+03	1.651E+03	5.848E+02	6.479E+03	1.181E+03	3.739E+03	2.252E+04	7.182E+02
F13	4.359E+05	3.112E+08	6.064E+08	1.080E+08	1.353E+08	3.584E+06	8.948E+04	**1.330E+04**	2.600E+07	2.747E+08	3.230E+08	3.880E+05
F14	2.052E+01	2.096E+01	2.133E+01	2.050E+01	2.101E+01	2.129E+01	2.119E+01	2.099E+01	**2.024E+01**	2.091E+01	2.114E+01	2.028E+01
F15	**2.293E+01**	3.008E+01	6.064E+01	4.543E+01	3.422E+01	3.241E+01	2.371E+01	2.581E+01	2.942E+01	5.928E+01	3.451E+01	2.487E+01
F16	**3.756E+01**	5.461E+02	1.134E+03	3.973E+02	4.962E+02	1.258E+02	5.384E+01	3.963E+02	1.496E+02	3.425E+02	8.355E+02	3.969E+01
F17	7.114E-01	3.090E+00	1.191E+01	1.670E+00	2.476E+00	1.106E+00	1.212E+00	**6.450E-01**	1.411E+00	1.904E+00	6.150E+00	7.205E-01
F18	7.623E+02	5.433E+03	4.502E+03	4.488E+03	6.505E+03	1.687E+03	**4.677E+02**	9.718E+03	1.281E+03	4.960E+03	6.437E+03	7.753E+02
F19	**3.662E+01**	1.663E+02	1.990E+02	1.394E+02	1.755E+02	1.347E+02	1.719E+02	1.591E+02	8.309E+01	1.427E+02	1.784E+02	3.688E+01
F20	8.180E+00	1.278E+01	9.741E+00	1.122E+01	1.492E+01	1.614E+01	1.058E+01	1.497E+01	**5.608E+00**	1.491E+01	9.588E+00	6.686E+00
F21	**1.317E+03**	2.591E+04	1.520E+05	1.754E+04	2.250E+04	4.072E+03	1.737E+03	1.032E+04	5.487E+03	1.807E+04	7.508E+04	1.714E+03
F22	**4.903E+06**	6.866E+09	1.172E+10	1.126E+09	2.786E+09	1.273E+08	9.483E+06	1.103E+07	2.435E+08	3.123E+09	1.046E+09	7.462E+06
F23	2.127E+01	2.102E+01	2.127E+01	**2.055E+01**	2.118E+01	2.124E+01	2.125E+01	2.124E+01	2.095E+01	2.127E+01	2.121E+01	2.125E+01
F24	**2.456E+01**	6.060E+01	1.180E+02	8.231E+01	6.892E+01	5.103E+01	5.531E+01	6.210E+01	6.712E+01	8.525E+01	8.732E+01	2.893E+01
F25	**2.398E+02**	7.772E+02	1.928E+03	8.994E+02	9.843E+02	5.195E+02	3.424E+02	6.200E+02	6.735E+02	8.485E+02	1.515E+03	2.414E+02
F26	1.181E+00	8.274E+00	8.598E+00	3.976E+00	5.758E+00	1.701E+00	1.639E+00	**1.057E+00**	3.254E+00	5.653E+00	6.929E+00	1.110E+00
F27	**1.113E+03**	2.480E+04	1.204E+05	1.563E+04	2.474E+04	3.585E+03	2.105E+03	1.701E+04	6.127E+03	1.447E+04	7.352E+04	1.411E+03
F28	**1.387E+02**	2.241E+02	2.533E+02	2.142E+02	2.386E+02	2.063E+02	2.102E+02	2.079E+02	2.135E+02	2.142E+02	2.428E+02	1.391E+02
F29	1.547E+01	1.522E+01	1.762E+01	1.674E+01	1.590E+01	1.739E+01	1.498E+01	1.665E+01	**1.248E+01**	1.904E+01	1.722E+01	1.666E+01

the standard deviation, where it outperformed the other metaheuristics with respect to stability in four out of ten functions namely F10, F11, F14 and F17, followed by the cSG, HHO, SMA, RUN, HGS, and SSAPSO which gained the second rank.

In the composition functions, denoted as F20-F29, the cSG showed dominant performance over this group by realizing the best average fitness values over most functions, i.e., F21, F22, F24, F25, F27, and F25. The HGS was the second best-performing method by achieving the optimal values across two functions (i.e., F20 and F29), whereas the SSA and RUN gained the third rank, which showed the pest performance over F23 and F26, respectively. In addition, the cSG was also dominant across this group and superior to the other competitors in achieving a low standard deviation over eight out of ten functions.

Furthermore, in terms of computational time of all methods, the cSG showed acceptable computational time in all

TABLE 4. Results of the standard deviation of the proposed method and the compared methods over global optimization functions.

Func.	cSG	OBLGOA	GOA	SSA	HHO	GBO	SMA	RUN	HGS	CoBiDE	SSAPSO	LSHADSP
F1	**1.22E+03**	6.27E+03	6.10E+04	1.90E+04	2.13E+04	1.34E+03	4.57E+03	1.27E+03	7.19E+03	6.59E+03	9.45E+03	1.47E+04
F2	**8.65E+02**	6.19E+03	4.16E+03	1.30E+04	9.06E+03	1.39E+03	7.45E+03	2.40E+03	2.45E+03	3.43E+03	1.18E+04	1.23E+03
F3	**9.00E+02**	8.77E+03	5.36E+03	1.48E+04	2.72E+04	4.32E+03	6.58E+03	1.94E+03	7.73E+03	7.93E+03	1.11E+04	2.26E+03
F4	**1.42E+01**	1.38E+02	1.38E+02	1.11E+02	5.33E+01	3.33E+01	2.61E+01	1.82E+01	3.10E+01	3.83E+01	1.44E+01	1.66E+01
F5	1.21E+02	5.05E+04	4.29E+05	4.77E+03	9.51E+03	4.54E+02	6.65E+01	6.26E+01	2.02E+03	6.47E+03	1.27E+04	**5.28E+01**
F6	5.02E+01	2.62E+02	4.51E+02	1.16E+02	1.28E+02	3.13E+01	**3.55E+00**	1.24E+02	9.69E+01	4.36E+01	5.21E+01	2.97E+01
F7	5.99E+01	1.65E+02	7.21E+01	1.09E+02	1.61E+02	1.95E+02	1.16E+02	1.16E+02	6.50E+01	1.38E+13	1.31E+02	**3.91E+01**
F8	1.27E+01	7.93E-01	9.22E+00	5.43E+00	**0.00E+00**	**0.00E+00**	1.66E-01	**0.00E+00**	**0.00E+00**	1.50E+01	3.33E-01	2.77E+01
F9	2.29E-04	8.77E-01	3.09E+00	1.25E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.20E-01	1.54E-02	**0.00E+00**
F10	1.93E+01	2.18E+00	1.96E+01	5.80E+00	**7.11E-15**	**7.11E-15**	**7.11E-15**	**7.11E-15**	**7.11E-15**	1.33E+01	5.36E-01	1.04E+01
F11	1.32E+00	3.06E+02	4.65E+02	1.47E+02	**0.00E+00**	**0.00E+00**	1.46E+00	2.67E+01	**0.00E+00**	2.43E+14	1.00E+02	1.49E+00
F12	**2.54E+01**	8.85E+02	1.59E+04	1.30E+03	9.81E+02	3.05E+02	1.52E+02	1.23E+03	3.11E+02	1.48E+03	3.32E+02	1.28E+02
F13	1.18E+05	2.62E+08	1.33E+09	4.24E+07	3.19E+07	1.18E+06	4.67E+04	**5.91E+03**	2.87E+07	8.39E+07	5.44E+08	3.71E+05
F14	3.72E-02	4.42E-02	1.30E-01	8.39E-02	7.98E-02	**2.44E-02**	7.08E-02	4.57E-01	5.95E-02	7.62E-02	6.06E-02	2.98E-02
F15	1.46E+00	3.61E+00	7.49E+00	5.92E+00	1.03E+00	3.73E+00	1.57E+00	1.84E+00	1.66E+00	8.46E+00	**2.63E-01**	2.30E+00
F16	**5.25E+00**	7.58E+01	2.70E+02	4.35E+01	4.62E+01	3.61E+01	7.74E+00	7.03E+01	4.85E+01	5.23E+01	1.81E+01	1.52E+01
F17	1.92E-01	7.56E-01	4.01E+00	1.14E-01	3.63E-01	**3.20E-02**	5.84E-02	2.05E-01	3.87E-01	2.90E-01	3.16E-01	1.87E-01
F18	1.67E+02	5.47E+02	1.11E+04	1.37E+03	1.38E+03	3.93E+02	**6.09E+01**	7.10E+02	4.85E+02	1.58E+03	9.70E+03	3.39E+02
F19	1.30E+01	9.79E+00	1.99E+01	4.61E+00	5.04E+00	1.15E+01	1.64E+01	1.25E+01	9.43E+00	4.00E+00	**3.47E+00**	5.91E+00
F20	3.03E-01	5.41E-01	2.08E+00	1.45E+00	7.98E-01	1.05E+00	1.09E+00	1.07E+00	4.91E-01	4.45E-01	3.68E+00	**2.61E-01**
F21	**3.97E+01**	1.26E+04	6.43E+04	5.37E+03	5.76E+03	1.69E+03	3.26E+02	2.77E+03	1.46E+03	3.98E+03	2.60E+03	4.77E+02
F22	**4.01E+06**	7.93E+09	1.81E+10	4.28E+08	7.81E+08	4.03E+07	9.79E+06	5.58E+06	2.45E+08	1.77E+09	5.12E+09	5.09E+06
F23	**2.00E-02**	6.90E-02	1.05E-01	7.76E-02	6.58E-02	7.52E-02	7.81E-02	3.14E-02	4.94E-02	3.46E-02	6.91E-02	6.36E-02
F24	3.82E+00	4.05E+00	1.95E+01	1.08E+01	1.78E+00	**1.46E+00**	8.11E+00	2.32E+00	6.66E+00	7.62E+00	2.99E+00	4.72E+00
F25	**2.12E+00**	8.98E+01	4.15E+02	1.27E+02	9.41E+01	6.58E+01	2.95E+01	9.29E+01	1.29E+02	7.99E+01	4.48E+01	7.01E+01
F26	**4.75E-02**	2.23E+00	3.90E+00	4.88E-01	7.01E-01	1.77E-01	9.09E-02	6.24E-02	1.63E+00	1.02E+00	8.82E+00	7.28E-02
F27	**2.62E+02**	5.18E+03	5.44E+04	4.43E+03	4.51E+03	4.09E+02	4.06E+02	2.10E+03	1.56E+03	4.79E+03	5.26E+03	3.68E+02
F28	**1.02E+00**	1.28E+01	2.74E+01	1.79E+01	2.12E+00	9.39E+00	2.56E+00	1.71E+01	1.90E+01	1.84E+01	6.91E+00	1.32E+01
F29	**7.81E-02**	9.99E-01	1.78E+00	3.02E+00	3.15E-01	7.61E-01	1.20E+00	1.25E+00	7.82E-01	6.03E-01	5.23E-01	9.96E-01

