



Improved Salp Swarm Algorithm with mutation schemes for solving global optimization and engineering problems

Bhaskar Nautiyal¹ · Rishi Prakash¹ · Vrinca Vimal² · Guoxi Liang³ · Huiling Chen⁴ 

Received: 2 August 2020 / Accepted: 18 December 2020 / Published online: 7 February 2021
© The Author(s), under exclusive licence to Springer-Verlag London Ltd. part of Springer Nature 2021

Abstract

Salp Swarm Algorithm (SSA) is a recent metaheuristic algorithm developed from the inspiration of salps' swarming behavior and characterized by a simple search mechanism with few handling parameters. However, in solving complex optimization problems, the SSA may suffer from the slow convergence rate and a trend of falling into sub-optimal solutions. To overcome these shortcomings, in this study, versions of the SSA by employing Gaussian, Cauchy, and levy-flight mutation schemes are proposed. The Gaussian mutation is used to enhance neighborhood-informed ability. The Cauchy mutation is used to generate large steps of mutation to increase the global search ability. The levy-flight mutation is used to increase the randomness of salps during the search. These versions are tested on 23 standard benchmark problems using statistical and convergence curves investigations, and the best-performed optimizer is compared with some other state-of-the-art algorithms. The experiments demonstrate the impact of mutation schemes, especially Gaussian mutation, in boosting the exploitation and exploration abilities.

Keywords Salp Swarm Algorithm · Gaussian mutation · Levy-flight mutation · Cauchy mutation

Guoxi Liang and Huiling Chen contributed equally to this work.

✉ Guoxi Liang
guoxiliang2017@gmail.com

✉ Huiling Chen
chenhuiling.jlu@gmail.com

Bhaskar Nautiyal
nautiyal_bhaskar@rediffmail.com

Rishi Prakash
rishi.prakas@gmail.com

Vrinca Vimal
vvimal@ec.iitr.ac.in

¹ Electronics and Communications Engineering, Graphic Era University, Dehradun 248002, India

² Computer Science and Engineering, Graphic Era Hill University, Dehradun 248002, India

³ Department of Information Technology, Wenzhou Polytechnic, Wenzhou 325035, China

⁴ Department of Computer Science, Wenzhou University, Wenzhou 325035, China

1 Introduction

There are many methodologies about how to handle searching for the best solution, such as robust optimization [102], single-objective optimization [30], large scale optimization [17, 19], multiobjective optimization [15], robust optimization [102], memetic optimization [42], many-objective optimization [16, 18], large scale optimization [17, 19], and fuzzy optimization [26]. There are many methods, but optimization cores are a required step in almost any kind of problem in data science and industry. Some examples are not limited to potential directions as image enhancement optimization [131], deployment optimization in sensor networks [14], Artificial Neural Network (ANN) [93, 161], parameter optimization [159], water-energy optimization [25], deep learning tasks [28, 71, 98–100], decision-making processes [77, 79, 139], sustainable development [48, 78, 178], mechanical parameters optimization [13], mechanical and temperature optimization [12], optimal resource allocation [150] and many other domains [21, 49, 84, 96, 143, 148, 149, 157]. Such wide application is that most of the tasks require higher accuracy and strong modeling to understand better the relations among the constraints and objectives systematically [2, 46, 47, 158, 181]. One main

category of optimization methods has a swarm and evolutionary basis. In some past years, swarm intelligence (SI) has gained enormous attention because of its simple and efficient search mechanism [107, 108, 116, 120, 135, 171]. It is the well-known and popular branch of population-based metaheuristic solvers, where multiple search agents participate in the search of the optimization task. Multiple search agents together are called “swarm” in SI-based algorithms [20]. These algorithms are used to solve several important real-life applications in science, engineering, and medicine. In the literature, several SI-based algorithms are developed by inspiring the food foraging and social behaviors of various creatures like bees, ants, hawks, etc.

Some comprehensive examples of SI-based algorithms along with their successful real-world applications are Particle Swarm Optimizer (PSO) [29, 153], Differential Evolution (DE) [118], Differential Search (DS) [75], Ant Colony Optimization (ACO) [173, 176, 177], Harris Hawks Optimizer (HHO)¹ [30, 35, 60], Slime Mould Algorithm (SMA)² [74], Grey Wolf Optimizer (GWO) [8, 22, 57, 58, 119, 175], Whale Optimizer (WOA) [21, 27, 59, 83, 86, 123, 125], Bacterial Foraging Optimization (BFO) [144], Moth-flame Optimizer (MFO) [61, 128, 145, 146, 156, 164, 165], Fruit Fly Optimization (FFO) [31, 36, 37, 110, 134, 155, 169, 170], and Salp Swarm Algorithm (SSA) [6].

Regardless of the variety and different search mechanisms of metaheuristic algorithms, there are two common features: exploration (diversification) and exploitation (intensification), which are responsible for the success of optimization process [44]. Different operators are applied to introduce both of these features in an algorithm and keep an appropriate balance between them. In exploration, the algorithm utilizes different search operators to perform a random search to explore various areas of the solution space deeply. Hence, the exploratory feature of search agents allows finding all possible promising areas of the solution space. On the other hand, the exploitation feature represents the capacity of neighborhood search around the search space’s located regions. This feature is generally performed after the exploration of all the algorithms. Hence, this exploitation can be used to perform a local search in the algorithm. A well-performed algorithm should be capable of establishing an appropriate balance between the exploitation and exploration, and the imbalance between them causes several issues like slow convergence speed, premature convergence, and prone towards the sub-optimal solutions.

The Salp Swarm Algorithm (SSA), inspired from the swarming behavior of salps, is introduced in 2017 by [92]. In the literature, researchers have proposed some modified

variants of the SSA by aiming to remove shortcomings present in the classical SSA. However, the classical SSA has shown a better convergence rate and enough exploration features during the search. Nevertheless, in some cases, it falls into sub-optimal solutions. Therefore, researchers have adopted different operators and search mechanisms to improve their search efficacy and provide better results. To improve the level of exploration as well as exploitation, the SSA is hybridized with PSO [67]. The hybrid algorithm is denoted by the SSAPSO, where both the SSA and PSO’s advantages are utilized to develop comparatively better optimizers. Sayed et al. [109] have embedded the chaos theory in the SSA to speed up the convergence and obtain more accurate optimization results. Utilization of the chaotic signals is a way they employed intending to inject pseudo-random motions into the searching behaviors, based on the well-known chaos-based properties [111, 130, 138, 140]. Tubishat et al. [124] have used the concepts of opposition based learning and new local search strategy to improve the swarm diversity and exploitation capability. Gupta et al. [52] have introduced a new variant of the SSA called harmonized salp chain-built optimization. In this variant, levy-flight search and opposition-based learning increase the convergence speed and avoid falling of salps into sub-optimal solutions. An inertia weight-based new search mechanism is introduced by Hegazy et al. [56] in SSA to adjust the present best salp. This inertia weight is adopted in the SSA to enhance solution accuracy, reliability, and convergence speed. Singh et al. [115] have hybridized the SCA search strategy into the SSA to improve the convergence rate and exploration capabilities. Wu et al. [137] have used the dynamic weight to update the state of salp, and adaptive mutation with an aim to achieve a better balance between exploration and exploitation. The SSA has been applied to intricate domains, and its enhanced variants have exposed trending exploratory searching patterns with global optimization [53, 162] and photovoltaic models [1]. There are also several in-depth studies on the structure and analysis of the SSA, including ensemble mutation-driven SSA with restart mechanisms proposed by [172], which shows a robust efficacy and it can be recognized as the best study on SSA. Also, a multi-strategy SSA proposed by [163] demonstrates that the results are much better than SSA in terms of local optima avoidance. Chaotic multi-swarm SSA [80], multi-objective dynamic SSA [9], time-varying hierarchical SSA [38], asynchronous binary SSA [7], and efficient binary SSA are some of the best research on this algorithm. Along with the improvement in the SSA, it is utilized to solve various real-life problems such as scheduling problems [117], image segmentation [142], feature selection [66], parameter estimation for soil water retention curve [160], training of neural network [4] etc. All these applications demonstrate the

¹ <https://aliasgharheidari.com/HHO.html>.

² <https://aliasgharheidari.com/SMA.html>.

wide applicability of the SSA. For a comprehensive survey on SSA, readers can refer to the literature review at [3, 39].

In the direction of modifying the SSA to obtain better results for the global optimization problem, this paper proposes a new version of the SSA based on mutation schemes. The motivation of this work can be supported with the fact given by No Free Lunch theorem, given by [136]. This theorem permits the modification of the developed algorithms to obtain better optimization results. The proposed strategies of the paper are tested and compared on 23 standard benchmark optimization problems. In addition to this, the better-performed optimizer among the SSA with Gaussian mutation, SSA with Cauchy mutation, and SSA with the levy-flight mutation is compared with some state-of-the-art optimization methods. The results illustrate that the mutation scheme is successful in improving the search efficacy of the classical SSA.

The rest of the paper is divided into four sections: A simple description of the SSA is provided in Sect. 2. Section 3 presents a brief description of the mutation schemes and the framework of proposed mutation-based SSA versions. Section 4 conducts the experiments to test and compare the performance of the proposed mutation-based SSA versions. Finally, Sect. 5 provides the conclusion of the study and suggests some future works.

2 Overview of the Salp Swarm Algorithm (SSA)

The SSA was developed in 2017 [92]. The inspiration behind the proposal of the SSA was the swarming behavior of salps. These salps are free-floating tunicates and barrel-shaped from the family of Salpidae. Generally, these salps float together in the form called a salp chain when foraging and navigating in oceans. The colony of salps moves in this form for better locomotion and foraging. Like other SI-based metaheuristic methods, the SSA also initializes the swarm with a predefined number of salps. Each salp in a swarm represents the search agent, which performs the search process for a targeted optimization problem. In the swarm of salps, two categories of salps are present: leading salps and follower salps. During the search procedure, follower salps follow the leading salps to allocate the optimal solution. The swarm S consisting of n salps is represented as follows:

$$S = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1D} \\ a_{21} & a_{22} & \cdots & x_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & x_{ND} \end{pmatrix} \tag{1}$$

In the mathematical model of the SSA, two concepts are adopted in which the work followed by the leading salps and follower salps are modeled mathematically. The leading salps update their states with the help of Eq. (2)

$$L_j = \begin{cases} F_j + r_1 \times (lb + r_2 \times (ub - lb)) & r_3 \geq 0.5 \\ F_j - r_1 \times (lb + r_2 \times (ub - lb)) & r_3 < 0.5 \end{cases} \tag{2}$$

where, L_j and F_j are the j th coordinates for the states of leading salps and food sources, respectively. ub and lb are the upper and lower boundary limits for the solution space. r_2 and r_3 are the random numbers between 0 and 1. r_1 is a variable decreases when the iterations increase. Its mathematical formulation is given by Eq. (3)

$$r_1 = 2 \times \exp \left[-\left(\frac{4t}{T} \right)^2 \right] \tag{3}$$

where t and T are the current iteration number and maximum iterations, respectively.

In the second phase of the search, the follower salps update their stats. They utilize Newton’s law of motion given by

$$X_{ij} = \frac{1}{2}at^2 + v_0t \tag{4}$$

where $a = v_f - v_0/\delta t$ and $v_0 = (x - x_0)/t$. The time in optimization process is referred as iteration and therefore, the discrepancy between iterations is 1. Considering $v_0 = 0$, Eq. 4 becomes Eq. (5)

$$X_{ij} = \frac{1}{2}(X_{ij} + X_{i-1,j}) \tag{5}$$

where, value of i is more than 1. X_{ij} and $X_{i-1,j}$ represents the j th coordinate of the follower salps i and $(i - 1)$, respectively. Hence, like other swarm intelligence based methods, SSA first initializes the swarm of salps within a provided solution space. In the second step, the leading salps and followers update their states to re-positioned at better locations. This process continues until the prefixed maximum iterations are not completed. The pseudo-code of the classical SSA is presented in Algorithm 1

Algorithm 1 Pseudo-code of the classical SSA

```

Initialize the swarm size and positions of salps
Initialize the algorithm parameters:  $r_1$  and maximum iterations  $T$ 
while ( $t < T$ ) do
    Calculate the fitness value of each salp
    Update the position of leading salps using eq. (2)
    Update the position of follower salps using eq. (5)
    Manage the boundary of salp positions if crossed the boundary
Return the best salp position  $x_{Best}$ 

```

3 Proposed improved Salp Swarm Algorithm with mutation strategies

Although the classical SSA enriches with some characteristics like fast convergence speed and simple implementation, it may trap at sub-optimal solutions easily in few cases when handling the more complex optimization problems. The interaction between the leading and follower salps characterize the performance of the SSA. If a single leading salp trap at the sub-optimal solution, a salp can prevent from that local solution by the pull effect of leading salps. However, when the whole swarm of salps falls into a sub-optimal solution, the algorithm is trapped at that local solution and eventually stagnate at that sub-optimal solution.

To explore the solution space more effectively, this paper introduces a strategy called mutation into SSA. The three different mutation schemes, namely, Cauchy Mutation, Gaussian Mutation, and levy-flight mutation, are embedded into the classical SCA. The developed versions are denoted by Cauchy-SSA (CSSA), Gaussian-SSA (GSSA), and levy-SSA (LSSA), respectively. In the proposed method, before applying the mutation scheme, the greedy search is adopted between the states X^t of t th iteration and X^{t+1} of $(t + 1)$ th iteration, using the following Eq. (6)

$$Y^{t+1} = \begin{cases} X^t & \text{if } f(X^t) < f(X^{t+1}) \\ X^{t+1} & \text{if } f(X^{t+1}) < f(X^t) \end{cases} \quad (6)$$

When the greedy search is completed corresponding to each salp, the mutation scheme is applied with mutation rate (m_r). The increasing value of the mutation rate is the cause of high diversity and helps complex and large-dimensional optimization problems. During the mutation scheme, new mutated salps are generated and compared with the parent salps. If the newly obtained mutated salp is found better than the parent salp in the sense of fitness, it replaces the parent salp otherwise discarded, and original salps are retained. This

rule is utilized in the proposed improved SSA during each mutation scheme.

