An effective solution to numerical and multi-disciplinary design optimization problems using chaotic slime mold algorithm

Dinesh Dhawale¹,² · Vikram Kumar Kamboj¹,⁴ · Priyanka Anand³

Received: 17 December 2020 / Accepted: 20 April 2021 / Published online: 30 May 2021
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

Abstract
Slime mold algorithm (SMA) is a recently developed meta-heuristic algorithm that mimics the ability of a single-cell organism (slime mold) for finding the shortest paths between food centers to search or explore a better solution. It is noticed that entrapment in local minima is the most common problem of these meta-heuristic algorithms. Thus, to further enhance the exploitation phase of SMA, this paper introduces a novel chaotic algorithm in which sinusoidal chaotic function has been combined with the basic SMA. The resultant chaotic slime mold algorithm (CSMA) is applied to 23 extensively used standard test functions and 10 multidisciplinary design problems. To check the validity of the proposed algorithm, results of CSMA has been compared with other recently developed and well-known classical optimizers such as PSO, DE, SSA, MVO, GWO, DE, MFO, SCA, CS, TSA, PSO-DE, GA, HS, Ray and Sain, MBA, ACO, and MMA. Statistical results suggest that chaotic strategy facilitates SMA to provide better performance in terms of solution accuracy. The simulation result shows that the developed chaotic algorithm outperforms on almost all benchmark functions and multidisciplinary engineering design problems with superior convergence.

Keywords Slime mold algorithm (SMA) · CSMA · Convergence rate

1 Introduction
Meta-heuristic algorithms being simple and easy to implement are effectively applied to continuous, discrete, constrained, or unconstrained problems which were found hard to solve using conventional methods such as conjugate gradient, quadratic programming, and quasi-network methods. These meta-heuristic algorithms are single solution based which provides only one solution during optimization or population-based which mimics mostly natural phenomena and evolves a set of solutions during each iteration. These meta-heuristic algorithms are mainly categorized into four main groups: evolutionary, physics-based, human-based, and swarm intelligence type algorithms. Evolutionary algorithms (EAs) such as genetic algorithm (GA), differential evolution (DE) [1], genetic programming (GP) [2], and evolution strategy (ES) [3] mimics behaviors such as selection, recombination and mutation. The second class utilizes some physical laws such as gravitational search algorithm (GSA) [4], big-bang big-crunch (BBBC) [5], multi-verse optimizer (MVO) [6] and sine cosine algorithm (SCA) [7]. The third category mimics certain human behaviors which includes some of the well-known algorithms such as Tabu search (TS) [8], teaching learning based optimization (TLBO) [9], socio evolution and learning optimization (SELO). The last category of P-meta-heuristics uses collective or social intelligence that artificially simulates the behaviors such moving in swarms, flocks and herds. This Swarm Intelligence includes particle swarm optimization (PSO) [10], ant colony optimization (ACO) [11], artificial bee colony (ABC) [12], machine learning (ML) [13], bat-inspired algorithm (BA) [14], grey wolf optimization (GWO) [15], moth flame alternative algorithms [16], and so on.
optimization (MFO) [16], artificial neural network (ANN) [17] and Harris Hawk optimizer (HHO) [18]. Machine-learning approach have great potential in solving numerical complexities involving large variables [19]. These algorithms have their own strategies but although they have two common search feature steps: diversification and intensification. The random search process is the explorative (diversification) process where the operators search the different regions more intensely to obtain the global optima. The local search process is the intensification step which is performed after the exploration phase to enhance solution accuracy. A proper balance between exploration and exploitation is the basic need to avoid any local minima entrapment. A large variety of meta-heuristic algorithms has been invented to solve various numerical and engineering optimization problems but still, there is always a scope of improvement in the existing research by utilizing no-free-lunch (NFL) theory. This theorem persuades that no method is the globally best and there is always scope for new more efficient algorithms. The researchers are continuously working on different variants to employ innumerable sorts of advanced methods on various problems. The concise surveys of the different

| Algorithm                      | Year  | References | Benchmark problems                      | Problem type              |
|--------------------------------|-------|------------|-----------------------------------------|---------------------------|
| Variable neighborhood search   | 2007  | [20]       | 16                                      | Open vehicle routing      |
| Biogeography-based optimization| 2008  | [21]       | 14                                      | Real world                |
| Gravitational search algorithm | 2009  | [4]        | 23                                      | NA                        |
| Firework algorithm             | 2010  | [22]       | 9                                       | NA                        |
| Krill Herd algorithm           | 2012  | [23]       | 20                                      | NA                        |
| Multi-start methods            | 2012  | [24]       | NA                                      | Standard benchmark        |
| Water cycle algorithm          | 2012  | [25]       | 19                                      | Engineering design optimization |
| Animal migration optimization  | 2013  | [26]       | 23                                      | NA                        |
| Cultural evolution algorithm   | 2013  | [27]       | 7                                       | Reliability engineering   |
| Grey wolf optimizer            | 2014  | [15]       | 29                                      | Engineering design optimization |
| Symbiotic organism search      | 2014  | [28]       | 26                                      | Engineering design optimization |
| Interior search algorithm      | 2014  | [29]       | 14                                      | Engineering design optimization |
| Binary PSO-GSA                 | 2014  | [30]       | 22                                      | NA                        |
| Competition over resources     | 2014  | [31]       | 8                                       | NA                        |
| Chaotic Krill Herd algorithm   | 2014  | [32]       | 14                                      | NA                        |
| Stochastic fractal search      | 2014  | [33]       | 23                                      | Engineering design optimization |
| Exchange market algorithm      | 2014  | [34]       | 12                                      | NA                        |
| Forest optimization algorithm  | 2014  | [35]       | 4                                       | Feature weighting         |
| Binary Gray Wolf optimization  | 2015  | [36]       | 18                                      | Design formulation        |
| Bird swarm algorithm           | 2015  | [37]       | 18                                      | NA                        |
| Elephant herding optimization  | 2015  | [38]       | 15                                      | NA                        |
| Electromagnetic field optimization | 2015 | [39]       | 30                                      | Global                    |
| Fuzzy optimization technique   | 2015  | [40]       | 29                                      | Optimization              |
| Lightning search algorithm     | 2015  | [41]       | 24                                      | NA                        |
| Moth-flame optimization algorithm | 2015 | [16]       | 29                                      | Engineering design        |
| Multi-verse optimizer          | 2015  | [6]        | 19                                      | Engineering optimization  |
| Grasshopper optimization algorithm | 2017 | [42]       | 19                                      | Global                    |
| GWO-SCA                        | 2017  | [43]       | 22                                      | Bio-medical optimization  |
| Lion optimization algorithm    | 2017  | [44]       | NA                                      | Engineering design optimization |
| Binary whale optimization algorithm | 2018 | [45]       | NA                                      | Unit commitment            |
| Coyote optimization algorithm  | 2018  | [46]       | 40                                      | Standard benchmark        |
| Self-adaptive differential artificial bee colony algorithm | 2019 | [47]       | 28                                      | Optimization              |
| The Sailfish optimizer         | 2019  | [48]       | 20                                      | Standard test function    |
| Synthetic minority over-sampling | 2019 | [49]       | NA                                      | Data communication        |
| Harris Hawks optimizer         | 2019  | [18]       | 29                                      | Standard benchmark        |
stochastic meta-heuristics and heuristic methods are presented in Table 1. The rest of the paper is arranged as: Sect. 2 includes a comprehensive review of the latest SMA variants. Section 3 comprises the basics of Physarum polycephalum and the mathematical model. In Sect. 4, standard benchmark test functions are included. Section 5 includes test results of the proposed algorithm and comparative analysis with well-known algorithms. Section 6 presents analysis of 10 multi-disciplinary problems. In Sect. 7, paper is concluded. Finally, limitations and future scope are explored in Sect. 8.

2 Literature survey of some recent SMA and chaotic variants

In this section, a comprehensive study of specific allied work has been presented to explore information regarding various recent advancements related to SMA and Chaotic strategies. The foraging behavior of Physarum polycephalum to discover new sites for food has been efficiently mimicked by many researchers to develop many new meta-heuristic algorithms with different platforms. Adamatzky et al. had experimentally proved that plasmodium (slime mold) could navigate through dissimilar channels without exploring all possible solutions. But, some irregularities are noticed that restrict the maze-solvers to compete with conventional architectures [50]. Nakagaki et al. presented mathematical statistics showing intelligent behavior of plasmodium wherein a tabular network is formed to find multiple food sources via the shortest path. It is observed that the proposed method not only explores different shapes but also provides overtime depending states [51]. Andrew Adamatzky and Jeff Jones effectively implemented the foraging capability to search optimal routes in ten urban areas and perform reconfiguring of transport networks [52]. Beekman et al. carried out an intensive study about the decision-making process of slime molds for choosing the optimal path for getting nutrition by searching the shortest path [53]. Burgin et al. designed the modal of a structural machine by adopting the inherent ability of molds to sense cell information. Further, it is noticed that structural machines have the potential to solve complex computational problems by implementing machine algorithms [54]. Daniel et al. had applied the SMA technique to solve the transportation issue due to growing urbanization. In this concern, SMA provides an optimal solution to minimize the travel path by searching the shortest path [55]. Houbraken et al. have developed an extended fault-tolerant algorithm by utilizing the slime mold concept to improve the fault-tolerant network in the telecommunication sector [56]. Kropat et al. presented a deterministic approach to solving single path and multi-path optimization problems under uncertainty. This research also explores the robustness of the SMA algorithm to tackle emergencies and avoiding disasters [57]. Abdel-Basset et al. have developed a hybrid algorithm by incorporating Harris’s Hawk’s algorithm with whale optimization algorithm (WOA). In this work, the image segmentation problem (ISP) related to the X-ray of an infected person due to Covid-19 has been improved [58]. Zhao et al. presented a new levy flight distributed parameters to enhance the performance of basic SMA [59]. Patino-Ramirez et al. described the influence of chemical composition on the morphology and dynamics of slime mold [60]. Kouadri et al. have analyzed the optimal power flow problem of a hybrid renewable system. This work also effectively improves system stability by integrating VAR compensators with the thermal-wind system [61]. GAO et al. have presented a hybrid algorithm by combining grey wolf optimizer with SMA and three different types of optimization problems [62]. Nguyen et al. incorporated SMA algorithm for handling hydropower generation and compared simulation results with other algorithms [63]. Davut et al. presented a novel solution to PID-controlled DC motor and AVR system by utilizing the exploitation capability of the SMA algorithm [64]. Chen et al. introduced the SMA algorithm for solving stochastic optimization problems and had effectively applied the SMA algorithm to solve benchmark mark and engineering design problems [65]. Recently, researchers are getting more attracted toward hybrid and chaotic variants as these strategies have the inherent capacity to enhance the local search process. A large number of chaos optimization algorithms (COAs) and hybrid COA have proven that chaos could easily escape from local minima as compared to classical stochastic optimization algorithms. There are various chaotic variants such as adaptive chaotic sine cosine algorithm [66], chaotic whale optimization algorithm [67], chaotic dragonfly algorithm [68], modified whale optimization algorithm [69], chaotic Krill Herd algorithm [32], binary grasshopper optimization algorithm [70], chaotic grey wolf optimization [71]. Some of the recent SMA and chaotic variants are explored in Table 2 to get more familiar with various concepts related to the proposed work.

From the above literature survey, it is noticed that a large variety of meta-heuristic and hybrid variants have been invented by the researchers to fix different types of stochastic complexities. Some real-world problems such as network foraging, fault-tolerant, transportation, structural machines, engineering design, image segmentation, optimal power flow, and feature selection were analyzed by various researchers using a heuristic approach. The solution accuracy of any algorithm depends on its capability to have a proper balance between intensification and diversification. Studies revealed that slow convergence is the common faintness of most heuristic algorithms. This ultimately gives rise to reduced computational efficiency. Thus, to improve the solution efficiency, a trend of developing hybrid algorithms is escalating fast. In addition, diverse chaotic strategies
Table 2  Review of some recent SMA and chaotic variants