functions, as recorded in Table 5, whereas, the HGS was faster than the compared methods and it was ranked first followed by the SSA, CoBiDE, SSAPSO, and GBO, respectively. However, the cSG was considered as a faster algorithm, it obtained the best fitness values and showed good stability in most cases.

Moreover, the convergence behaviors for all methods are illustrated in Figure 2. This figure shows and compares the

TABLE 5. The Computational time of the proposed method and the compared methods over global optimization functions.

Func.	cSG	OBLGOA	GOA	SSA	HHO	GBO	SMA	RUN	HGS	CoBiDE	SSAPSO	LSHADSP
F1	7.9035	16.2121	11.6088	2.7437	6.4407	3.2523	4.9480	5.1601	**2.6131**	2.8544	3.1079	3.6181
F2	9.0128	17.9911	14.5992	6.3065	14.8807	6.8374	8.5667	11.0868	**6.2190**	6.3943	6.6772	9.8918
F3	7.7716	15.9368	11.5191	2.7383	6.3601	3.2133	5.0055	5.1996	**2.6440**	2.8043	3.1235	3.5697
F4	8.7057	15.8299	11.1402	2.3033	5.2277	2.7861	4.5133	4.4730	**2.2071**	2.3654	2.6861	3.0885
F5	10.2859	19.7144	17.9711	9.3481	21.0249	9.8826	11.6211	16.0770	**9.2393**	9.4476	9.7800	15.1654
F6	7.4435	15.9803	10.9508	2.4159	5.3849	2.9121	4.6398	4.6456	**2.3149**	2.4822	2.7826	3.2772
F7	5.8044	15.6850	11.1770	2.3180	5.4075	2.8235	4.5189	4.5040	**2.1995**	2.4086	2.6818	3.1830
F8	8.9219	16.1593	11.5699	2.7142	5.6668	3.1814	4.9448	5.1227	**2.6324**	2.7739	3.0867	3.4802
F9	7.6433	15.7010	11.1157	2.2866	4.7206	2.7632	4.4982	4.4710	**2.2096**	2.3367	2.6487	3.1496
F10	7.3275	16.0695	11.2205	2.7532	5.8133	3.2405	5.0276	5.1683	**2.6758**	2.8556	3.1177	3.5855
F11	7.9102	15.7680	10.9200	2.2657	4.6829	2.7709	4.5033	4.3983	**2.1663**	2.3272	2.6495	2.9277
F12	9.6433	18.0956	14.4508	5.8483	12.5375	6.3277	8.1115	10.1620	**5.7436**	5.9242	6.1948	9.2047
F13	9.7189	17.7412	14.4040	5.8772	12.5520	6.3642	8.2006	10.1475	**5.7744**	5.9741	6.2276	9.3184
F14	8.4026	17.8169	14.4963	5.8845	14.3025	6.3709	8.1283	10.5479	**5.7792**	5.9624	6.2591	9.6693
F15	9.8550	18.1116	14.2801	5.8069	13.2176	6.2967	8.0510	10.1937	**5.7078**	5.8758	6.1995	9.4733
F16	9.9950	17.8064	13.9588	5.8164	12.7053	6.3066	7.9973	10.0570	**5.6760**	5.8716	6.1672	9.1756
F17	9.7384	17.8983	14.3931	5.9800	12.7399	6.4832	8.2087	10.3116	**5.8725**	6.0434	6.3481	9.3050
F18	10.2393	18.0063	14.1261	5.9881	12.9741	6.4902	8.3122	10.3519	**5.8924**	6.0773	6.3539	9.2261
F19	10.6310	17.6370	14.6307	5.9280	14.1997	6.4104	8.2322	10.5282	**5.8218**	5.9842	6.2915	9.3866
F20	8.3468	17.6035	14.5977	5.9188	14.3076	6.4153	8.1804	10.6435	**5.8197**	6.0180	6.3099	9.1296
F21	9.4892	17.6906	14.6074	5.8729	12.6391	6.3805	8.1204	10.1600	**5.7761**	5.9606	6.2624	9.2533
F22	8.5984	17.9586	15.0649	5.9194	12.7963	6.3922	8.0523	10.2584	**5.8024**	5.9976	6.2897	9.5341
F23	8.4411	17.9091	14.5296	5.9251	14.2953	6.4136	8.1545	10.5896	**5.8158**	6.0060	6.3087	9.2251
F24	9.4469	17.7251	14.4768	5.9319	13.7921	6.3998	8.0927	10.4174	**5.8145**	5.9873	6.3026	9.4399
F25	8.7755	18.2051	14.2106	6.2220	13.2599	6.5679	8.3009	10.3873	**5.8433**	6.0465	6.3600	9.0334
F26	9.7914	18.2702	14.5574	6.2448	13.5528	6.7065	8.6097	10.7954	**6.0937**	6.4032	6.6207	8.8435
F27	9.6724	18.4391	14.8646	**6.1691**	13.3999	6.7760	8.4839	10.8597	6.1755	6.4332	6.5945	9.7991
F28	10.3516	17.7830	14.5247	6.0442	14.3460	6.5110	8.3371	10.6866	**5.9407**	6.1312	6.4101	11.8683
F29	8.2718	17.7868	14.1780	6.0037	14.5569	6.4770	8.2985	10.7095	**5.8940**	6.1058	6.3761	13.1479

convergence curves to evaluate the behavior of each method in reaching the optimal values. From the figure, we can see that the cSG showed fast convergence (the black curve) to the optimal value and improved its behavior in the course of iterations than the compared methods, especially in F19 and F28. The LSHADSP also showed good convergence in most of the functions. In function F23, the SMA and HGS showed fast convergence at the beginning of the iterations; then, they did not show any convergence until the last quarter of iterations; they also showed good convergence in F29. In general, all algorithms were able to update their search domains during the process, except the original GOA took a long time to update its populations; therefore, the proposed cSG effectively improved the local search of the original GOA.