3.1 Gaussian-SSA (GSSA)

In this section, the Gaussian mutation [62], often used in GA and PSO, is used to mutate the salp based on mutation rate. We aim to do not apply the process by which every salp of the classical SSA travels to another state within the solution space with a predefined mutation rate without being affected by other salps but leave a certain vagueness in the transition to the next iteration due to Gaussian mutation. This mutation follows the following eq.

$$\hat{x}_i = x_i \times (1 + \text{Gaussian}(\delta)) \quad (7)$$

where x_i denotes the i th salp and $\text{Gaussian}(\delta)$ is a random number generated using the Gaussian distribution. The density function of Cauchy distribution is given by Eq. (8).

$$f_{\text{Gaussian}(0,\sigma^2)}(\beta) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(\beta - \mu)^2}{2\sigma^2}\right) \quad (8)$$

where σ^2 is a variance for each salp. To generate random numbers, the above equation is reduced with a Avg $\mu = 0$ and standard deviation σ to 1. The Gaussian mutation is integrated to cope with the diversity loss during the search process. In our approach, this mutation is used to locally explore the search space around the visited regions of the solution space.

3.2 Cauchy-SSA (CSSA)

Similar to other swarm intelligence methods and other metaheuristics, the SSA also tends to fall into a sub-optimal solution due to insufficient diversity and escaping ability from the sub-optimal regions. Therefore, we should adopt some strategy that provides a high jump sometimes throughout the search process. The Cauchy distribution can help in

this situation as it generates large values infrequently, which can provide a large step-size of mutation [51, 147]. In this Cauchy-SSA, a random number is generated, and if its value allows the mutation based on the mutation rate, then each salp of the swarm is mutated as follows

$$\hat{x}_i = x_i \times (1 + \text{Cauchy}(\delta)) \tag{9}$$

where x_i denotes the i th salp and $\text{Cauchy}(\delta)$ is a random number generated using the Cauchy distribution function given by

$$y = \frac{1}{\pi} \arctan\left(\frac{\alpha - \alpha_0}{\gamma}\right) + \frac{1}{2} \tag{10}$$

and the density function (DF) is given by equation (11)

$$f_{\text{cauchy}(0,\gamma)}(\alpha) = \frac{1}{\pi} \frac{\gamma}{\gamma^2 + \alpha^2} \tag{11}$$

where y is a uniformly distributed random number within (0,1). $\alpha = 0$ is a location parameter and $\gamma = 1$ is a scale parameter [126]. This Cauchy mutation generates higher chances of making longer jumps as compared to the Gaussian mutation.

3.3 Levy-SSA (LSSA)

In this section, the levy-mutation [82, 151] is used to improve salps diversity in the SSA. The levy-mutation can handle the global search more effectively by mutating the

salps when the mutation rate allows. Each salp in the levy-SSA is mutated as follows

$$\hat{x}_i = x_i \times (1 + \text{Levy}(\delta)) \tag{12}$$

where x_i denotes the i th salp and $\text{Levy}(\delta)$ is a random number generated using the levy distribution function. A simplified version of the levy distribution is defined by Eq. (13)

$$\text{Levy}(\beta) \sim y = t^{-\beta-1}, \quad 0 < \beta \leq 1 \tag{13}$$

where β is stability index. The Levy-distributed random number can be obtained using

$$\text{Levy}(\beta) \sim \frac{\psi \times u}{|v|^{1/\beta}} \tag{14}$$

where u and v are standard normal distribution. The value of ψ is defined by Eq. (15)

$$\psi = \left[\frac{\Gamma(1 + \beta) \sin(\pi\beta/2)}{\Gamma((\frac{1+\beta}{2}) \times \beta \times 2^{(\beta-1)/2})} \right]^{1/\beta} \tag{15}$$

The value of β is fixed to 1.5. Normally, the levy-mutation generates different offspring salps as it is long tailed distribution. This feature is helpful to jump out from sub-optimal regions when the stagnation occurs during the search.

A general framework of all the mutation-based SSA is presented in Algorithm 2.

Algorithm 2 Pseudo-code of the mutation-based SSA:GSSA, CSSA and LSSA

```

Initialize the swarm size and positions of salps
Initialize the algorithm parameters:  $r_1$ , mutation rate  $m_r$  and maximum iterations  $T$ 
while ( $t < T$ ) do
    Calculate the fitness value of each salp
    Update the position of leading salps using equation (2)
    Update the position of follower salps using equation (5)
    Manage the boundary of salp positions if crossed the boundary
    Apply the greedy search corresponding to each salp position
    if  $rand < m_r$  then
        for (each salp ( $x_i$ )) do
            Apply one of the mutation strategy to generate mutant salp as explained in Subsections 3.1, Subsections 3.2 and Subsections 3.3 using one of the equations
             $\hat{x}_i = x_i \times (1 + \text{Gaussian}(\delta))$ 
             $\hat{x}_i = x_i \times (1 + \text{Cauchy}(\delta))$ 
             $\hat{x}_i = x_i \times (1 + \text{Levy}(\delta))$ 
            Replace the parent salp with the generated mutant salp if its fitness found better
    Return the elite salp  $x_{Best}$ 
    
```

Table 1 Description of unimodal benchmark functions

Function	Dimensions	Range	f_{\min}
$f_1(x) = \sum_{i=1}^D x_i^2$	30,100	[-100, 100]	0
$f_2(x) = \sum_{i=1}^D ix_i^2$	30,100	[-10, 10]	0
$f_3(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	30,100	[-10, 10]	0
$f_4(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j\right)^2$	30,100	[-100, 100]	0
$f_5(x) = \max_i \{ x_i , 1 \leq i \leq D\}$	30,100	[-100, 100]	0
$f_6(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30,100	[-30, 30]	0
$f_7(x) = \sum_{i=1}^D ([x_i + 0.5])^2$	30,100	[-100, 100]	0
$f_8(x) = \sum_{i=1}^D ix_i^4$	30,100	[-1.28, 1.28]	0
$f_9(x) = \sum_{i=1}^D ix_i^4 + \text{random}[0, 1]$	30,100	[-1.28, 1.28]	0
$f_{10}(x) = \sum_{i=1}^D x_i ^{i+1}$	30,100	[-1, 1]	0
$f_{11}(x) = \sum_{i=1}^D -x_i \sin(\sqrt{ x_i })$	30,100	[-500, 500]	$-418.9829 \times D$
$f_{12}(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30,100	[-5.12, 5.12]	0
$f_{13}(x) = -20 \exp(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$	30,100	[-32, 32]	0
$f_{14}(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30,100	[-600, 600]	0
$f_{15}(x) = \frac{\pi}{D} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_D - 1)^2 \right\}$ $+ \sum_{i=1}^D u(x_i, 10, 100, 4) y_i = 1 + \frac{x_i+1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 - a & < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30,100	[-50, 50]	0
$f_{16}(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^D (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_D - 1)^2 [1 + \sin^2(2\pi x_D)] \right\}$ $+ \sum_{i=1}^D u(x_i, 5, 100, 4)$	30,100	[-50, 50]	0
$f_{17}(x) = \sum_{i=1}^D x_i \sin(x_i) + 0.1 x_i $	30,100	[-10, 10]	0
$f_{18}(x) = \sum_{i=1}^D 0.1 D - \left(0.1 \sum_{i=1}^D \cos(5\pi x_i) - \sum_{i=1}^D x_i^2\right)$	30,100	[-1, 1]	0
$f_{19}(x) = \sum_{i=1}^{D-1} (x_i^2 + 2x_{i+1}^2)^{0.25} \times [1 + \sin(50(x_i^2 + x_{i+1}^2)^{0.1})]^2$	30,100	[-10, 10]	0
$f_{20}(x) = \sum_{i=1}^D (10^6)^{(i-1)/(D-1)} x_i^2$	30,100	[-100, 100]	0
$f_{21}(x) = (-1)^{D+1} \prod_{i=1}^D \cos(x_i) \times \exp(-\sum_{i=1}^D (x_i - \pi)^2)$	30,100	[-100, 100]	0
$f_{22}(x) = 1 - \cos\left(2\pi \sqrt{\sum_{i=1}^D x_i^2}\right) + 0.1 \sqrt{\sum_{i=1}^D x_i^2}$	30,100	[-100, 100]	0
$f_{23}(x) = 0.5 + \frac{\sin^2\left(\sqrt{\sum_{i=1}^D x_i^2}\right) - 0.5}{1 + 0.001\left(\sum_{i=1}^D x_i^2\right)^2}$	30,100	[-100, 100]	0

3.4 Computational complexity

To determine the computational complexity of the proposed mutation-based SSA, mainly seven components are used: Initialization of salp swarm, fitness evaluation of each salp, position update of leading salps, position update of follower salps, fitness evaluation of updated salps, greedy search, mutation scheme and memorization of elite salp. The complexity of initialization of swarm is $O(N \times D)$, fitness

evaluation of salp utilizes $O(N)$ computational effort. The complexity of the position update process of leading and follower salp is $O(N \times D)$, fitness evaluation of updated salp takes $O(N)$ computational effort, the complexity of both the schemes greedy search and mutation scheme is $O(N)$. The computational effort of the memorization of elite salp is $O(N)$. Hence, by summing all, the complexity of the proposed mutation-based SSA is $O(N \times D \times T)$ same to the classical SSA.

4 Experimental results and validation of proposed mutation-based variants

In this section, the proposed mutation-based SSA variants are evaluated and tested in three phases. In the first phase, a comparison is conducted among classical SSA and all mutation-based variants. The best performer optimizer is selected from here, which is better than the remaining mutation-based variants and classical SSA. In the second phase, the winner best performer variant of the SSA is compared with some state-of-the-art optimization methods. In the third and last phase, all the SSA variants, classical SSA, and other state-of-the-art algorithms are used to solve some real engineering test cases. The benchmark comparison was conducted on a set of 23 scalable benchmark problems. These benchmarks are provided in Table 1. The source of these benchmark problems is [54, 55, 81]. We follow the same testing way for all the compared methods [112–114]. This is an accepted way to ensure all methods take the same advantage (or no competitive advantage) according to the user system and conditions [85, 94, 154, 166, 167]. Experimental results obtained from different SSA versions based on mutation scheme in terms of the Best, average (Avg.), and standard deviation (Std.) of the objective function are used to evaluate these versions' potential. The best results are highlighted in bold. Furthermore, a non-parametric statistical test Wilcoxon signed-rank test [45] at 0.05 significance level is employed to investigate that the achieved results are significantly better or not. To represent these statistical results, “+ / - / =” symbols are used, which are presented in Table 4 and indicate that our proposed method is superior, worse, and statistically same to its competitive optimization method, respectively.

4.1 Comparison among classical SSA, GSSA, CSSA, and LSSA

This section compares the classical SSA with mutation-based SSA versions such as GSSA, CSSA, and LSSA on a set of 23 benchmarks given in Table 1. Many numerical optimization methods also use these test problems. Furthermore, a comparison between the GSSA, CSSA, and LSSA is also conducted to select the best performer optimization method. In these experiments, the dimension of test functions is taken to 30 and 100. The swarm size is set to 30. Maximum iterations and maximum function evaluations are set to 500 and 15,000, respectively. As can be observed from Tables 2–3 that the proposed mutation-based SSA variants outperform the classical SSA on approximately 92% at dimension 30 problems with the same results on remaining one problem and worse results on one problem F7 only. However, on dimension 100 problems, the mutation-based

SSA variants have outperformed on 100% of problems. It is noticed that the GSSA has the smallest Avg. objective function value than other mutation variants on 20 problems out of 23 for both dimensions 30 and 100. In addition to this, the GSSA has provided a near-optimal solution to most of the test problems. Hence, due to this comparison, the GSSA is selected for future comparison perspectives with other swarm-intelligence-based methods. To demonstrate the superiority of the proposed GSSA in terms of convergence rate, the convergence curves are plotted in Figs. 1 and 2. In this figure, the Avg. value of best objective function obtained in 30 trials is shown and compared for classical SCA, GSSA, CSSA, and LSSA. From these figures, it can be seen that according to the convergence rate, the GSSA takes first place, followed successively by CSSA, LSSA, and classical SSA. Obviously, the fast convergence rate results from the applied mutation scheme and greedy search approach in the proposed method. Hence, from the curves, it can be seen that the mutation scheme has improved the convergence rate of the proposed method, but the more effective is due to the Gaussian mutation rule. Furthermore, the Wilcoxon signed-rank test is utilized to determine whether the GSSA is performed better than the other mutation variants or not? The results obtained by employing this test between A and B (A/B) methods are listed by the symbols “+ / - / =” to indicate that the A is significantly better, worse, or equal to its competitive method. All the results are listed in Table 4 corresponding to the dimension 30 and 100. By this table, it has been found that the GSSA has outperformed classical SCA on 20 problems for 30 dimensions and 21 problems for 100 dimensions. Compared with CSSA and LSSA, GSSA is excellent in providing significantly better results or statistically the same results. It is clear from the table that GSSA is not worse, even on a single problem. Moreover, out of the 23 test problems, CSSA has outperformed LSSA on 15 problems, statistically the same on seven problems and worse on only one problem.

