Neural Computing and Applications (2022) 34:4081–4110
https://doi.org/10.1007/s00521-021-06747-4

REVIEW

Meta-heuristic optimization algorithms for solving real-world mechanical engineering design problems: a comprehensive survey, applications, comparative analysis, and results

Laith Abualigah[1,2] · Mohamed Abd Elaziz[3,8,9,10] · Ahmad M. Khasawneh[1] · Mohammad Alshinwan[1] · Rehab Ali Ibrahim[3] · Mohammed A. A. Al-qaness[4] · Seyedali Mirjalili[5,6] · Putra Sumari[2] · Amir H. Gandomi[7]

Received: 12 May 2021 / Accepted: 11 November 2021 / Published online: 16 January 2022

Abstract
Real-world engineering design problems are widespread in various research disciplines in both industry and industry. Many optimization algorithms have been employed to address these kinds of problems. However, the algorithm's performance substantially reduces with the increase in the scale and difficulty of problems. Various versions of the optimization methods have been proposed to address the engineering design problems in the literature efficiently. In this paper, a comprehensive review of the meta-heuristic optimization methods that have been used to solve engineering design problems is proposed. We use six main keywords in collecting the data (meta-heuristic, optimization, algorithm, engineering, design, and problems). It is worth mentioning that there is no survey or comparative analysis paper on this topic available in the literature to the best of our knowledge. The state-of-the-art methods are presented in detail over several categories, including basic, modified, and hybrid methods. Moreover, we present the results of the state-of-the-art methods in this domain to figure out which version of optimization methods performs better in solving the problems studied. Finally, we provide remarkable future research directions for the potential methods. This work covers the main important topics in the engineering and artificial intelligence domain. It presents a large number of published works in the literature related to the meta-heuristic optimization methods in solving various engineering design problems. Future researches can depend on this review to explore the literature on meta-heuristic optimization methods and engineering design problems.

Keywords Meta-heuristic optimization algorithms · Differential evolution · Real-world engineering design problems · Optimization problems · Algorithm · Benchmark

1 Introduction

Optimization problems can be found in pretty much any field of study [1]. Some of the most popular areas are: medical image, engineering problem [2–4], signal processing [5], cloud computing [6–12], feature selection [13–15], deep learning [16–19], text mining [20–25], photovoltaic models [26–30], and urban development [31, 32], others [33–35].

There are several techniques developed to tackling the different optimization problems [36–39]. Each algorithm has a particular behavior and process, which suite some problems by solving them efficiently and not suitable for others [40–42]. However, most of them can be classified into two groups. The first group is the traditional methods such as gradient descent and Newton [43]. These methods, in general, are easy and simple, but they are considered time-consuming methods, and they provide only one solution at each iteration. The methods belong to the second group of techniques that aim to avoid the limitation of the traditional method, these methods named meta-heuristic (MH) [44, 45]. In general, these MH techniques are inspired by nature, physics laws, human behavior. These methods can be classified into swarm-based, evolutionary-based, human-based, and natural phenomena-based methods [46, 47].

The swarm-based methods emulate the behavior of animals, fish, birds and other swarms during phase to find their food [48, 49]. This kind of MH techniques including salp swarm algorithm (SSA) [50], squirrel search algorithm

Extended author information available on the last page of the article

(SSA) [51], krill herd algorithm (KHA) [52], gray wolf optimization (GWO) [53], and particle swarm optimization (PSO) [54, 55], whereas the natural phenomena-based methods simulate the phenomena from nature such as spiral, light, wind and rain. These methods include water cycle algorithm (WCA) [56], wind-driven optimization (WDO) [57], simulated annealing (SA) [58], gray wolf optimizer (GWO) [59], firefly algorithm (FA) [60], genetic algorithm (GA) [61], symbiotic organisms search (SOS) [62], water drops algorithm (WDA) [63], glowworm swarm optimization algorithm (GSOA) [64], bat algorithm (BA) [65], teaching–learning optimization algorithm (TLO) [66], gravitational search algorithm (GSA) [67], Aquila optimizer (AO) [68], chemical reaction optimization (CRO) [69] Electromagnetism algorithm [70], and others [71–73].

The human-based methods emulate the behavior of human as an optimization methods. These methods are including seeker optimization algorithm (SOA) [74], teaching learning-based optimization (TLBO) [75], volleyball premier league algorithm (VPL) [76], poor and rich optimization algorithm (PROA) [77], chimp optimization algorithm (COA) [78], human mental search (HMS) [79], league championship algorithm (LCA) [80], and cultural algorithm (CA) [81].

Evolutionary-based methods inspirit their mechanism from simulating the concepts of natural genetic which applied imitates the principle of natural genetic [82, 83]. These methods depend on using crossover, mutation, and natural selection as operators. Some examples of these methods are evolutionary programming (EP) [84], genetic algorithm (GA) [85, 86], genetic programming (GP) [87], differential evolution (DE) [88], interior search algorithm (ISA) [89], evolution strategies (ES) [90], and arithmetic optimization algorithm (AOA) [91].

The main contributions of this study can be summarized as:

1. Reviewed the applications of meta-heuristic optimization methods to tackling a different set of engineering problems.
2. Evaluated several variants of meta-heuristic optimization methods that applied to engineering problems.
3. Defined the problems and challenges in the domain of engineering problems for using meta-heuristic optimization methods.

The rest of this paper is organized as follows. Section 2 presents the definitions and procedures of the most common engineering design optimization problems. Section 3 presents the recent optimization algorithms that have been used to solve different engineering design optimization problems. In Sect. 4 discussions and more informations regarding the algorithms and problems are given. Finally, in Sect. 6, the conclusions and potential feature work directions are given.

2 Definitions of the most common engineering design problems

In this section, the most common engineering design problems are presented. The mathematical representation and definition are given to explain the problems clearer [92, 93]. These problems are given as follows.

- The problem with welded beam design
- The problem with tension/compression spring design
- The problem with pressure vessel design
- The problem with the 3-bar truss design
- The problem with speed reducer
- The problem with cantilever beam design problem
- The problem with multiple disc clutch brake problem

These are the most well-known engineering design problems in the real world. Usually, it needs an active method to find its optimal parameters to find the optimal design. Each problem has some parameters (variables) that need to be adjusted. Also, some constraints are given to provide the values of the variables within the given range. The following description shows the general details of the optimization problem.

Generally, bound-constrained and common constrained optimization problems are decided to investigate the utility of any optimization method. Every design vector is regularly demanded to give the bound-constrained for any engineering or optimization problem [94–96]:

$$LB_{(j)} \leq x_{(i,j)} \leq UB_{(j)}, \quad j = 1, 2, 3, \ldots, n \tag{1}$$

where $LB_{(j)}$ and $UB_{(j)}$ are the lower bound and upper bound of the given problem for the position $x_{(i,j)}$, and n is the length of given positions (vector size). Furthermore, the common constrained optimization problem can usually be given as follows.

$$\begin{aligned} & minf(X) \\ & X = \{x_{(110}, x_{(1j)}, \ldots, x_{(1n)}\} \\ & s.t. \quad g_i(X) \leq 0, j = 1, 2, 3, \ldots, m \\ & h_k(X) = 0, k = 1, 2, 3, \ldots, l \\ & LB_{(j)} \leq x_{(ij)} \leq UB_{(j)}, \quad j = 1, 2, 3, \ldots, n \end{aligned} \tag{2}$$

where m presents the number of constraints given in the problem, and l presents the number of equality/equilibrium constraints.