| Sr. no. | Algorithm                                                      | References | Year | Main findings related to proposed work                                                                                                                                 |
|---------|---------------------------------------------------------------|------------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1       | HSMA_WOA                                                      | [58]       | 2020 | In this work, image segmentation problem (ISP) related to X-ray of an infected person due to Covid-19 was examined                                                                 |
| 2       | K-means clustering and chaotic slime mold algorithm           | [72]       | 2020 | This work deals with parameter setting using two different techniques. Eight benchmark problems are simulated on 6 different datasets using the proposed algorithm           |
| 3       | MOSMA: multi-objective slime mould algorithm                 | [73]       | 2020 | In this research, enlist sorting strategy was employed to improve the convergence rate. Forty-one different multi-dimensional design are tested to validate the proposed method |
| 4       | Chaotic slime mold algorithm with Chebyshev map               | [74]       | 2020 | In this work, 100 Monte Carlo experiments were performed using SMA and Chebyshev mapping. To check the validity of the proposed method, some standard benchmark functions were simulated |
| 5       | Chaotic Salp swarm algorithm                                 | [75]       | 2020 | An extensive study was carried by authors to study breast abnormalities in thermal images using CSSA algorithm with a proper balance between exploration and exploitation phases |
| 6       | Modified whale optimization algorithm                         | [76]       | 2020 | In this paper, the Tent chaos map and tournament selection strategy are presented. Six standard functions were tested for the truss problem analysis with lesser iterations, and the minimum weight |
| 7       | Adaptive chaotic sine cosine algorithm                        | [66]       | 2020 | This paper presents an improved SCA based using adaptive parameters and a chaotic approach. Two mechanisms were incorporated with SCA and tested on 31 benchmark functions for solving a constrained optimization problem |
| 8       | Chaotic whale optimization algorithm                          | [67]       | 2020 | In this research, combined heat and power economic dispatch was analyzed using a chaotic base whale optimization algorithm to minimize fuel costs as well as emissions. Two different nonlinear realistic power areas have been utilized to explore global challenges |
| 9       | Chaotic particle swarm optimization                           | [77]       | 2019 | In this work, the chaotic PSO method was implemented to solve the power system problem concerned with electric vehicles using MATLAB and CRUISE software. The result reveals that the parameters of the optimal function can be achieved for balancing the power performance and provides economic operation |
| 10      | Chaotic harmony search algorithm                              | [78]       | 2019 | In the research, properties such as uniform distribution to generate random numbers, employing virtual harmony memories, and dynamically tuning the algorithm parameters are explored. Combined economic emission dispatch problems were analyzed for Six test systems having 6, 10, 13, 14, 40, and 140 units |
| 11      | Binary grasshopper optimization algorithm                     | [79]       | 2019 | This paper presents binary grasshopper algorithm and comparative results of five well-known swarm-based algorithms used in feature selection problems for 20 data sets with various sizes |
| 12      | Chaotic dragonfly algorithm                                  | [80]       | 2019 | In this paper, the Chaotic Dragonfly Algorithm using ten chaotic maps were implemented by adjusting the main parameters of dragonflies’ activities to increase the convergence rate and enhance the competence of DA |
| 13      | Modified dolphin swarm algorithm                              | [81]       | 2019 | In this paper, chaotic mapping was incorporated with DSA. Rastrigin function with an optimal chaotic map was explored among eight chaotic maps. Rotated Hyper-Ellipsoid function and Sum Squares function, respectively, were used for high-dimensional Levy function |
| 14      | Genetic algorithm using theory of chaos                       | [82]       | 2019 | In this paper, chaotic strategy is applied to solve optimization problems. The results of experiments were found to be the average of all task results related to the three individual types of functions |
have been effectively incorporated by many researchers to optimize their specific objective function. The ultimate aim of these techniques is to provide an optimal solution for a pre-defined objective function. Recently, a chaotic variant of SMA using the “Chebyshev function” was presented by Zhao et al. [74]. In this work, 100 Monte Carlo experiments were performed using SMA and Chebyshev mapping. Only the “Sargan function” of the uni-modal test function was simulated for 100 iterations. “Sine Wave function” was simulated for multi-modal test function for the same number of iterations. It was noticed that the solution of these benchmark functions is not exploited to an appreciable level. To check the validity of this method, only two standard benchmark functions had been tested. It was noticed that the results given by Chebyshev and sine wave function were not capable of giving efficient solutions. In most cases, simulation results are subjected to premature convergence. Although the methodology of the proposed CSMA is similar to the basic variant but differs in terms of the selection of chaos map. In the proposed research the local search capability of basic SMA has been enhanced using “sinusoidal chaotic function”. The CSMA method has been effectively employed to evaluate global optimization, standard benchmark, and engineering design problems. The comparative

| Sr. no. | Algorithm                                      | References | Year | Main findings related to proposed work                                                                 |
|--------|-----------------------------------------------|------------|------|--------------------------------------------------------------------------------------------------------|
| 15     | Chaotic genetic algorithm                     | [82]       | 2019 | In this research, eight different chaotic variants were applied to improve the search ability of the basic system |
| 16     | Chaotic whale optimization algorithm          | [83]       | 2018 | Twenty benchmark functions were tested to endorse the applicability of the suggested scheme with 30 and 50 iterations |
| 17     | Chaotic grasshopper optimization algorithms   | [84]       | 2018 | In this research, the author has clubbed GOA with 10 different chaotic maps. Ten shifted and biased functions were considered with 30-dimensional and 50-dimensional benchmark problems. Further three truss bar designs were investigated and the results are compared with authentic algorithms |
| 18     | Cat swarm algorithm                           | [85]       | 2017 | In this study, Chaos Quantum-behaved Cat Swarm Optimization (CQCSO) algorithm has been projected to improve the accuracy of the CSO, by introducing a tent map for escaping local optimum. Further, CQCSO has been tested for five different test functions |
| 19     | Chaotic fruit fly algorithm                   | [86]       | 2017 | In this study, the chaotic element adjusts the fruit fly swarm location to search for food sources. Two separate procreative methods were implemented for new food sources, for local global search based on swarm location |
| 20     | Chaotic grey wolf algorithm                   | [71]       | 2017 | Ten different chaotic maps were tested for 13 standard benchmark functions. Further, five engineering design problems were tested using the CGWO algorithm |
| 21     | Chaotic particle swarm algorithm              | [70]       | 2016 | In this work, the chaotic dynamics property was combined with PSO to enhance the diversity of solutions for escaping from premature convergence. Four multi-modal functions were tested to check the optimality of the suggested Chaotic PSO technique |
| 22     | CS-PSO: chaotic particle swarm algorithm      | [87]       | 2016 | In this study, combinatorial optimization problems are solved by utilizing the periodicity of the chaotic maps |
| 23     | Cooperative optimization algorithm            | [88]       | 2015 | This paper presents, chaotic ant swarm algorithm for analyzing the dynamic characteristics of a distributed system in a multi agent system at micro level for allocation in a networked multi agent system |
| 24     | Swarm optimization with various chaotic maps  | [89]       | 2014 | In this paper, effects of nine chaotic maps on the performance of system. For all problems, swarm size was set to 20, while the number of dimensions was set to 30 and 50 with maximum iterations of 2000 and 3000 |
| 25     | Chaotic invasive weed algorithm               | [90]       | 2014 | In this research, the standard IEEE 30-bus system is tested using chaos, and optimal settings of Power flow control is explored with non-smooth and non-convex generator fuel cost curves |
analysis demonstrated in the result section revealed that the suggested method gives outstanding performance in terms of fitness evaluation and solution accuracy.

3 Chaotic slime mold algorithm

3.1 Background of proposed research

It was found that the organism’s behavior could be easily adopted and statistically modeled to handle unconstrained and non-convex mathematics. Investigators have endeavored to mimic the working guidelines to progress computations and algorithms. Slime molds have acknowledged ample courtesy in contemporary years. The slime mold points out and algorithms. Slime molds have acknowledged ample

3.2 Basic slime mold algorithm

Step 1 In this step, mathematics for the slime mold behavior is formed and following rule is assigned to find updated position of during search for food. The criteria for this depends upon r and p. This is the contraction mode of mold:

$$X(t + 1) = \begin{cases} X_B(t) + \overline{v}_b \cdot \left( \frac{W \cdot X_A(t) - X_B(t)}{\overline{v}_c \cdot X(t)} \right) & r < p \\ \overline{v}_c \cdot X(t) & r \geq p \end{cases}$$

where $\overline{v}_b$ is a parameter with a range of $[-a,a]$ , $\overline{v}_c$ is the parameter which approaches linearly toward zero. $t$ is the current iteration, $X_A$ is the location of each particle in region where odor is maximum, $X$ is the mold’s location. $\bar{X}_A$ and $\bar{X}_B$ are the randomly selected variables from the swarm, $W$ is the measure of weights of masses.

The maximum limit of $p$ is as follows:

$$p = \tan h[s(i) - DF], \quad (2)$$

where $i \in 1, 2, \ldots, n, S(i)=$ fitness of $X$, DF=overall fitness from all steps.

The equation of $\overline{v}_b$ as follows:

$$\overline{v}_b = [-a,a] \quad (3)$$

$$a = \arctan h \left(-\frac{t}{\text{max} t} + 1\right). \quad (4)$$

The equation of $\bar{W}$ is listed as follows:

$$W(\text{smell index}(i)) = \begin{cases} 1 + r \cdot \log \left( \frac{bF-S(i)}{bF-wF} + 1 \right), & \text{condition} \\ 1 - r \cdot \log \left( \frac{bF-S(i)}{bF-wF} + 1 \right), & \text{others} \end{cases}$$

$$\text{Smell Index} = \text{sort}(S), \quad (5)$$

where $S(i)$ rank first half of the population, $r$ is the random value in the interval of $[0,1]$, $bF$ is the optimal fitness obtained in the current iterative process, $wF$ is the worst fitness value obtained in the iterative process, and Sort (s) function sorts fitness values.
Step 2 The equation for upgrading the positions of agents (i.e. to wrap food) is given as follows:

\[
X^i = \begin{cases} 
    \text{rand} \cdot (\text{UB} - \text{LB}) + \text{LB}, & \text{rand} < z \\
    \frac{X_y(i) + \text{rand} \cdot (W \cdot X_i(t) - X_y(i))}{\text{rand} \cdot X_i(t)}, & r < p \\
    \frac{X_y(i) + \text{rand} \cdot (W \cdot X_i(t) - X_y(i))}{\text{rand} \cdot X_i(t)}, & r \geq p 
\end{cases}
\]

where LB and UB are the search limits, and rand and r denote the random value in [1].

Step 3 With the upgradation in the search process, the value of \(\text{rand} \cdot X_i(t)\) varies between \([-a, a]\) and \(\text{rand} \cdot X_i(t)\) varies between \([-1, 1]\) and at last shrinks to zero. This is known to be as ‘grabbling of food’.

3.3 Types of chaotic functions

The concept of probability distribution is captured by lot of meta-heuristics algorithms to gain randomness. Chaotic maps could be beneficial if randomness due to ergodicity, idleness and molding properties are replaced. These criteria’s are full filled by the following equation:

\[y_{k+1} = f(y_k).\]  

(8)

In Eq. (8), \(y_{k+1}\) and \(f(y_k)\) are the \((k + 1)\)th and \(K\)th chaotic number, respectively. The action of chaotic function is dependent on initial value \(y_0\). The particular type of chaotic function will generate a solution within the standardized equations as shown in Table 3.

### Table 3  Chaotic map functions

| Sr. no. | Chaotic name | Mathematical description | Chaotic description |
|---------|--------------|--------------------------|--------------------|
| 1       | Chebyshev    | \(y_{i+1} = \cos(\cos^{-1}(y_i))\) | \([-1, 1]\)        |
| 2       | Iterative    | \(y_{i+1} = \sin(\frac{a \pi}{y_i}), a = 0.7\) | \([-1, 1]\)        |
| 3       | Sinusoidal   | \(y_{i+1} = a \sin(\pi yi), a = 2.3\) | \([0, 1]\)         |
| 4       | Sine         | \(y_{i+1} = \frac{a}{2} \sin(\pi yi), a = 4\) | \([0, 1]\)         |
| 5       | Circle       | \(y_{i+1} = \text{mod}(y_i + b - (\frac{a \pi}{2}) \sin(2\pi y_i), 1); a = 0.5, b = 0.2\) | \([0, 1]\)         |
| 6       | Piecewise    | \[
    \begin{cases}
    y_i/p & 0 \leq y_i < p \\
    (y_i-p)/(0.5-p) & 0.5 \leq y_i < 1-p \\
    (1-p-y_i)/(0.5-p) & 0.5 \leq y_i < 1-p \\
    (1-y_i)/p & 1-p \leq y_i \leq 1, p = 0.4 \\
    \end{cases}
\] | \([0, 1]\)         |
| 7       | Gauss/mouse  | \[
    \begin{cases}
    1 & y_i = 0 \\
    1 \mod (\pi, 1) & \text{otherwise} \\
    \end{cases}
\] | \([0, 1]\)         |
| 8       | Singer       | \(y_{i+1} = \mu (7.02 y_i - 23.3 y_i^3 + 28.7 y_i^5 - 13.301875 y_i^7), \mu = 1.07\) | \([0, 1]\)         |
| 9       | Logistic     | \(y_{i+1} = a y_i (1 - y_i), a = 4\) | \([0, 1]\)         |
| 10      | Tent         | \[
    \begin{cases}
    y_{i+1} = \begin{cases}
    (y_i/0.7), & y_i < 0.7 \\
    (10/3)(1-y), & y_i \geq 0.7 \\
    \end{cases}
\] | \([0, 1]\)         |

3.4 Algorithm of proposed work

The basic SMA is combined with the sinusoidal chaotic approach to further enhance the performance. The pseudo-code for method proposed is as shown in Fig. 2.

There are several complex optimization problems wherein mathematical reformulations restrict to perform algorithms efficiently. The proposed Chaotic SMA variant is advantageous over standard algorithms along with standard SMA. The convergence rate of stochastic methods was noticed not much efficient due to premature convergence. Whenever some additional functionality is incorporated with the original system, the performance starts getting degraded. After a thorough review of various SMA and chaotic variants, it is observed that the sinusoidal function to be the most appropriate to improve exploitation. The proposed algorithm utilizes a sinusoidal chaotic function to intensify the search capacity of classical SMA and optimize the objective fitness of various problems. The chaotic approach enables SMA to regulate the initial parameters of the search and thus rectifies the local entrapment of molds. It is seen that some algorithms lack global search capability. For the efficient performance of any algorithm, there should a proper balance between its local and global search capabilities. In the proposed research, to incorporate these local and global search requirements, no composite operations are involved. Second, the efficacy of any algorithm is judged by the simulation time required to simulate a particular objective function. The experimental section of the paper reveals that the suggested method has improved solution efficacy to a greater extend.
Lastly, during the cumulative run process, as the simulation successive progresses, premature convergence “local area stagnation” is the most common problem of many algorithms. This local area stagnation problem is omitted in the proposed method by utilizing the properties of ergodicity. The basic feature of Chaotic SMA is to enhance the optimization process by improving local search capabilities. Twenty-three benchmark problems including uni-modal, multi-modal and fixed dimensions are tested using the suggested chaotic slime algorithm.