4.2 Comparison with other metaheuristic methods

The comparison conducted above illustrates the superior solution accuracy by the GSSA among classical SCA and other mutation-based variants. In this section, the same set of benchmark problems with dimension 100 is used to compare the results of the GSSA with other metaheuristic methods on the same parameter environment (population size and function evaluations) as utilized in the previous section. The metaheuristic methods which are used for comparison in this section are: Firefly Algorithm (FA) [151], Grey Wolf Optimizer (GWO) [91], Moth-flame Optimizer (MFO) [89], Sine Cosine Algorithm (SCA) [90], Teaching-learning-based Optimization (TLBO) [103], Hybrid SSA with SCA (mod-SSA) [115] and Improved Salp Swarm Algorithm (ISSA)

Table 2 Optimization results on 23 problems for the dim 30

Problem	Results	SSA	GSSA	CSSA	LSSA
F1	Best	3.39E-08	6.84E-124	5.02E-93	3.66E-67
	Avg	2.76E-07	3.25E-109	1.27E-84	1.21E-56
	Std.	6.18E-07	1.20E-108	3.14E-84	3.86E-56
F2	Best	4.88E-02	7.94E-122	1.89E-91	1.15E-64
	Avg	2.86E+00	1.56E-109	1.29E-84	3.79E-56
	Std.	3.28E+00	4.50E-109	4.99E-84	2.03E-55
F3	Best	1.19E-01	3.21E-61	5.16E-47	7.14E-34
	Avg	2.37E+00	4.08E-54	1.22E-42	3.04E-29
	Std.	1.64E+00	2.05E-53	3.44E-42	8.89E-29
F4	Best	4.83E+02	2.20E-123	1.80E-90	2.58E-63
	Avg	1.78E+03	3.88E-106	8.04E-83	3.37E-55
	Std.	1.27E+03	2.11E-105	4.36E-82	1.10E-54
F5	Best	4.49E+00	2.51E-60	7.92E-48	7.90E-33
	Avg	1.21E+01	7.76E-56	6.56E-42	1.54E-28
	Std.	3.51E+00	2.46E-55	2.51E-41	7.90E-28
F6	Best	2.48E+01	2.89E+01	2.89E+01	2.89E+01
	Avg	2.85E+02	2.90E+01	2.90E+01	2.90E+01
	Std.	6.26E+02	2.28E-02	3.25E-02	2.44E-02
F7	Best	3.40E-08	1.66E-01	1.43E-01	2.57E-01
	Avg	4.17E-07	9.66E-01	9.14E-01	9.38E-01
	Std.	1.46E-06	5.77E-01	4.96E-01	6.20E-01
F8	Best	8.43E-18	1.58E-251	2.96E-192	1.37E-134
	Avg	1.64E-11	9.69E-215	3.81E-170	7.98E-117
	Std.	3.45E-11	0.00E+00	0.00E+00	3.23E-116
F9	Best	7.14E-02	7.75E-05	5.12E-05	7.75E-05
	Avg	1.74E-01	4.16E-04	5.02E-04	1.10E-03
	Std.	6.43E-02	3.69E-04	4.75E-04	9.85E-04
F10	Best	3.55E-07	1.21E-181	6.33E-148	3.89E-101
	Avg	2.04E-06	2.78E-165	4.11E-129	7.83E-87
	Std.	2.10E-06	0.00E+00	2.21E-128	2.29E-86
F11	Best	- 8.77E+03	- 9.47E+03	- 8.84E+03	- 9.47E+03
	Avg	- 7.58E+03	- 7.46E+03	- 7.59E+03	- 7.54E+03
	Std.	6.82E+02	8.53E+02	8.17E+02	7.50E+02
F12	Best	3.48E+01	0.00E+00	0.00E+00	0.00E+00
	Avg	5.78E+01	0.00E+00	0.00E+00	0.00E+00
	Std.	1.58E+01	0.00E+00	0.00E+00	0.00E+00
F13	Best	1.34E+00	8.88E-16	8.88E-16	8.88E-16
	Avg	2.62E+00	8.88E-16	8.88E-16	8.88E-16
	Std.	6.76E-01	0.00E+00	0.00E+00	0.00E+00
F14	Best	7.72E-04	0.00E+00	0.00E+00	0.00E+00
	Avg	1.63E-02	0.00E+00	0.00E+00	0.00E+00
	Std.	1.28E-02	0.00E+00	0.00E+00	0.00E+00
F15	Best	2.20E+00	4.81E-03	3.39E-03	7.99E-03
	Avg	7.20E+00	2.34E-02	2.64E-02	5.46E-02
	Std.	3.48E+00	1.56E-02	2.62E-02	6.32E-02
F16	Best	4.21E-02	9.99E-02	6.69E-02	1.12E-01
	Avg	1.62E+01	2.43E+00	2.14E+00	2.62E+00
	Std.	1.55E+01	1.16E+00	1.33E+00	9.68E-01
F17	Best	1.08E+00	3.75E-61	1.12E-46	4.16E-33
	Avg	3.87E+00	3.42E-56	1.35E-42	9.47E-30
	Std.	1.63E+00	9.44E-56	4.93E-42	2.35E-29

Table 2 (continued)

Problem	Results	SSA	GSSA	CSSA	LSSA
F18	Best	4.43E-01	0.00E+00	0.00E+00	0.00E+00
	Avg	1.22E+00	0.00E+00	0.00E+00	0.00E+00
	Std.	3.87E-01	0.00E+00	0.00E+00	0.00E+00
F19	Best	1.99E+01	5.37E-124	1.04E-94	1.47E-63
	Avg	5.95E+01	5.35E-106	6.97E-81	9.31E-55
	Std.	3.18E+01	2.28E-105	3.79E-80	4.67E-54
F20	Best	3.01E+06	4.58E-116	4.97E-86	3.37E-60
	Avg	1.29E+07	3.40E-105	5.53E-78	7.37E-53
	Std.	8.01E+06	1.47E-104	1.90E-77	1.96E-52
F21	Best	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Avg	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F22	Best	1.30E+00	1.12E-52	3.46E-37	3.53E-22
	Avg	1.93E+00	1.81E-43	1.08E-28	2.46E-12
	Std.	4.04E-01	9.82E-43	5.84E-28	9.42E-12
F23	Best	2.73E-01	0.00E+00	0.00E+00	0.00E+00
	Avg	3.99E-01	0.00E+00	0.00E+00	3.96E-03
	Std.	4.25E-02	0.00E+00	0.00E+00	4.80E-03

[56]. Each of these algorithms is independently 30 times implemented on a benchmark set, and the simulated results in terms of Best, Avg, and Std. are presented in Table 5. To validate that the GSSA has outperformed other metaheuristic algorithms, a non-parametric Wilcoxon signed rank test is used at a 0.05 significance level. These statistical results are presented in Table 6 with p-values and symbols “+ / - / ≈” to indicate that the GSSA is significantly superior, equal or same as its competitive method.

The table indicates that the proposed GSSA has outperformed FA, GWO, MFO, SCA, TLBO, mod-SSA, and ISSA on 22, 19, 21, 22, 14, 19, and 18 problems and inferior to them on 0, 3, 1, 0, 5, 2 and 3 problems, respectively. Thus, the results conclude that the proposed Gaussian mutation-based SSA (GSSA) is superior in solution accuracy to its competitive metaheuristic methods.

4.3 Application of proposed GSSA on engineering design problems

In this subsection, the proposed GSSA is applied to optimize three engineering design cases with constraints such as three-bar truss design, pressure vessel design, and speed reducer design problem. These optimization cases consist of some inequality and equality constraints [182], so the constraint handling method should be employed in the GSSA. The methods based on giving a penalty to the objective function to construct a fitness function can be used to tackle such situations. In this study, the death penalty is a popular and easiest method to deal with the constraints [34]. In this approach, the SSA will

automatically discard the infeasible solutions. This approach has the advantages of small computation and simple calculation. However, this approach does not take advantage of the information of infeasible solutions, which may be useful in solving problems with dominated infeasible areas of the solution space. To verify its efficacy, GSSA is merged with the death penalty approach to solving constrained engineering cases.

4.3.1 Three-bar truss design

This design case was firstly introduced by Nowacki [95]. In this, the objective function is given by minimizing the volume of a statically loaded 3-bar truss by minimizing the cross-sectional area $f(A_1, A_2)$ with restrictions in the form of stress constraints on each truss member. The mathematical formulation of the problem is given as follows:

$$\text{Minimize } f(A_1, A_2) = (2\sqrt{2}) \times A_1 + A_2 \times l \tag{16}$$

subject to

$$g_1(A_1, A_2) = \frac{\sqrt{2}A_1 + A_2}{2A_1A_2 + \sqrt{2}A_1^2} \times P - \sigma \leq 0 \tag{17}$$

$$g_2(A_1, A_2) = \frac{A_2}{2A_1A_2 + \sqrt{2}A_1^2} \times P - \sigma \leq 0 \tag{18}$$

Table 3 Optimization results on 23 problems for the dim 100

Problem	Results	SSA	GSSA	CSSA	LSSA
F1	Best	8.08E+02	3.34E–118	2.79E–90	1.39E–64
	Avg	1.42E+03	6.41E–104	6.61E–81	6.21E–55
	Std.	3.81E+02	2.41E–103	3.58E–80	1.96E–54
F2	Best	5.18E+02	1.49E–116	5.51E–90	1.18E–62
	Avg	8.87E+02	5.40E–106	8.33E–79	1.21E–54
	Std.	2.43E+02	2.28E–105	4.56E–78	4.04E–54
F3	Best	3.67E+01	1.25E–60	1.05E–45	1.40E–33
	Avg	4.65E+01	6.21E–54	2.58E–41	4.47E–28
	Std.	7.30E+00	2.27E–53	9.54E–41	8.96E–28
F4	Best	1.88E+04	2.15E–115	1.30E–89	3.24E–62
	Avg	5.06E+04	2.22E–105	1.47E–78	3.58E–51
	Std.	2.22E+04	1.11E–104	7.98E–78	1.93E–50
F5	Best	1.96E+01	3.77E–59	1.55E–45	3.03E–31
	Avg	2.77E+01	9.98E–53	6.19E–42	1.52E–28
	Std.	3.99E+00	5.30E–52	2.33E–41	5.07E–28
F6	Best	7.00E+04	9.89E+01	9.89E+01	9.89E+01
	Avg	1.91E+05	9.90E+01	9.90E+01	9.90E+01
	Std.	7.35E+04	1.51E–02	2.27E–02	1.74E–02
F7	Best	6.96E+02	1.04E+01	1.06E+01	1.10E+01
	Avg	1.61E+03	1.50E+01	1.54E+01	1.50E+01
	Std.	4.80E+02	2.82E+00	2.64E+00	2.65E+00
F8	Best	9.56E–02	4.11E–241	1.23E–185	2.99E–139
	Avg	3.10E–01	8.98E–214	1.04E–168	1.08E–114
	Std.	1.75E–01	0.00E+00	0.00E+00	5.63E–114
F9	Best	1.36E+00	2.15E–05	2.22E–05	2.90E–05
	Avg	2.70E+00	3.86E–04	3.67E–04	1.02E–03
	Std.	6.21E–01	3.40E–04	2.91E–04	1.03E–03
F10	Best	4.18E–07	1.77E–179	1.86E–145	3.35E–100
	Avg	3.15E–06	7.05E–157	9.07E–129	1.70E–84
	Std.	2.81E–06	3.86E–156	4.83E–128	8.38E–84
F11	Best	– 2.44E+04	– 2.49E+04	– 2.38E+04	– 2.49E+04
	Avg	– 2.15E+04	– 2.06E+04	– 2.09E+04	– 2.14E+04
	Std.	1.58E+03	1.81E+03	1.48E+03	2.02E+03
F12	Best	1.41E+02	0.00E+00	0.00E+00	0.00E+00
	Avg	2.50E+02	0.00E+00	0.00E+00	0.00E+00
	Std.	4.99E+01	0.00E+00	0.00E+00	0.00E+00
F13	Best	7.06E+00	8.88E–16	8.88E–16	8.88E–16
	Avg	1.06E+01	8.88E–16	8.88E–16	8.88E–16
	Std.	1.41E+00	0.00E+00	0.00E+00	0.00E+00
F14	Best	9.37E+00	0.00E+00	0.00E+00	0.00E+00
	Avg	1.44E+01	0.00E+00	0.00E+00	0.00E+00
	Std.	2.87E+00	0.00E+00	0.00E+00	0.00E+00
F15	Best	1.69E+01	8.55E–02	7.87E–02	1.08E–01
	Avg	3.36E+01	1.42E–01	1.30E–01	1.48E–01
	Std.	9.54E+00	3.33E–02	3.22E–02	3.13E–02
F16	Best	2.17E+02	9.99E+00	9.99E+00	9.99E+00
	Avg	9.62E+03	1.00E+01	9.99E+00	1.00E+01
	Std.	1.70E+04	2.44E–03	2.67E–03	2.30E–03
F17	Best	1.54E+01	2.06E–60	1.75E–46	1.54E–32
	Avg	2.87E+01	4.76E–55	4.13E–43	8.84E–29
	Std.	4.82E+00	2.35E–54	1.18E–42	3.55E–28

Table 3 (continued)

Problem	Results	SSA	GSSA	CSSA	LSSA
F18	Best	3.61E+00	0.00E+00	0.00E+00	0.00E+00
	Avg	4.89E+00	0.00E+00	0.00E+00	0.00E+00
	Std.	7.22E−01	0.00E+00	0.00E+00	0.00E+00
F19	Best	1.53E+03	3.50E−119	7.06E−93	6.18E−65
	Avg	1.79E+03	1.36E−108	2.48E−82	4.10E−55
	Std.	1.89E+02	4.91E−108	1.08E−81	1.74E−54
F20	Best	3.10E+07	7.60E−112	3.54E−85	9.74E−58
	Avg	1.06E+08	1.90E−101	2.77E−75	6.57E−51
	Std.	5.28E+07	6.26E−101	1.52E−74	2.31E−50
F21	Best	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Avg	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F22	Best	1.02E+01	1.42E−51	6.80E−36	3.16E−22
	Avg	1.21E+01	5.84E−45	9.22E−27	2.36E−10
	Std.	1.07E+00	2.77E−44	3.52E−26	1.15E−09
F23	Best	4.98E−01	0.00E+00	0.00E+00	0.00E+00
	Avg	4.99E−01	0.00E+00	0.00E+00	5.60E−03
	Std.	2.81E−04	0.00E+00	0.00E+00	4.82E−03

$$g_3(A_1, A_2) = \frac{1}{A_1 + \sqrt{2}A_2} \times P - \sigma \leq 0 \tag{19}$$

where $0 \leq A_1, A_2 \leq 1$, $l = 100$ cm, $P = \sigma = 2$ KN/cm².