By implementing the cost function in the evaluation of the used optimization method, all the given constrained problems in Eq. (2) are outlined into the bound-constrained structure. The used cost function can be merged into the used objective function for each infeasible solution. The cost function is calculated to be a result of its homogeneity in situations. It only requires an assistant cost function and [97–100] is satisfactory for all different problems.

2.1 Problem 1: Welded beam design

The welded beam design is one of the common engineering optimization problems, and it is unconstrained in life. In this problem, four decision variables need to be optimized by finding its optimal values, namely, *h*, *l*, *t*, and *b*. These variables are positive integers that fall inside a fixed range. The drew of the welded beam design is presented in Fig. 1. This design problem intends to reduce the cost of the welded beam in real life. The mathematical representation of this problem is presented as follows.

Consider $\overrightarrow{x} = [x_1\ x_2\ x_3\ x_4] = [\ h\ l\ t\ p]$,

Minimize $f(\overrightarrow{x}) = 1.10471x_1^2\ x_1{+}0.04811x_3\ x_4(14.0{+}x_2)$

Subject to $g_1(\overrightarrow{x}) = \tau\ (\overrightarrow{x}) - \tau_{\max}\ \leq\ 0$,

$g_2(\overrightarrow{x}){=}\sigma\ (\overrightarrow{x}) - \sigma_{\max}\ \leq\ 0$,

$g_3(\overrightarrow{x}) = \delta\ (\overrightarrow{x})\ {-}\delta_{\max}\ \leq\ 0$,

$g_4(\overrightarrow{x}) = x_1\ -\ x_4\ \leq\ 0$,

$g_5(\overrightarrow{x}) = p$ - $p_c(\overrightarrow{x})\ \leq\ 0$,

$g_6(\overrightarrow{x}) = 0.125$ - $x_1\ \leq\ 0$,

$g_7(\overrightarrow{x}) = 1.10471x_1^2 + 0.04811x_3\ x_4(14.0 + x_2)\ {-}0.5\ \leq$ 0,

Variables range ($0.1\ \leq\ x_1, x_4\ \leq\ 2$), ($0.1\ \leq\ x_2, x_3\ \leq$ 10), where $\tau(\overrightarrow{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')}$, $\tau' = \frac{P}{\sqrt{2x_1x_2}}$,

$\tau'' = \frac{MR}{J}$, M = P(L $+\frac{x_2}{2}$), R = $\sqrt{\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2}$,

$P_c(\overrightarrow{x}) = \frac{\sqrt[4.013E]{\frac{x_3^2x_4^6}{0}}}{L^2}$, $(\ 1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}})$, $\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$,

$J = 2\ \left\{\sqrt{2x_1x_2}\left[\frac{x_x^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2\right]\right\}$, $\sigma\ (\overrightarrow{x}) = \frac{6PL}{x_4x_3^2}$, δ $(\overrightarrow{x}){=}\frac{6PL^3}{Ex_4x_3^2}$,

Note that, $P = 6000$ lb, $L = 14$, $\delta_{\max} = 0.25$, $E = 30 \times 1^6 psi$, $G = 12 \times 10^6 psi$, $\tau_{\max} = 13600\ psi$, and $\sigma_{\max}{=}\ 30000$ *psi*.

2.2 Problem 2: Tension/compression spring design problem

The tension/compression spring design is also one of the common engineering optimization problems, and it is unconstrained in life. In this problem, three decision variables need to be optimized by finding their minimum values, namely, *d*, *D*, and *N*. These variables are positive integers that take numbers in fixed ranges. The drew of the tension/compression spring design problem is presented in Fig. 2. This design problem intends to find the minimum cost of the tension/compression spring in real life. The mathematical representation of this problem is provided as follows.

Consider $\overrightarrow{x} = [x_1\ x_2\ x_3] = [d\ D\ N]$,

Minimize $f(\overrightarrow{x}) = (x_3{+}2)x_2\ x_1^2$,

Subject to $g_1(\overrightarrow{x}) = 1{-}\frac{x_3x_2^3}{71785x_1^4}\ \leq\ 0$,

$g_2 = (\overrightarrow{x}) = \frac{4x_2^2-x_1x_1}{12566(x_2x_1^3-x_1^4)}{+}\frac{1}{5108x_1^2}\ \leq\ 0$,

$g_3 = (\overrightarrow{x}) = 1{-}\frac{140.54x_1}{x_2^2x_3}\ \leq\ 0$,

$g_4 = (\overrightarrow{x}) = \frac{x_1x_2}{1.5}{-}1\ \leq\ 0$,

Variables range ($0.05\ \leq\ x_1\ \leq\ 2$), ($0.25\ \leq\ x_2\ \leq\ 1.30$), ($2.00\ \leq\ x_3\ \leq\ 15$),

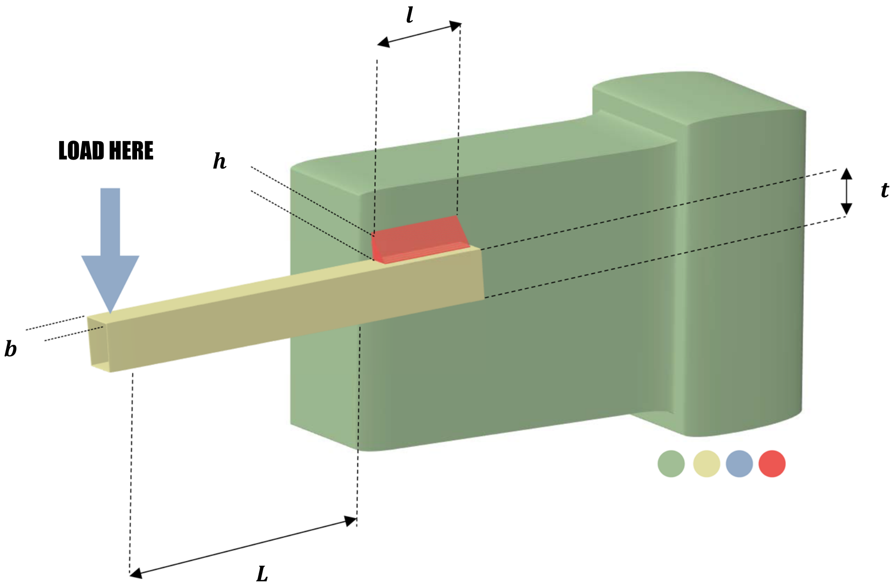


Fig. 1 Welded beam design problem

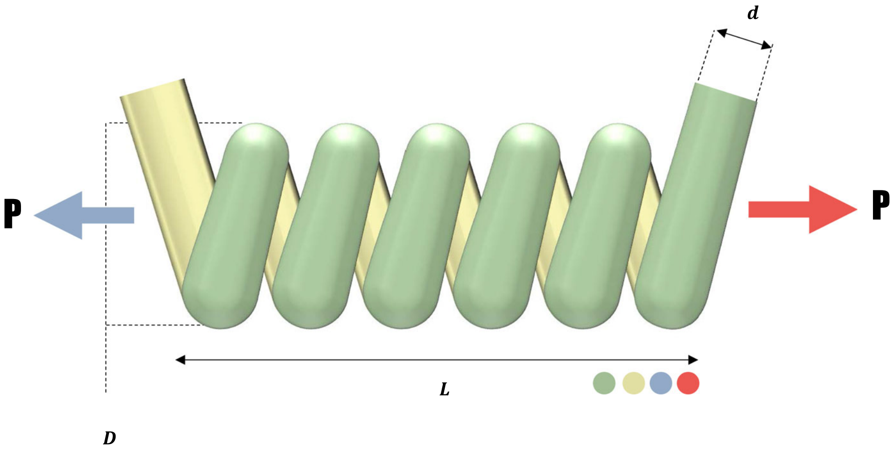


Fig. 2 Tension/compression spring design problem

2.3 Problem 3: Pressure vessel design problem

The pressure vessel design is a famous unconstrained engineering optimization problem in the real world. In this problem, four decision variables need to be optimized by finding their minimum values, namely, T_s, T_h, R, and L. The problem variables are positive integers that come inside fixed ranges. The drew of the pressure vessel design problem is presented in Fig. 3. This engineering problem intends to find the minimum cost of the pressure vessel. The mathematical presentation of this problem is provided as follows.