### 4 Test benchmark functions

The developed Chaotic SMA algorithm has been simulated on Intel Core TM, i5-3320 M CPU@2.60 GHz system. These standard benchmark function are characterized by their objective fitness in parameter space within a particular dimension (Dim), range, and frequency ($f_{\text{min}}$). In the entire work, $F_1$–$F_7$ represent as uni-modal test functions (U-Modal), $F_8$–$F_{13}$ are multi-modal test functions (M-Modal) and $F_{14}$–$F_{23}$ are fixed dimension (FD) functions. The effectiveness of the proposed Chaotic SMA optimization technique is examined by referring benchmark functions are taken [91, 92]. The characteristics of benchmark functions differ from each other. Some functions show better performance in exploring local search while a few functions are found to excellent in determining global optima. Table 4 illustrates equations of $F_1$–$F_7$ with their name, dimension, range, and frequency. On similar grounds, Tables 5 and 6 explore the details of multi-modal and fixed dimensions functions.

In the whole research study, 30 search agents are taken into considerations and the proposed algorithm is simulated for maximum iterations of 500. Figures 3, 4, and 5 illustrate the 3D view along with their objective space representing convergence for $F_1$–$F_7$, $F_8$–$F_{13}$ and $F_{14}$–$F_{23}$, respectively. It is clear from the various

| Uni-modal test function | Name                        | Dim | Limit       | $f_{\text{min}}$ |
|-------------------------|-----------------------------|-----|-------------|------------------|
| $f_1(y) = \sum_{i=1}^{n}y_i^2$ | Sphere function             | 30  | $[-100, 100]$ | 0                |
| $f_2(y) = \sum_{i=1}^{n}|y_i| \prod_{i=1}^{n}|y_i|$ | Schwefel absolute function | 30  | $[-10, 10]$  | 0                |
| $f_3(y) = \sum_{i=1}^{n} \left( \sum_{j=1}^{n}y_j \right)^2$ | Schwefel double sum function | 30  | $[-100, 100]$ | 0                |
| $f_4(y) = \max \{\{y_i|, 1 \leq i \leq n\}$ | Schwefel max. function      | 30  | $[-100, 100]$ | 0                |
| $f_5(y) = \sum_{i=1}^{n} \left[100(y_{i+1} - y_i)^2 + (y_i - 1)^2\right]$ | Rosenbrock function         | 30  | $[-30, 30]$  | 0                |
| $f_6(y) = \sum_{i=1}^{n} \left(y_i + 0.5\right)^2$ | The step function           | 30  | $[-100, 100]$ | 0                |
| $f_7(y) = \sum_{i=1}^{n} y_i^4 + \textrm{random}[0, 1]$ | Quartic random function     | 30  | $[-1.28, 1.28]$ | 0                |
Table 5 Multimodal test function (M-modal)

| Multimodal test function | Name                  | Dim | Limit     | $f_{\text{min}}$ |
|--------------------------|-----------------------|-----|-----------|------------------|
| $f_8(y) = \sum_{i=1}^{n} -y_i \sin(\sqrt{|y_i|})$ | Schwefel sine function | 30  | [-500, 500] | -418.98295       |
| $f_9(y) = \sum_{i=1}^{n} [y_i^2 - 10 \cos(2\pi y_i) + 10]$ | Rastrigin function    | 30  | [-5.12, 5.12] | 0                |
| $f_{10}(y) = -20 \exp \left( -0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} y_i^2} \right) - \exp \left( \frac{1}{n} \sum_{i=1}^{n} \cos(2\pi y_i) + 20 + c \right)$ | The Ackley function | 30  | [-32, 32]   | 0                |
| $f_{11}(y) = 1 + \frac{\sqrt{x}}{4000} - \prod_{i=1}^{n} \cos \left( \frac{\sqrt{y_i}}{\sqrt{i}} \right)$ | Griewank function     | 30  | [-600, 600] | 0                |
| $f_{12}(y) = \frac{1}{n} \left( 10 \sin(y_1) + \sum_{i=2}^{n} (y_i - 1)^2 \left[ 1 + 10 \sin^2(\pi y_i + 1) \right] + (z_n - 1)^2 \right) + \sum_{i=1}^{n} u(y_i, 10, 100, 4)$ | Penalized Penalty#1 function | 30  | [-50, 50]   | 0                |

$z_1 = 1 + \frac{y_{i+1}}{4}$

$u(y_i, a, k, m) = \begin{cases} 
  k(y_i - a) & y_i > a \\
  0 & -a < y_i < a \\
  k(-y_i - a) & y_i < -a 
\end{cases}$

$F_{13}(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^{n} \left( x_i - 1 \right)^2 \left[ 1 + \sin^2(3\pi x_i + 1) \right] \right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$ | Levi N. 13 function | 30  | [-50, 50]   | 0                |
Table 6  Fixed dimension test function

| FD test function | Name                      | Dim | Limit          | f/min |
|------------------|---------------------------|-----|----------------|-------|
| $f_{14}(y) = \frac{1}{500} + \sum_{j=1}^{5}\frac{1}{\sum_{i=1}^{6}(y_i-a_{ij})^2}$ | Shekel foxhole Function | 2   | $[-65.536, 65.536]$ | 1     |
| $f_{15}(y) = \sum_{i=1}^{11}(a_i - \frac{y_i(b_i+b_{2i})}{b_i+b_{2i}y_i})^2$ | Brad function            | 4   | $[-5, 5]$       | 0.00030 |
| $f_{16}(y) = 4y_1^2 - 2.1y_1^4 + \frac{1}{3}y_1^6 + y_1y_2 - 4y_2^2 + 4y_2^4$ | Camel function—six hump  | 2   | $[-5, 5]$       | -1.0316 |
| $f_{17}(y) = (y_2 - \frac{4}{4e^{y_1^2}} + \frac{4}{2y_1} - 6)^2 + 10\left(1 - \frac{1}{8}\right)\cos y_1 + 10$ | Branin RCOS function     | 2   | $[-5, 5]$       | 0.398  |
| $f_{18}(y) = \left[1 + (y_1 + y_2 + 1)^2\right]$ | Goldstein-price function | 2   | $[-2, 2]$        | 3      |
| $f_{19}(y) = \left[19 - 14y_1 + 3y_1^2 - 14y_2 + 6y_1y_2 + 3y_2^2\right]$ | Goldstein-price function | 2   | $[-2, 2]$        | 3      |
| $f_{20}(y) = -\sum_{i=1}^{4}c_i \exp\left(-\sum_{j=1}^{3}a_{ij}(y_j - q_{ij})^2\right)$ | Hartman 3 function       | 3   | $[1, 3]$         | -3.32  |
| $f_{21}(y) = -\sum_{i=1}^{4}c_i \exp\left(-\sum_{j=1}^{6}a_{ij}(y_j - q_{ij})^2\right)$ | Hartman 6 function       | 6   | $[0, 1]$         | -3.32  |
| $f_{22}(y) = -\sum_{i=1}^{5}\left[(y-a_i)^T + d_i\right]$ | Hybrid composition function #1 | 4   | $[0, 10]$       | -10.1532 |
| $f_{23}(y) = -\sum_{i=1}^{7}\left[(y-a_i)^T + d_i\right]$ | Hybrid composition function #2 | 4   | $[0, 10]$       | -10.4028 |
| $f_{24}(y) = -\sum_{i=1}^{10}\left[(y-a_i)^T + d_i\right]$ | Hybrid composition function #3 | 4   | $[0, 10]$       | -10.5363 |
Fig. 3 Three-dimensional view of F1–F7 along with convergence curve for SMA and CSMA
comparative results that the newly developed chaotic strategy appreciably increases convergence rate and thus improves its ability to easily escape from local minima entrapment.

5 Results of proposed algorithm

In this section, test results for benchmark functions are discussed with their average, best, worst, median, standard deviation, and $p$ value have been taken into account. The
stochastic complexity of the proposed algorithm is justified and analyzed by running the algorithm for 30 trial checks and 500 iterations. To analyze the feasibility of the solution, Wilcoxon sum test has been taken into account. The parameter setting for the proposed CSMA method is illustrated in Table 7. On similar grounds, results are compared with other universally validated systems.

Fig. 5 Three-dimensional view of F14–F23 along with convergence curve for SMA and CSMA
5.1 Testing of uni-modal functions (U-modal)

The search process for the best position depends upon the capability of the search agents to reach closer to origin. During the search process by various agents, there may be the possibility of getting entrapped far or nearby and accordingly defined in terms of exploration and exploitation. Exploration comes under the global search process and exploitation falls under the local search category. The statistical outcomes of U-Modal (F1–F7) have a few pick points with increased convergence validates the effectiveness of the proposed algorithm. Figure 6 illustrates a comparison between Chaotic SMA and different methods. It can be seen from the convergence curves that the suggested algorithm converges to optima much earlier.

To check the appropriateness of the proposed algorithm, each test function is simulated with SMA and CSMA. Table 8 illustrates statistical outcomes of the uni-modal benchmark function in terms of average, standard deviation, the best value, worst value, median value, and p value. In the search space, there are some regions of global optima whereas some regions are stagnated to local optima. The global search process determines the exploration phase whereas in the local search process, exploitation phase is evaluated. The performance of any algorithm is judged by its ability to attain the maxima or minima with less computation time. Table 9 shows the computational time in terms of best, mean, and worst time. Table 10 shows the comparison of the CSMA method with other techniques such as PSO [93], GWO [15], GSA [94], BA [95], FA [96], GA [97], BDA [98], BPSO [99], MFO [16], MVO [6], BGSA [100], SMS

| Parameter setting                              | CSMA |
|-----------------------------------------------|------|
| Number of search agents                       | 30   |
| Number of iterations for U-Modal, M-modal, and F-Modal | 500  |
| Number of iterations for engineering optimization design problems | 500  |
| Number of trial runs test functions           | 30   |
| Number of trial engineering design problems   | 30   |

Fig. 5 (continued)
Comparative Analysis of CSMA algorithm for F1

Comparative Analysis of CSMA algorithm for F2

Comparative Analysis of CSMA algorithm for F3

Comparative Analysis of CSMA algorithm for F4

Comparative Analysis of CSMA algorithm for F5

Comparative Analysis of CSMA algorithm for F6

Fig. 6 Convergence curve for U-Modal test function showing comparison of CSMA with other algorithms

Table 8 Test results for U-modal function using CSMA

| Functions | Average value | STD  | Best value | Worst value | Median value | p value |
|-----------|---------------|------|------------|-------------|--------------|---------|
| F1        | 1.2E−280      | 0    | 0          | 3.5E−279    | 0            | 0.5     |
| F2        | 3.4E−156      | 1.7E−155 | 3E−258    | 9.4E−155    | 1.3E−188     | 1.7344E−06 |
| F3        | 0             | 0    | 0          | 0           | 0            | 1       |
| F4        | 5.1E−134      | 2.8E−133 | 1.5E−269  | 1.5E−132    | 2.7E−190     | 1.7344E−06 |
| F5        | 5.035453      | 9.27916 | 0.044388  | 28.19006    | 1.219173     | 1.7344E−06 |
| F6        | 0.004431      | 0.003059 | 2.06E−05  | 0.016714    | 0.004434     | 1.7344E−06 |
| F7        | 0.0003        | 0.000211 | 2.17E−05  | 0.000935    | 0.000274     | 1.7344E−06 |
search algorithms such as PSO, GWO, GSA, are illustrated in Fig. 7. Comparative analysis revealed that method, independent trial runs for each benchmark function convergence. Further, to verify the effectiveness of the proposed former method are found to be subjected to premature con- is increased with higher efficiency. On the other hand, the search phase more intensively.

The characteristics of benchmark functions vary from each other. These test functions have different exploration and exploitation search capacities. Griewank, levy, Ackley, Rastrigin, Schwefel functions, sphere function, explore many local minima points while sum square function, Zakharov are applicable to explore global minima points. In this regard, test assessment of seven uni-modal benchmark functions (F1–F7) are analyzed. The test results for each function in terms of average and standard deviations over 30 independent trial runs and 500 iterations are recorded. The scalability assessment is carried out to investigate the impact of chaos on the solutions of SMA. The statistical results shown in Table 10 reveals an appreciable gap between CSMA and other methodologies. As it can be seen in Table 10, that by introducing a chaotic function, the exploration and exploitation phase of SMA has been improved. The results of CSMA when compared with PSO, GWO, DE, FA, GA, BBO, MFO, SCA, and SSA show remarkable performance in dealing with F3, F4, and F7 test functions. As per the convergence curves in Fig. 6, it is observed that the optimality of results is increased with higher efficiency. On the other hand, the former method are found to be subjected to premature convergence. Further, to verify the effectiveness of the proposed method, independent trial runs for each benchmark function are illustrated in Fig. 7. Comparative analysis revealed that the sinusoidal chaotic function facilitates to explore the local search phase more intensively.