The problem formulation shows that this is a non-linear optimization problem with the continuous nature of decision variables. This problem has been solved by [105], and by [121]. In the study of [97] this problem is also attempted to solve. Furthermore, the cuckoo search metaheuristic method [43] is also adopted to solve this problem. [152] have used the bat algorithm (BA) to solve this problem. [50] have used the classical sine cosine algorithm (SCA) and their modified SCA (m-SCA) to solve this problem. In this paper, the GSSA is applied to this problem using 25 salp swarm size and 1000 iterations. Numerical results of all the optimization methods are presented in Table 7. The table indicates that the GSSA is superior to provide the best results than other methods. It also verified from this study that the proposed GSSA could deal with the constraints of this optimization case more effectively.

4.3.2 Tension-compression spring design

Another engineering case optimization problem is a tension-compression spring design. In this problem, the optimization task is described by the minimization of the heaviness of a spring. Three decision variables: diameter d , mean coil diameter D , and the number of dynamic coils N are involved

in this problem. The mathematical description of this case problem is as follows:

$$\text{Minimize } f(d, D, N) = (N + 2)d^2D \tag{20}$$

subject to

$$g_1(d, D, N) = 1 - \frac{D^3N}{71785d^4} \leq 0 \tag{21}$$

$$g_2(d, D, N) = \frac{4D^2 - dD}{12566(d^3D - d^4)} + \frac{1}{5108d^2} \leq 0 \tag{22}$$

$$g_3(d, D, N) = 1 - \frac{140.45d}{D^3N} \leq 0 \tag{23}$$

$$g_4(d, D, N) = \frac{d + D}{1.5} - 1 \leq 0 \tag{24}$$

where

$$0.05 \leq d \leq 2.0, 0.25 \leq D \leq 1.30, 2.0 \leq N \leq 15.0.$$

In the previous studies, several metaheuristic methods are utilized to solve this case of the optimization problem. In our experiments, we have performed 15,000 searches with 25 salp swarm sizes, and the results obtained by the proposed GSSA are presented in Table 8. In this table, results of various other methods such as GSA ([104]), ES ([87]), GA ([33]), mathematical optimization ([11]) and Constraint Correction [10] are presented for comparison of the GSSA. The table indicates that the GSSA has provided better results than other compared method.

Table 4 Statistical comparison through Wilcoxon signed rank test for 23 benchmark problems

Problem	Dimension	SSA/GSSA	SSA/CSSA	SSA/LSSA	GSSA/CSSA	GSSA/LSSA	CSSA/LSSA
F1	30	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
	100	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
F2	30	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
	100	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
F3	30	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
	100	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
F4	30	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
	100	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
F5	30	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
	100	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
F6	30	1.48E−04 (+)	1.36E−04 (+)	1.48E−04 (+)	4.05E−01 (≈)	1.11E−02 (+)	7.71E−04 (−)
	100	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	2.62E−01 (≈)	4.78E−01 (≈)	4.95E−02 (−)
F7	30	1.73E−06 (−)	1.73E−06 (+)	1.73E−06 (+)	6.44E−01 (≈)	2.89E−01 (≈)	6.29E−01 (≈)
	100	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.59E−01 (≈)	7.97E−01 (≈)	1.78E−01 (≈)
F8	30	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
	100	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
F9	30	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	2.71E−01 (≈)	4.07E−05 (+)	1.36E−04 (+)
	100	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	8.13E−01 (≈)	2.83E−04 (+)	6.64E−04 (+)
F10	30	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
	100	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
F11	30	4.65E−01 (≈)	7.34E−01 (≈)	7.66E−01 (≈)	9.06E−02 (≈)	4.89E−01 (≈)	7.11E−01 (≈)
	100	5.45E−02 (≈)	9.78E−02 (≈)	5.44E−01 (≈)	2.67E−01 (≈)	2.99E−02 (≈)	1.27E−01 (≈)
F12	30	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1 (≈)	1 (≈)	1 (≈)
	100	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1 (≈)	1 (≈)	1 (≈)
F13	30	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1 (≈)	1 (≈)	1 (≈)
	100	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1 (≈)	1 (≈)	1 (≈)
F14	30	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1 (≈)	1 (≈)	1 (≈)
	100	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1 (≈)	1 (≈)	1 (≈)
F15	30	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	8.45E−01 (≈)	3.61E−03 (+)	4.99E−03 (+)
	100	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	7.19E−02 (≈)	2.71E−01 (≈)	8.73E−03 (+)
F16	30	3.06E−04 (+)	8.19E−05 (+)	3.06E−04 (+)	3.93E−01 (≈)	5.32E−03 (+)	3.06E−04 (+)
	100	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	6.56E−02 (≈)	1.57E−02 (+)	5.71E−04 (+)
F17	30	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
	100	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
F18	30	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1 (≈)	1 (≈)	1 (≈)
	100	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1 (≈)	1 (≈)	1 (≈)
F19	30	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
	100	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
F20	30	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
	100	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
F21	30	1 (≈)	1 (≈)	1 (≈)	1 (≈)	1 (≈)	1 (≈)
	100	1 (≈)	1 (≈)	1 (≈)	1 (≈)	1 (≈)	1 (≈)
F22	30	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
	100	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
F23	30	1.69E−06 (+)	1.69E−06 (+)	1.73E−06 (+)	1 (≈)	1.32E−04 (+)	1.32E−04 (+)
	100	1.65E−06 (+)	1.65E−06 (+)	1.73E−06 (+)	1 (≈)	8.30E−06 (+)	8.30E−06 (+)

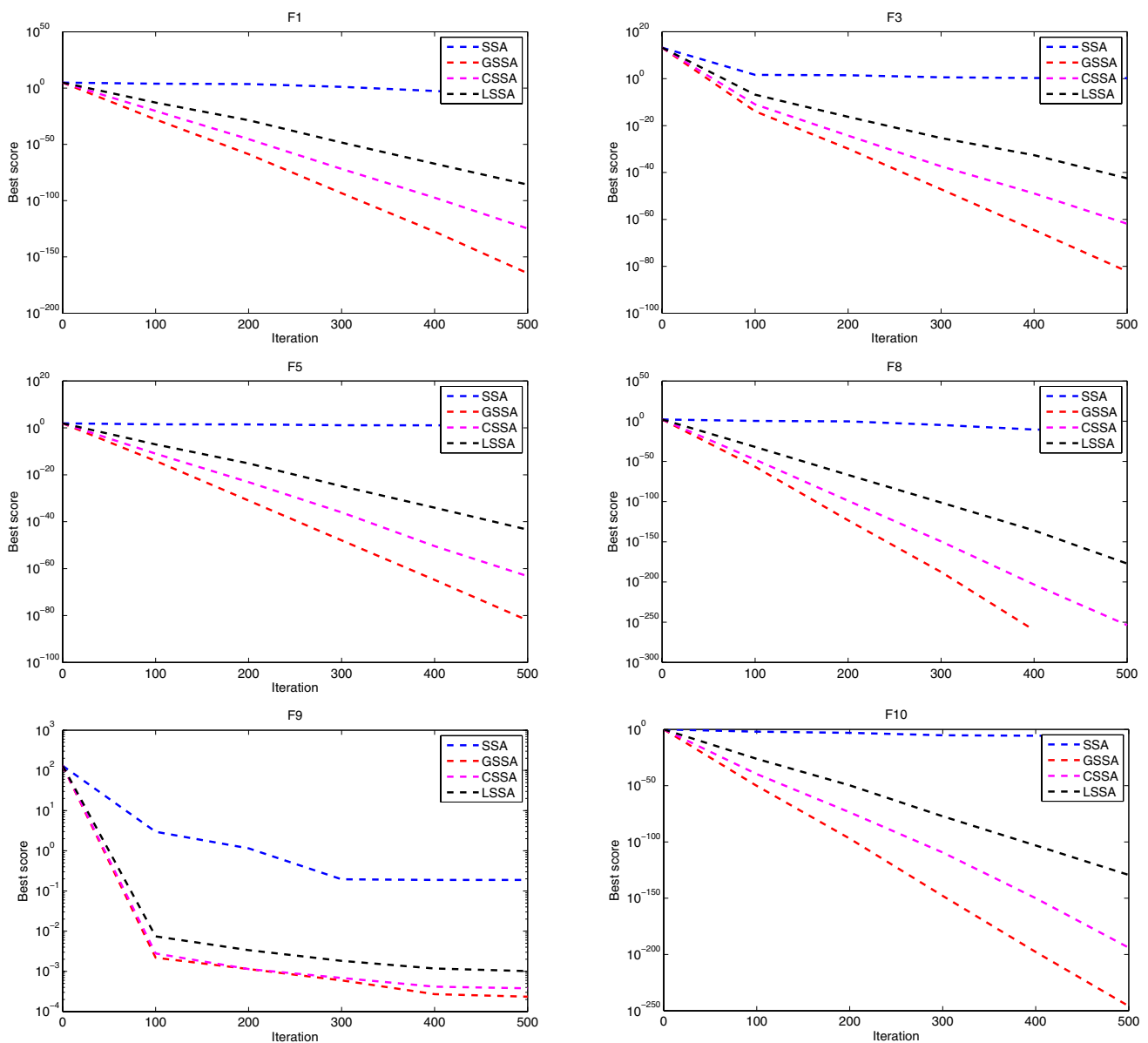


Fig. 1 Convergence curves for benchmark problems

4.3.3 Speed reducer design

In this subsection, the proposed GSSA is applied to the optimization task of designing a speed reducer, where the weight of the speed reducer is minimized. This problem is considered as a structural optimization problem with the decision variable: module of teeth m , face width b , the number of

teeth on pinion z , length of shaft-I and shaft-II between bearings l_1 and l_2 , respectively, the diameter of shaft-I and shaft-II d_1 and d_2 , respectively. The constraints in this problem are applied on bending stress of the gear teeth, transverse deflections of shafts 1 and 2 due to transmitted force, surface stress, and stresses in shafts-I and shaft-II. In the mathematical form, the problem is stated as follows:

$$\begin{aligned} \text{Minimize } f(b, m, z, l_1, l_2, d_1, d_2) = & 0.7854bm^2(14.9334z + 3.3333z^2 - 43.0934) \\ & - 1.508b(d_1^2 + d_2^2) + 0.7854(l_1d_1^2 + l_2d_2^2) + 7.477(d_1^3 + d_2^3) \end{aligned} \tag{25}$$

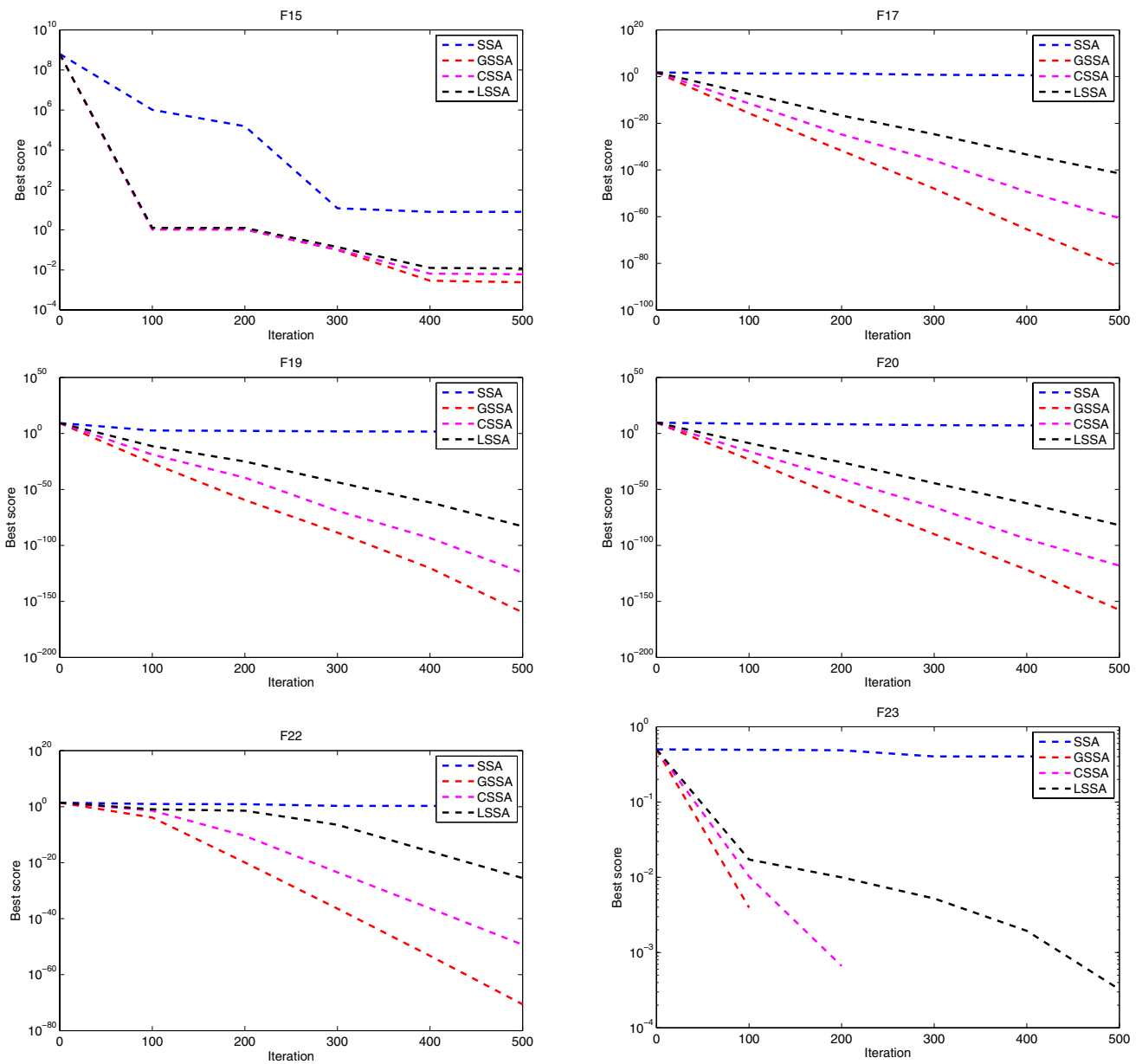


Fig. 2 Convergence curves for benchmark problems

subject to

$$g_1 = \frac{27}{bz^2m^2} \times P - 1 \leq 0 \tag{26}$$

$$g_2 = \frac{397.50}{bz^2m^2} - 1 \leq 0 \tag{27}$$

$$g_3 = \frac{1.93}{mz l_1^3 d_1^4} - 1 \leq 0 \tag{28}$$

$$g_4 = \frac{1.93}{mz l_1^3 d_2^4} - 1 \leq 0 \tag{29}$$

$$g_5 = \frac{\sqrt{1.69 \times 10^6 + (745l_1/mz)^2}}{110d_1^3} - 1 \leq 0 \tag{30}$$