Consider $\overrightarrow{x} = [x_1\ x_2\ x_3\ x_4] = [\ T_s\ T_h\ R\ L]$,

Minimize $f(\overrightarrow{x}) = 0.6224x_1\ x_3\ x_4 + 1.7781x_2\ x_3^2 + 3.1661x_1^2\ x_4 + 19.84x_1^2\ x_3$,

Subject to $g_1(\overrightarrow{x}) = -x_1 + 0.0193x_3 \leq 0$,

$g_2(\overrightarrow{x}) = -\ x_3 + 0.000954x_3 \leq 0$,

$g_3(\overrightarrow{x}) = -\ \pi\ x_3^2\ x_4 - \frac{4}{3}\ \pi\ x_3^3 + 1296000 \leq 0$,

$g_4(\overrightarrow{x}) = x_4 - 240 \leq 0$,

Variables range ($0 \leq x_1, x_2 \leq 99$), ($10 \leq x_3, x_4 \leq 200$),

2.4 Problem 4: 3-bar truss design

The 3-bar truss design is also a famous unconstrained engineering optimization problem in the real world. In this problem, three main decision variables need to be optimized by finding their minimum values, namely, $A1$, $A2$, and $A3$, where $A1 = A3$. All the given variables are positive integers that come inside fixed ranges. The 3-bar truss design is given in Fig. 4. This design problem intends to find the minimum cost weight of the 3-bar truss. The mathematical representation of this problem is provided as follows.

Minimize $f(x) = (2\sqrt{2x_1} + x_2)\times\ l$

Subject to:

$g_1\mathrm{x} = \frac{\sqrt{x1}x_1+x_2}{\sqrt{2}x_1^2+2x_1x_2}\ P - \sigma \leq 0$

$g_2\mathrm{x} = \frac{x_2}{\sqrt{2}x_1^2+2x_1x_2}\ P\text{-}\sigma \leq 0$

$g_3\mathrm{x} = \frac{1}{\sqrt{2}x_2+x_1}\ P\text{-}\sigma \leq 0$

$l = 100\mathrm{cm}$, $P = 2\ \mathrm{kN}/cm^2$, $\sigma = 2\ \mathrm{kN}/cm^2$,

Variables range ($0 \leq x_i \leq 1$, $i = 1,2$)

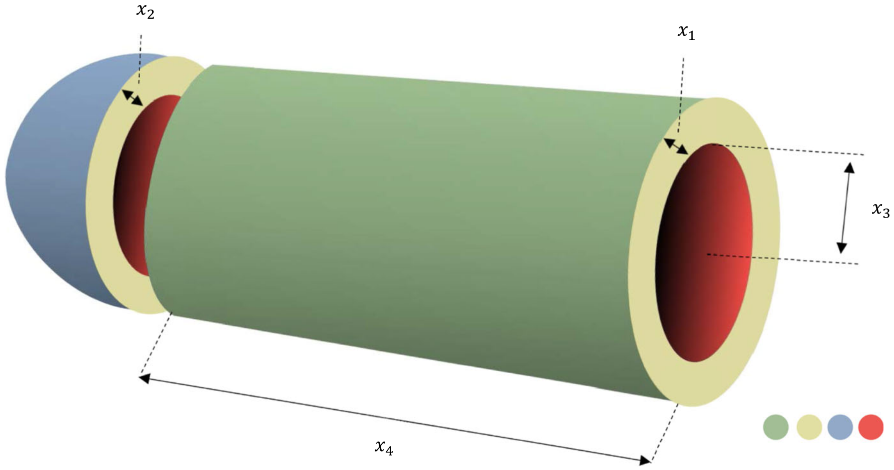


Fig. 3 Pressure vessel design problem

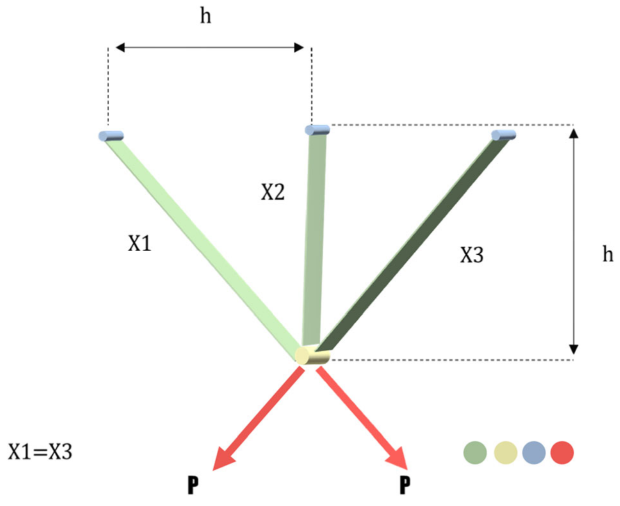


Fig. 4 3-bar truss design problem

2.5 Problem 5: Speed reducer

The speed reducer design is also a widespread unconstrained engineering optimization problem in the real world. In this problem, seven main decision variables need to be optimized by finding their minimum values, namely, m, z, l_2, l_1, d_1, d_2, b, and b. All the given variables are positive integers that come inside fixed ranges. The drew of the speed reducer design is given in Fig. 5. This design problem intends to find the minimum cost weight of the speed reducer. The mathematical representation of this problem is provided as follows.

Consider $\overrightarrow{x} = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.4777x_6^3 + x_7^3 + 0.7854x_4x_6^2 + x_5x_7^2$

Subject to:

$g(1) = \frac{27}{x_1x_2^2x_3} - 1 \leq 0,$

$g(2) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0,$

$g(3) = \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \leq 0,$

$g(4) = \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \leq 0,$

$g(5) = \frac{1}{110x_6^3}\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 16.9 \times 10^6} - 1 \leq 0,$

$g(6) = \frac{1}{85x_7^3}\sqrt{\left(\frac{745x_5}{x_2x_3}\right)^2 + 157.5 \times 10^6} - 1 \leq 0,$

$g(7) = \frac{x_2x_3}{40} - 1 \leq 0,$

$g(8) = \frac{5x_2}{x_1} - 1 \leq 0,$

$g(9) = (x_1/12x_2) - 1 \leq 0,$

$g(10) = \frac{1.5x_6+1.9}{x_4} - 1 \leq 0,$

$g(11) = \frac{1.1x_7+1.9}{x_5} - 1 \leq 0$

Variables range $(2.6 \leq x_1 \leq 3.6)$, $(0.7 \leq x_2 \leq 0.8)$, $(17 \leq x_3 \leq 28)$, $(7.3 \leq x_4 \leq 8.3)$, $(7.3 \leq x_5 \leq 8.3)$, $(2.9 \leq x_6 \leq 3.9)$, $(5 \leq x_7 \leq 5.5)$

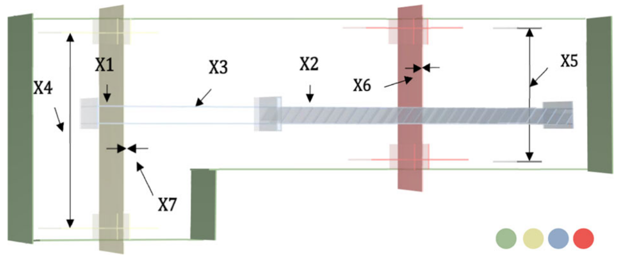


Fig. 5 Construction of a speed reducer

2.6 Problem 6: Cantilever beam design

The cantilever beam design is also a widespread unconstrained engineering optimization problem in the real world. In this problem, five main decision variables need to be optimized by finding their minimum values, namely, $x1$, $x2$, $x3$, $x4$, and $x5$. All the given variables are positive integers that come inside fixed ranges. The drew of the speed reducer design is given in Fig. 6. This design problem intends to find the minimum cost weight of the cantilever beam. The mathematical representation of this problem is provided as follows.