### 5.2 Testing of multi-modal test functions (M-modal)

Figure 8 illustrates comparison between Chaotic SMA and other methods for multi-modal benchmark (F8–F13). The multi-modal function is tested for 30 introductory attempts and 500 iterations and results are shown in Table 11. Simulation time for M-Modal utilizing CSMA appears in Table 12. Table 13 shows compared results with other meta-heuristics search algorithms such as PSO, GWO, GSA, BA, FA, GA, BDA, BPSO, MFO, MVO, BGSA, SMS, FPA, DE, ALO, and WOA in terms of average value and std. deviation. It can be seen in Fig. 8 that test outcomes of M-Modal (F8–F13) have some pick points with increased convergence using CSMA, justifies the effectiveness of algorithm in solving multi-modal test functions.

From the statistical data analysis illustrated in Table 13, it is observed that the optimality of multi-modal test functions is marginally improved by implementing the sinusoidal chaotic function. According to the best and standard deviation results, CSMA shows better performance for almost all seven test functions. As can be seen from convergence curves shown in Fig. 8, CSMA gives optimal convergences except for a few of the test functions. It can be understood from the convergence comparison that CSMA converges faster and seizes the run as soon as it reaches the stop criterion. As per comparative curves shown in Fig. 8, it is observed that the suggested algorithm shows superior performance in dealing with F9, F10 and F11 and comparative operation in the case of F8, F12 and F13. Trial run accuracy matrices shown in Fig. 9 reveal that proposed CSMA significantly searches the local and global space more intensively.

### 5.3 Testing of fixed dimension function (F-modal)

Fixed dimension test functions (F14–F23) are tested for 30 trial runs and 500 iterations as shown in Fig. 11. Simulation results for FD test function using CSMA are shown in Table 14. Table 15 illustrates simulation results of F-Modal. It can be seen that results for F-Modal have many pick points with better convergence. Simulation time for F-Modal benchmark problems using CSMA is shown in Table 15. Table 16 illustrates CSMA results compared with others variants such as GWO, PSO, GSA, DE, and FEP in terms of average (AVG) and standard deviation (SD).

The statistical data analysis illustrated in Table 16 shows that solution accuracy of fixed dimension test functions is appreciably improved by implementing the sinusoidal chaotic function. Rendering to the best and standard deviation results, CSMA shows enhanced performance for nearly all ten test functions. As can be seen from convergence curves shown in Fig. 10, CSMA gives optimal convergences except for a few of the test functions. It can be implicit from the convergence evaluation that CSMA converges faster and seizes the run as soon as it reaches the halt criterion. As per comparative curves shown in Fig. 8, it is observed that the suggested algorithm shows superior performance in dealing with F14, F18, F19 and F20 and comparative operation in the case of F15, F21, F22 and F23. Trial runs shown in Fig. 11 reveal that proposed CSMA significantly searches the local and global space for finding the optimal solution.
In this section, ten real-world design problems are tested which includes “3-bar truss problem, pressure –vessel design, compression design, welded beam, cantilever beam design, gear train design problem, speed reducer problem, Belleville spring problem, rolling element problem and multidisc clutch brake problem” [112]. Each design problem is simulated with CSMA algorithm. The abbreviations for various multidisciplinary engineering functions (EFs) has been shown in Table 17. The comparison of the engineering design problem with their average, standard deviation, best, worst and p value has been elucidated in Table 18.
simulation time for best, mean and average values are shown in Table 19.

6.1 Three-bar truss design problem

The proposed chaotic SMA method is applied for solving problem of truss design as shown in Fig. 12 [113, 114]. Truss design problem depends on two variables and three parameters. The main focus of truss design problem is to minimize weight. The various constraints involved in truss bar design problem are warping, deflection and stress. These constraints are optimized to achieve the desired objective. The mathematical modeling of 3-bar truss are illustrated through Eqs. (9.1) to (9.1d) subject to various constraints. The solutions of CSMA were compared with existing methods and illustrated in Table 20. It is seen that the suggested method appreciably improves the objective of cost minimization. The design problem is modeled as shown below:

Consider,

\[
\tilde{y} = [y_1, y_2] = [A_1, A_2].
\]  

(9.1)

Minimize,

\[
f(\tilde{y}) = (2\sqrt{2}y_1 + y_2) \times l.
\]  

(9.1a)

Subjected to:

\[
g_1(\tilde{y}) = \frac{\sqrt{2}y_1 + y_2}{\sqrt{2}y_1^2 + 2y_1y_2} P - \sigma \leq 0
\]  

(9.1b)

\[
g_2(\tilde{y}) = \frac{y_2}{\sqrt{2}y_1^2 + 2y_1y_2} P - \sigma \leq 0
\]  

(9.1c)

\[
g_3(\tilde{y}) = \frac{1}{\sqrt{2}y_2 + y_1} P - \sigma \leq 0.
\]  

(9.1d)

6.2 Pressure vessel engineering problem

The design specification for this kind of engineering problem as illustrated in Fig. 13 [113, 114] is selected for reference. The chaotic SMA is applied to diminish the expense which includes the material cost and welding cost to form the vessel in cylindrical form. The four variables used to design the pressure vessel are: (i) shell thickness (\(T_s\)), (ii) head thickness (\(T_h\)), (iii) length of cylindrical unit (\(L_h\)), (iv) without head thickness (R). These four variables are modeled as \(y_1\), \(y_2\), \(y_3\), \(y_4\). The numerical formulation of this kind of problem is shown in Eqs. (9.2) through (9.2e). Table 21 shows the result assessment of suggested CSMA with some recent algorithms. From the comparative analysis, it is found that CSMA effectively reduces the expense of design by controlling the design variables:

Consider:

\[
\tilde{y} = [y_1y_2y_3y_4] = [T_sT_hRL_h].
\]  

(9.2)

Minimize;

\[
f(\tilde{y}) = 0.6224y_1y_2y_4 + 1.7781y_2y_3^2 + 3.1661y_1y_4 + 19.84y_1y_3
\]  

(9.2a)

Subject to:
Comparative Analysis of CSMA algorithm for F8

Comparative Analysis of CSMA algorithm for F9

Comparative Analysis of CSMA algorithm for F10

Comparative Analysis of CSMA algorithm for F11

Comparative Analysis of CSMA algorithm for F12

Comparative Analysis of CSMA algorithm for F13

Fig. 8 Convergence curve for M-Modal test function showing comparison of CSMA with other algorithms

Table 11 Testing of multimodal using CSMA

| Functions | Average value | STD     | Best value | Worst value | Median value | p value   |
|-----------|--------------|---------|------------|-------------|--------------|-----------|
| F8        | − 12,569.1   | 0.319584| − 12,569.5 | − 12,568.1  | − 12,569.2   | 1.7344E−06|
| F9        | 0            | 0       | 0          | 0           | 0            | 1         |
| F10       | 8.88E−16     | 0       | 8.88E−16   | 8.88E−16    | 8.88E−16     | 4.32046E−08|
| F11       | 0            | 0       | 0          | 0           | 0            | 1         |
| F12       | 0.003937     | 0.006237| 5.64E−07   | 0.032017    | 0.002066     | 1.7344E−06|
| F13       | 0.00664      | 0.00989 | 7.428E−05  | 0.05034892  | 0.00270      | 1.7344E−06|
6.3 Compression spring engineering design

Compression spring design concerned with mechanical engineering [113, 114] is shown in Fig. 14. The main intention of this type of problem is minimization of the spring weight. There are three design variables: (1) no. of active coils \(N_c\), (2) wire diameter \(d\), and (3) mean coil diameter \(D_m\). The formulation is shown in the Eqs. (9.3) through (9.5f). The proposed method is applied to solve compression spring design problem and results are compared with other methods shown in Table 22 for validation. It is clearly seen from the analysis that CSMA method is efficient for reducing spring weight marginally.

Consider

\[
\begin{align*}
\mathbf{y} &= [y_1 y_2 y_3] = [d_i D_m N_c], \\
\text{Minimize} & f(\mathbf{y}) = (y_3 + 2)y_2^2 y_1, \\
\text{Subject to:} & \\
g_1(\mathbf{y}) &= 1 - \frac{y_3^2 y_1}{71785y_1^4} \leq 0, \quad (9.3b) \\
g_2(\mathbf{y}) &= \frac{4y_2^2 - y_1 y_2}{12566(y_2 y_1^3 - y_1^4)} + \frac{1}{5108y_1^2} \leq 0, \quad (9.3c)
\end{align*}
\]

\[
g_3(\mathbf{y}) = 1 - \frac{140.45 y_1}{y_2 y_3} \leq 0, \quad (9.3e)
\]

\[
g_4(\mathbf{y}) = \frac{y_1 + y_2}{1.5} - 1 \leq 0. \quad (9.3f)
\]

Variable range \(0.005 \leq y_1 \leq 2.00, 0.25 \leq y_2 \leq 1.30, 2.00 \leq y_3 \leq 15.0\).

6.4 Welded beam design

In welded beam design, welding is carried out by fusing different sections by molten metal as shown in Fig. 15 [113, 114]. The main focus is on minimization of the making cost of the welded beam. The four variables are: (1) bar thickness \(b\) is specified by \(y_1\), (2) bar length \(l\) is specified by \(y_2\), (3) weld thickness \(h\) is specified by \(y_3\), and (4) the bar height \(h\) is specified by \(y_4\) which is subject to constraints, such as Buckling of bar \(P_c\), side constraints, End deflection of beam \(d\), bending stress of the beam \(h\) and stress of shear \(s\). The equations of the above-mentioned design problem are noted by Eqs. (9.4) through (9.4n). The results are compared with other methods as shown in Table 23. The comparative analysis shows that proposed method is competent for handling beam design problem more precisely.

Consider

\[
\begin{align*}
\mathbf{y} &= [y_1 y_2 y_3 y_4] = [hltb]. \\
\text{Minimize} & f(\mathbf{y}) = 1.10471 y_2^3 y_2 + 0.04811 y_3 y_4 (14.0 + y_2) \\
\text{Subject to} & \\
g_1(\mathbf{y}) &= \tau(\mathbf{y}) - \tau_{\text{max}} \leq 0, \quad (9.4b) \\
g_2(\mathbf{y}) &= \sigma(\mathbf{y}) - \sigma_{\text{max}} \leq 0, \quad (9.4c) \\
g_3(\mathbf{y}) &= \delta(\mathbf{y}) - \delta_{\text{max}} \leq 0, \quad (9.4d) \\
g_4(\mathbf{y}) &= y_1 - y_4 \leq 0, \quad (9.4e) \\
g_5(\mathbf{y}) &= P - P_c(\mathbf{y}) \leq 0, \quad (9.4f) \\
g_6(\mathbf{y}) &= 0.125 - y_1 \leq 0. \quad (9.4g)
\end{align*}
\]
\[ g(\vec{y}) = 1.10471y_1^2 + 0.04811y_3y_4(14.0 + y_2) - 5.0 \leq 0. \]  
\( (9.4h) \) 

Variable range \( 0.1 \leq y_1 \leq 2, \ 0.1 \leq y_2 \leq 10, \ 0.1 \leq y_3 \leq 10, \ 0.1 \leq y_4 \leq 2, \)

where
\[ \tau(\vec{y}) = \frac{\tau}{\tau^2 + \frac{y_2}{\sqrt{2R}}} \]

\[ \tau = \frac{P_i}{\sqrt{2y_1y_2}}, \quad \tau'' = \frac{MR}{J}, \quad M = P_i\left(L + \frac{y_3}{2}\right), \]

\[ R = \sqrt{\frac{y_2^2}{4} + \left(\frac{y_1 + y_3}{2}\right)^2}, \]

\[ J = 2\left\{\sqrt{2y_1y_2} \left[\frac{y_2^2}{4} + \left(\frac{y_1 + y_3}{2}\right)^2\right]\right\}, \]

\[ \sigma(\vec{y}) = \frac{6P_iL}{y_4y_3}, \quad \delta(\vec{y}) = \frac{6P_iL^3}{Ey_2^2y_4}, \]

**Fig. 9** Trial runs of M-Modal functions

**Table 14** Simulation results for fixed dimension test function using CSMA

| Functions | AVG    | SD      | Best value | Worst value | Median value | \( p \) value |
|-----------|--------|---------|------------|-------------|--------------|---------------|
| F14       | 0.998004 | 9.26E−13 | 0.998004   | 0.998004    | 0.998004     | 1.7344E−06    |
| F15       | 0.00055  | 0.000244 | 0.00031    | 0.001223    | 0.000469     | 1.7344E−06    |
| F16       | −1.03163 | 1.51E−09 | −1.03163   | −1.03163    | −1.03163     | 1.7344E−06    |
| F17       | 0.397887 | 6.82E−08 | 0.397887   | 0.397888    | 0.397887     | 1.7344E−06    |
| F18       | 3       | 8.43E−12 | 3          | 3           | 3            | 1.7344E−06    |
| F19       | −3.86278 | 4.21E−07 | −3.86278   | −3.86278    | −3.86278     | 1.7344E−06    |
| F20       | −3.25824 | 0.060654 | −3.20008   | −3.20309    | −3.20309     | 1.7344E−06    |
| F21       | −10.1528 | 0.000274 | −10.1519   | −10.1529    | −10.1529     | 1.7344E−06    |
| F22       | −10.4026 | 0.000208 | −10.4027   | −10.4027    | −10.4027     | 1.7344E−06    |
| F23       | −10.536  | 0.000299 | −10.5364   | −10.5354    | −10.5361     | 1.7344E−06    |