Table 5 Comparison with other metaheuristic methods on dim 100

Problem	Results	FA	GWO	MFO	SCA	TLBO	mod-SSA	ISSA	GSSA
F1	Best	8.47E+03	1.65E−13	3.73E+04	1.55E+03	1.12E−34	9.74E−17	1.17E−12	3.34E−118
	Avg	1.23E+04	1.64E−12	6.32E+04	1.50E+04	2.33E−33	2.59E−14	1.69E−12	6.41E−104
	Std.	2.07E+03	1.04E−12	1.39E+04	1.19E+04	3.09E−33	6.98E−14	2.13E−13	2.41E−103
F2	Best	4.30E+03	1.72E−13	1.72E+04	7.23E+02	8.86E−35	3.69E−17	5.89E−13	1.49E−116
	Avg	6.16E+03	5.35E−13	3.02E+04	4.46E+03	6.16E−34	3.21E−15	8.34E−13	5.40E−106
	Std.	1.03E+03	3.25E−13	8.34E+03	3.35E+03	6.05E−34	3.54E−15	1.08E−13	2.28E−105
F3	Best	9.68E+01	1.96E−08	1.78E+02	3.38E−01	8.96E−18	1.50E−13	8.72E−07	1.25E−60
	Avg	1.33E+02	3.88E−08	2.45E+02	7.00E+00	2.78E−17	6.33E−12	1.03E−06	6.21E−54
	Std.	2.48E+01	1.25E−08	3.02E+01	6.66E+00	1.59E−17	9.14E−12	7.98E−08	2.27E−53
F4	Best	5.79E+04	4.93E+01	1.64E+05	1.42E+05	1.76E−08	4.36E+01	6.55E−12	2.15E−115
	Avg	1.36E+05	5.31E+02	2.38E+05	2.59E+05	8.11E−06	2.40E+04	4.62E−11	2.22E−105
	Std.	4.60E+04	5.71E+02	5.90E+04	5.67E+04	2.17E−05	3.81E+04	3.80E−11	1.11E−104
F5	Best	2.76E+01	7.59E−02	8.81E+01	8.44E+01	2.09E−15	9.18E+01	2.86E−07	3.77E−59
	Avg	3.54E+01	1.01E+00	9.34E+01	8.98E+01	5.60E−15	9.62E+01	3.40E−07	9.98E−53
	Std.	3.81E+00	1.29E+00	2.10E+00	2.68E+00	2.64E−15	1.49E+00	2.86E−08	5.30E−52
F6	Best	2.01E+06	9.64E+01	5.41E+07	2.07E+07	9.58E+01	9.72E+01	9.87E+01	9.89E+01
	Avg	3.56E+06	9.79E+01	1.67E+08	1.20E+08	9.71E+01	9.85E+01	9.88E+01	9.90E+01
	Std.	1.18E+06	6.23E−01	7.07E+07	4.60E+07	6.83E−01	4.30E−01	4.36E−02	1.51E−02
F7	Best	8.68E+03	8.28E+00	3.09E+04	1.91E+03	3.12E+00	1.79E+01	1.62E+01	1.04E+01
	Avg	1.30E+04	9.99E+00	5.98E+04	1.23E+04	4.87E+00	1.91E+01	1.87E+01	1.50E+01
	Std.	2.32E+03	1.04E+00	1.39E+04	9.46E+03	7.89E−01	4.95E−01	9.31E−01	2.82E+00
F8	Best	2.51E+00	5.20E−27	5.91E+01	4.31E+01	2.38E−73	3.20E−30	4.34E−32	4.11E−241
	Avg	5.77E+00	2.99E−25	2.94E+02	1.49E+02	6.36E−71	2.45E−24	1.06E−31	8.98E−214
	Std.	1.78E+00	5.57E−25	1.65E+02	6.67E+01	2.04E−70	1.09E−23	3.74E−32	0.00E+00
F9	Best	2.17E+00	3.48E−03	1.04E+02	2.88E+00	1.07E−03	1.64E−03	6.62E−06	2.15E−05
	Avg	5.28E+00	7.45E−03	2.57E+02	1.42E+02	8.29E−03	1.19E−02	1.30E−04	3.86E−04
	Std.	1.80E+00	2.48E−03	9.17E+01	7.87E+01	4.39E−03	1.37E−02	1.15E−04	3.40E−04
F10	Best	8.63E−07	1.49E−84	3.08E−05	1.84E−02	4.87E−113	1.22E−02	2.84E−24	1.77E−179
	Avg	3.99E−05	8.16E−64	1.37E−02	2.08E−01	2.67E−105	1.43E+00	9.13E−22	7.05E−157
	Std.	5.27E−05	4.47E−63	3.40E−02	1.49E−01	1.40E−104	4.78E−01	1.52E−21	3.86E−156
F11	Best	− 1.47E+04	− 1.97E+04	− 2.84E+04	− 8.20E+03	− 2.58E+04	− 9.01E+03	− 8.35E+03	− 2.49E+04
	Avg	− 1.15E+04	− 1.62E+04	− 2.33E+04	− 6.85E+03	− 2.20E+04	− 6.96E+03	− 6.44E+03	− 2.06E+04
	Std.	2.49E+03	2.43E+03	2.17E+03	5.53E+02	2.08E+03	8.63E+02	6.97E+02	1.81E+03
F12	Best	6.70E+02	5.51E−11	7.50E+02	6.39E+01	0.00E+00	0.00E+00	2.27E−13	0.00E+00
	Avg	7.38E+02	8.42E+00	8.53E+02	2.66E+02	0.00E+00	8.72E−01	3.87E−13	0.00E+00
	Std.	4.84E+01	5.83E+00	6.80E+01	1.06E+02	0.00E+00	4.78E+00	1.02E−13	0.00E+00
F13	Best	1.10E+01	4.64E−08	1.95E+01	8.85E+00	4.44E−15	2.00E+01	1.40E−07	8.88E−16
	Avg	1.23E+01	1.29E−07	1.99E+01	1.91E+01	7.88E−15	2.00E+01	1.67E−07	8.88E−16
	Std.	5.59E−01	4.33E−08	1.21E−01	3.83E+00	6.49E−16	2.82E−04	1.15E−08	0.00E+00
F14	Best	7.72E+01	1.31E−13	3.07E+02	1.35E+01	0.00E+00	0.00E+00	8.89E−13	0.00E+00
	Avg	1.12E+02	2.49E−03	5.51E+02	1.19E+02	0.00E+00	1.54E−03	1.59E−12	0.00E+00
	Std.	1.87E+01	8.11E−03	1.47E+02	6.63E+01	0.00E+00	6.58E−03	5.18E−13	0.00E+00
F15	Best	8.61E+01	1.58E−01	9.04E+07	3.79E+07	3.67E−02	6.11E−01	4.83E−01	8.55E−02
	Avg	2.74E+04	3.31E−01	3.53E+08	3.24E+08	6.06E−02	7.72E−01	6.22E−01	1.42E−01
	Std.	4.62E+04	9.73E−02	2.17E+08	1.45E+08	1.23E−02	8.38E−02	6.02E−02	3.33E−02
F16	Best	3.16E+05	6.08E+00	1.27E+08	2.10E+08	5.29E+00	8.83E+00	9.97E+00	9.99E+00
	Avg	1.99E+06	6.91E+00	5.55E+08	5.88E+08	6.62E+00	9.21E+00	9.97E+00	1.00E+01
	Std.	1.11E+06	4.23E−01	2.19E+08	2.69E+08	7.93E−01	2.53E−01	3.26E−03	2.44E−03

Table 5 (continued)

Problem	Results	FA	GWO	MFO	SCA	TLBO	mod-SSA	ISSA	GSSA
F17	Best	6.20E+01	5.94E−04	4.31E+01	6.09E+00	1.04E−18	5.30E−12	8.83E−08	2.06E−60
	Avg	8.04E+01	3.46E−03	6.66E+01	2.22E+01	2.92E−18	8.17E−01	1.04E−07	4.76E−55
	Std.	8.67E+00	1.65E−03	1.41E+01	1.07E+01	1.32E−18	4.24E+00	6.36E−09	2.35E−54
F18	Best	8.18E+00	1.42E−14	1.07E+01	2.36E− E−01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Avg	9.13E+00	3.06E−14	1.32E+01	3.04E+00	0.00E+00	5.92E−17	0.00E+00	0.00E+00
	Std.	5.01E−01	1.01E−14	1.50E+00	2.09E+00	0.00E+00	3.24E−16	0.00E+00	0.00E+00
F19	Best	1.05E+13	4.52E+01	2.23E+03	4.00E+02	1.51E+01	1.98E+00	1.97E−14	3.50E−119
	Avg	8.02E+13	9.31E+01	2.70E+03	6.62E+02	8.86E+01	6.63E+01	2.63E−13	1.36E−108
	Std.	4.33E+13	3.06E+01	2.37E+02	1.62E+02	4.32E+01	6.76E+01	4.31E−13	4.91E−108
F20	Best	3.33E+08	2.06E−10	4.03E+08	1.44E+06	1.03E−30	2.03E−14	6.68E−08	7.60E−112
	Avg	6.72E+08	3.06E−09	1.37E+09	1.77E+07	9.57E−30	4.25E−11	1.34E−07	1.90E−101
	Std.	1.97E+08	2.79E−09	7.74E+08	1.62E+07	1.34E−29	1.04E−10	5.16E−08	6.26E−101
F21	Best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Avg	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F22	Best	1.15E+01	3.00E−01	2.77E+01	6.10E+00	9.99E−02	9.99E−02	1.18E−07	1.42E−51
	Avg	1.32E+01	3.53E−01	3.49E+01	1.22E+01	1.27E−01	1.73E−01	1.31E−07	5.84E−45
	Std.	8.31E−01	5.07E−02	3.34E+00	3.29E+00	4.48E−02	4.50E−02	7.16E−09	2.77E−44
F23	Best	5.00E−01	3.72E−02	5.00E−01	4.98E−01	9.72E−03	9.72E−03	1.15E−12	0.00E+00
	Avg	5.00E−01	9.30E−02	5.00E−01	5.00E−01	1.89E−02	3.22E−02	1.78E−12	0.00E+00
	Std.	5.38E−05	2.82E−02	1.24E−06	3.54E−04	1.32E−02	1.46E−02	2.87E−13	0.00E+00

Table 6 Statistical comparison through Wilcoxon signed rank test for 23 benchmark problems

Problem	GSSA/FA	GSSA/GWO	GSSA/MFO	GSSA/SCA	GSSA/TLBO	GSSA/mod-SSA	GSSA/wSSA
F1	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
F2	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
F3	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
F4	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
F5	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
F6	1.73E−06 (+)	1.73E−06 (−)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (−)	1.73E−06 (−)	1.73E−06 (−)
F7	1.73E−06 (+)	1.73E−06 (−)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (−)	5.22E−06 (+)	3.18E−06 (+)
F8	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
F9	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.92E−06 (+)	1.73E−06 (+)	1.60E−04 (−)
F10	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
F11	1.73E−06 (+)	2.88E−06 (+)	1.80E−05 (−)	1.73E−06 (+)	8.22E−03 (−)	1.73E−06 (+)	1.73E−06 (+)
F12	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.00E+00 (≈)	5.90E−05 (+)	1.18E−06 (+)
F13	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	6.80E−08 (+)	1.73E−06 (+)	1.73E−06 (+)
F14	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.00E+00 (≈)	3.78E−06 (+)	1.73E−06 (+)
F15	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (−)	1.73E−06 (+)	1.73E−06 (+)
F16	1.73E−06 (+)	1.73E−06 (−)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (−)	1.73E−06 (−)	1.73E−06 (−)
F17	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
F18	1.73E−06 (+)	1.67E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.00E+00 (≈)	1.00E+00 (≈)	1.00E+00 (≈)
F19	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
F20	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
F21	1.00E+00 (≈)	1.00E+00 (≈)	1.00E+00 (≈)	1.00E+00 (≈)	1.00E+00 (≈)	1.00E+00 (≈)	1.00E+00 (≈)
F22	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.7344E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)
F23	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.7344E−06 (+)	1.73E−06 (+)	1.73E−06 (+)	1.73E−06 (+)

Table 7 Comparison results of the GSSA for the three-bar truss design problem

Algorithm	Optimum variable		Optimum cost
	A ₁	A ₂	
GSSA	0.788703186	0.408168954	263.895844
[105]	0.795	0.395	264.3
[121]	0.788	0.408	263.68 (infeasible)
[97]	0.78879	0.40794	263.8965
CS	0.78867	0.40902	263.9716
BA	0.78863	0.40838	263.8962
SCA	0.78669	0.41426	263.9348
m-SCA	0.81915	0.36956	263.8972

Table 8 Comparison results of the GSSA for tension-compression spring design problem

Algorithm	Optimum variable			Optimum cost
	d	D	N	
GSSA	0.050806005	0.33584285	12.62658005	0.0126797
GSA	0.050276	0.32368	13.52541	0.0127022
ES	0.051989	0.363965	10.890522	0.012681
GA	0.05148	0.351661	11.632201	0.0127048
Math- emati- cal Optimi- zation (Bel- egundu)	0.053396	0.39918	9.1854	0.0127303
Constraint cor- rection (Arora)	0.05	0.3159	14.25	0.0128334

$$g_6 = \frac{\sqrt{157.50 \times 10^6 + (745l_1/mz)^2}}{85d_2^3} - 1 \leq 0 \tag{31}$$

Table 9 Comparison results of the GSSA for speed reducer design problem

Algorithm	Optimum variable							Optimum cost
	b	m	z	l ₁	l ₂	d ₁	d ₂	
GSSA	3.500957	0.7	17	7.331317	7.806692	3.351851	5.28669	2997.5658
CS	3.5015	0.7	17	7.605	7.8181	3.352	5.2875	3000.981
[105]	3.514185	0.700005	17	7.497343	7.8346	2.9018	5.0022	2732.9006 (infeasible)
[87]	3.506163	0.700831	17	7.460181	7.962143	3.3629	5.309	3025.005
[5]	3.506122	0.700006	17	7.549126	7.85933	3.365576	5.289773	3008.08
[69]	3.6	0.7	17	7.3	7.8	3.4	5	2876.117623 (infeasible)
[63]	3.5197	0.7039	17.3831	7.3	7.7152	3.3498	5.2866	3,007.80

$$g_7 = \frac{mz}{40} \times -1 \leq 0 \tag{32}$$

$$g_8 = \frac{5m}{B-1} \times -1 \leq 0 \tag{33}$$

$$g_9 = \frac{b}{12m} \times -1 \leq 0 \tag{34}$$

where

$$2.60 \leq b \leq 3.60, 0.70 \leq m \leq 0.80, 17 \leq z \leq 28$$

$$7.30 \leq l_1 \leq 8.30, 7.80 \leq l_2 \leq 8.30$$

$$2.90 \leq d_1 \leq 3.90, 5.00 \leq d_2 \leq 5.50$$

In our study, we have fixed the swarm size to 50 and iterations to 5,000 for obtaining the solution of this problem. In the previous literature, many studies [5, 63, 69, 88, 105] are done to perform this optimization task. [43] have applied the CS to solve this problem. The simulated results on this problem by the GSSA is presented in Table 9, which indicates that the GSSA than other results produces the superior results.