Consider $x = [x_1x_2x_3x_4x_5]$

Minimize $f(x) = 0.6224(x_1 + x_2 + x_3 + x_4 + x_5)$,

subject to:

$g(x) = \frac{60}{x_1^3} + \frac{27}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0$

Variables range

$0.01 \leq x_1, x_2, x_3, x_4, x_5 \leq 100.$

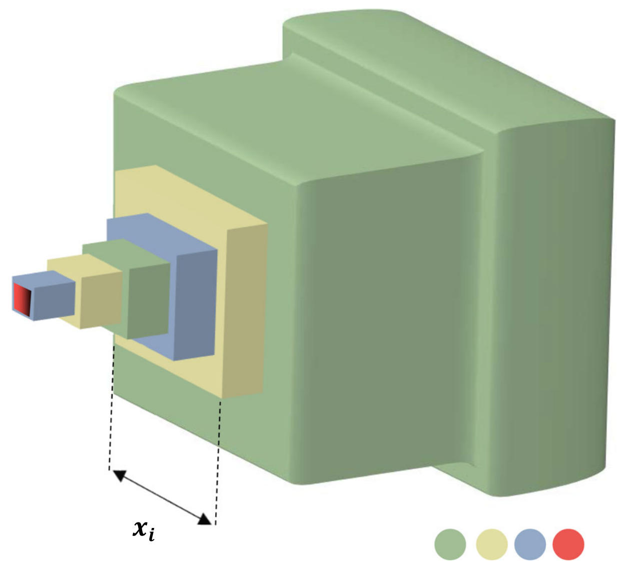


Fig. 6 Cantilever beam design problem

2.7 Problem 7: Multiple disc clutch brake

The multiple disc clutch brake design is also a widespread unconstrained engineering optimization problem in the real world. In this problem, five main decision variables need to be optimized by finding their minimum values, namely, F, r_i, r_0, t, and δ. All the given variables are positive integers that come inside fixed ranges. The drew multiple disc clutch brake design is given in Fig. 7. This design problem intends to find the minimum cost weight of the multiple disc clutch brake. The mathematical representation of this problem is provided as follows.

Consider x = $[x_1 x_2 x_3 x_4 x_5]$

minimize $f(x) = \Pi(r_o^2 - r_i^2)t(Z+1)\rho$

subject to

$$g_1(x) = r_o - r_i - \Delta r \geq 0$$
$$g_2(x) = l_{\max} - (Z+1)(t+\delta) \geq 0$$
$$g_3(x) = Pmax - Prz \geqslant 0$$
$$g_4(x) = P_{\max}\ v_{sr\ max} - P_{rz}\ v_{sr} \geq 0$$
$$g_5(x) = v_{sr\ max} - v_{sr} \geq 0$$
$$g_6 = T_{\max} - T \geq 0$$
$$g_7(x) = M_h - sM_s \geq 0$$
$$g_8(x) = T \geq 0$$

where

$$M_h = \frac{2}{3}\mu FZ\frac{r_o^3 - r_i^2}{r_o^2 - r_i^3}, P_{rz} = \frac{F}{\Pi(r_o^2 - r_i^2)},$$
$$v_{rz} = \frac{2\Pi(r_o^3 - r_i^3)}{90(r_o^2 - r_i^2)}, T = \frac{I_z\ \Pi\ n}{30(M_h + M_f)}$$

$\Delta r = 20$ mm, $I_z = 55$ kgmm^2, $P_{\max} = 1\ MPa$, $F_{\max} = 1000$ N,
$T_{\max} = 15\ s$, $\mu = 0.5$, s $= 1.5$, $M_s = 40$ Nm,
$M_f = 3$Nm, $n = 250$ rpm,
$v_{sr\ max} = 10$ m/s, $l_{\max} = 30\ mm$, $r_{i\ min} = 60$,
$r_{i\ max} = 80, r_{o\ min} = 90$,
$r_{o\ max} = 110$, $t_{\min} = 1.5, t_{\max} = 3$, $F_{\min} = 600$,
$F_{\max} = 1000$, $Z_{\min} = 2$, $Z_{\max} = 9$

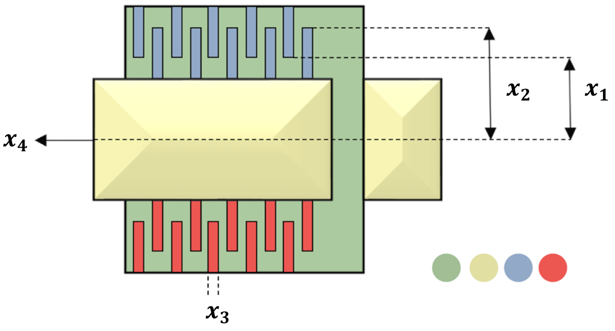


Fig. 7 Multiple disc clutch brake problem

3 Engineering design methods: a review

In this section, we review all optimization-based methods that have been used to solve engineering design problems.

3.1 Particle swarm optimization (PSO)

The particle swarm optimization (PSO) method is one of the simple and easy optimizers to implement. PSO imitates the behavior of flying birds. PSO is used widely to solve a variety of engineering problems. In this paper [101], an improved approach is presented based on the splitting phenomenon of astrophysics, which is implemented to move the abilities between local exploitation and global exploration. Furthermore, the Cauchy mutation is employed to evade local optima to a particular dimension of the best particle. The simulation results show that this algorithm has a practical and superior performance. In this paper [102], the authors suggest a hybridization between three algorithms genetic algorithm (GA), ant colony optimization (ACO), and PSO. This algorithm utilizes the ACO continuous and discrete probability distribution method, which will support the exploratory search of the GA algorithm. Furthermore, two methods are proposed, called mutation operators and distinctive crossover, to enhance local exploitation and global exploration. The PSO algorithm is proposed to support the exploitative search. The experimental result shows a competitive performance compared with other methods. In this paper [103], another hybridized with the sequential quadratic programming (SQP). In this paper [104], a dynamic adaptive inactivity weighting element based on a combination of GA and PSO is introduced called DAPSO-GA. In this model, the GA operators are combined into the PSO algorithm and adopted to improve the PSO performance. A few particles in the swarm that match the GA standard in the selection process with different time probability are chosen to update the particles' positions. The experimental results show that this approach improves convergence speed and robustness.