D. EXPERIMENT 3: SOLVING FEATURE SELECTION PROBLEMS

In this experiment, eleven datasets, including Wine, Breast-cancer, Glass, Lymphography, Waveform, Spect, Zoo, Ecoli, Vote, Breastw, and Ionosphere, were used to evaluate the proposed cSG method. These benchmark datasets were chosen

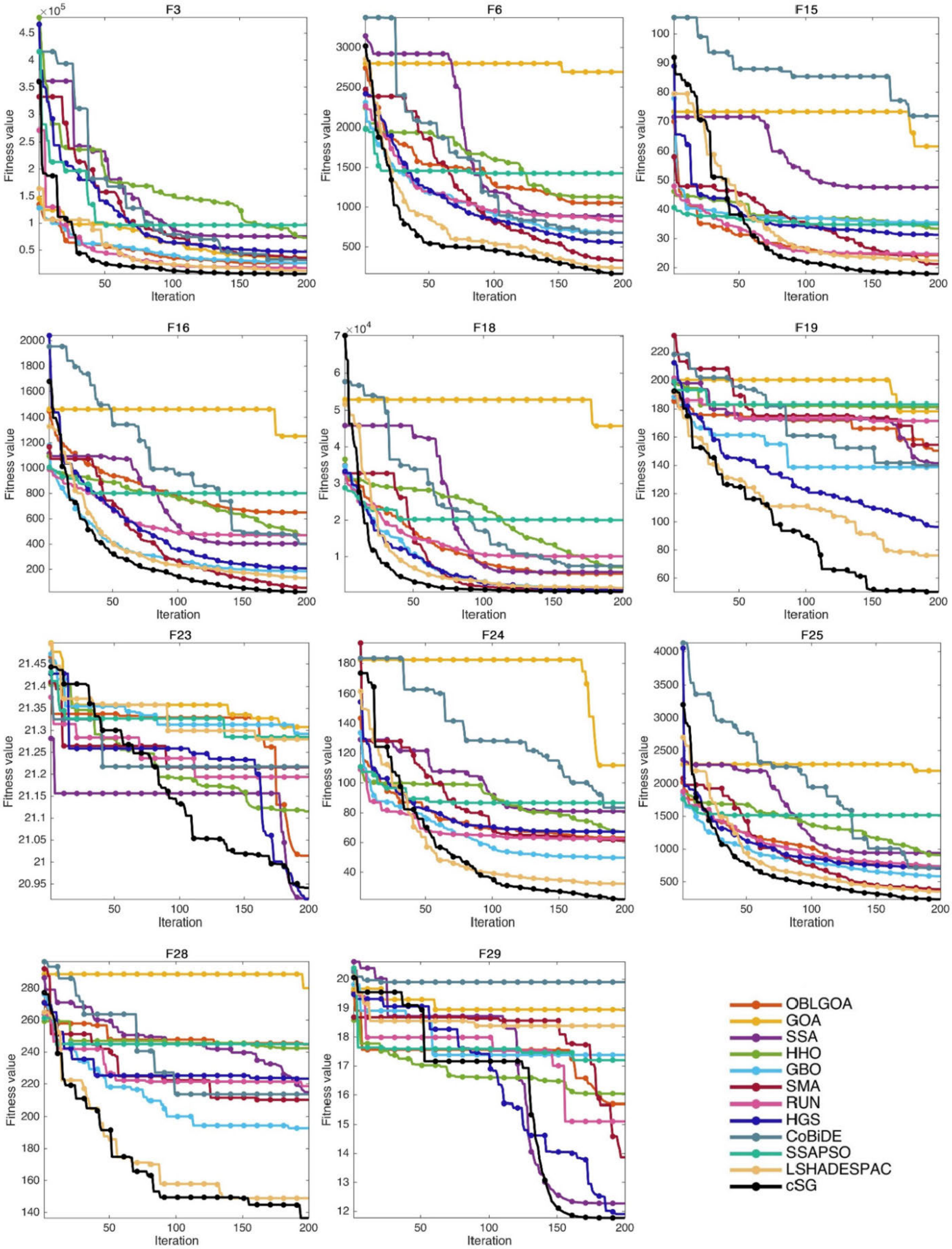


FIGURE 2. Example of the convergence curves for all methods in global optimization experiment.

from the UCI Repository Machine Learning Database [71], which all have distinct properties as shown in Table 6. In order to prove the performance of suggested cSG algorithm in solving the general feature selection problems, we compared it to some recent and well-known competitive optimizers, including: SSA, GOA, GA, PSO, and MVO.

The experimental results are discussed based on some performance measures, namely: 1) The fitness function value as in Eq. (10). 2) The number of selected features formulated in Eq. 12. 3) The classification accuracy as in Eq. (13). 4) The computation time. 5) The Wilcoxon rank-sum test as a statistical test.

The number of selected features is calculated as:

$$NF = \left[1 - \frac{\sum_{i=1}^{S} e_i}{S}\right], \quad e_i \leftarrow \begin{cases} 1 & \textit{if } e_i \textit{ is selected} \\ 0 & \textit{if } e_i \textit{ is neglected} \end{cases} \tag{12}$$

where, S is the feature number. e_i denotes the selected attribute.

The accuracy (ACC) is calculated as:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

Here TP symbolizes the true positive instances, FP indicates the false positive instances, FN symbolizes the false negative instances, TN indicates the true negative instances.

In terms of the fitness function values, according to Table 7, the cSG method achieved the lowest fitness values in 8 datasets out of 11: Glass (0.1266), Lymphography (0.2605), Waveform (0.6325), Spect (0.3120), Ecoli (0.2077), Vote (0.1274), Breastw (0.1009), and Ionosphere (0.1620). Furthermore, the PSO ranked second in three datasets namely Wine (0.0042), Breastcancer (0.1682), and ZOO (0.0037). It is also noted that the cSG algorithm similarly acts with PSO in Breastcancer and Zoo datasets, achieving RMSE of 0.1898 and 0.0047, respectively. It is observed that the GOA realized the last rank by showing the highest error rates across all the tested datasets. These results reflect the stability of the cSG algorithm over the different runs. Figure 3 illustrates the average results of fitness function value.

In terms of classification accuracy, as observed in Table 8, the cSG method realized the highest accuracy over 9 datasets: Breastcancer (0.9705), Glass (0.7862), Lymphography (0.9305), Spect (0.8972), Zoo (1), Ecoli (0.8492), Vote (0.9833), Breastw (0.989), and Ionosphere (0.9735). These results reveal the general capability of the algorithm to effectively search the promising regions within the search space. Furthermore, the classification accuracy results of cSG are very close to these of PSO in two datasets that are Wine and Waveform, respectively. Accordingly, the PSO algorithm took the second rank after the proposed method with respect to classification accuracy, while the GOA also achieved the last rank between the other compared models. Moreover, the F-measure for all methods was also calculated and presented in Table 9. It is observed from this table that the proposed cSG method obtained the best F-measure results in 8 out of 11 datasets. Whereas, the PSO ranked second, it realized the best results over 3 datasets with a slight difference from cSG. The SSA and MVO were ranked third and fourth, respectively followed by the GA and GOA.

TABLE 6. Description of the benchmark datasets.

Dataset	Classes number	Instances number	Features number
Wine	3	178	13
Breastcancer	2	699	9
Glass	7	214	9
Lymphography	2	148	18
Waveform	3	5000	40
Spect	2	267	22
Zoo	6	101	16
Ecoli	8	336	7
Vote	2	300	16
Breastw	2	569	30
Ionosphere	2	351	34

TABLE 7. Results of fitness function value.

Dataset	cSG	SSA	GOA	GA	PSO	MVO
Wine	0.0131	0.0487	0.1506	0.0168	**0.0042**	0.0209
Breastcancer	0.1898	0.2112	0.3240	0.2152	**0.1682**	0.2113
Glass	**0.1266**	0.1484	0.2229	0.1499	0.1394	0.1471
Lymphography	**0.2605**	0.2952	0.5109	0.3558	0.2776	0.3249
Waveform	**0.6325**	0.6500	0.7285	0.6516	0.6333	0.6564
Spect	**0.3120**	0.3519	0.4710	0.3642	0.3284	0.3578
Zoo	0.0047	0.0471	0.1155	0.0157	**0.0037**	0.0143
Ecoli	**0.2077**	0.2284	0.3363	0.2249	0.2192	0.2242
Vote	**0.1274**	0.1871	0.3350	0.2017	0.1682	0.1841
Breastw	**0.1009**	0.1056	0.2164	0.1229	0.1042	0.1279
Ionosphere	**0.1620**	0.2221	0.3173	0.2276	0.1831	0.2199

In terms of computational time, as seen in Table 10, the MVO method ranked first based on the average of all datasets, whereas the PSO, cSG, and SSA methods showed similar times to some extent followed by GOA and GA, respectively. That means the cSG is faster than the original version of the GOA due to the effective balance of the SSA and the GOA operators. Figure 4 illustrates the average results of the classification accuracy for all methods.