5 Conclusions and future directions

In this study, the recently proposed salp swarm algorithm’s performance is enhanced using new search rules based on greedy search and mutation strategies. Significantly, to improve the exploitation feature and to balance exploration and exploitation in the algorithm. Further, the swarm diversity is managed by the different mutation rules: Gaussian, Cauchy, and levy. Based on these mutation rules, the different variants GSSA, CSSA, and LSSA are proposed. These rules have significantly improved the SSA’s exploitation and exploration abilities. Experimental results have also shown that these rules are fruitful in preventing local optima and faster convergence. The experiments for evaluating these variants and choosing the best mutation rule for the SSA, 23 benchmark test problems with dimension 30 and 100 are utilized. Convergence analysis and statistical tests ensure

that the GSSA is the best variant to solve global optimization problems. Furthermore, the best-chosen method, the GSSA, is used to solve some engineering design optimization cases. The results and comparison indicate the superiority of the GSSA over other studies that are done to find the solution to these engineering cases.

There are many future domains for the utilized and suggested SSA-based optimizers according to the many domains that need optimization and solution finding in general. We have initialized research works to employ the mutation-based SSA for the sensor networks [40, 41], structural health assessment and promising optimization foundation [32], possible optimization features of the solar systems [129, 132], GIS [122, 179, 180, 183], and a set of industrial tasks in the power engineering [65, 106]. As the proposed method has a more stable exploratory basis, we suggest the application of it to the data modeling of the location-based services (LBS) [72], landslide prediction and forecasting [133], prediction methods [68, 101], and modeling in the environmental scenarios [23, 73, 168]. In the future, applying the proposed mutation-based variant to solve more practical engineering cases such as multiobjective optimization, binary optimization is a worthwhile research direction. The proposed method and its resulted variations can be a suitable tool for operation under intelligent systems, and medical diagnosis cases [24, 64, 70, 76, 127, 141, 174]. The proposed mutation-based variants can also be hybridized with other metaheuristic methods to propose a new better optimization method. The proposed mutation-based SSA can also be used in other areas, such as optimizing machine learning models' structure and weights.

References

- Abbassi A, Abbassi R, Heidari AA, Oliva D, Chen H, Habib A, Jemli M, Wang M (2020) Parameters identification of photovoltaic cell models using enhanced exploratory salp chains-based approach. *Energy* 198:117,333. <https://doi.org/10.1016/j.energy.2020.117333>
- Abedini M, Zhang C (2020) Performance assessment of concrete and steel material models in ls-dyna for enhanced numerical simulation, a state of the art review. *Arch Comput Methods Eng*:1–22
- Abualigah L, Shehab M, Alshinwan M, Alabool H (2019) Salp swarm algorithm: a comprehensive survey. *Neural Comput Appl*:1–21
- Abusnaina AA, Ahmad S, Jarrar R, Mafarja M (2018) Training neural networks using salp swarm algorithm for pattern classification. In: *Proceedings of the 2nd international conference on future networks and distributed systems*, pp 1–6
- Akhtar S, Tai K, Ray T (2002) A socio-behavioural simulation model for engineering design optimization. *Eng Optim* 34(4):341–354
- Ala'M AZ, Heidari AA, Habib M, Faris H, Aljarah I, Hassounah MA (2020) Salp chain-based optimization of support vector machines and feature weighting for medical diagnostic information systems. Springer, Berlin, pp 11–34
- Aljarah I, Mafarja M, Heidari AA, Faris H, Zhang Y, Mirjalili S (2018) Asynchronous accelerating multi-leader salp chains for feature selection. *Appl Soft Comput* 71:964–979
- Aljarah I, Mafarja M, Heidari AA, Faris H, Mirjalili S (2019) Clustering analysis using a novel locality-informed grey wolf-inspired clustering approach. *Knowl Inf Syst*. <https://doi.org/10.1007/s10115-019-01358-x>
- Aljarah I, Habib M, Faris H, Al-Madi N, Heidari AA, Mafarja M, Abd Elaziz M, Mirjalili S (2020) A dynamic locality multi-objective salp swarm algorithm for feature selection. *Comput Ind Eng* 147(106):628
- Arora JS (2004) *Introduction to optimum design*. Elsevier, Oxford
- Belegundu AD, Arora JS (1985) A study of mathematical programming methods for structural optimization part i: Theory. *Int J Num Methods Eng* 21(9):1583–1599
- Cai C, Gao X, Teng Q, Kiran R, Liu J, Wei Q, Shi Y (2020a) Hot isostatic pressing of a near a-ti alloy: Temperature optimization, microstructural evolution and mechanical performance evaluation. *Mater Sci Eng A*:140426
- Cai C, Wu X, Liu W, Zhu W, Chen H, Qiu JCD, Sun CN, Liu J, Wei Q, Shi Y (2020b) Selective laser melting of near-a titanium alloy ti-6al-2zr-1mo-1v: Parameter optimization, heat treatment and mechanical performance. *J Mater Sci Technol*
- Cao B, Zhao J, Gu Y, Fan S, Yang P (2019a) Security-aware industrial wireless sensor network deployment optimization. *IEEE Trans Ind Inform* 16(8):5309–5316
- Cao B, Zhao J, Yang P, Gu Y, Muhammad K, Rodrigues JJ, de Albuquerque VHC (2019b) Multiobjective 3-d topology optimization of next-generation wireless data center network. *IEEE Trans Ind Inform* 16(5):3597–3605
- Cao B, Dong W, Lv Z, Gu Y, Singh S, Kumar P (2020a) Hybrid microgrid many-objective sizing optimization with fuzzy decision. *IEEE Trans Fuzzy Syst*
- Cao B, Fan S, Zhao J, Yang P, Muhammad K, Tanveer M (2020b) Quantum-enhanced multiobjective large-scale optimization via parallelism. *Swarm Evol Comput* 57(100):697. <https://doi.org/10.1016/j.swevo.2020.100697>
- Cao B, Wang X, Zhang W, Song H, Lv Z (2020c) A many-objective optimization model of industrial internet of things based on private blockchain. *IEEE Netw* 34(5):78–83
- Cao B, Zhao J, Gu Y, Ling Y, Ma X (2020d) Applying graph-based differential grouping for multiobjective large-scale optimization. *Swarm Evol Comput* 53(100):626
- Cao Y, Li Y, Zhang G, Jermsittiparsert K, Nasser M (2020e) An efficient terminal voltage control for pemfc based on an improved version of whale optimization algorithm. *Energy Rep* 6:530–542
- Cao Y, Wang Q, Cheng W, Nojavan S, Jermsittiparsert K (2020f) Risk-constrained optimal operation of fuel cell/photovoltaic/battery/grid hybrid energy system using downside risk constraints method. *Int J Hydrogen Energy* 45(27):14,108–14,118. <https://doi.org/10.1016/j.ijhydene.2020.03.090>
- Chantar H, Mafarja M, Alsawalqah H, Heidari AA, Aljarah I, Faris H (2020) Feature selection using binary grey wolf optimizer with elite-based crossover for Arabic text classification. *Neural Comput Appl* 32(16):12,201–12,220
- Chao L, Zhang K, Li Z, Zhu Y, Wang J, Yu Z (2018) Geographically weighted regression based methods for merging satellite and gauge precipitation. *J Hydrol* 558:275–289
- Chen HL, Wang G, Ma C, Cai ZN, Liu WB, Wang SJ (2016) An efficient hybrid kernel extreme learning machine approach for early diagnosis of Parkinson's disease. *Neurocomputing* 184:131–144

25. Chen Y, He L, Guan Y, Lu H, Li J (2017) Life cycle assessment of greenhouse gas emissions and water-energy optimization for shale gas supply chain planning based on multi-level approach: case study in Barnett, Marcellus, Fayetteville, and Haynesville shales. *Energy Convers Manag* 134:382–398
26. Chen H, Qiao H, Xu L, Feng Q, Cai K (2019a) A fuzzy optimization strategy for the implementation of rbf lssvr model in vis-nir analysis of pomelo maturity. *IEEE Trans Ind Inform* 15(11):5971–5979
27. Chen H, Yang C, Heidari AA, Zhao X (2019b) An efficient double adaptive random spare reinforced whale optimization algorithm. *Expert Syst Appl*. <https://doi.org/10.1016/j.eswa.2019.113018>
28. Chen H, Chen A, Xu L, Xie H, Qiao H, Lin Q, Cai K (2020a) A deep learning cnn architecture applied in smart near-infrared analysis of water pollution for agricultural irrigation resources. *Agric Water Manag* 240(106):303
29. Chen H, Fan DL, Fang L, Huang W, Huang J, Cao C, Yang L, He Y, Zeng L (2020b) Particle swarm optimization algorithm with mutation operator for particle filter noise reduction in mechanical fault diagnosis. *Int J Pattern Recogn Artif Intell*:2058012
30. Chen H, Heidari AA, Chen H, Wang M, Pan Z, Gandomi AH (2020) Multi-population differential evolution-assisted Harris Hawks optimization: framework and case studies. *Future Gen Comput Syst* 111:175–198. <https://doi.org/10.1016/j.future.2020.04.008>
31. Chen H, Li S, Heidari AA, Wang P, Li J, Yang Y, Wang M, Huang C (2020d) Efficient multi-population outpost fruit fly-driven optimizers: framework and advances in support vector machines. *Expert Syst Appl* 142(112):999
32. Chen H, Zhang G, Fan D, Fang L, Huang L (2020e) Nonlinear lamb wave analysis for microdefect identification in mechanical structural health assessment. *Measurement*:108026
33. Coello CAC (2000) Use of a self-adaptive penalty approach for engineering optimization problems. *Comput Ind* 41(2):113–127
34. Coello CAC (2002) Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Comput Methods Appl Mech Eng* 191(11–12):1245–1287
35. Elaziz MA, Heidari AA, Fujita H, Moayedi H (2020) A competitive chain-based Harris Hawks optimizer for global optimization and multi-level image thresholding problems. *Appl Soft Comput J*. <https://doi.org/10.1016/j.asoc.2020.106347>
36. Fan Y, Wang P, Heidari AA, Wang M, Zhao X, Chen H, Li C (2020a) Boosted hunting-based fruit fly optimization and advances in real-world problems. *Expert Syst Appl*:113502
37. Fan Y, Wang P, Heidari AA, Wang M, Zhao X, Chen H, Li C (2020b) Rationalized fruit fly optimization with sine cosine algorithm: a comprehensive analysis. *Expert Syst Appl*:113486
38. Faris H, Heidari AA, Ala'M AZ, Mafarja M, Aljarah I, Eshtay M, Mirjalili S (2020a) Time-varying hierarchical chains of salps with random weight networks for feature selection. *Expert Syst Appl* 140(112):898
39. Faris H, Mirjalili S, Aljarah I, Mafarja M, Heidari AA (2020b) Salp swarm algorithm: theory, literature review, and application in extreme learning machines. Springer, Berlin, pp 185–199
40. Fu X, Yang Y (2020) Modeling and analysis of cascading node-link failures in multi-sink wireless sensor networks. *Reliab Eng Syst Saf* 197(106):815
41. Fu X, Fortino G, Li W, Pace P, Yang Y (2019) Wsns-assisted opportunistic network for low-latency message forwarding in sparse settings. *Future Gen Comput Syst* 91:223–237
42. Fu X, Pace P, Aloï G, Yang L, Fortino G (2020) Topology optimization against cascading failures on wireless sensor networks using a memetic algorithm. *Comput Netw*:107327
43. Gandomi AH, Yang XS, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Eng Comput* 29(1):17–35
44. Gao N, Luo D, Cheng B, Hou H (2020) Teaching-learning-based optimization of a composite metastructure in the 0–10 kHz broadband sound absorption range. *J Acoust Soc Am* 148(2):EL125–EL129
45. García S, Fernández A, Luengo J, Herrera F (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Inf Sci* 180(10):2044–2064
46. Gholipour G, Zhang C, Mousavi AA (2020a) Nonlinear numerical analysis and progressive damage assessment of a cable-stayed bridge pier subjected to ship collision. *Mar Struct* 69(102):662
47. Gholipour G, Zhang C, Mousavi AA (2020b) Numerical analysis of axially loaded rc columns subjected to the combination of impact and blast loads. *Eng Struct* 219(110):924
48. Guo J, Zhang X, Gu F, Zhang H, Fan Y (2020a) Does air pollution stimulate electric vehicle sales? empirical evidence from twenty major cities in china. *J Clean Prod* 249(119):372
49. Guo L, Sriyakul T, Nojavan S, Jermsittiparsert K (2020b) Risk-based traded demand response between consumers' aggregator and retailer using downside risk constraints technique. *IEEE Access* 8:90,957–90,968
50. Gupta S, Deep K (2019a) A hybrid self-adaptive sine cosine algorithm with opposition based learning. *Expert Syst Appl* 119:210–230
51. Gupta S, Deep K (2019b) A novel random walk grey wolf optimizer. *Swarm Evol Comput* 44:101–112
52. Gupta S, Deep K, Heidari AA, Moayedi H, Chen H (2019a) Harmonized salp chain-built optimization. *Eng Comput*:1–31
53. Gupta S, Deep K, Heidari AA, Moayedi H, Chen H (2019b) Harmonized salp chain-built optimization. *Eng Comput*. <https://doi.org/10.1007/s00366-019-00871-5>
54. Gupta S, Deep K, Heidari AA, Moayedi H, Wang M (2020a) Opposition-based learning harris hawks optimization with advanced transition rules: principles and analysis. *Expert Syst Appl*:113510
55. Gupta S, Deep K, Mirjalili S (2020b) An efficient equilibrium optimizer with mutation strategy for numerical optimization. *Appl Soft Comput* 96(106):542
56. Hegazy AE, Makhlof M, El-Tawel GS (2020) Improved salp swarm algorithm for feature selection. *J King Saud Univ Comput Inf Sci* 32(3):335–344
57. Heidari AA, Pahlavani P (2017) An efficient modified grey wolf optimizer with lévy flight for optimization tasks. *Appl Soft Comput* 60:115–134
58. Heidari AA, Abbaspour RA, Chen H (2019a) Efficient boosted grey wolf optimizers for global search and kernel extreme learning machine training. *Appl Soft Comput* 81(105):521
59. Heidari AA, Aljarah I, Faris H, Chen H, Luo J, Mirjalili S (2019b) An enhanced associative learning-based exploratory whale optimizer for global optimization. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-019-04015-0>
60. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019c) Harris hawks optimization: algorithm and applications. *Future Gen Comput Syst* 97:849–872. <https://doi.org/10.1016/j.future.2019.02.0280>
61. Heidari AA, Yin Y, Mafarja M, Jalali SMJ, Dong JS, Mirjalili S (2020) Efficient moth-flame-based neuroevolution models. Springer, Berlin, pp 51–66
62. Higashi N, Iba H (2003) Particle swarm optimization with gaussian mutation. In: Proceedings of the 2003 IEEE swarm intelligence symposium. SIS'03 (Cat. No. 03EX706), IEEE, pp 72–79