In this paper [105], a new approach is suggested to solve the drawbacks of PSO. The suggested algorithm build based on a combination of the three algorithms: PSO, sine-cosine algorithm (SCA), and elder-Mead simplex optimization (NMS), and named PSOSCANMS. The positions of particles are updated based on the optimal position founded by NMS and SCA. The results demonstrate that PSOSCANMS achieves an outperformance compared with other algorithms (i.e., PSO, GWO, GA, etc.). In this paper [106], an improved PSO version is introduced, this algorithm called multi-population ensemble particle swarm optimizer (MPEPSO). MPEPSO consists of three searching strategies. The particles are separated into four subpopulations, including three-pointer subpopulations and one recompense subpopulation. The velocities of the particles are updated in each pointer subpopulation. Further, at each learning stage, the improved function values are saved. At each learning stage, the recompense subpopulation is allocated to the optimal strategy. The CEC 2014 test suit is used to evaluate the performance of MPEPSO, which shows superior performance.

3.2 Salp swarm algorithm (SSA)

In this paper [50], Mirjalili et al. propose a novel metaheuristic optimizer named Salp Swarm Algorithm (SSA) and a Multi-objective SSA (MSSA). This SSA mimics the swarming attitude of salps during searching and seeking in seas. In this paper [107], a new method combines several selection schemes based on the SSA with the hill climbing (HC) algorithm named (HSSA). This algorithm comprises two steps. Firstly, to improve the exploitation of basic SSA, it combines with HC local search. Secondly, to enhance the exploration process of SSA, the selection system is applied. Furthermore, proportional selection is proposed to be a selection scheme, this hybridization with SSA called PHSSA. The results demonstrate that the HSSA enhances the local search execution in the SSA. On the other hand, the PHSSA improves the performance of the SSA by balance exploitation and exploration.

In this paper, [108], a suggested approach combines a Multi-objective method of Spotted Hyena Optimizer called (MOSHO), SSA, and the Emperor Penguin Optimizer (EPO), namely, MoSSE. The MOSHO algorithm employs to explore the search space. The SSA is used for the selection process in terms of obtaining the global solution. EPO was implemented for the improvement of the next solution. MoSSE was validated in ten IEEE CEC- 9 functions and compared with other algorithms. The results show that the proposed algorithm achieves excellent performance in accuracy, convergence speed, and searchability.

3.3 Whale optimization algorithm (WOA)

In this paper [109], a novel approach for solving engineering problems is proposed based on the well-known algorithm Whale Optimization Algorithm (WOA). The WOA was modified to solve the multi-objective problems with Non-Dominated Sorting WOA called (NSWOA). The NSWOA gathers each non-dominated Pareto best solution until the evolution reaches the last iteration. Then, the crowding measure method is based on the coverage of solutions and the bubble-net fishing approach. In this paper [110], a modified version of the WOA algorithm was proposed based on integrating the Lévy flight algorithm and chaotic local search into the WOA optimizer named BWOA. This approach aims to lead the swarm and improve the balance between the neighborhood-informed and conventional method abilities of the conventional method and examine the core searching abilities of WOA. The experimental results provide highly competitive outcomes in terms of conventional speed and accuracy. In this paper [111], an enhanced version of the WOA is proposed to overcome its drawbacks. The proposed method is MIWOALS, which stands for multi-objective improved WOA combined by a local search operator. The distance factor of WOA will be generated dynamically in this method. To avoid local optimal, the arrangement between moving to the contrast of the optimal solution and its actual values is based upon specific probability. The Pareto Archived Evolution Strategy (PAES) and Nelder-Mead method were introduced to improve the convergence speed and coverage. The MIWOALS algorithm's performance was validated on several benchmarks multi-objective functions such as CEC 2009, GLT, and DTLZ. The MIWOALS shows an outperformance comparing to other algorithms.

3.4 Teaching-learning-based optimization (TLBO)

The teaching-learning-based optimization (TLBO) mimics the classroom teaching aspect and consists of the teacher and learner stages. In this paper [112], a combination proposed of TLBO and neural network algorithm (NNA) called TLNNA. The experimental results demonstrate that the TLNNA achieves robustness performance in terms of accuracy and computational capability. In this paper [113], a new method suggested improving the TLBO algorithm (a reformative TLBO (RTLBO) method), excluding the bias of converging to the source and improving the exploration process through convergence time. Further, a self-learning stage is proposed to support the capability of exploration after convergence. Additionally, a mutation phase is presented to limiting untimely convergence and reliable combination capacity among the population. The results provide highly competitive performance better than other algorithms.

3.5 Artificial bee colony (ABC)

ABC became a new NIA based on the swarming concept. ABC is an effective and fast method for addressing global optimization problems. In [114], authors suggest an updated version of the ABC. The concept behind the updated version Dichotomous ABC (DABC) is to shift variably in two ways to establish a different trial stage. The validity of the suggested method is demonstrated by the findings of a trade analysis conducted on three main functional optimization problems based on the literature.

In [115], a combination technique that utilizes the ABC with Evolution Strategies (ES) from the Evolutionary PSO (EPSO) is proposed. This method combines ABC's efficiency with Evolution Strategies' detailed local search mechanism to create an algorithm that achieves high performance with a shorter time. Numerous benchmark methods and restricted optimization-engineering problems serve as an evaluation setting for this new algorithm. The ABC+ES findings are compared to ABC along with a variety of many other optimization methods.

In [116], a new homogeneous multi-agent optimization (HMAO) has been introduced. A computer-aided molecular (CAMD) problem is used to evaluate the framework. The CAMD is an explanation selection problem developed as a mixed-integer programming problem. Different effective ant colony optimization (EACO) method is treated as separate agents in developing the HMAO system. This method is demonstrated with a practical example of the best solvent design for liquid–liquid acetic acid extraction from a waste process source. The mixture properties are calculated using the UNIFAC system based on the infinite dilution activity coefficient. According to the findings, the accuracy of the objective method efficiency gains is increased. Compared to the solvents suggested in the literature, the introduced solvents have treatment thermodynamic properties highly.

An ABC meta-heuristic continued enhanced in [117] to optimize goal programming engineering design (GP) issues. The presented scheme firstly became modified to assist the GP through minimizing variations from determined goals and next enhanced with the excellent deluge meta-heuristic, particularly during the observer bee stage, to improve its local search performance. The introduced algorithm is named the great deluge artificial bee colony (GD-ABC). Tests on different methods demonstrated that the GD-ABC converges effectively addressed these issues. The comparison was developed using the nonparametric Mann–Whitney U test.

The ABC design imitates bees' intelligent foraging performance. There are three types of bees in ABC. Employed bees carry out discovery, and observer bees carry out exploitation, while explorer bees are in charge of looking for food sources in the available area at random. ABC is commonly employed to overcome multifaceted optimization issues due to its simplicity and lack of control parameters. ABC is effective at discovery than it is at exploitation. These two antagonistic variables determine any non-traditional algorithm's performance. With this limitation in view, the basic ABC food locations are improved in [118] utilizing the OBL principle. I-ABC greedy is a version that has been enhanced by integrating greediness into the searching behavior. The improvements help to preserve demographic diversity while also increasing exploitation. Seven mechanical engineering (ME) design issues are employed to validate the introduced algorithm. The findings demonstrated that the presented algorithms outperform state-of-the-art algorithms.

3.6 Cuckoo search (CS)

Mlakar et al. [119] explored how the hybrid self-adaptive cuckoo Search algorithm (HSA-CS) performs when faced with constraint engineering design optimization problems. These problems remain popular in research, such as welded beam, design of pressure vessels, speed reducer, and springs design. The findings are compared with the literature, which shows that the HSA-CS produced better results. It concluded from the findings that the HSA-CS is valid for use in real-world engineering applications.