In terms of the selected attributes number, checking the results of Table 11, the cSG method was capable of selecting the significant features over 2 datasets out of 11 (namely Spect and Breastw) and acts like the ISSALD approach in one dataset (named Exactly2 dataset). Although the GOA method realized the last rank in terms of RMSE and classification accuracy, it comes in the first place with regard to the number of selected attributes by realizing the best results over 5 datasets (namely Wine, Glass, Waveform, Zoo, and Ecoli). In addition, the PSO model achieved the third rank by showing the best results over 3 datasets (namely Breastcancer, Breastw, and Ionosphere). Moreover, the SSA model took the fourth rank by achieving the least number of selected attributes in only 1 dataset (namely Lymphography).

Furthermore, Table 12 records the statistical results of all methods using the Wilcoxon rank-sum test for inspecting if there is a significant difference between the cSG and the other algorithms or not at p-value < 0.05. The Wilcoxon rank-sum is a statistical test implemented for assessing the performance significance. It is non-parametric test that sets

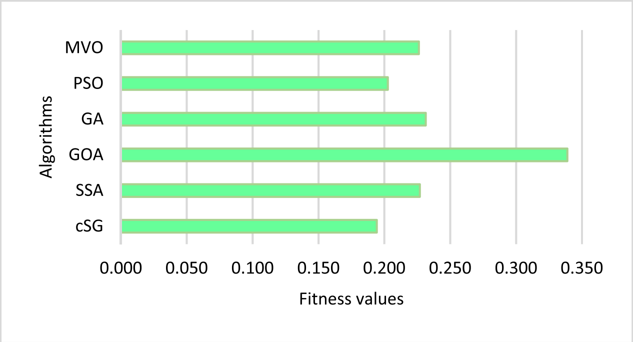


FIGURE 3. Average results of the fitness function value.

TABLE 8. Results of the classification accuracy.

Dataset	cSG	SSA	GOA	GA	PSO	MVO
Wine	0.9982	0.9735	0.9034	0.9949	**0.9987**	0.9937
Breastcancer	**0.9705**	0.9533	0.8917	0.9517	0.9696	0.9536
Glass	**0.7862**	0.6908	0.5173	0.6095	0.7013	0.6205
Lymphography	**0.9305**	0.9108	0.4039	0.7080	0.8132	0.5788
Waveform	0.7936	0.7870	0.7431	0.7832	**0.7939**	0.7820
Spect	**0.8972**	0.8735	0.7745	0.8631	0.8901	0.8725
Zoo	**1.0000**	0.7078	0.2589	0.9011	0.9956	0.9044
Ecoli	**0.8492**	0.8393	0.8185	0.8357	0.8367	0.8401
Vote	**0.9833**	0.9615	0.8800	0.9574	0.9709	0.9644
Breastw	**0.9890**	0.9861	0.9511	0.9828	0.9883	0.9818
Ionosphere	**0.9735**	0.9479	0.8952	0.9460	0.9643	0.9476

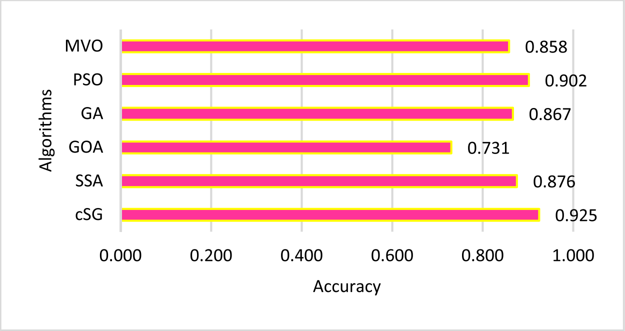


FIGURE 4. Average results of the accuracy measure.

ranks for the scores. It ranks all the scores in one group then it sums the ranks of the groups. In this two-sample test, the null assumption assumes that the samples are from the same population. Accordingly, if there is a difference in two rank sums, the sampling error will be the reason of that. From Table 12 we can notice that, the cSG has a significant difference to the compared algorithms in most cases namely: 64% with SSA and GA, 100% with GOA, 45% with PSO, and 55% with MVO.

The results of this experiment indicate that the cSG is a strong algorithm in solving the global optimization problems. The superiority of the algorithm above all the other applied algorithms is due to the integration of both genetic crossover operator and SSA which enhanced the exploitation and exploration of both algorithms. This hybridization added more flexibility for both algorithms and ensured finding the best solution for the optimization problem quickly and efficiently.

TABLE 9. F-measure results for all methods.

Dataset	cSG	SSA	GOA	GA	PSO	MVO
Wine	0.9925	0.9738	0.9038	0.9952	**0.9988**	0.9937
Breastcancer	0.9764	0.9647	0.9211	0.9638	**0.9768**	0.9651
Glass	**0.7452**	0.7280	0.5885	0.6726	0.7263	0.6943
Lymphography	**0.9473**	0.9248	0.6715	0.8455	0.9328	0.8701
Waveform	0.7932	0.7863	0.7425	0.7825	**0.7933**	0.7813
Spect	**0.7427**	0.6518	0.4377	0.6284	0.7004	0.6673
Zoo	**1.0000**	0.8841	0.6277	0.9478	0.9940	0.9662
Ecoli	**0.6202**	0.5509	0.3793	0.5529	0.5689	0.5827
Vote	**0.9785**	0.9520	0.8511	0.9472	0.9636	0.9557
Breastw	**0.9848**	0.9809	0.9328	0.9765	0.9839	0.9749
Ionosphere	**0.9798**	0.9600	0.9228	0.9588	0.9724	0.9599

TABLE 10. Results of the computational time.

Dataset	cSG	SSA	GOA	GA	PSO	MVO
Wine	**6.005**	6.413	6.732	7.182	6.455	6.436
Breastcancer	6.302	**6.032**	6.690	6.752	6.040	6.065
Glass	**5.902**	6.672	6.787	7.327	6.577	6.587
Lymphography	5.728	6.122	6.145	6.672	5.734	**5.623**
Waveform	18.40	18.15	18.41	20.33	18.18	**18.13**
Spect	6.066	6.071	6.471	6.679	6.039	**6.017**
Zoo	6.408	6.373	6.462	7.286	**6.172**	6.314
Ecoli	5.286	5.231	5.265	5.838	**5.220**	5.234
Vote	**6.209**	6.448	6.717	7.276	6.559	6.515
Breastw	6.697	6.355	6.951	7.069	**6.304**	6.333
Ionosphere	6.416	6.183	6.809	6.859	6.171	**6.154**

TABLE 11. Results of the selected features by each algorithm.

Dataset	cSG	SSA	GOA	GA	PSO	MVO
Wine	8.7	7.0	**6.5**	7.6	7.6	7.3
Breastcancer	15.9	15.1	16.3	15.8	**14.9**	15.6
Glass	5.8	5.4	**4.9**	5.3	5.0	5.5
Lymphography	10.8	**8.7**	9.4	10.1	9.2	9.8
Waveform	16.0	13.5	**11.7**	13.2	13.9	13.1
Spect	**10.3**	11.0	11.6	11.0	10.4	11.1
Zoo	10.1	9.2	**8.4**	9.2	9.4	9.4
Ecoli	5.2	4.9	**3.6**	4.9	5.0	5.3
Vote	**7.2**	7.2	8.3	7.9	7.7	7.5
Breastw	17.7	15.8	15.4	15.8	**15.3**	15.7
Ionosphere	17.3	14.6	17.2	15.7	**14.0**	15.7

E. EXPERIMENT 4: SOLVING DIFFERENT PROBLEMS OF ENGINEERING APPLICATIONS

This experiment evaluates the proposed cSG using six popular engineering problems including: (1) Welded beam design. (2) Pressure vessel design. (3) Three-bar truss. (4) Tension/compression spring. (5) Rolling element bearing. (6) Multi-plate disc clutch brake. These problems were solved by the proposed cSG and some meta-heuristic methods formerly

TABLE 12. Statistical results of the proposed cSG method with respect to wilcoxon rank-sum test.