**Table 15** Simulation time for fixed dimension using CSMA

| Functions | Best time | Mean time | Worst time |
|-----------|-----------|-----------|------------|
| F14       | 1.140625  | 1.215104  | 1.921875   |
| F15       | 0.671875  | 0.788021  | 1.34375    |
| F16       | 0.53125   | 0.623438  | 1.171875   |
| F17       | 0.5       | 0.582813  | 1.21875    |
| F18       | 0.5       | 0.595313  | 1.125      |
| F19       | 0.59375   | 0.680729  | 1.265625   |
| F20       | 0.859375  | 0.971354  | 1.46875    |
| F21       | 0.8125    | 0.941146  | 1.4375     |
| F22       | 0.859375  | 0.972917  | 1.40625    |
| F23       | 0.9375    | 1.065104  | 1.59375    |
6.5 Cantilever beam design

This is civil engineering problem in which main focus is minimization of beam weight as shown in Fig. 16. In beam design, there are five elements \( l_1, l_2, l_3, l_4 \) and \( l_5 \) [114]. The main aim is minimization of the weight of the beam shown in Eq. (9.5). Taking care that displacement of vertical constraint not to disturb during finishing process of the beam for final optimal solution shown by Eq. (9.5a) to Eq. (9.5b). The results shown in Table 24 validate that CSMA algorithm efficiently reduces the weight of the beam. The formulation of design is given below:

Consider

\[
\bar{x} = [l_1 l_2 l_3 l_4 l_5].
\]  

Minimize

\[
f(\bar{x}) = 0.6224(l_1 + l_2 + l_3 + l_4 + l_5),
\]  

Subject to

\[
g(\bar{x}) = \frac{61}{\bar{P}_1} + \frac{37}{\bar{P}_2} + \frac{19}{\bar{P}_3} + \frac{7}{\bar{P}_4} + \frac{1}{\bar{P}_5} \leq 1.
\]  

6.6 Gear train design

In this method, the four variables \( g_1, g_2, g_3, \) and \( g_4 \) are reformed to diminish the scalar value and teeth ratio as shown in Fig. 17 [113]. Teeth on each gear are the decision variables in designing process. The gear train design problem is formulated through Eq. (9.6a) to Eq. (9.6b). The simulation results shown in Table 25 reveal that CSMA method gives comparison of results with other methods. From the assessment of test results, it is seen that proposed method effectively evaluates the gear train ratio.

Let us consider:

\[
\tilde{\vec{y}} = [g_1 g_2 g_3 g_4] = [M_A M_B M_C M_D]
\]  

Minimizing

\[
P_c(\tilde{\vec{y}}) = \frac{4.013E^{\frac{\gamma_3^2}{36}}}{L^2} \left(1 - \frac{\gamma_3}{2L} \sqrt{\frac{E}{4G}}\right)
\]  

\[
P_i = 6000 lb, L = 14 in, \delta_{\text{max}, i} = 0.25 in,
E = 30 \times 10^6 \text{ psi}, G = 12 \times 10^6 \text{ psi},
r_{\text{max}, i} = 13600 \text{ psi}, \sigma_{\text{max}, i} = 3000 \text{ psi}.
Subject to: $12 \leq g_1, g_2, g_3, g_4 \leq 60$.

6.7 Speed reducer engineering design problem

The fundamental problem is to limit the heaviness of the speed reducer. This type of design problem consist of seven variables as shown in Fig. 18 [113]. The seven variables are face width ($x_1$), teeth module ($x_2$), pinion teeth ($x_3$), first shaft length ($x_4$), second shaft length ($x_5$), the first shaft diameter ($x_6$) and second shaft diameter ($x_7$). The results shown in Table 26 shows that optimum fitness has been improved to some extend from previous evaluation. The equations are formulated as given below:

Minimizing

\[
\begin{align*}
\hat{f}(\mathbf{x}) &= 0.7854x_1x_2(3.3333x_3^2 + 14.9334x_3 - 43.0934) \\
&- 1.508x_1(x_5^2 + x_6^2) + 7.4777(x_3^3 + x_7^3) \\
&+ 0.7854(x_4^2 + x_5^2)
\end{align*}
\]

Subject to:

\[
\begin{align*}
g_1(\mathbf{x}) &= \frac{27}{x_1x_2x_3} - 1 \leq 0 \\
g_2(\mathbf{x}) &= \frac{397.5}{x_1x_2x_3} - 1 \leq 0 \\
g_3(\mathbf{x}) &= \frac{1.93x_4^3}{x_2x_3x_6} - 1 \leq 0
\end{align*}
\]
\[ g_4(\vec{x}) = \frac{1.93x^3}{x_2x_3x_4} - 1 \leq 0 \]

(9.7d)

\[ g_5(\vec{x}) = \frac{1}{110x^2} \sqrt{\left(\frac{745.0x_4}{x_2x_3}\right)^2 + 16.9 \times 10^6} - 1 \leq 0 \]

(9.7e)

\[ g_6(\vec{x}) = \frac{1}{85x^2} \sqrt{\left(\frac{745.0x_5}{x_2x_3}\right)^2 + 157.5 \times 10^6} - 1 \leq 0 \]

(9.7f)

\[ g_7(\vec{x}) = \frac{x_2x_3}{40} - 1 \leq 0 \]

(9.7g)

\[ g_8(\vec{x}) = \frac{5x_2}{x_1} - 1 \leq 0 \]

(9.7h)

\[ g_9(\vec{x}) = \frac{x_1}{12x_2} - 1 \leq 0 \]

(9.7i)

\[ g_{10}(\vec{x}) = \frac{1.5x_6 + 1.9}{12x_2} - 1 \leq 0 \]

(9.7j)

\[ g_{11}(\vec{x}) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0 \]

(9.7k)

where \( 2.6 \leq x_1 \leq 3.6, 0.7 \leq x_2 \leq 0.8, 17 \leq x_3 \leq 28, 7.3 \leq x_4 \leq 8.3, 7.8 \leq x_5 \leq 8.3, 2.9 \leq x_6 \leq 3.9 \) and \( 5 \leq x_7 \leq 5.5 \).

6.8 Belleville spring design

This type of design problem is shown in Fig. 19 [113]. In this method, one of the design parameter is selected according to variable ratio. The focus of this method is to minimize weight within certain constraints. The designed variables are
internal diameter of the spring (DIMI), external diameter of the spring (DIME), spring height (SH) and spring thickness (ST). The results are compared with other methods as shown in Table 27. The assessment of CHHO with other methods reveals that present method is effective in solving spring design problem more precisely. The mathematical formulation for spring design are given below:

Minimizing

\[
\text{Subject to:}
\]

\[
\begin{align*}
\text{(9.8a)} \quad f(x) &= 0.07075 \pi (\text{DIM}_E^2 - \text{DIM}_I^2)t \\
\text{(9.8b)} \quad b_1(x) &= G - \frac{4P\lambda_{\text{max}}}{(1 - \delta^2)\alpha\text{DIM}_E} \left[ \delta(S_H - \frac{\lambda_{\text{max}}}{2}) + \mu t \right] \geq 0 \\
\text{(9.8c)} \quad b_2(x) &= \left( \frac{4P\lambda_{\text{max}}}{(1 - \delta^2)\alpha\text{DIM}_E} \left[ (S_H - \frac{\lambda_{\text{max}}}{2})(S_H - \lambda_{\text{max}}r + r^2) \right] \right)_{\lambda_{\text{max}}} - P_{\text{MAX}} \geq 0 \\
\text{(9.8d)} \quad b_3(x) &= \lambda_1 - \lambda_{\text{max}} \geq 0 \\
\text{(9.8e)} \quad b_4(x) &= H - S_H - t \geq 0
\end{align*}
\]

Fig. 11 Trial run of FD benchmark functions
where \( \mu = \frac{6}{\pi^2} \ln \left( \frac{J - 1}{2} \right) \)  

\[ b_5(x) = \text{DIM}_\text{MAX} - \text{DIM}_E \geq 0 \]  

\[ b_6(x) = \text{DIM}_E - \text{DIM}_I \geq 0 \]  

\[ b_7(x) = 0.3 - \frac{S_H}{\text{DIM}_E - \text{DIM}_I} \geq 0 \] 

where \( \alpha = \frac{6}{\pi \ln J} \left( \frac{J - 1}{J} \right)^2 \)  

\[ \delta = \frac{6}{\pi \ln J} \left( \frac{J - 1}{\ln J - 1} \right) \]

### Table 17: Abbreviations for 10 types of design problems

| Engineering function (EF) | Type of problem |
|---------------------------|-----------------|
| EF1                       | 3-bar truss problem |
| EF2                       | Pressure vessel |
| EF3                       | Compression design |
| EF4                       | Welded beam |
| EF5                       | Cantilever beam design |
| EF6                       | Gear train |
| EF7                       | Speed reducer problem |
| EF8                       | Belleville spring |
| EF9                       | Rolling element bearing |
| EF10                      | Multiple disk clutch brake (discrete variables) |

### Table 18: Engineering design problems by CSMA

| Engineering function (EF) | Mean  | STD value | Best value | Worst value | Median value | \( p \) value |
|---------------------------|-------|-----------|------------|-------------|--------------|---------------|
| EF1                       | 270.7824 | 1.791805  | 265.4599 | 273.3948 | 271.2534 | 1.7344E–06 |
| EF2                       | 2994.48  | 0.005837  | 2994.474 | 2994.495 | 2994.479 | 1.7344E–06 |
| EF3                       | 6427.41  | 531.9222  | 5885.341 | 7318.996 | 6195.263 | 1.7344E–06 |
| EF4                       | 0.014245 | 0.001415  | 0.012715 | 0.017524 | 0.013955 | 1.7344E–06 |
| EF5                       | 1.740409 | 0.052373  | 1.724899 | 2.00749  | 1.726646 | 1.7344E–06 |
| EF6                       | – 85.534.4 | 10.61208   | – 85.539.2 | – 85.498.2 | – 85.538.7 | 1.7344E–06 |
| EF7                       | 0.392818 | 0.005457  | 0.389654 | 0.404654 | 0.39664 | 1.7344E–06 |
| EF8                       | 3.34E–11 | 6.46E–11  | 4.82E–14 | 2.91E–10 | 7.56E–12 | 1.7344E–06 |
| EF9                       | 5.24E+22 | 5.87E+22  | 1.57E+21 | 1.79E+23 | 1.21E+22 | 1.7344E–06 |
| EF10                      | 1.303713 | 0.000368  | 1.303281 | 1.30519  | 1.303612 | 1.7344E–06 |

### Table 19: Computation time for engineering function (EF) using CSMA

| Engineering function (EF) | Best time | Mean time | Worst time |
|---------------------------|-----------|-----------|-----------|
| EF1                       | 0.5       | 0.58496   | 1.390625  |
| EF2                       | 0.984375  | 1.063021  | 1.453125  |
| EF3                       | 0.671875  | 0.814063  | 1.359375  |
| EF4                       | 0.578125  | 0.663542  | 1.3125    |
| EF5                       | 0.671875  | 0.829167  | 1.515625  |
| EF6                       | 1.21875   | 1.301042  | 1.796875  |
| EF7                       | 0.734375  | 0.829688  | 1.28125   |
| EF8                       | 0.6875    | 0.784375  | 1.5       |
| EF9                       | 0.671875  | 0.782813  | 1.359375  |
| EF10                      | 0.75      | 0.833333  | 1.359375  |

Rolling element bearing design

The major aspect of this kind of design is to improve the dynamic load carrying capacity of rolling bearing element as illustrated in Fig. 20 [89]. There are ten parameters which
decides the optimum design of bearing for improving the load bearing power. Out of these ten variables only five variables are of much consideration. These major variables are (1) diameter of the ball (DIMB), (2) diameter pitch (DIMP), (3) ball numbers \( N_b \), (4) outer curvature coefficient and (5) inner curvature coefficient. Rest of five variables only affect indirectly to the internal portion of the geometry. The proposed algorithm is applied to solve rolling design problem and outputs are compared with other methods as illustrated in Table 28. From the result assessment, it can be seen that
CSMA gives analogous outcomes. The design equations are formulated through the following equations:

Maximizing

\[ C_D = f_c N^{2/3} \frac{\text{DIM}_b^{1.8}}{} \]  

\[ (9.9a) \]

if \( \text{DIM} \leq 25.4 \text{ mm} \)

\[ C_D = 3.647 f_c N^{2/3} \frac{\text{DIM}_b^{4.4}}{} \]  

\[ (9.9b) \]

if \( \text{DIM} \geq 25.4 \text{ mm} \)