Pauline et al. [120] suggested an adaptive cuckoo search algorithm (ACSA). In its diversification method, ACSA employs an optimized phase size selection approach. This method achieves the convergence characteristic by maintaining the CSA's balance between diversification and intensification efficiency. Three structural engineering problems show the ACSA's efficiency in solving structural optimization problems. Performance evaluation reveals that the ACSA performed the CSA standard and other literature approaches in several case studies.

The bat algorithm (BA) performs best at solving small-scale engineering problems and has the downside of being stuck in a local minimum with slow convergence when applied to more significant issues. Pathak et al. [121] introduced an enhanced bat algorithm named UBCSIW. In the UBCSIW, the bat method can influence optimum solutions and merge with cuckoo search to discover optimal solutions globally utilizing Levy fight. Next, the bat introduces a unique speed and location search formula to find the best candidate solution. This phase balances exploration and abilities and efficiently enhances performance using the greedy technique for optimal solution selection. Lastly, the Sugeno fuzzy inertia measurement is integrated with the speed update formula, which increases accuracy. The proposed UBCSIW algorithm's outputs are compared to those of standard BA and other state-of-the-art swarm-based algorithms.

3.7 Firefly algorithm (FA)

Kasdirin et al. [122] introduced a Hybrid Invasive Weed Firefly Optimization (HIWFO) method for solving engineering optimization design issues. The FA has a problem getting stuck at local optimums, while the invasive weed optimization (IWO) method is efficient in global exploration accuracy. The firefly approach is inserted into the invasive weed optimization in the proposed algorithm to improve the local exploration functionality of the IWO method, which already has strong searchability. The HIWFO method has better-searching efficiency and robustness, according to the analysis.

The firefly method is an effective swarm-intelligence approach influenced by fireflies' social technique, which is focused on their flashing and attraction features. In [123], authors examined how to use the firefly algorithm to address limited global optimization problems to incorporate a dynamic penalty approach. Some benchmark issues of engineering design optimization have been employed to evaluate the applicability and efficiency of the introduced method.

In [124], an efficient fruit fly optimization algorithm (named DSLC-FOA) has been presented using critical linear and logic chaos mapping to overcome optimization problems. The DSLC-FOA is more reliable than the literature to find the best solution to four standard benchmark functions. Various engineering design optimization issues are listed as applications, with approximate optimal results given by DSLC-FOA. Results indicated that the introduced DSLC-FOA method outperforms the standard FOA algorithms.

The FA is a well-known method to tackle a variety of global optimization problems. A survey of constraint management techniques is presented by Balande et al. [125]. It provides a hybrid algorithm for tackling restrained real-world engineering optimization issues, called the Stochastic Ranking with Improved Firefly Algorithm (SRIFA). The stochastic ranking method is widely employed to keep the weight functions and penalty in balance. FA is widely utilized because it converges high performance to other meta-heuristic methods. Furthermore, SRIFA employs feasibility-based principles to keep the penalty and objective roles in check. To test and analyze the efficiency of SRIFA, SRIFA is tested with the optimization of 24 standard CEC 2006 methods and five engineering problems. It has been shown that SRIFA's overall calculation results are higher than the simple FA findings.

3.8 Genetic algorithm (GA)

Genetic algorithms (GA) have been used to solve several optimization issues with great success. They solve complex problems that cannot or are difficult to address using traditional linear or nonlinear optimization methods. The Constrained Stochastic Optimization (CSO) problem is an example of one of these problems [126].

Engineering design challenges are challenging by nature as they include multiple variables and constraints in their essential objective functions. Engineers must ensure that the imposed requirements are met while keeping production costs down. Furthermore, the approach used may differ depending on the design issue. The key issue is determining which method is best for optimization. In the past, traditional optimization methods such as gradient search, evolutionary optimization, and random search were used to solve design problems. These are referred to as classical techniques. As a result, the approaches are less reliable. Soft-computing methods are also commonly used for function optimization. These are a lot more durable. One such approach is the genetic algorithm. In the field of stochastic optimization, it is a valuable method (non-classical). The algorithm generates a vast number of strings to arrive at the best result. Basakper et al. in [127] conducted a study to use a genetic algorithm to optimize engineering design problems and to investigate how the algorithm approaches the optimal solution efficiently and precisely. For the design variables, it selects the mathematical expression and optimizes it with the GA.

The engineering design problem is among the traditional optimization problems with complex constraint conditions and qualitative variable parameters. Genetic algorithm has made huge progress in the area of overcoming constrained optimization problems. The effects of using a traditional mathematical optimization approach are often substandard. One hybrid search strategy was developed in [128] to address the limitations of a simple genetic algorithm. The algorithm is less likely to trap optimally locally as a result of improvement. In addition, the simulation test demonstrated that the proposed method for engineering design problems had a significant impact on the reliability and precision of convergence.

xia et al. [129] presented a new approach, the optimal solution function approximation by kriging model (OSFAKM). In the kriging scheme, the optimum solution formula is approximated. A comprehensive test shows that OSFAKM achieves high results than BLGAs. To explore the effectiveness of the suggested technique, various test problems are provided.

3.9 Gravitational search algorithm (GSA)

In [130], Rather et al. investigated the performance of a chaotic gravitational search algorithm (CGSA) when addressing automated engineering scheme structures. To improve the optimization power of the gravitational search (GSA) method, various chaotic maps have been merged with gravitational constant in this analysis. CGSA has also been employed to keep the gravitational constant's adaptive capacity. To statistically check the findings, nonparametrical statistical tests, Wilcoxon rank-sum testing, were conducted at the level of 5% significance. Besides, 11 meta-heuristic methods were employed to verify the experimental outcomes, strengthening the experimental setup's authenticity.

Rather et al. [131] proposed a stochastic hybrid optimization technique known as the Constriction Coefficient-based Particle Swarm Optimization and Chaotic Gravitational Search Algorithm (CPSOCGSA). The PSO combination is caused by the theoretical activity of bird fluctuation and GSA, a physical heuristic technology stimulated by the Newton law on universal gravity. CPSOCGSA's advantages have been integrated into a variety of mechanical and civil engineering design frameworks. Compared to other algorithms, the proposed method effectively decreased the cost method.

3.10 Sine cosine algorithm (SCA)

Tawhid et al. [132] introduced the multi-objective sine-cosine algorithm (MO-SCA) using the sine-cosine exploration method algorithm. MO-SCA employs the elitist non-dominated sorting and crowding distance approaches to achieve various non-domination levels and support flexibility with the optimal explication range. The expected outcome is calculated by implementing multi-target comparison problems. The suggested technique has been evaluated for problems with distinctive features in the multi-objective engineering framework. Besides, it demonstrated that the proposed method outperforms other well-known methods.

Rizk et al. [133] introduced a novel technique for tackling engineering design issues called a multi-orthogonal sine cosine algorithm (MOSCA). The suggested MOSCA combines the benefits of the SCA and MOSS to reduce the shortcomings of the SCA. The suggested MOSCA is divided into two phases. The first phase begins with the discovery method to improve investigation ability. Second, the MOSS process begins its exploration by looking for SCA that has already been discovered to increase exploitation tendencies. As an alternative, the MOSS stage will help the SCA stage searching based on in-depth exploration/exploitation models. As a result, the MOSCA could be further stable. The MOSCA algorithm is tested on various benchmark issues and four engineering design issues. The findings show that MOSCA is an effective method and, in many cases, exceeds the other techniques.