63. Hsu YL, Liu TC (2007) Developing a fuzzy proportional-derivative controller optimization engine for engineering design optimization problems. *Eng Optim* 39(6):679–700
64. Hu L, Hong G, Ma J, Wang X, Chen H (2015) An efficient machine learning approach for diagnosis of paraquat-poisoned patients. *Comput Biol Med* 59:116–124
65. Hu X, Ma P, Gao B, Zhang M (2019) An integrated step-up inverter without transformer and leakage current for grid-connected photovoltaic system. *IEEE Trans Power Electron* 34(10):9814–9827
66. Ibrahim HT, Mazher WJ, Ucan ON, Bayat O (2017) Feature selection using salp swarm algorithm for real biomedical datasets. *IJCSNS* 17(12):13–20
67. Ibrahim RA, Ewees AA, Oliva D, Abd Elaziz M, Lu S (2019) Improved salp swarm algorithm based on particle swarm optimization for feature selection. *J Ambient Intell Hum Comput* 10(8):3155–3169
68. Jiang Q, Wang G, Jin S, Li Y, Wang Y (2013) Predicting human microRNA-disease associations based on support vector machine. *Int J Data Min Bioinform* 8(3):282
69. Ku KJ, Rao SS, Chen L (1998) Taguchi-aided search method for design optimization of engineering systems. *Eng Optim* 30(1):1–23
70. Li C, Hou L, Sharma BY, Li H, Chen C, Li Y, Zhao X, Huang H, Cai Z, Chen H (2018) Developing a new intelligent system for the diagnosis of tuberculous pleural effusion. *Comput Methods Programs Biomed* 153:211–225
71. Li T, Xu M, Zhu C, Yang R, Wang Z, Guan Z (2019a) A deep learning approach for multi-frame in-loop filter of hevcc. *IEEE Trans Image Process* 28(11):5663–5678
72. Li X, Zhu Y, Wang J (2019b) Highly efficient privacy preserving location-based services with enhanced one-round blind filter. *IEEE Trans Emerg Top Comput*. <https://doi.org/10.1109/TETC.2019.29263851>
73. Li C, Sun L, Xu Z, Wu X, Liang T, Shi W (2020) Experimental Investigation and Error Analysis of High Precision FBG Displacement Sensor for Structural Health Monitoring. *Int J Struct Stab Dyn* 20(06):2040011
74. Li S, Chen H, Wang M, Heidari AA, Mirjalili S (2020) Slime mould algorithm: a new method for stochastic optimization. *Future Gen Comput Syst* 111:300–323. <https://doi.org/10.1016/j.future.2020.03.0552>
75. Liu J, Wu C, Wu G, Wang X (2015) A novel differential search algorithm and applications for structure design. *Appl Math Comput* 268:246–269
76. Liu D, Wang S, Huang D, Deng G, Zeng F, Chen H (2016a) Medical image classification using spatial adjacent histogram based on adaptive local binary patterns. *Comput Biol Med* 72:185–200
77. Liu S, Chan FT, Ran W (2016b) Decision making for the selection of cloud vendor: an improved approach under group decision-making with integrated weights and objective/subjective attributes. *Expert Syst Appl* 55:37–47
78. Liu E, Li W, Cai H, Peng S (2019) Formation mechanism of trailing oil in product oil pipeline. *Processes* 7(1):7
79. Liu S, Yu W, Chan FTS, Niu B (2020a) A variable weight-based hybrid approach for multi-attribute group decision making under interval-valued intuitionistic fuzzy sets. *Int J Intell Syst*. <https://doi.org/10.1002/int.223293>
80. Liu Y, Shi Y, Chen H, Asghar Heidari A, Gui W, Wang M, Chen H, Li C (2020b) Chaos-assisted multi-population salp swarm algorithms: framework and case studies. *Expert Syst Appl*. <https://doi.org/10.1016/j.eswa.2020.114369>
81. Long W, Jiao J, Liang X, Tang M (2018) An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization. *Eng Appl Artif Intell* 68:63–80
82. Luo J, Chen H, Xu Y, Huang H, Zhao X et al (2018) An improved grasshopper optimization algorithm with application to financial stress prediction. *Appl Math Model* 64:654–668
83. Luo J, Chen H, Heidari AA, Xu Y, Zhang Q, Li C (2019) Multi-strategy boosted mutative whale-inspired optimization approaches. *Appl Math Model* 73:109–123. <https://doi.org/10.1016/j.apm.2019.03.0465>
84. Lv Z, Kumar N (2020) Software defined solutions for sensors in 6g/5g. *Comput Commun* 153:42–47
85. Lv Z, Qiao L (2020) Deep belief network and linear perceptron based cognitive computing for collaborative robots. *Appl Soft Comput*:106300
86. Mafarja M, Heidari AA, Habib M, Faris H, Thaher T, Aljarah I (2020) Augmented whale feature selection for iot attacks: structure, analysis and applications. *Future Gen Comput Syst* 112:18–40. <https://doi.org/10.1016/j.future.2020.05.0206>
87. Mezura-Montes E, Coello CAC (2008) An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. *Int J Gen Syst* 37(4):443–473
88. Mezura-Montes E, Coello CC, Landa-Becerra R (2003) Engineering optimization using simple evolutionary algorithm. In: *Proceedings. 15th IEEE international conference on tools with artificial intelligence, IEEE*, pp 149–156
89. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. *Knowl Based Syst* 89:228–249
90. Mirjalili S (2016) Sca: a sine cosine algorithm for solving optimization problems. *Knowl Based Syst* 96:120–133
91. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
92. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. *Adv Eng Softw* 114:163–191
93. Mousavi AA, Zhang C, Masri SF, Gholipour G (2020) Structural damage localization and quantification based on a Ceemdan Hilbert transform neural network approach: A model steel truss bridge case study. *Sensors* 20(5):1271
94. Ni T, Chang H, Song T, Xu Q, Huang Z, Liang H, Yan A, Wen X (2020) Non-intrusive online distributed pulse shrinking-based interconnect testing in 2.5d ic. *IEEE Transact Circuits Syst II Express Briefs* 67(11):2657–2661. <https://doi.org/10.1109/TCSII.2019.29628247>
95. Nowacki H (1973) Optimization in pre-contract ship design. In: *International conference on computer applications in the automation of shipyard operation and ship design*, held by IFIP/IFAC/JSNA, Tokyo, Japan, Aug 28–30, 1973
96. Pang R, Xu B, Kong X, Zou D (2018) Seismic fragility for high cfrds based on deformation and damage index through incremental dynamic analysis. *Soil Dyn Earthq Eng* 104:432–436
97. Park Y, Chang M, Lee TY (2007) A new deterministic global optimization method for general twice-differentiable constrained nonlinear programming problems. *Eng Optim* 39(4):397–411
98. Qian J, Feng S, Li Y, Tao T, Han J, Chen Q, Zuo C (2020) Single-shot absolute 3D shape measurement with deep-learning-based color fringe projection profilometry. *Opt Lett* 45(7):1842
99. Qian J, Feng S, Tao T, Hu Y, Li Y, Chen Q, Zuo C (2020) Deep-learning-enabled geometric constraints and phase unwrapping for single-shot absolute 3D shape measurement. *APL Photonics* 5(4):046105
100. Qiu T, Shi X, Wang J, Li Y, Qu S, Cheng Q, Cui T, Sui S (2019) Deep learning: a rapid and efficient route to automatic metasurface design. *Adv Sci* 6(12):1900,128
101. Qu K, Wei L, Zou Q (2019) A review of DNA-binding proteins prediction methods. *Current Bioinform* 14(3):246–254

102. Qu S, Han Y, Wu Z, Raza H (2020) Consensus modeling with asymmetric cost based on data-driven robust optimization. *Group Decis Negot*:1–38
103. Rao RV, Savsani VJ, Vakharia D (2011) Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. *Comput Aid Des* 43(3):303–315
104. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) Gsa: a gravitational search algorithm. *Inf Sci* 179(13):2232–2248
105. Ray T, Saini P (2001) Engineering design optimization using a swarm with an intelligent information sharing among individuals. *Eng Optim* 33(6):735–748
106. Ridha HM, Gomes C, Hizam H, Ahmadipour M, Heidari AA, Chen H (2020a) Multi-objective optimization and multi-criteria decision-making methods for optimal design of standalone photovoltaic system: A comprehensive review. *Renew Sustain Energy Rev* 135(110):202
107. Ridha HM, Heidari AA, Wang M, Chen H (2020) Boosted mutation-based Harris Hawks optimizer for parameters identification of single-diode solar cell models. *Energy Convers Manag* 209(112):660. <https://doi.org/10.1016/j.enconman.2020.112660>
108. Rodríguez-Esparza E, Zanella-Calzada LA, Oliva D, Heidari AA, Zaldivar D, Pérez-Cisneros M, Foong LK (2020) An efficient Harris Hawks-inspired image segmentation method. *Expert Syst Appl* 155(113):428. <https://doi.org/10.1016/j.eswa.2020.113428>
109. Sayed GI, Khoriba G, Haggag MH (2018) A novel chaotic salp swarm algorithm for global optimization and feature selection. *Appl Intell* 48(10):3462–3481
110. Shen L, Chen H, Yu Z, Kang W, Zhang B, Li H, Yang B, Liu D (2016) Evolving support vector machines using fruit fly optimization for medical data classification. *Knowl Based Syst* 96:61–75
111. Shi K, Tang Y, Liu X, Zhong S (2017) Non-fragile sampled-data robust synchronization of uncertain delayed chaotic lurie systems with randomly occurring controller gain fluctuation. *ISA Trans* 66:185–199
112. Shi K, Tang Y, Zhong S, Yin C, Huang X, Wang W (2018) Non-fragile asynchronous control for uncertain chaotic Lurie network systems with Bernoulli stochastic process. *Int J Robust Nonlinear Control* 28(5):1693–1714
113. Shi K, Wang J, Tang Y, Zhong S (2020a) Reliable asynchronous sampled-data filtering of t-s fuzzy uncertain delayed neural networks with stochastic switched topologies. *Fuzzy Sets Syst* 381:1–25
114. Shi K, Wang J, Zhong S, Tang Y, Cheng J (2020b) Non-fragile memory filtering of ts fuzzy delayed neural networks based on switched fuzzy sampled-data control. *Fuzzy Sets Syst* 394:40–64
115. Singh N, Chiclana F, Magnot JP et al (2020) A new fusion of salp swarm with sine cosine for optimization of non-linear functions. *Eng Comput* 36(1):185–212
116. Song S, Wang P, Heidari AA, Wang M, Zhao X, Chen H, He W, Xu S (2020) Dimension decided harris hawks optimization with gaussian mutation: Balance analysis and diversity patterns. *Knowl Based Syst*:106425
117. Sun ZX, Hu R, Qian B, Liu B, Che GL (2018) Salp swarm algorithm based on blocks on critical path for reentrant job shop scheduling problems. In: *International conference on intelligent computing*, Springer, pp 638–648
118. Sun G, Yang B, Yang Z, Xu G (2019) An adaptive differential evolution with combined strategy for global numerical optimization. *Soft Comput*:1–20
119. Tang H, Xu Y, Lin A, Heidari AA, Wang M, Chen H, Luo Y, Li C (2020) Predicting green consumption behaviors of students using efficient firefly grey wolf-assisted k-nearest neighbor classifiers. *IEEE Access* 8:35,546–35,562. <https://doi.org/10.1109/ACCESS.2020.29737630>
120. Thaher T, Heidari AA, Mafarja M, Dong JS, Mirjalili S (2020) Binary Harris Hawks optimizer for high-dimensional, low sample size feature selection. Springer, Berlin, pp 251–272
121. Tsai JF (2005) Global optimization of nonlinear fractional programming problems in engineering design. *Eng Optim* 37(4):399–409
122. Tsai Y-H, Wang J, Chien W-T, Wei C-Y, Wang X, Hsieh S-H (2019) A BIM-based approach for predicting corrosion under insulation. *Autom Constr* 107:102923
123. Tu J, Chen H, Liu J, Heidari AA, Zhang X, Wang M, Ruby R, Pham QV (2020) Evolutionary biogeography-based whale optimization methods with communication structure: towards measuring the balance. *Knowl Based Syst*:106642
124. Tubishat M, Idris N, Shuib L, Abushariah MA, Mirjalili S (2020) Improved salp swarm algorithm based on opposition based learning and novel local search algorithm for feature selection. *Expert Syst Appl* 145(113):122
125. Wang M, Chen H (2020) Chaotic multi-swarm whale optimizer boosted support vector machine for medical diagnosis. *Appl Soft Comput J*. <https://doi.org/10.1016/j.asoc.2019.1059461>
126. Wang H, Li H, Liu Y, Li C, Zeng S (2007) Opposition-based particle swarm algorithm with cauchy mutation. In: *2007 IEEE congress on evolutionary computation, IEEE*, pp 4750–4756
127. Wang SJ, Chen HL, Yan WJ, Chen YH, Fu X (2014) Face recognition and micro-expression recognition based on discriminant tensor subspace analysis plus extreme learning machine. *Neural Process Lett* 39(1):25–43
128. Wang M, Chen H, Yang B, Zhao X, Hu L, Cai Z, Huang H, Tong C (2017) Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses. *Neurocomputing* 267:69–84
129. Wang G, Yao Y, Chen Z, Hu P (2019) Thermodynamic and optical analyses of a hybrid solar cpv/t system with high solar concentrating uniformity based on spectral beam splitting technology. *Energy* 166:256–266
130. Wang B, Zhang B, Liu X (2020a) An image encryption approach on the basis of a time delay chaotic system. *Optik* 225(165):737
131. Wang B, Zhang B, Liu X, Zou F (2020b) Novel infrared image enhancement optimization algorithm combined with dfocs. *Optik* 224(165):476
132. Wang M, Zhao X, Heidari AA, Chen H (2020c) Evaluation of constraint in photovoltaic models by exploiting an enhanced ant lion optimizer. *Solar Energy* 211:503–521
133. Wang S, Zhang K, van Beek LP, Tian X, Bogaard TA (2020d) Physically-based landslide prediction over a large region: scaling low-resolution hydrological model results for high-resolution slope stability assessment. *Environ Model Softw* 124(104):607
134. Wang X, Chen H, Heidari AA, Zhang X, Xu J, Xu Y, Huang H (2020) Multi-population following behavior-driven fruit fly optimization: a Markov chain convergence proof and comprehensive analysis. *Knowl Based Syst* 210(106):437
135. Wei Y, Lv H, Chen M, Wang M, Heidari AA, Chen H, Li C (2020) Predicting entrepreneurial intention of students: an extreme learning machine with gaussian barebone Harris Hawks optimizer. *IEEE Access* 8:76,841–76,855. <https://doi.org/10.1109/ACCESS.2020.2982796>
136. Wolpert DH, Macready WG, et al. (1995) No free lunch theorems for search. Technical Report SFI-TR-95-02-010, Santa Fe Institute, Tech rep
137. Wu J, Nan R, Chen L (2019a) Improved salp swarm algorithm based on weight factor and adaptive mutation. *J Exp Theor Artif Intell* 31(3):493–515
138. Wu T, Cao J, Xiong L, Zhang H (2019b) New stabilization results for semi-markov chaotic systems with fuzzy sampled-data control. *Complexity*