Subjected to:

\[ r_1(x) = \frac{\theta_0}{2 \sin^{-1} \left( \frac{\text{DIM}_b}{\text{DIM}_{\text{max}}} \right)} - N + 1 \geq 0 \]  

\[ (9.9c) \]

\[ r_2(x) = 2\text{DIM}_b - K_{\text{DIM}_{\text{min}}} \left( \text{DIM} - \text{dim} \right) \geq 0 \]  

\[ (9.9d) \]

\[ r_3(x) = K_{\text{DIM}_{\text{max}}} \left( \text{DIM} - \text{dim} \right) \geq 0 \]  

\[ (9.9e) \]

\[ r_4(x) = \beta B_w - \text{DIM}_b \leq 0 \]  

\[ (9.9f) \]

\[ r_4(x) = \text{DIM}_{\text{MAX}} - 0.5(\text{DIM} + \text{dim}) \geq 0 \]  

\[ (9.9g) \]

\[ r_5(x) = \text{DIM}_{\text{MAX}} - 0.5(\text{DIM} + \text{dim}) \geq 0 \]  

\[ (9.9h) \]

\[ r_6(x) = (0.5 + \text{re})(\text{DIM} + \text{dim}) \geq 0 \]  

\[ (9.9i) \]

\[ r_7(x) = 0.5(\text{DIM} - \text{DIM}_{\text{MAX}} - \text{DIM}_b) - a \text{DIM}_b \geq 0 \]  

\[ (9.9j) \]

\[ r_8(x) = f_i \geq 0.515 \]  

\[ (9.9k) \]

\[ r_9(x) = f_0 \geq 0.515 \]  

\[ (9.9l) \]

where

\[ f_c = \text{37.91} \left[ 1 + \left( \frac{1 - \epsilon}{1 + \epsilon} \right)^{1.72} \left( \frac{f_i (2f_0 - 1)}{f_0 (2f_1 - 1)} \right)^{0.41} \right]^{10/3} - \epsilon^{0.3} (1 + \epsilon)^{1.39} \left[ \frac{2f_i}{2f_1 - 1} \right]^{0.41} \]
\[
\theta_0 = 2\pi - 2\cos^{-1}\left(\frac{((\text{DIM} - \text{dim})/2 - 3(t/4))^2 + (\text{DIM}/2 - t/4 - \text{DIM}_B)^2 - (\text{dim}/2 + t/4)^2}{2((\text{DIM} - \text{dim})/2 - 3(t/4))(\text{DIM}/2 - t/4 - \text{DIM}_B)}\right)
\]

**Fig. 14** Compression spring engineering design

**Table 22** Comparison of CSMA with other methods

| Method       | CSMA | GWO [15] | GSA [94] | CPSO [126] | ES [127] | GA [128] | HS [18] | DE [1] | MO [129] | CC [130] |
|--------------|------|----------|----------|------------|----------|----------|---------|--------|----------|----------|
| Optimized value for variables |      |          |          |            |          |          |         |        |          |          |
| \(d\)'      | 0.05 | 0.0516   | 0.0503   | 0.0517     | 0.052    | 0.0515   | 0.0512  | 0.0516 | 0.0534   | 0.05     |
| \(D\)'      | 0.3174| 0.3567   | 0.3237   | 0.3576     | 0.364    | 0.3517   | 0.3499  | 0.3547 | 0.3992   | 0.3159   |
| \(N\)'      | 14.0278| 11.2889  | 13.5254  | 11.2445    | 10.8905  | 11.6322  | 12.0764 | 11.4108| 9.1854   | 14.25    |
| Optimum weight | 0.012715| 0.01267  | 0.0127   | 0.01267    | 0.01268  | 0.0127  | 0.01267 | 0.01273| 0.01283  |          |

Bold values indicate the results of the Chaotic variant of the Slime Mould Algorithm

**Fig. 15** Welded beam engineering design

**Table 23** Comparative analysis of welded beam design with other methods

| Method       | CSMA | GSA [105] | GA1 [128] | GA2 [131] | HS [132] | Random [133] | Simplex [133] | David [133] | APPROX [133] |
|--------------|------|-----------|-----------|-----------|---------|---------------|---------------|-------------|-------------|
| Optimum variables |      |           |           |           |         |               |               |             |             |
| \(h\)       | 0.2057| 0.1821    | 0.24890   | 0.2088    | 0.2442  | 0.4575        | 0.2792        | 0.2434      | 0.2444      |
| \(l\)       | 3.4710| 3.857     | 6.17300   | 3.4205    | 6.2231  | 4.7313        | 5.6256        | 6.2552      | 6.2189      |
| \(t\)       | 9.0366| 10        | 8.1789    | 8.9975    | 8.2915  | 5.0853        | 7.7512        | 8.2915      | 8.2915      |
| \(h\)       | 0.2057| 0.2024    | 0.2533    | 0.2100    | 0.2443  | 0.66          | 0.2796        | 0.2444      | 0.2444      |
| Optimal cost | 1.7248| 1.88      | 2.4334    | 1.7583    | 2.3807  | 4.1185        | 2.5307        | 2.3841      | 2.3815      |

Bold values indicate the results of the Chaotic variant of the Slime Mould Algorithm
6.10 Multidisc-clutch design

Brake design is one of the most crucial problems in engineering and is shown in Fig. 21 [143]. The clutch-design problem is mainly fabricated to minimize the overall weight. The five design variables are inner surface radius \(R_i\), outer surface radius \(R_o\), thickness of disc’s (Th), actuating force (Fac) and count of friction surface \(S_f\). In Table 29, results are compared and observed that optimum fitness is found to be better than other methods. The equations for Multidisc brake problem are given below:

Minimizing

\[
\begin{align*}
\min f(R_i, R_o, S_f, Th) &= \pi Th \left( R_o^2 - R_i^2 \right)(S_f + 1) \\
&= \pi Th \left( R_o^2 - R_i^2 \right)
\end{align*}
\]  

(9.10a)

Table 24 Comparative analysis of beam problem with other methods

| Method     | CSMA | ALO [134] | SOS [28] | CS [135] | MMA [136] | GCA_I [136] | GCA_II [136] |
|------------|------|-----------|----------|----------|-----------|-------------|--------------|
| Optimal values for variables |      |           |          |          |           |             |              |
| \(l_1\)   | 5.9700 | 6.0181 | 6.0188 | 6.0089 | 6.01 | 6.01 | 6.01 |  |
| \(l_2\)   | 4.8841 | 5.3114 | 5.3034 | 5.3049 | 5.3 | 5.304 | 5.3 |  |
| \(l_3\)   | 4.4544 | 4.4884 | 4.4959 | 4.5023 | 4.49 | 4.49 | 4.49 |  |
| \(l_4\)   | 3.4738 | 3.4975 | 3.499 | 3.5077 | 3.49 | 3.498 | 3.49 |  |
| \(l_5\)   | 2.1571 | 2.1583 | 2.1556 | 2.1504 | 2.15 | 2.15 | 2.15 |  |
| Optimum weight | 1.30328 | 1.3399 | 1.3396 | 1.33999 | 1.34 | 1.34 | 1.34 |  |

Bold values indicate the results of the Chaotic variant of the Slime Mould Algorithm

Table 25 Comparison of gear train problem with other methods

| Method | CSMA | Gene AS [121] | Kannan and Kramer [121] | Sandgren [121] |
|--------|------|---------------|-------------------------|----------------|
| Optimal values for variables |      |               |                         |                |
| \(x_1\) | 41   | 50            | 41                      | 60             |
| \(x_2\) | 33   | 33            | 33                      | 45             |
| \(x_3\) | 15   | 14            | 15                      | 22             |
| \(x_4\) | 13   | 17            | 13                      | 18             |
| Optimum fitness | 0.144124 | 0.144124 | 0.144124 | 0.146667 |
where  \( R_i \in 60, 61, \ldots, 80; \ R_o \in 90, 91, \ldots, 110; \ Th \in 1, 1.5, 2, 2.5, 3; \ F_{ax} \in 600, 610, 620, 1000; \ S_i \in 2, 3, 4, 5, 6, 7, 8, 9. \)

Subject to:

\[
m_1 = R_o - R_i - \Delta R \geq 0
\]  

(9.10b)
\[
m_2 = L_{\text{MAX}} - (S_f + 1)(Th + \alpha) \geq 0
\]
(9.10c)
\[
m_3 = PM_{\text{MAX}} - PM_{\pi} \geq 0
\]
(9.10d)
\[
m_4 = PM_{\text{MAX}}Y_{\text{MAX}} + PM_{\pi}Y_{\text{SR}} \geq 0
\]
(9.10e)
\[
m_5 = Y_{\text{SRMAX}} - Y_{\text{SR}} \geq 0
\]
(9.10f)

\[
m_6 = t_{\text{MAX}} - t \geq 0
\]
(9.10g)
\[
m_7 = DC_{h} - DC_{t} \geq 0
\]
(9.10h)
\[
m_8 = t \geq 0
\]
(9.10i)

To check the effectiveness of chaotic SMA, algorithm is tested for 30 trial runs. The algorithm is tested with respect for best value, worst value, \( p \) value and standard deviation along with the trial runs. Furthermore, a comparative analysis with recent optimization methods is provided for justifying the validity of tested results for each of the standard function’s and design problems. Figure 22a and b shows 30

Table 27 Comparative analysis of Belleville spring design variables with other methods

| Method | CSMA   | TLBO [9] | MBA [113] |
|--------|--------|----------|-----------|
| Values for variables |        |          |           |
| \( x_1 \) | 8.83686 | 12.01    | 12.01     |
| \( x_2 \) | 4.81595 | 10.0304  | 10.0304   |
| \( x_3 \) | 0.2    | 0.20414  | 0.20414   |
| \( x_4 \) | 0.2    | 0.2      | 0.2       |
| Optimum fitness | 0.0572 | 0.19896  | 0.19896   |

Bold values indicate the results of the Chaotic variant of the Slime Mould Algorithm

Table 28 Comparative analysis of rolling element design variables

| Method | CSMA   | WCA [140] | SCA [141] | MFO [16] | MVO [142] |
|--------|--------|-----------|-----------|----------|-----------|
| Values for variables |        |           |           |          |           |
| \( r_1 \) | 125.7227 | 125.72    | 125       | 125      | 125.6002  |
| \( r_2 \) | 21.4233  | 21.4230   | 21.03287  | 21.03287 | 21.32250  |
| \( r_3 \) | 11.00116 | 10.01030  | 10.96571  | 10.96571 | 10.97338  |
| \( r_4 \) | 0.515    | 0.515000  | 0.515     | 0.515    | 0.515000  |
| \( r_5 \) | 0.515    | 0.515000  | 0.515     | 0.515000 | 0.515000  |
| \( r_6 \) | 0.4944   | 0.401514  | 0.5       | 0.5      | 0.5       |
| \( r_7 \) | 0.6986   | 0.659047  | 0.7       | 0.67584  | 0.68782   |
| \( r_8 \) | 0.3      | 0.300032  | 0.3       | 0.300214 | 0.301348  |
| \( r_9 \) | 0.03346  | 0.040045  | 0.027780  | 0.02397  | 0.03617   |
| \( r_{10} \) | 0.60049  | 0.600000  | 0.62912   | 0.61001  | 0.61061   |
| Optimum fitness | \(-85,534.166\) | 85,538.48 | 83,431.11 | 84,002.524 | 84,491.266 |

Bold values indicate the results of the Chaotic variant of the Slime Mould Algorithm
trial runs and for 10 multidisciplinary engineering problems to check the optimality of the algorithm.

7 Conclusion

In the proposed research, the exploitation phase of the classical SMA has been enhanced by incorporating sinusoidal chaotic function. The resultant chaotic SMA has been applied to 23 standard benchmark problems. In the set of experiments, CSMA was compared with basic SMA. The test results of benchmark functions are also compared with other algorithms in terms of mean and standard deviation. To check the soundness of the proposed algorithm, results of CSMA have been compared with others recently developed and well-known classical optimizers such as PSO, DE, SSA, MVO, GWO, DE, MFO, SCA, CS, TS, PSO-DE, GA, HS, Ray and Sain, MBA, ACO, and MMA. Experimental results suggest that chaotic strategy enables SMA to improve the exploitation phase with better convergence. Simulation results show that the developed chaotic SMA algorithm outperforms on almost all benchmark functions. Furthermore, the CSMA is applied to solve 10 real-world engineering design problems. Each design problem with a specific objective function has been simulated by implementing CSMA. Engineering design problems are mostly analyzed in terms of weight minimization and reduced manufacturing cost. The comparative analysis reveals that the proposed method effectively explores the search space to maximize the objective function and proofs that CSMA can demonstrate good results not only on unrestricted issues but also on restricted issues. It is seen that the resultant chaotic slime mold algorithm is capable of giving more optimistic and convergent results. Thus, the proposed CSMA may be a good choice for solving numerical optimization problems.

8 Limitation and future scope

The proposed optimizer is giving a powerful and optimal solution depending on the type of chaotic strategy opted, if chaotic strategies are not properly decided, it may lead to insignificant results also. Therefore, it is recommended that proper selection of chaotic strategy is required for significant results, so that the exploitation phase of the existing algorithm can be explored in the most significant way. In the future, the proposed chaotic variant may be significantly used to solve the various engineering and design optimization problems including power system optimization problems such as Economic Load Dispatch, Generation Scheduling problem, Unit commitment problem, and Automatic
generation control and load frequency issues of realistic power system and in more deeper sense, it may be applied to solve power system dispatch and unit commitment problems considering electric and hybrid electric vehicles including uncertainty of wind and solar power.

References

1. Storn R, Price K (1997) Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11:341–359

2. Koza JR, Poli R (2005) Genetic programming. Search methodologies. Springer, Boston, MA, pp 127–164

3. Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans Evol Comput 13:398–417. https://doi.org/10.1109/TEVC.2008.927706

4. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179:2232

5. Kaveh Ali (2014) Advances in metaheuristic algorithms for optimal design of structures. Springer International Publishing, Switzerland

6. Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization.