To solve engineering design problems, a hybrid SCA and SSGA have been presented in [134]. This approach combines the value of SCA's discovery and SSGA's exploitation capacity to reduce susceptibility to initial concentration, accelerate the quest method, and speed up confluence in an efficient time. The proposed method combines SSGA and SCA principles to produce a new generation using SSGA and SCA mechanisms. To test the reliability of the introduced method, it uses two engineering design issues. According to the findings, the proposed method outperforms other optimization methods.

3.11 Gray wolf optimizer (GWO)

Kumar et al. [135] introduced a modified technique for avoiding the premature exploration and convergence of a gray wolf optimizer (GWO) on optimization problems. To improve the efficiency of the existing GWO, three new strategies are employed. The first technique makes use of the prey weight principle. The second technique employs astrophysics principles; the gray wolves are guided into the advancement areas of the quest space. The strength of this technique is that it allows each wolf to discover from the sun's movement in the exploration room and the fact that the wolves are required to identify at the same time. The third strategy incorporates both first and second techniques to benefit from astrophysical and prey weight strategies. The suggested GWO enhancements were validated using thirteen benchmark test functions. The suggested improvements have been compared with five well-known techniques. The experimental findings demonstrate the importance of engineering solutions of the modified technique compared with other methods.

In [136], this research aims to find an optimum synergy exploration/exploitation SCA. The exploration capabilities are enhanced, firstly with integrating the social and the cognitive element and secondly with the gray wolf optimizer (GWO) maintaining the balance between exploration and operation. The introduced method is called SC-GWO. A collection of benchmark issues are used to evaluate results. The dimensions of benchmark testing issues vary from 30 to 100 to demonstrate the SC-GWO's scalability. The analysis of findings reveals that the proposed SC-GWO has an accurate result compared with well-known methods.

An Improved Gray Wolf Optimizer (I-GWO) has been suggested in [93] to tackle global optimization and engineering design issues. The I-GWO method takes advantage of an unusual movement approach. DLH takes a modified way to identify neighbors for each wolf; the first is exchanging knowledge between neighbors. This learning method, employed in the DLH discovery method, helps preserve diversity while balancing local and global search. The suggested I-GWO technique is tested using the CEC 2018 benchmark dataset. I-GWO is compared with well-known methods validating the effectiveness of the proposed engineering design algorithm shows.

3.12 Moth–flame optimization (MFO)

The MFO technique has been considered a new nature-inspired heuristic model. This algorithm's key source of motivation is the transverse orientation exploration system used by moths. Moths usually fly at night by having a fixed point with the moon, a powerful method for traveling long distances in a direct route. An enhanced variant of the MFO method called LMFO has been suggested in [137]. Levy flights could improve population diversity and enhance the efficiency of the introduced technique. This method can help deliver more reliable performance in MFO's exploration/exploitation abilities, making LMFO faster and more stable than MFO. A comparison with DA, GGSA, BA, ABC, and PSOGSA is examined and demonstrates the performance of LMFO.

Jangir et al. [138] introduced a Non-Dominated Sorting Moth Flame Optimizer (NSMFO). The suggested NSMFO method gathers all optimum solutions obtained in non-dominated Pareto before the last limit of iteration evolves. From the collection of Pareto optimum solutions, the best solutions are chosen based on a crowded distance method, as a navigating technique, to direct moths to the dominant search areas. A collection of emotional, restrained, and engineering design issues is applied to verify the efficiency of the developed NSMFO method. NSMFO's findings are compared to well-known techniques. The effectiveness of the suggested NSMFO method validates the performance of NSMFO in terms of faster convergence in the standard unconstrained, high-reach coverage, and restricted and engineering design issue.

Sayed et al. [139] introduced a hybrid combination of MFO and SA algorithms (SA-MFO) as it combines the benefits of both techniques. To control neighbor identification, it can avoid local optimum mechanisms of SA and achieve a quick discovering learning mechanism. The suggested SA-MFO technique is evaluated, and the findings illustrate that the introduced method is efficient. Moreover, SA-MFO is compared to current meta-heuristic methods. The findings demonstrate that SA-MFO can compete with MFO and other meta-heuristic algorithms.

Shehab et al. [140] presented two levels of improvement the standard Moth Flame Optimization(MFO) method. The first stage hybridized the MFO and the local algorithm, Hill Climbing (HC), known as the MFOHC. The suggested technique takes advantage of HC to optimize searches and enhance learning to find optimal solutions for standard MFOs. The second stage adds six typical selection schemes for improvements in the quality of the selected solution by providing an opportunity to choose with high fitness and diversity. The experimental results demonstrate that the proportional selection strategy (PMFOHC) with MFOHC outperforms the other suggested algorithms.

3.13 Differential evolution (DE)

In this section, the basic, modified, and hybrid differential evolution methods are presented that have been used in the literature to solve various engineering design problems. In evolutionary methods, differential evolution is a process that optimizes any optimization problem by iteratively seeking to develop a candidate solution concerning a provided measurement of quality [141].

3.13.1 Basic differential evolution

A fast differential evolution (FDE) framework is developed to solve constrained engineering design optimization problems. To achieve a compromise between population discovery and exploitation, Shen et al. [142] introduced a 'DE/current-to-ppbest/bin.' Furthermore, to update the population, a ranking-based selection process chooses the most promising individuals from a mix of parents and offspring. FDE can achieve very competitive efficiency, according to experimental results on five instances extracted from engineering design. In terms of solution efficiency, FDE is competitive with other existing approaches. FDE achieves better performance compared with other state-of-the-art approaches when it comes to convergence speed. FDE is a good option when the number of function evaluations is small, or the cost of function evaluation is high.

Dervis et al. [143] compared the achievement of the differential evolution (DE) algorithm with some other popular, efficient versions of genetic algorithms represented by PGA, Grefensstette, Eshelman. They tested the algorithm on some popular tested functions called De Jong's functions, where the experiments depended on measuring the speed of convergence of the algorithms. The authors used different types of tested functions like convex, non-convex, discontinuous, stochastic, and multimodal.

Kim et al. [144] modified the differential evolution mechanism by adding some parameters and modifying them; this is for the global optimization problem, which is constrained. The modification depended on utilizing a scaling factor in the initialization phase. Such factor value is fixed in the range (0, 2] where there is no previous work concerned the value of this factor, and they used the value gained empirically, which ranges from 0.4 to 1. The authors also tried to handle the constraints to penalize the violations of these constraints by utilizing a penalty function. Where to handle the constraints, there are several criteria for selection, when two solutions are feasible, the solution with best fitness value is selected, and when there is only one feasible solution, it is selected. As well as, when the two solutions are infeasible, then the one with the lower violation is selected. After such modifications, they tested the machine on some engineering problems. The experimental results proved the superiority of the proposed technique, which also applied to the robust design of a gas circuit breaker to decrease the performance variation and enhance the ability of interruption.

Pant et al. [145] enhanced the differential evolution algorithm, where their algorithm called Modified Differential Evolution (MDE), where the enhancement depended on utilizing the technique of the opposition-based learning for initializing the starting population individuals, which differs from the traditional randomness techniques. Also, in the standard DE, the individual of mutation is generated randomly, but in MDE, it is tournament-best, and finally, MDE uses just one population set in opposite for two sets in the standard DE. The authors compared the proposed MDE and the standard DE according to the average fitness of function values, standard deviation, and t-values and studied changing the dimension of the problem and its influence at the dimensions 10, 30, and 50. In terms of the results obtained, MDE surpasses DE on 11 cases, while DE outperforms MDE in 4 cases out of 15 cases in terms of average fitness.