Dataset	SSA	GOA	GA	PSO	MVO
Wine	1.0E+00	**3.5E-05**	3.6E-01	3.6E-01	5.1E-01
Breastcancer	**4.3E-04**	**5.3E-05**	**8.2E-04**	3.2E-01	**1.5E-04**
Glass	**7.1E-03**	**7.4E-05**	**5.0E-03**	**2.6E-02**	**3.5E-03**
Lymphography	**5.4E-04**	**6.5E-05**	**5.8E-05**	**2.6E-02**	**5.7E-04**
Waveform	**2.0E-03**	**7.9E-05**	**2.5E-03**	1.1E-01	**1.2E-02**
Spect	**9.1E-03**	**3.5E-05**	**4.0E-03**	3.2E-01	1.6E-01
Zoo	1.5E-01	**3.8E-04**	8.5E-02	1.0E+00	2.0E-01
Ecoli	4.7E-01	**2.4E-04**	3.2E-01	4.7E-01	2.9E-01
Vote	**1.8E-02**	**5.0E-05**	**4.5E-03**	**1.7E-03**	**4.6E-02**
Breastw	7.1E-01	**8.9E-05**	**1.9E-02**	**4.8E-02**	**1.2E-02**
Ionosphere	**3.4E-02**	**1.4E-04**	6.5E-02	**2.2E-02**	3.1E-01

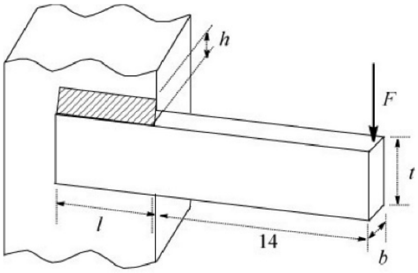


FIGURE 5. Problem of welded beam design.

implemented in the literature. The following subsections compare the solutions reached by cSG and the other methods to solve these optimization problems.

1) PROBLEM OF WELDED BEAM DESIGN

This problem tries to minimize fabrication cost for welded beam as shown in Figure 5. There are four variables to be optimized namely the weld's thickness of (h), the bar's thickness (bb), length of an attached part (l), and the bar's height (tt).

The results in Table 13 point out that the cSG algorithm obtained the best result among the compared methods for the welded beam structure through finding the lowest cost namely 1.72586; this cost represents the minimum optimum cost among all other compared algorithms. The GWO, MVO, and WOA achieved 1.72624, 1.72645, and 1.73049 and were ranked as second, third, and fourth, respectively. The highest cost was reported by the GA which reached 2.43312. It is apparent that the cSG significantly outperforms the aforementioned techniques in solving this problem.

2) PROBLEM OF PRESSURE VESSEL DESIGN

This experiment is intended to minimize the cylindrical pressure vessel's cost. Figure 6 shows the design of this problem. Four variables in this experiment need to be optimized namely: both thickness of a shell (T_s) and head (T_h), the cylindrical section length (L), and the inner radius (R).

TABLE 13. Results of the proposed method for solving the problem of welded beam design.

Algorithm	Optimal values for the variables				Optimal cost
	h	*l*	*tt*	*bb*	
Proposed cSG	0.20504	3.48701	9.03768	0.20572	**1.72586**
MVO [67]	0.20546	3.47319	9.04450	0.20569	1.72645
GSA [67]	0.18212	3.85697	10.00	0.20237	1.87995
GA [72]	0.24890	6.17300	8.17890	0.25330	2.43312
WOA [13]	0.20539	3.48429	9.03742	0.20627	1 .73049
HHO [17]	0.20403	3.53106	9.02746	0.20614	1.731991
GWO [73]	0.20567	3.47837	9.03681	0.20577	1.72624

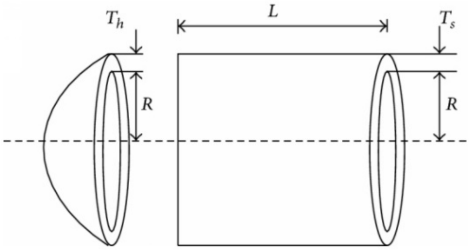


FIGURE 6. Pressure vessel design problem.

TABLE 14. Results of the proposed method for solving the problem of pressure vessel design.

Algorithm	Optimal values for the variables				Optimal cost
	Ts(x1)	*Th(x2)*	*R(x3)*	*L(x4)*	
Proposed cSG	0.830556	0.41064	43.0338	165.9827	**5995.783**
MVO [67]	0.8125	0.4375	42.09074	176.7387	6060.807
GSA [67]	1.125	0.625	55.98865	84.4542	8538.835
GA [72]	0.8125	0.4375	42.0974	176.6541	6059.946
WOA [13]	0.8125	0.4375	42.09827	176.639	6059.741
HHO [17]	0.817584	0.40729	42.09175	176.7196	6000.463
GWO [73]	0.8125	0.4345	42.08918	176.7587	6051.564

This subsection gives a comparison on the optimal results taken for the problem of pressure vessel design by cSG and the other previously mentioned models. As shown in Table 14, the lowest cost was reported by the cSG algorithm which is 1.72586. It is also seen that the GWO, MVO, and WOA came in the second, third, and fourth rank with the cost of 1.72624, 1.72645, 1.73049, respectively. In this regard, the GA reported the highest cost of 2.43312 among the other algorithms. These results demonstrate the merits of cSG algorithm in solving this design problem.

3) PROBLEM OF THREE-BAR TRUSS

The aim of this experiment is minimizing the truss weight. Two variables in this experiment need to be optimized namely A_1 and A_2. Variable $A_3 = A_1$ as shown in Figure 7.

The results of the cSG algorithm for solving the design problem of three-bar truss in comparison to some algorithms are provided in Table 15. It can be noticed that the algorithm of cSG is competitive. It achieved the lowest cost of 263.896. It is also clear that the MVO is competitive to the GOA and MFO by reporting lower cost of 263.896 while the two other algorithms reported 263.896 and 263.896, respectively. The highest cost was reached by the CS algorithm which achieved a value of 263.972. From these results, it is apparent that cSG

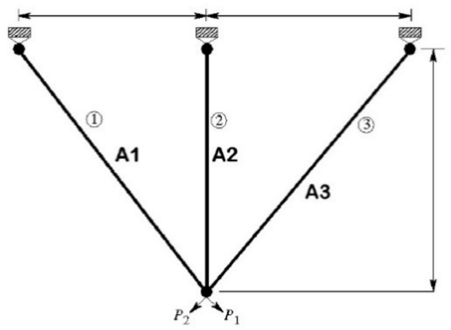


FIGURE 7. Three-bar truss problem.

TABLE 15. Results of the proposed method for solving the design problem of three-bar truss.

Algorithm	Optimal values for the variables		Optimal cost
	x_1	x_2	
Proposed cSG	0.78868267	0.408226977	**263.8958434**
GOA [14]	0.788898	0.40762	263.8958815
MVO [67]	0.788603	0.408453	263.8958499
CS [74]	0.78867	0.40902	263.9715600
MFO [75]	0.7882447	0.409466905	263.8959797

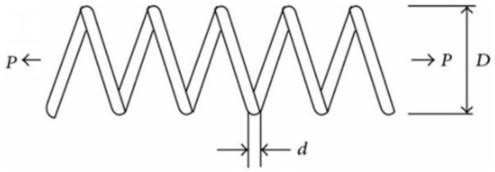


FIGURE 8. Tension/compression spring problem.

significantly outperforms the other compared meta-heuristic algorithms when solving this design problem.

4) PROBLEM OF TENSION/COMPRESSION SPRING

The objective of this experiment is to minimize the tension/compression spring weight. Three variables in this experiment need to be optimized namely the mean coil diameter (D), the active coils number (NN), and the wire diameter (dd). These variables are illustrated in Figure 8.

The problem of tension/compression spring design problem was extensively addressed through various bio-inspired optimization algorithms including: MVO, GSA, PSO, WOA, GWO, MFO, SSA and RO. Table 16 presents the comparison results between the proposed cSG and the other competitors' algorithms with regard to the values of design variables and the cost value for this problem. Accordingly, the results in Table 16 shows that the proposed cSG is capable of finding the optimal design for this problem achieving a minimum cost of 0.012665. Such cost is slightly lower than those given by other algorithms. The WOA and SSA showed similar cost of 0.01268 whereas the MVO algorithm showed the highest cost of 0.01279. These results confirm that cSG performs better than other meta-heuristic optimization methods in reaching the optimum solution for solving this design problem.

TABLE 16. Results of the proposed method for solving the design problem of tension/compression spring.