139. Wu C, Wu P, Wang J, Jiang R, Chen M, Wang X (2020a) Critical review of data-driven decision-making in bridge operation and maintenance. *Struct Infrastruct Eng*:1–24
140. Wu T, Xiong L, Cheng J, Xie X (2020) New results on stabilization analysis for fuzzy semi-Markov jump chaotic systems with state quantized sampled-data controller. *Inf Sci* 521:231–250
141. Xia J, Chen H, Li Q, Zhou M, Chen L, Cai Z, Fang Y, Zhou H (2017) Ultrasound-based differentiation of malignant and benign thyroid nodules: an extreme learning machine approach. *Comput Methods Progr Biomed* 147:37–49
142. Xing Z, Jia H (2019) Multilevel color image segmentation based on glcm and improved salp swarm algorithm. *IEEE Access* 7:37,672–37,690
143. Xiong Z, Xiao N, Xu F, Zhang X, Xu Q, Zhang K, Ye C (2020) An equivalent exchange based data forwarding incentive scheme for socially aware networks. *J Signal Process Syst*:1–15
144. Xu X, Chen HL (2014) Adaptive computational chemotaxis based on field in bacterial foraging optimization. *Soft Comput* 18(4):797–807
145. Xu Y, Chen H, Heidari AA, Luo J, Zhang Q, Zhao X, Li C (2019a) An efficient chaotic mutative moth-flame-inspired optimizer for global optimization tasks. *Expert Syst Appl* 129:135–155. <https://doi.org/10.1016/j.eswa.2019.03.0433>
146. Xu Y, Chen H, Luo J, Zhang Q, Jiao S, Zhang X (2019b) Enhanced moth-flame optimizer with mutation strategy for global optimization. *Inf Sci* 492:181–203. <https://doi.org/10.1016/j.ins.2019.04.0224>
147. Xu Y, Chen H, Luo J, Zhang Q, Jiao S, Zhang X (2019c) Enhanced moth-flame optimizer with mutation strategy for global optimization. *Inf Sci* 492:181–203
148. Xu B, Pang R, Zhou Y (2020) Verification of stochastic seismic analysis method and seismic performance evaluation based on multi-indices for high cfrds. *Eng Geol* 264(105):412
149. Yan J, Pu W, Zhou S, Liu H, Bao Z (2020a) Collaborative detection and power allocation framework for target tracking in multiple radar system. *Inf Fus* 55:173–183
150. Yan J, Pu W, Zhou S, Liu H, Greco MS (2020b) Optimal resource allocation for asynchronous multiple targets tracking in heterogeneous radar networks. *IEEE Trans Signal Process* 68:4055–4068
151. Yang XS (2010) Firefly algorithm, levy flights and global optimization. In: *Research and development in intelligent systems XXVI*, Springer, pp 209–218
152. Yang XS, Gandomi AH (2012) Bat algorithm: a novel approach for global engineering optimization. *Eng Comput*
153. Yang L, Chen H (2019) Fault diagnosis of gearbox based on rbf-pf and particle swarm optimization wavelet neural network. *Neural Comput Appl* 31(9):4463–4478
154. Yang S, Deng B, Wang J, Li H, Lu M, Che Y, Wei X, Loparo KA (2019) Scalable digital neuromorphic architecture for large-scale biophysically meaningful neural network with multi-compartment neurons. *IEEE Trans Neural Netw Learn Syst* 31(1):148–162
155. Yang Y, Chen H, Li S, Heidari AA, Wang M (2020) Orthogonal learning harmonizing mutation-based fruit fly-inspired optimizers. *Appl Math Model* 86:368–383. <https://doi.org/10.1016/j.apm.2020.05.0195>
156. Yu C, Heidari AA, Chen H (2020) A quantum-behaved simulated annealing algorithm-based moth-flame optimization method. *Appl Math Model* 87:1–19. <https://doi.org/10.1016/j.apm.2020.04.0196>
157. Yue H, Wang H, Chen H, Cai K, Jin Y (2020) Automatic detection of feather defects using lie group and fuzzy fisher criterion for shuttlecock production. *Mech Syst Signal Process* 141(106):690. <https://doi.org/10.1016/j.ymssp.2020.1066907>
158. Zhang C, Ou J (2015) Modeling and dynamical performance of the electromagnetic mass driver system for structural vibration control. *Eng Struct* 82:93–103
159. Zhang C, Ou J, Zhang J (2006) Parameter optimization and analysis of a vehicle suspension system controlled by magnetorheological fluid dampers. *Struct Control Health Monit* 13(5):885–896
160. Zhang J, Wang Z, Luo X (2018) Parameter estimation for soil water retention curve using the salp swarm algorithm. *Water* 10(6):815
161. Zhang X, Wang Y, Chen X, Su CY, Li Z, Wang C, Peng Y (2018) Decentralized adaptive neural approximated inverse control for a class of large-scale nonlinear hysteretic systems with time delays. *IEEE Trans Syst Man Cybern Syst* 49(12):2424–2437
162. Zhang Q, Chen H, Heidari AA, Zhao X, Xu Y, Wang P, Li Y, Li C (2019) Chaos-induced and mutation-driven schemes boosting salp chains-inspired optimizers. *Ieee Access* 7:31243–31261
163. Zhang H, Cai Z, Ye X, Wang M, Kuang F, Chen H, Li C, Li Y (2020a) A multi-strategy enhanced salp swarm algorithm for global optimization. *Eng Comput*. <https://doi.org/10.1007/s00366-020-01099-48>
164. Zhang H, Heidari AA, Wang M, Zhang L, Chen H, Li C (2020b) Orthogonal nelder-mead moth flame method for parameters identification of photovoltaic modules. *Energy Convers Manag*. <https://doi.org/10.1016/j.enconman.2020.1127649>
165. Zhang H, Li R, Cai Z, Gu Z, Heidari AA, Wang M, Chen H, Chen M (2020c) Advanced orthogonal moth flame optimization with broyden–fletcher–goldfarb–shanno algorithm: framework and real-world problems. *Expert Syst Appl*:113617
166. Zhang H, Qiu Z, Cao J, Abdel-Aty M, Xiong L (2020d) Event-triggered synchronization for neutral-type semi-markovian neural networks with partial mode-dependent time-varying delays. *IEEE Trans Neural Netw Learn Syst* 31(11):4437–4450. <https://doi.org/10.1109/TNNLS.2019.29552870>
167. Zhang H, Wang Z, Chen W, Heidari AA, Wang M, Zhao X, Liang G, Chen H, Zhang X (2020e) Ensemble mutation-driven salp swarm algorithm with restart mechanism: framework and fundamental analysis. *Expert Syst Appl* 165(113):897
168. Zhang K, Ruben GB, Li X, Li Z, Yu Z, Xia J, Dong Z (2020f) A comprehensive assessment framework for quantifying climatic and anthropogenic contributions to streamflow changes: a case study in a typical semi-arid north china basin. *Environ Model Softw*:104704
169. Zhang X, Xu Y, Yu C, Heidari AA, Li S, Chen H, Li C (2020g) Gaussian mutational chaotic fruit fly-built optimization and feature selection. *Expert Syst Appl* 141(112):976
170. Zhang Y, Liu R, Heidari AA, Wang X, Chen Y, Wang M, Chen H (2020h) Towards augmented kernel extreme learning models for bankruptcy prediction: algorithmic behavior and comprehensive analysis. *Neurocomputing*. <https://doi.org/10.1016/j.neuco.2020.10.0381>
171. Zhang Y, Liu R, Wang X, Chen H, Li C (2020i) Boosted binary harris hawks optimizer and feature selection. *Eng Comput*. <https://doi.org/10.1007/s00366-020-01028-52>
172. Zhang H, Wang Z, Chen W, Heidari AA, Wang M, Zhao X, Liang G, Chen H, Zhang X (2021) Ensemble mutation-driven salp swarm algorithm with restart mechanism: framework and fundamental analysis. *Expert Syst Appl* 165:113897. <https://doi.org/10.1016/j.eswa.2020.1138973>
173. Zhao X, Li D, Yang B, Ma C, Zhu Y, Chen H (2014) Feature selection based on improved ant colony optimization for online detection of foreign fiber in cotton. *Appl Soft Comput* 24:585–596

174. Zhao X, Li D, Yang B, Chen H, Yang X, Yu C, Liu S (2015) A two-stage feature selection method with its application. *Comput Electr Eng* 47:114–125
175. Zhao X, Zhang X, Cai Z, Tian X, Wang X, Huang Y, Chen H, Hu L (2019) Chaos enhanced grey wolf optimization wrapped elm for diagnosis of paraquat-poisoned patients. *Comput Biol Chem* 78:481–490. <https://doi.org/10.1016/j.compbiolchem.2018.11.0174>
176. Zhao D, Liu L, Yu F, Heidari AA, Wang M, Liang G, Muhammad K, Chen H (2020a) Chaotic random spare ant colony optimization for multi-threshold image segmentation of 2d kapur entropy. *Knowl Based Syst*:106510
177. Zhao D, Liu L, Yu F, Heidari AA, Wang M, Oliva D, Muhammad K, Chen H (2020b) Ant colony optimization with horizontal and vertical crossover search: Fundamental visions for multi-threshold image segmentation. *Expert Syst Appl*:114122
178. Zhu B, Su B, Li Y (2018) Input-output and structural decomposition analysis of India's carbon emissions and intensity, 2007/08–2013/14. *Appl Energy* 230:1545–1556
179. Zhu J, Wang X, Chen M, Wu P, Kim MJ (2019) Integration of BIM and GIS: IFC geometry transformation to shapefile using enhanced open-source approach. *Autom Constr* 106:102859
180. Zhu J, Wang X, Wang P, Wu Z, Kim MJ (2019) Integration of BIM and GIS: Geometry from IFC to shapefile using open-source technology. *Autom Constr* 102:105–119
181. Zhu L, Kong L, Zhang C (2020) Numerical study on hysteretic behaviour of horizontal-connection and energy-dissipation structures developed for prefabricated shear walls. *Appl Sci* 10(4):1240
182. Zhu G, Wang S, Sun L, Ge W, Zhang X (2020) Output Feedback Adaptive Dynamic Surface Sliding-Mode Control for Quadrotor UAVs with Tracking Error Constraints. *Complexity* 2020:1–23
183. Zhu J, Wu P, Chen M, Kim MJ, Wang X, Fang T (2020) Automatically Processing IFC Clipping Representation for BIM and GIS Integration at the Process Level. *Appl Sci* 10(6):2009

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.