Fig. 22 Trial run test for engineering design problems
Neural Comput Appl 27:495–513. https://doi.org/10.1007/s00521-015-1870-7
7. Moghdani R, Salimifard K (2018) Volleyball premier league algorithm. Appl Soft Comput J 64:161–185. https://doi.org/10.1016/j.asoc.2017.11.043
8. Glover F (1989) Tabu search—part I. ORSA J Comput 1(3):190–206
9. Satapathy SC, Naik A, Parvati K (2013) A teaching learning based optimization based on orthogonal design for solving global optimization problems. Springerplus 2:1–12. https://doi.org/10.1186/2193-1801-2-130
10. Eberhart R, Kennedy J (1995) Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks, vol. 4
11. Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. IEEE Comput Intell Mag 1:28–39. https://doi.org/10.1109/MCI.2006.329691
12. Brajevic I, Tuba M (2013) An upgraded artificial bee colony (ABC) algorithm for constrained optimization problems. J Intell Manuf 24:729–740. https://doi.org/10.1007/s10845-011-0621-6
13. Verma C, Stoffova V, Illes Z, Tanwar S, Kumar N (2020) Machine learning-based student’s native place identification for real-time. IEEE Access 8:130840–130854. https://doi.org/10.1109/ACCESS.2020.3008830
14. Yang XS (2011) Bat algorithm for multi-objective optimisation. Int J Bioinspired Comput 3:267–274. https://doi.org/10.1504/IJBIC.2011.042259
15. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. J Exp Algorithms 69:46–61. https://doi.org/10.1016/j.advensoft.2014.12.007
16. Mirjalili S (2015) Moth–flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl Based Syst 89:228–249. https://doi.org/10.1016/j.knosys.2015.07.006
17. Verma C, Stoffova V, Illés Z (2019) Prediction of students’ awareness level towards ICT and mobile technology in Indian and Hungarian University for the real-time: preliminary results. Heliyon. https://doi.org/10.1016/j.heliyon.2019.e01806
18. Heidari AA, Mirjalili S, Faris H, Aljarahe I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications. Future Gener Comput Syst. https://doi.org/10.1016/j.future.2019.02.028
19. Verma C, Stoffova V, Illés Z (2018) An Ensemble approach to identifying the student gender towards information and communication technology awareness in European schools using machine learning. Int J Eng Technol 7:3392–3396. https://doi.org/10.14419/ijet.v7i4.14045
20. Fleszar K, Osman IH, Hindi KS (2009) A variable neighbourhood search algorithm for the open vehicle routing problem. Eur J Oper Res 195:803–809. https://doi.org/10.1016/j.ejor.2007.06.064
21. Simon D (2008) Biogeography-based optimization. IEEE Trans Evol Comput 12:702–713
22. Tan Y, Zhu Y (2010) Fireworks algorithm for optimization. In: International conference in swarm intelligence. Springer, Berlin, Heidelberg
23. Gandomi AH, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simul 17:4831–4845. https://doi.org/10.1016/j.cnsns.2012.05.010
24. Marti R, Resende MGC, Ribeiro CC (2013) Multi-start methods for combinatorial optimization. Eur J Oper Res 226:1–8. https://doi.org/10.1016/j.ejor.2012.10.012
25. Eskandar H, Sadollah A, Bahreininejad A, Hamdi M (2012) Water cycle algorithm—a novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput Struct 110–111:151–166. https://doi.org/10.1016/j.compstruc.2012.07.010
26. Li X, Zhang J, Yin M (2014) Animal migration optimization: An optimization algorithm inspired by animal migration behavior. Neural Comput Appl 24:1867–1877. https://doi.org/10.1007/s00521-013-1433-8
27. Kuo HC, Lin CH (2013) Cultural evolution algorithm for global optimizations and its applications. J Appl Res Technol 11:510–522. https://doi.org/10.1016/S1665-6423(13)71155-X
28. Cheng MY, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. Comput Struct 139:98–112. https://doi.org/10.1016/j.compstruc.2014.03.007
29. Gandomi AH (2014) Interior search algorithm (ISA): a novel approach for global optimization. ISA Trans 53:1168–1183. https://doi.org/10.1016/j.isatra.2014.03.018
30. Mirjalili S, Wang GG, Coelho LS (2014) Binary optimization using hybrid particle swarm optimization and gravitational search algorithm. Neural Comput Appl 25:1423–1435. https://doi.org/10.1007/s00521-014-1629-6
31. Mohseni S, et al. (2014) Competition over resources: a new optimization algorithm based on animals behavioral ecology. In: 2014 International Conference on Intelligent Networking and Collaborative Systems. IEEE
32. Wang GG, Guo L, Gandomi AH, Hao GS, Wang H (2014) Chaotic Krill Herd algorithm. Inf Sci (Ny) 274:17–34. https://doi.org/10.1016/j.ins.2014.02.012
33. Salimi H (2015) Stochastic fractal search: a powerful metaheuristic algorithm. Knowl Based Syst 75:1–18. https://doi.org/10.1016/j.knosys.2014.07.025
34. Ghorbani N, Babaei E (2014) Exchange market algorithm. Appl Soft Comput J 19:177–187. https://doi.org/10.1016/j.asoc.2014.02.006
35. Ghaemi M, Feizi-Derakhshi MR (2014) Forest optimization algorithm. Expert Syst Appl 41:6676–6687. https://doi.org/10.1016/j.eswa.2014.05.009
36. Gray B, Optimization W (2015) Author’s accepted manuscript binary gray wolf optimization approaches for feature selection. Neurocomputing. https://doi.org/10.1016/j.neucom.2015.06.083
37. Meng XB, Gao XZ, Lu L, Liu Y, Zhang H (2016) A new bio-inspired optimisation algorithm: Bird Swarm Algorithm. J Exp Theor Artif Intell 28:673–687. https://doi.org/10.1007/109528.15X.2015.042530
38. Wang GG, Suash D, Coelho LDS (2015) Elephant herding optimization. In: 2015 3rd International Symposium on Computational and Business Intelligence (ISCBI). IEEE
39. Abedinpourshotorban H, Mariyam Shamsuddin S, Beheshti Z, Jawawi DNA (2016) Electromagnetic field optimization: a physics-inspired metaheuristic optimization algorithm. Swarm Evol Comput 26:8–22. https://doi.org/10.1016/j.swevo.2015.07.002
40. Shahriar MS, Rana J, Asif MA, Hasan M, Hawlader M (2015) Optimization of Unit Commitment Problem for wind-thermal generation using Fuzzy optimization technique. In: 2015 International conference on advances in electrical engineering (ICAEE), pp. 88–92. IEEE
41. Shareef H, Ibrahim AA, Mutlag AH (2015) Lightning search algorithm. Appl Soft Comput J 36:315–333. https://doi.org/10.1016/j.asoc.2015.03.028
42. Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. Adv Eng Softw 105:30–47. https://doi.org/10.1016/j.advengsoft.2017.01.004
43. Singh N, Singh SB (2017) A novel hybrid GWO-SCA approach for optimization problems. Eng Sci Technol Int J 20:1586–1601. https://doi.org/10.1016/j.jestch.2017.11.001
44. Gohil NB, Dwivedi VV (2017) A review on lion optimization. Nat Inspired Evol Algorithm 7:340–352
45. Reddy SK, Panwar L, Panigrahi BK, Kumar R (2018) Binary whale optimization algorithm: a new metaheuristic approach for
profit-based unit commitment problems in competitive electricity markets. Eng Optim. https://doi.org/10.1080/0305215X.2018.1463527

46. Pierzan J, Coelho LDS (2018) Coyote optimization algorithm: a new metaheuristic for global optimization problems. In: 2018 IEEE congress on evolutionary computation (CEC). IEEE

47. Chen X, Tianfeld H, Li K (2019) BASE DATA. Swarm Evol Comput. https://doi.org/10.1016/j.swevo.2019.01.003

48. Shadravan S, Naji HR, Bardsiri VK (2019) The Salishsh optimization: a novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. Eng Appl Artif Intell 80:20–34. https://doi.org/10.1016/j.engappai.2019.01.001

49. Verma C, Illes Z, Stoffova V (2019) Age group predictive models for the real time prediction of the university students using machine learning: Preliminary results. In: 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT). IEEE

50. Adamatzky A (2012) Slime mold solves maze in one pass, assisted by gradient of chemo-attractants. IEEE Trans Nanobiosci 11:131–134. https://doi.org/10.1109/TNB.2011.2181978

51. Nakagaki T, Kobayashi R, Nishiura Y, Ueda T (2004) Obtaining multiple separate food sources: behavioural intelligence in the Physarum plasmodium. Proc R Soc B Biol Sci 271:2305–2310. https://doi.org/10.1098/rspb.2004.2856

52. Adamatzky A, Jones J (2010) Road planning with slime mould: if Physarum built motorways it would route M6/M74 through Newcastle. Int J Bifurc Chaos 20:3065–3084. https://doi.org/10.1142/S0218127410027568

53. Beekman M, Latty T (2015) Brainless but multi-headed: decision making by the acellular slime mould Physarum polycephalum. J Mol Biol 427:3734–3743. https://doi.org/10.1016/j.jmb.2015.07.007

54. Burgin M, Adamatzky A (2017) Structural machines and slime mould computation. Int J Gen Syst 46:201–224. https://doi.org/10.1080/03054140.2017.1300585

55. Cuevas E, González M, Zaldívar D, Pérez-Cisneros M, García G (2012) An algorithm for global optimization inspired by collective animal behavior. Discret Dyn Nat Soc. https://doi.org/10.1155/2012/638275

56. Houbraken M, Demeyer S, Staessens D, Audenaert P, Colle D, Pickavet M (2013) Fault tolerant network design inspired by Physarum built motorways it would route M6/M74 through Newcastle. Int J Bifurc Chaos 20:3065–3084. https://doi.org/10.1142/S0218127410027568

57. Kropat E, Meyer-Nieberg S (2014) Slime mold inspired evolving networks under uncertainty (SLIMO). In: 2014 47th Hawaii International Conference on System Sciences (HICSS), pp. 1153–1161. IEEE Computer Society

58. Abdel-basset M, Chang V, Mohamed R (2020) HSMA_WOA: A hybrid novel Slime mould algorithm with whale optimization algorithm fortackling the image segmentation problem of chest X-ray images. Applied Soft Computing 95:1–14. https://doi.org/10.1016/j.asoc.2019.05.001

59. Zhao J, Gao ZM, Sun W (2020) The improved slime mould algorithm with Levy flight. J Phys Conf Ser. https://doi.org/10.1088/1742-6596/1617/1/012033

60. Patino-Ramirez F, Boussard A, Arson C, Dussutour A (2019) Substrate composition directs slime moulds behavior. Sci Rep 9:1–14. https://doi.org/10.1038/s41598-019-50872-z

61. Kouaddr R, Slimani L, Bouktir T (2020) Slime mould algorithm for practical optimal power flow solutions incorporating stochastic wind power and static var compensator device. Electr Eng Electromech. https://doi.org/10.20998/2074-272x.2020.6.07

62. Gao ZM, Zhao J, Yang Y, Tian XJ (2020) The hybrid grey wolf optimization-slime mould algorithm. J Phys Conf Ser. https://doi.org/10.1088/1742-6596/1617/1/012034

63. Nguyen TT, Wang HJ, Hao TK, Pan JS, Liu JH, Weng S (2020) An improved slime mold algorithm and its application for optimal operation of cascade hydropower stations. IEEE Access 8:226754–226772. https://doi.org/10.1109/ACCESS.2020.3045975

64. Izcí D, Ekinç S (2021) Comparative performance analysis of slime mould algorithm for efficient design of proportional–integral–derivative controller. Electraca 21:151–159. https://doi.org/10.5152/ELECTRICA.2021.20077

65. Li S, Chen H, Wang M, Heidari AA, Mirjalili S (2020) Slime mould algorithm: a new method for stochastic optimization. Futur Gener Comput Syst. https://doi.org/10.1016/j.future.2020.03.055

66. Ji Y, Tu J, Zhou H, Gui W, Liang G, Chen H, Wang M (2020) An adaptive chaotic sine cosine algorithm for constrained and unconstrained optimization. Complexity. https://doi.org/10.1155/2020/6084917

67. Paul C, Roy PK, Mukherjee V (2020) Chaotic whale optimization algorithm for optimal solution of combined heat and power economic dispatch problem incorporating wind. Renew Energy Focus. https://doi.org/10.1016/j.ref.2020.06.008

68. Chuang LY, Hsiao CJ, Yang CH (2011) Chaotic particle swarm optimization for data clustering. Expert Syst Appl 38:14555–14563. https://doi.org/10.1016/j.eswa.2011.05.027

69. Mane SU, Narsingrao MR (2021) A chaotic-based improved many-objective jaya algorithm for many-objective optimization problems. Int J Ind Eng Comput 12:49–62. https://doi.org/10.5267/j.ijiec.2020.10.001

70. Dong N, Fang X, Wu AG (2016) A novel chaotic particle swarm optimization algorithm for parking space guidance. Math Probl Eng. https://doi.org/10.1155/2016/5126808

71. Kohli M, Arora S (2018) Chaotic grey wolf optimization algorithm for constrained optimization problems. J Comput Des Eng 5:458–472. https://doi.org/10.1007/s41190-017-0200-5