Algorithm	Optimal values for the variables			Optimal cost
	d	D	N	
Proposed cSG	0.051619	0.355034	11.38839	**0.012665**
MVO [67]	0.05251	0.3760	10.3351	0.01279
GSA [67]	0.050276	0.323680	13.525410	0.0127022
WOA [13]	0.051207	0.345215	12.004032	0.0126763
GWO [73]	0.05169	0.356737	11.28885	0.012666
MFO [75]	0.0519945	0.364109	10.8684	0.01267
SSA [60]	0.051207	0.345215	12.004032	0.0126763
RO [76]	0.051370	0.349096	11.76279	0.0126788

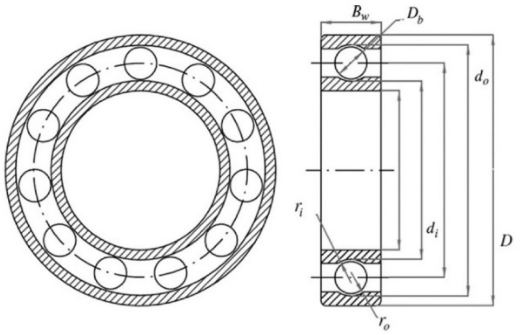


FIGURE 9. Rolling element bearing problem.

5) PROBLEM OF ROLLING ELEMENT BEARING

The aim of this experiment is maximizing the ability of the dynamic load carrying. In this experiment, ten variables used for assembling and restrictions of geometric, need to be obtimized. Figure 9 illustrates an overview of this problem.

A comparison between the cSG and the HHO, MVO, PVS, and TLBO, for solving the rolling element bearing design problem is illustrated in Table 17. Regarding the optimal costs in Table 17, cSG presented the best value for that design problem with 85446.7489. The MVO came in the second rank by reporting a cost value equals 83535.147 followed by the HHO which showed a cost equals 83011.883. On the other side, the PVS and TLBO algorithmic models reported almost similar cost values. These results reflect that the cSG outperforms the other meta-heuristic optimization techniques in reaching the best optimal solution for this design problem.

6) PROBLEM OF MULTI-PLATE DISC CLUTCH BRAKE

The aim of this experiment is minimizing the multiple disc clutch brake weight. Five variables in this experiment need to be optimized namely outer radius, discs thickness, inner radius, actuating force, and friction surfaces. Figure 10 illustrates an overview of this problem.

The cSG algorithm is used to solve multi-plate disc clutch brake problem and compared to MVO, PVS, WCA, and TLBO. Table 18 reports the results of this comparison to get the best cost found by such meta-heuristics. The findings report that the proposed cSG algorithm outperformed all other methods in finding the optimum minima solution of this design. It showed the lowest cost of 0.2598 followed by

TABLE 17. Results of the proposed method for solving the design problem of rolling element bearing.

Variable	Proposed cSG	HHO [17]	MVO	PVS [77]	TLBO [78]
D_m	125.5126	125.000	126.1035	125.719	125.719
D_b	21.4233	21.000	21.03794	21.42559	21.42559
Z	10.98428	11.092	11.13977	11.0000	11.0000
f_i	0.51500	0.51500	0.51500	0.5150	0.515000
f_o	0.51500	0.515	0.52286	0.5150	0.515000
K_{dmin}	0.49385	0.4000	0.44355	0.40043	0.424266
K_{dmax}	0.613865	0.6000	0.67521	0.68016	0.633948
ε	0.300329	0.3000	0.30108	0.3000	0.300000
e	0.078389	0.0505	0.08852	0.07999	0.068858
ξ	0.635913	0.6000	0.61468	0.7000	0.799498
Max cost	**85446.748**	83011.88	83535.14	81859.74	81859.74

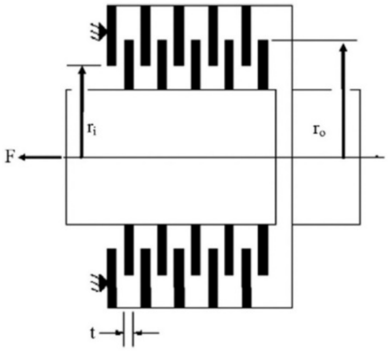


FIGURE 10. Multi-plate disc clutch brake problem.

TABLE 18. Results of the proposed method for solving multi-plate disc clutch brake problem.

Algorithm	Optimal values for the variables					Optimal cost
	r_i	r_o	t	F	z	
Proposed cSG	70.000	90	1	999.99	2.312783	**0.2597690**
MVO [67]	69.993	90	1	999.30	2.31676	0.2601551
PVS [77]	70	90	1	980.00	3	0.3136600
WCA [79]	70	90	1	910.00	3	0.3136560
TLBO [78]	70	90	1	810.00	3	0.313656

the MVO with 0.2606. The other algorithms namely WCA, TLBO, and PVS showed roughly similar values for the cost of 0.3136560, 0.313656, and 0.313660, respectively. These results confirm that the proposed cSG algorithm is superior to the other meta-heuristic techniques and demonstrates very competitive results compared to all others when solving the problem of multi-plate disc clutch brake.

According to the previous experiments, the superiority and dominance of the proposed cSG algorithm can be summarized by two factors. The first is the use of crossover operators to prevent being caught in a local optimum through increasing the diversity for population of the SSA. The second one is the utilization of crossover SSA as a local search for GOA, which leads to promoting the GOA's capacity to maintain its population as well as find the best solution. Switching between these factors enhances the cSG's behavior and allows it to retain the benefits of the crossover operators, SSA, and GOA. In addition, the simplicity of the cSG, few predefined parameters and its ability to escape from local optima, help in giving stable results.

V. CONCLUSION

In this work, an improved grasshopper optimization algorithm (GOA) is proposed by applying the crossover operators to maintain the population of the salp swarm algorithm (SSA), then the improved SSA is applied as local search to the original GOA. This improvement prevents the proposed method from premature convergence, the high computation time, besides getting trapped in local minima. The proposed method was applied to solve 29 global optimization problems, feature selection (FS) tasks, and real-engineering problems. For FS problems, eleven well-known benchmark datasets and four performance measures were selected for the experiment. Additionally, the performance of the proposed cSG method was compared with several algorithms in the state-of-the-art. The experimental results demonstrated the superiority of the proposed method compared to other optimization algorithms in all metrics. Moreover, the results showed that using crossover operators with SSA improved the performance of GOA effectively and enhanced the exploration behavior. In future, it would be attractive to examine the performance of cSG method on more advanced science, machine learning tasks in dealing with other datasets, and further improve its complexity with no effect on the performance.

REFERENCES

[1] Q. Al-Tashi, S. J. Abdulkadir, H. M. Rais, S. Mirjalili, and H. Alhussian, "Approaches to multi-objective feature selection: A systematic literature review," *IEEE Access*, vol. 8, pp. 125076–125096, 2020.

[2] Z. M. Elgamal, N. B. M. Yasin, M. Tubishat, M. Alswaitti, and S. Mirjalili, "An improved Harris Hawks optimization algorithm with simulated annealing for feature selection in the medical field," *IEEE Access*, vol. 8, pp. 186638–186652, 2020.

[3] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, Jul. 2018.

[4] X. Zhang, C. Mei, D. Chen, and Y. Yang, "A fuzzy rough set-based feature selection method using representative instances," *Knowl.-Based Syst.*, vol. 151, no. 1, pp. 216–229, Jul. 2018.

[5] M. Mafarja and S. Mirjalili, "Whale optimization approaches for wrapper feature selection," *Appl. Soft Comput.*, vol. 62, pp. 441–453, Jan. 2018.

[6] N. Neggaz, A. A. Ewees, M. A. Elaziz, and M. Mafarja, "Boosting Salp swarm algorithm by sine cosine algorithm and disrupt operator for feature selection," *Expert Syst. Appl.*, vol. 145, May 2020, Art. no. 113103.

[7] A. T. Sahlol, P. Kollmannsberger, and A. A. Ewees, "Efficient classification of white blood cell leukemia with improved swarm optimization of deep features," *Sci. Rep.*, vol. 10, no. 1, pp. 1–11, Dec. 2020.

[8] A. Sharafati, M. Haghbin, N. K. Tiwari, S. K. Bhagat, N. Al-Ansari, K.-W. Chau, and Z. M. Yaseen, "Performance evaluation of sediment ejector efficiency using hybrid neuro-fuzzy models," *Eng. Appl. Comput. Fluid Mech.*, vol. 15, no. 1, pp. 627–643, Jan. 2021.

[9] W. Jing, Z. M. Yaseen, S. Shahid, M. K. Saggi, H. Tao, O. Kisi, S. Q. Salih, N. Al-Ansari, and K.-W. Chau, "Implementation of evolutionary computing models for reference evapotranspiration modeling: Short review, assessment and possible future research directions," *Eng. Appl. Comput. Fluid Mech.*, vol. 13, no. 1, pp. 811–823, Jan. 2019.

[10] Y. Che and D. He, "A hybrid whale optimization with seagull algorithm for global optimization problems," *Math. Problems Eng.*, vol. 2021, pp. 1–31, Jan. 2021.

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

[12] G. G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Comput. Appl.*, vol. 31, no. 7, pp. 1995–2014, 2019.

[13] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, Feb. 2016.

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

[15] H. Faris, M. M. Mafarja, A. A. Heidari, and I. Aljarah, "An efficient binary Salp swarm algorithm with crossover scheme for feature selection problems," *Knowl.-Based Syst.*, vol. 154, pp. 43–67, Aug. 2018.

[16] G.-G. Wang, "Moth search algorithm: A bio-inspired Metaheuristic algorithm for global optimization problems," *Memetic Comput.*, vol. 10, no. 2, pp. 151–164, Jun. 2018.

[17] 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.

[18] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Gener. Comput. Syst.*, vol. 111, pp. 300–323, Oct. 2020.

[19] I. Ahmadianfar, O. Bozorg-Haddad, and X. Chu, "Gradient-based optimizer: A new Metaheuristic optimization algorithm," *Inf. Sci.*, vol. 540, pp. 131–159, Nov. 2020.

[20] Q. Askari, M. Saeed, and I. Younas, "Heap-based optimizer inspired by corporate rank hierarchy for global optimization," *Expert Syst. Appl.*, vol. 161, Dec. 2020, Art. no. 113702.

[21] Y. Yang, H. Chen, A. A. Heidari, and A. H. Gandomi, "Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts," *Expert Syst. Appl.*, vol. 177, Sep. 2021, Art. no. 114864.

[22] I. Ahmadianfar, A. A. Heidari, A. H. Gandomi, X. Chu, and H. Chen, "RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method," *Expert Syst. Appl.*, vol. 181, Nov. 2021, Art. no. 115079.

[23] J. Tu, H. Chen, M. Wang, and A. H. Gandomi, "The colony predation algorithm," *J. Bionic Eng.*, vol. 18, no. 3, pp. 674–710, May 2021.

[24] M. H. Mozaffari, H. Abdy, and S. H. Zahiri, "IPO: An inclined planes system optimization algorithm," *Comput. Inf.*, vol. 35, no. 1, pp. 222–240, May 2016.

[25] M. H. Mozaffari and W. Lee, "Convergent heterogeneous particle swarm optimisation algorithm for multilevel image thresholding segmentation," *IET Image Process.*, vol. 11, no. 8, pp. 605–619, Aug. 2017.

[26] H. Faris, I. Aljarah, and S. Mirjalili, "Improved monarch butterfly optimization for unconstrained global search and neural network training," *Int. J. Speech Technol.*, vol. 48, no. 2, pp. 445–464, Feb. 2018.

[27] M. Alweshah *et al.*, "The monarch butterfly optimization algorithm for solving feature selection problems," *Neural Comput. Appl.*, pp. 1–15, 2020, doi: 10.1007/s00521-020-05210-0.

[28] O. M. Dorgham, M. Alweshah, M. H. Ryalat, J. Alshaer, M. Khader, and S. Alkhalaileh, "Monarch butterfly optimization algorithm for computed tomography image segmentation," *Multimedia Tools Appl.*, vol. 80, pp. 30057–30090, Jan. 2021.

[29] I. Strumberger, E. Tuba, N. Bacanin, M. Beko, and M. Tuba, "Hybridized moth search algorithm for constrained optimization problems," in *Proc. Int. Young Eng. Forum (YEF-ECE)*, May 2018, pp. 1–5.

[30] T. Kundu and H. Garg, "A hybrid ITLHHO algorithm for numerical and engineering optimization problems," *Int. J. Intell. Syst.*, pp. 1–81, 2021, doi: 10.1002/int.22707.

[31] M. Premkumar, P. Jangir, and R. Sowmya, "MOGBO: A new multi-objective gradient-based optimizer for real-world structural optimization problems," *Knowl.-Based Syst.*, vol. 218, Apr. 2021, Art. no. 106856.

[32] A. M. Helmi, M. A. A. Al-qaness, A. Dahou, R. Damaševičius, T. Krilavičius, and M. A. Elaziz, "A novel hybrid gradient-based optimizer and grey wolf optimizer feature selection method for human activity recognition using smartphone sensors," *Entropy*, vol. 23, no. 8, p. 1065, Aug. 2021.

[33] T.-T. Nguyen, H.-J. Wang, T.-K. Dao, J.-S. Pan, J.-H. Liu, and S. Weng, "An improved slime mold algorithm and its application for optimal operation of cascade hydropower stations," *IEEE Access*, vol. 8, pp. 226754–226772, 2020.

[34] M. Abdel-Basset, V. Chang, and R. Mohamed, "HSMA_WOA: A hybrid novel slime mould algorithm with whale optimization algorithm for tackling the image segmentation problem of chest X-ray images," *Appl. Soft Comput.*, vol. 95, Oct. 2020, Art. no. 106642.

[35] M. Abdel-Basset, R. Mohamed, M. Elhoseny, R. K. Chakrabortty, and M. J. Ryan, "An efficient heap-based optimization algorithm for parameters identification of proton exchange membrane fuel cells model: Analysis and case studies," *Int. J. Hydrogen Energy*, vol. 46, no. 21, pp. 11908–11925, Mar. 2021.

[36] F. Kutlu Onay and S. B. Aydemir, "Chaotic hunger games search optimization algorithm for global optimization and engineering problems," *Math. Comput. Simul.*, vol. 192, pp. 514–536, Feb. 2022.

[37] R. Manjula Devi, M. Premkumar, P. Jangir, B. Santhosh Kumar, D. Alrowaili, and K. S. Nisar, "BHGSO: Binary hunger games search optimization algorithm for feature selection problem," *Comput., Mater. Continua*, vol. 70, no. 1, pp. 557–579, 2022.

[38] B. Shi, H. Ye, L. Zheng, J. Lyu, C. Chen, A. A. Heidari, Z. Hu, H. Chen, and P. Wu, "Evolutionary warning system for COVID-19 severity: Colony predation algorithm enhanced extreme learning machine," *Comput. Biol. Med.*, vol. 136, Sep. 2021, Art. no. 104698.

[39] M. Kohli and S. Arora, "Chaotic grey wolf optimization algorithm for constrained optimization problems," *J. Comput. Des. Eng.*, vol. 5, no. 4, pp. 458–472, 2018.

[40] Q.-T. Bui, Q.-H. Nguyen, X. L. Nguyen, V. D. Pham, H. D. Nguyen, and V.-M. Pham, "Verification of novel integrations of swarm intelligence algorithms into deep learning neural network for flood susceptibility mapping," *J. Hydrol.*, vol. 581, Feb. 2020, Art. no. 124379.

[41] Y. Meraihi, A. B. Gabis, S. Mirjalili, and A. Ramdane-Cherif, "Grasshopper optimization algorithm: Theory, variants, and applications," *IEEE Access*, vol. 9, pp. 50001–50024, 2021.

[42] Z. Xu, Z. Hu, A. A. Heidari, M. Wang, X. Zhao, H. Chen, and X. Cai, "Orthogonally-designed adapted grasshopper optimization: A comprehensive analysis," *Expert Syst. Appl.*, vol. 150, Jul. 2020, Art. no. 113282.

[43] S. S. Guo, J. S. Wang, W. Xie, M. W. Guo, and L. F. Zhu, "Improved grasshopper algorithm based on gravity search operator and pigeon colony landmark operator," *IEEE Access*, vol. 8, pp. 22203–22224, 2020.

[44] A. D. Falehi, "Optimal robust disturbance observer based sliding mode controller using multi-objective grasshopper optimization algorithm to enhance power system stability," *J. Ambient Intell. Hum. Comput.*, vol. 11, pp. 5045–5063, Nov. 2020.

[45] P. Mishra, V. Goyal, and A. Shukla, "An improved grasshopper optimization algorithm for solving numerical optimization problems," in *Advances in Intelligent Computing and Communication*. Singapore: Springer, 2020, pp. 179–188.