72. Chen Z, Liu W (2020) An efficient parameter adaptive support vector regression using K-Means clustering and chaotic slime mould algorithm. IEEE Access 8:156851–156862. https://doi.org/10.1109/ACCESS.2020.3018866

73. Premkumar M, Jangir P, Sowmya R, Alhelou HH, Heidari AA, Chen H (2021) MOSMA: multi-objective slime mould algorithm based on elitist non-dominated sorting. IEEE Access 9:3229–3248. https://doi.org/10.1109/ACCESS.2020.3047936

74. Zhai J, Gao ZM (2020) The chaotic slime mould algorithm with Chebyshev. Map J Phys Conf Ser. https://doi.org/10.1088/1742-6596/1613/1/012071

75. Majhi SK, Mishra A, Pradhan R (2019) A chaotic salp swarm algorithm based on quadratic integrate and fire neural model for function optimization. Prog Artif Intell 8:343–358. https://doi.org/10.1007/s13748-019-00184-0

76. Li Y, Han M, Guo Q (2020) Modified whale optimization algorithm based on tent chaotic mapping and its application in structural optimization. KSCE J Civ Eng 24:3703–3713. https://doi.org/10.1007/s12205-020-0504-5

77. Zhu T, Zheng H, Ma Z (2019) A chaotic particle swarm optimization algorithm for solving optimal power system problem of electric vehicle. Adv Mech Eng 11:1–9. https://doi.org/10.1177/1687814019833500

78. Rezaie H, Kazemi-Rahbar MH, Vahidi B, Rastegar H (2019) Solution of combined economic and emission dispatch problem using a novel chaotic improved harmony search algorithm. J Comput Des Eng 6:447–467. https://doi.org/10.1016/j.jcdes.2018.08.001

79. Hichem H, Elkamel M, Rafik M, Messeoud MT, Ouahiba C (2019) A new binary grasshopper optimization algorithm for feature selection problem. J King Saud Univ Comput Inf Sci. https://doi.org/10.1016/j.jksuci.2019.11.007
80. Sayed GI, Tharwat A, Hassanien AE (2019) Chaotic dragonfly algorithm: an improved metaheuristic algorithm for feature selection. Appl Intell 49:188–205. https://doi.org/10.1007/s10489-018-1261-8
81. Qiao W, Yang Z (2019) Modified dolphin swarm algorithm based on chaotic maps for solving high-dimensional function optimization problems. IEEE Access 7:110472–110486. https://doi.org/10.1109/ACCESS.2019.2931910
82. Fuertes G, Vargas M, Alfaro M, Soto-Garrido R, Sabattin J, Peralta MA (2019) Chaotic genetic algorithm and the effects of entropy in performance optimization. Chaos. https://doi.org/10.1063/1.5048299
83. Kaur G, Arora S (2018) Chaotic whale optimization algorithm. J Comput Des Eng 5:275–284. https://doi.org/10.1016/j.jcde.2017.12.006
84. Saxena A, Shekhawat S, Kumar R (2018) Application and development of enhanced chaotic grasshopper optimization algorithms. Model Simul Eng. https://doi.org/10.1155/2018/4945157
85. Nie X, Wang W, Nie H (2017) Chaos quantum-behaved bat swarm optimization algorithm and its application in the PV MPPT. Comput Intell Neurosci. https://doi.org/10.1155/2017/1583847
86. Ye F, Lou XY, Sun LF (2017) An improved chaotic fruit fly optimization based on a mutation strategy for simultaneous feature selection and parameter optimization for SVM and its applications. PLoS ONE. https://doi.org/10.1371/journal.pone.0173516
87. Xu X, Rong H, M. Tropfati M, Liptrott M, Bessis N (2018) CS-PSO: chaotic particle swarm optimization algorithm for solving combinatorial optimization problems. Soft Comput 22:783–795. https://doi.org/10.1007/s00500-016-2383-8
88. Ge F, Hong L, Wu Q, Shi L (2015) A cooperative optimization algorithm inspired by chaos-order transition. Math Probl Eng. https://doi.org/10.1155/2015/984047
89. Zhang Y, Ji G, Dong Z, Wang S, Phillips P (2015) Comment on “an investigation into the performance of particle swarm optimization with various chaotic Maps.” Math Probl Eng 2015:11–14. https://doi.org/10.1155/2015/815370
90. Ghasemi M, Ghaavide S, Akbari E, Vahed AA (2014) Solving non-linear, non-smooth and non-convex optimal power flow problems using chaotic invasive weed optimization algorithms based on chaos. Energy 73:340–353. https://doi.org/10.1016/j. energ y.2014.06.026
91. Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. IEEE Trans Evol Comput 3:82
92. Digalakis JG, Margaritis KG (2007) On benchmarking functions for genetic algorithms. Int J Comput Math 77:481–506. https://doi.org/10.1080/00207160500805080
93. Wang J, Wang D (2008) Particle swarm optimization with a leader and followers. Prog Nat Sci 18:1437–1443. https://doi.org/10.1016/j.pnsc.2008.03.029
94. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci (Ny) 179:2232–2248. https://doi.org/10.1016/j.ins.2009.03.004
95. Xie J, Zhou YQ, Chen H (2013) A bat algorithm based on Lévy flights trajectory. Moshi Shibie Yu Rengong Zhineng Pattern Recognit Artif Intell 26:829–837
96. Yang XS (2010) Firefly algorithm. Eng Optim 221
97. Kazarlis SA (1996) A genetic algorithm solution to the unit commitment problem. IEEE Trans Power Syst 11:83–92
98. Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput Appl 27:1053–1073. https://doi.org/10.1007/s00521-015-1920-1
99. Nezamabadi-pour H, Rostami-sharbabaki M, Maghfoori-Farsangi M (2008) Binary particle swarm optimization: challenges and new solutions. CSI J Comput Sci Eng 6:21–32
100. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2010) BGSA: binary gravitational search algorithm. Nat Comput 9:727–745. https://doi.org/10.1007/s11047-009-9175-3
101. Cuevas E, Echavarria A, Ramirez-Ortégon MA (2014) An optimization algorithm inspired by the States of Matter that improves the balance between exploration and exploitation. Appl Intell 40:256–272. https://doi.org/10.1007/s10489-013-0458-0
102. Ang X-S, Karamanoglu M, He X (2014) Flower pollination algorithm: a novel approach for multiobjective optimization. Eng Optim 46:12
103. Jagodziński D, Jarosław A (2017) A differential evolution strategy. In: 2017 IEEE Congress on Evolutionary Computation (CEC). IEEE
104. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98. https://doi.org/10.1016/j.advengsoft.2015.01.010
105. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008
106. Jafari S, Bozorg-Haddad O, Chu X (2018) Cuckoo optimization algorithm (COA). Stud Comput Intell 720:39–49. https://doi.org/10.1007/978-981-10-5221-7_5
107. Kennedy J, Eberhart RC (1997) A discrete binary version of the particle swarm algorithm. IEEE Int Conf Syst Man Cybern Cybern Simul 5:4104–4108. https://doi.org/10.1109/ICSMC.1997.637339
108. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98. https://doi.org/10.1016/j.advengsoft.2015.01.010
109. 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. https://doi.org/10.1016/j.advengsoft.2017.07.002
110. John H (1992) Holland, adaptation in natural and artificial systems. MIT Press, Cambridge
111. Chopard B, Tomassini M (2018) Particle swarm optimization. Nat Comput Ser. https://doi.org/10.1007/978-3-319-93073-2_6
112. Kamboj VK, Nandi A, Bhadoria A, Sehgal S (2020) An intensify and dispatch of thermal generating units considering impact of wind penetration using hGWO-RES algorithm. Appl Intell. https://doi.org/10.1007/s10489-018-1325-9
113. Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2013) Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. Appl Soft Comput J 89:106018. https://doi.org/10.1016/j.asoc.2019.106018
114. Bhadoria A, Kamboj VK (2018) Optimal generation scheduling and dispatch of thermal generating units considering impact of wind penetration using hGWO-RES algorithm. Appl Intell. https://doi.org/10.1007/s10489-018-1325-9
115. Heidari AA, Mirjalili S, Faris H, Aljahar I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications. Future Gener Comput Syst 97:849–872. https://doi.org/10.1016/j.future.2019.02.028
116. Gandomi AH, Yang XS, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 29:17–35. https://doi.org/10.1007/s10440-011-0241-y
117. Ray T, Saini P (2001) Engineering design optimization using a swarm with an intelligent information sharing among individuals. Eng Optim 33:735–748. https://doi.org/10.1080/03052150108940941
118. Tsai JFA (2005) Global optimization of nonlinear fractional programming problems in engineering design. Eng Optim 37:399–409. https://doi.org/10.1080/03052150500066737
119. Hameed IA, Bye RT, Olsen OL (2016) Grey wolf optimizer (GWO) for automated offshore crane design. In: 2016 IEEE symposium series on computational intelligence (SSCI). IEEE
120. Ariables V (2015) The butterfly particle swarm optimization (butterfly PSO/BF-PSO) technique and its variables. Int J Soft Comput Math Control (IJSCMC) 4:23–39
121. Cagnina LC, Esquivel SC, Nacional U, Luis DS, Luis S, Coello CAC (2008) Solving engineering optimization problems with the simple constrained particle swarm optimizer. Informatica 32:319–326
122. Deb K (1996) A combined genetic adaptive search (GeneAS) for engineering design. Comput Sci Inform 26:30–45
123. Wang L, Li LP (2010) An effective differential evolution with level comparison for constrained engineering design. Struct Multidisciplinary Optimization 41(6):947–963
124. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Soft 95:51–67
125. Kamboj VK et al (2020) An intensify Harris Hawks optimizer for numerical and engineering optimization problems. Appl Soft Comput 89:106018
126. He Q, Wang L (2007) An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng Appl Artif Intell 20:89–99. https://doi.org/10.1016/j.engappai.2006.03.003
127. Mezura-Montes E, Coello Coello CA (2005) A simple multi-membered evolution strategy to solve constrained optimization problems. IEEE Trans Evol Comput 9:1–17. https://doi.org/10.1109/TEVC.2004.836819
128. Deb K (1990) Optimal design of a class of welded structures via genetic algorithms. In: 31st Structures, Structural Dynamics and MaterialsConference, p. 1179.
129. Mahdavi M, Fesanghary M, Damangir E (2007) An improved harmony search algorithm for solving optimization problems. Appl Math Comput 188:1567–1579. https://doi.org/10.1016/j.amc.2006.11.033
130. Wu G, Pedrycz W, Suganthan PN, Mallipeddi R (2015) A variable reduction strategy for evolutionary algorithms handling equality constraints. Appl Soft Comput J 37:774–786. https://doi.org/10.1016/j.asoc.2015.09.007
131. Coello Coello CA (2000) Use of a self-adaptive penalty approach for engineering optimization problems. Comput Ind 41:113–127. https://doi.org/10.1016/S0166-3615(99)00046-9
132. Lee KS, Geem ZW (2004) A new structural optimization method based on the harmony search algorithm. Comput Struct 82:781–790. https://doi.org/10.1016/j.compstruc.2004.01.002
133. Ragsdell KM, Phillips DT (1976) Optimal design of a class of welded structures using geometric programming. J Manuf Sci Eng Trans ASME 98:1021–1025. https://doi.org/10.1115/1.3438995
134. Cuevas E, Echavarría A (2013) An optimization algorithm inspired by the States of Matter that improves the balance between exploration and exploitation. Appl Intell. https://doi.org/10.1007/s10489-013-0458-0
135. Shankar K, Eswaran P (2016) RGB-based secure share creation in visual cryptography using optimal elliptic curve cryptography technique. J Circuits Syst Comput 25:1650138. https://doi.org/10.1142/S0218126616501383
136. Chickermane H, Gea HC (2002) Structural optimization using a new local approximation method. Int J Numer Methods Eng 39:829–846. https://doi.org/10.1002/sic.1097-0207(19960315)39:5%3c829::aid-nme884%3e3.0. co;2-u
137. Zhao W, Zhang Z, Wang L (2020) Manta ray foraging optimization: an effective bio-inspired optimizer for engineering applications. Eng Appl Artif Intell 87:103300. https://doi.org/10.1016/j.engappai.2019.103300
138. Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11:341–359. https://doi.org/10.1023/A:1008202821328
139. Niu B, Li L (2008) A novel PSO-DE-based hybrid algorithm for global optimization (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). Lect Notes Comput Sci 5227:156–163. https://doi.org/10.1007/978-3-540-85984-0_20
140. Sadollah A, Eskandar H, Bahreininejad A, Kim JH (2015) Water cycle algorithm for solving multi-objective optimization problems. Soft Comput 19:2587–2603. https://doi.org/10.1007/s00500-014-1424-4
141. Hafez AI, Zawbaa HM, Emary E, Hassanien AE (2016) Sine cosine optimization algorithm for feature selection.In: 2016 international symposium on innovations in intelligent systems and applications (INISTA). IEEE
142. Sayed GI, Darwish A, Hassanien AE (2018) A new chaotic multi-verse optimization algorithm for solving engineering optimization problems. J Exp Theor Artif Intell 30:293–317. https://doi.org/10.1080/0952813X.2018.1430858
143. Abderazek H, Ferhat D, Ivana A (2016) Adaptive mixed differential evolution algorithm for bi-objective tooth profile spur gear optimization. Int J Adv Manuf Technol. https://doi.org/10.1007/s00170-016-9523-2

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