[46] N. Razmjooy, S. Razmjooy, Z. Vahedi, V. V. Estrela, and G. G. de Oliveira, "Skin color segmentation based on artificial neural network improved by a modified grasshopper optimization algorithm," in *Metaheuristics and Optimization in Computer and Electrical Engineering*. Cham, Switzerland: Springer, 2021, pp. 169–185.

[47] A. A. Ewees, M. A. Elaziz, and E. H. Houssein, "Improved grasshopper optimization algorithm using opposition-based learning," *Expert Syst. Appl.*, vol. 112, pp. 156–172, Dec. 2018.

[48] A. Bala, I. Ismail, R. Ibrahim, S. M. Sait, and D. Oliva, "An improved grasshopper optimization algorithm based echo state network for predicting faults in airplane engines," *IEEE Access*, vol. 8, pp. 159773–159789, 2020.

[49] J. Huang, C. Li, Z. Cui, L. Zhang, and W. Dai, "An improved grasshopper optimization algorithm for optimizing hybrid active power filters' parameters," *IEEE Access*, vol. 8, pp. 137004–137018, 2020.

[50] J. Li, W. Shi, and D. Yang, "Color difference classification of dyed fabrics via a kernel extreme learning machine based on an improved grasshopper optimization algorithm," *Color Res. Appl.*, vol. 46, no. 2, pp. 388–401, Apr. 2021.

[51] S. Z. T. Motlagh and A. Akbari Foroud, "Power quality disturbances recognition using adaptive chirp mode pursuit and grasshopper optimized support vector machines," *Measurement*, vol. 168, Jan. 2021, Art. no. 108461.

[52] 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.

[53] R. A. Ibrahim, A. A. Ewees, D. Oliva, M. Abd Elaziz, and S. Lu, "Improved Salp swarm algorithm based on particle swarm optimization for feature selection," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 8, pp. 3155–3169, Aug. 2019.

[54] A. E. Hegazy, M. A. Makhlouf, and G. S. El-Tawel, "Feature selection using chaotic salp swarm algorithm for data classification," *Arabian J. Sci. Eng.*, vol. 44, no. 4, pp. 3801–3816, Apr. 2019.

[55] Z. M. Yaseen, H. Faris, and N. Al-Ansari, "Hybridized extreme learning machine model with Salp swarm algorithm: A novel predictive model for hydrological application," *Complexity*, vol. 2020, pp. 1–14, Feb. 2020.

[56] G. I. Sayed, G. Khoriba, and M. H. Haggag, "A novel chaotic salp swarm algorithm for global optimization and feature selection," *Int. J. Speech Technol.*, vol. 48, no. 10, pp. 3462–3481, Oct. 2018.

[57] A. G. Hussien, A. E. Hassanien, and E. H. Houssein, "Swarming behaviour of Salps algorithm for predicting chemical compound activities," in *Proc. 8th Int. Conf. Intell. Comput. Inf. Syst. (ICICIS)*, Dec. 2017, pp. 315–320.

[58] S. M. Ismael, S. H. E. A. Aleem, A. Y. Abdelaziz, and A. F. Zobaa, "Practical considerations for optimal conductor reinforcement and hosting capacity enhancement in radial distribution systems," *IEEE Access*, vol. 6, pp. 27268–27277, 2018.

[59] S. Ekinci, B. Hekimoglu, and S. Kaya, "Tuning of PID controller for AVR system using Salp swarm algorithm," in *Proc. Int. Conf. Artif. Intell. Data Process. (IDAP)*, Sep. 2018, pp. 1–6.

[60] Z. Xing and H. Jia, "Multilevel color image segmentation based on GLCM and improved Salp Swarm algorithm," *IEEE Access*, vol. 7, pp. 37672–37690, 2019.

[61] R. Abbassi, A. Abbassi, A. A. Heidari, and S. Mirjalili, "An efficient salp swarm-inspired algorithm for parameters identification of photovoltaic cell models," *Energy Convers. Manage.*, vol. 179, pp. 362–372, Jan. 2019.

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

[63] N. A. Khan, S. P. Ghoshal, and S. Ghosh, "Optimal allocation of distributed generation and shunt capacitors for the reduction of total voltage deviation and total line loss in radial distribution systems using binary collective animal behavior optimization algorithm," *Electr. Power Compon. Syst.*, vol. 43, no. 2, pp. 119–133, Jan. 2015.

[64] R. Purushothaman, S. P. Rajagopalan, and G. Dhandapani, "Hybridizing gray wolf optimization (GWO) with grasshopper optimization algorithm (GOA) for text feature selection and clustering," *Appl. Soft Comput.*, vol. 96, Nov. 2020, Art. no. 106651.

[65] S. Mirjalili, "Genetic algorithm," in *Evolutionary Algorithms and Neural Networks* (Studies in Computational Intelligence), vol. 780. Cham, Switzerland: Springer, 2019, doi: 10.1007/978-3-319-93025-1_4.

[66] S. Forrest, "Genetic algorithms," *ACM Comput. Surv.*, vol. 28, no. 1, pp. 77–80, 1996.

[67] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: A nature-inspired algorithm for global optimization," *Neural Comput. Appl.*, vol. 27, no. 2, pp. 495–513, 2015.

[68] Y. Wang, H.-X. Li, T. Huang, and L. Li, "Differential evolution based on covariance matrix learning and bimodal distribution parameter setting," *Appl. Soft. Comput.*, vol. 18, pp. 232–247, May 2014.

[69] A. W. Mohamed, A. A. Hadi, A. M. Fattouh, and K. M. Jambi, "LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2017, pp. 145–152.

[70] P. Awad, N. Ali, M. Liang, J. Qu, and B. Suganthan, "Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective real-parameter numerical optimization," Nanyang Technol. Univ. Jordan Univ. Sci. Technol. Zhengzhou Univ. Singap. Zhenzhou, China, Tech. Rep. 201611, pp. 1–16, 2016.

[71] D. Dua and C. Graff, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, Irvine, CA, USA, 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[72] K. Deb, "Optimal design of a welded beam via genetic algorithms," *AIAA J.*, vol. 29, no. 11, pp. 2013–2015, 1991.

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

[74] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: A Metaheuristic approach to solve structural optimization problems," *Eng. Comput.*, vol. 29, no. 1, pp. 17–35, Jan. 2013.

[75] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015.

[76] A. Kaveh and M. Khayatazad, "A new meta-heuristic method: Ray optimization," *Comput. Struct.*, vols. 112–113, pp. 283–294, Dec. 2012.

[77] P. Savsani and V. Savsani, "Passing vehicle search (PVS): A novel Metaheuristic algorithm," *Appl. Math. Model.*, vol. 40, nos. 5–6, pp. 3951–3978, Mar. 2016.

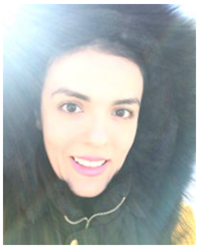
[78] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems," *Comput.-Aided Des.*, vol. 43, no. 3, pp. 303–315, Mar. 2011.

[79] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, "Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems," *Comput. Struct.*, vols. 110–111, pp. 151–166, Nov. 2012.

AHMED A. EWEES (Senior Member, IEEE) received the Ph.D. degree from Damietta University, Egypt. He is currently working as an Associate Professor in computer science with Damietta University. He co-supervises the master's and Ph.D. students and leads and supervises various graduation projects. He has published over 100 scientific research papers in international journals and conferences. His research interests include machine learning, artificial intelligence, text mining, natural language processing, image processing, and metaheuristic optimization techniques.

MARWA A. GAHEEN received the Ph.D. degree from Damietta University, Egypt. She is currently working as an Assistant Professor at Damietta University. She supervises various students graduation projects. She has published different scientific research papers. Her current research interests include image processing, pattern recognition, and artificial intelligence.

ZAHER MUNDHER YASEEN is currently a Lecturer and a Researcher in the field of civil engineering. He is also an Adjunct Research Fellow with the University of Southern Queensland. He has published over 250 research articles within international journals and total number of citations over 7000 (Google Scholar H-index = 48). He has collaborated with over 45 international countries and more than 475 researchers. He has served as a reviewer for more than 135 international journals. His research interests include covering water resources engineering, environmental engineering, knowledge-based system development, and the implementation of data analytic and artificial intelligence models.



RANIA M. GHONIEM received the Ph.D. degree in integrating artificial intelligence techniques in medical diagnosis from Mansoura University, Egypt. She is currently working with the Department of Information Technology, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia. Her research interests include computer-aided diagnostic systems, medical image and sound processing, natural language processing, deep learning, and meta-heuristic optimization methods.