The authors then investigated the efficiency of the MDE algorithm by applying it to two real-world problems: transformer architecture [146], and transistor modeling [146], which are restricted and unconstrained problems, respectively. Consequently, the findings revealed that the proposed algorithm can solve problems of various dimensions in less time and with fewer feature evaluations without sacrificing solution efficiency.

Melo et al. [147] minimized the weight of a tension spring to test DE to solve constrained engineering design problems and compare the findings with some current optimization methods. The results revealed that both mechanisms could get very close to the right solution for each problem after a limited number of feature tests. Furthermore, small standard deviation values may suggest the mechanism's supremacy. Moreover, DE produced better results with a lower mean (after 50 runs) and several objective evaluations when DE compared to other standard methods.

In [148], the DE has been applied to solve mixed-integer constrained process engineering problems. In addition, the results of this advanced method are compared with Branch-and-Bound as an exact method and a genetic algorithm. The experimental results consist of two analytical formulations for the batch plant design problem and examples gathering the three issues as mentioned earlier. The results established the high performance of DE to solve this kind of problem against Branch and-Bound and GA.

3.13.2 Modified differential evolution

A modified DE was proposed in [145], which has three different ways from the traditional DE. The first one is to use opposition-based learning instead of uniform random numbers for the initialization. The mutant individual is tournament-best, while in the traditional DE is random. Finally, two sets of population are used in the traditional DE, where this modified version uses only one set. This modified version was evaluated in different problems, using ten benchmark functions and two real applications. It was also compared to the traditional DE, and it reached better performance in all tests except one benchmark test.

Xiao et al. [149] proposed an improved evolutionary dynamic membrane algorithm for solving restricted engineering design problems based on the improvement of particle swarm and differential evolution (IDMEA-PSO/DE). The PSO/DE search strategy is combined with the complex membrane structure of P systems in this approach. Several engineering design issues have been used to estimate IDMEA-PSO/DE results. Simulation outcomes revealed that the introduced method outclasses various algorithms presented in the literature.

Ao and Chi [150] presented a new modified version of the DE using a mutation operator that generates multiple trial vectors that makes the best of selected multiple parents to enhance the probability of creating optimal offspring. Also, they proposed a new adaptive control parameter to improve the capacity of adaptation. It was applied in different constrained engineering optimization problems and compared to related methods. The modified version achieved significant performance, and it can be considered an efficient optimization algorithm that outperforms the traditional DE.

Ali et al. [151] proposed two modified versions of the DE, namely, DE with Parent Centric Crossover (DEPCX) and DE with probabilistic Parent Centric Crossover (Pro. DEPCX). The main modification of these two modified versions is to apply parent centric mechanism as a crossover operator in the mutation phase. It was tested with different nonlinear engineering design problems in comparison with the traditional DE. The outcomes confirmed that the two modified versions enhanced the convergence rate of the traditional DE.

In [152], another modified DE was proposed depending on the global competitiveness ranking. In this modified version, a modified mutation, self-adaptive control parameters, modified selection, and inversion operation are introduced to obtain global solutions. The global competitive ranking is incorporated to obtain the fitness of individual solutions. Sixteen engineering design optimization problems were utilized to evaluate the modified DE, which showed competitive performance compared to other methods.

In [147], a modified DE with a simple penalty function was applied to solve constrained engineering problems, using five mutation strategies. These strategies are employed with a straightforward strategy to handle constraints. It was evaluated on minimizing the weight of a tension/compression spring and the pressure vessel design. Also, it was compared to several well-known algorithms, and it reached significant performance, especially in terms of standard deviations.

In [153], a Multi-View DE approach was proposed for solving constrained and unconstrained engineering optimization tasks. This DE variant uses several mutation strategies to create different views for each iteration in the current population. After that, views are combined based on the winner-takes-all scheme, which automatically leads to balance exploitation and exploration.

In [154], a new modified self-adaptive strategy to control parameters in DE (ISADE), is proposed. Three mutation operators of the DE were chosen as candidates to be employed to individuals in the generated population because of their ability to improve solutions with different characteristics. It showed significant performance in solving constrained engineering optimization problems.

Muangkote in [155] proposed an improved constrained DE, called ICDE to handle constrained optimization problems in ME. The developed ICDE uses an archiving-based adaptive tradeoff model to deal with constraints. Also, another modification was applied to the ICDE using the penalty technique to handle the constraints. These two modifications, called ArATM-ICDE and Penalty-ICDE, were evaluated in ME constrained problems, and the ArATM-ICDE showed better performance.

Gong et al. [156] presented an improved constrained DE for solving constrained engineering problems. They proposed two improvements to enhance the traditional DE. The first one is to apply a ranking-based mutation operator to enhance converge of the DE. The second improvement is by applying a dynamic diversity mechanism is applied to maintain solutions. Twenty-four benchmark functions were utilized to test the proposed rank-iMDDE, and the results showed that it improved the quality of solutions.

To handle a nonlinear computing and engineering design problems, an improved differential evolution algorithm (IDEA) was presented by tsai [157]. The Taguchi form, sliding stages, and DEA algorithm are combined in the introduced IDEA. The DEA holds a strong microspace exploration capability with fewer control parameters. The orthogonal design with sliding stage and acknowledgment table's hierarchical reasoning capacity is employed to identify the best individuals on microspace as future offspring. As a result, the presented IDEA on exploration and exploitation is well improved and developed. It examines the performance of the evolutionary parameters on the IDEA's results. The IDEA demonstrates its efficiency comparing with DEA and the real-coded genetic algorithm. The IDEA was employed to overcome engineering design optimization issues and showed that it is more reliable. Based on the obtained results, the proposed IDEA achieves more reliable results than the introduced in the literature.

Moreover, Mohamed [158] developed a new DE method, called NDE for constrained optimization problems. A new triangular mutation rule was applied to boost the performance of the traditional DE. The NDE was tested on five CEC-2006 benchmark functions and compared to different methods. It had competitive performance using several evaluation metrics.

Mohamed et al. [159] developed an enhanced DE, called EDDE, using information given by reasonable individual solutions and bad individual solutions in the population. This new mutation process improved the efficiency of the balance between exploration and exploitation phases.

Kizilay et al. [160] presented a Q-Learning differential evolution method (DE-QL) to tackle engineering problem design (EDPs) issues. The output of a DE algorithm relies on a mutation technique and its variables of control, namely crossover and mutation rates. Therefore, the proposed DE-QL produces the test population via the QL method to facilitate selecting the mutation strategy between four different strategies and crossover and mutation rates of the Q table.

Sun et al. [161] proposed an enhanced atom discovery optimization with a nonlinear inertia weight function, a neighbor learning variable, and updating mechanisms. First, a nonlinear inertia weight factor is calculated to identify the prior speed and acceleration retained to better balance discovery and exploitation. Second, a neighbor learning element is inserted into the updating speed formula to significantly enhance the exchange of information among atoms, which is essential for improving the exploitation capability. Third, an update method is suggested. A greedy collecting method is used to increase the chance of identifying the best selection. Finally, the performance evaluation of the proposed algorithm was demonstrated using the state-of-the-art CEC2017 benchmark datasets and various real-world engineering design issues. The findings indicate that the introduced method outperforms the comparative methods.

3.13.3 Hybrid differential evolution

In [162], a hybrid DE algorithm (MDE) with a set of mechanisms, most of them are inspirit from particle swarm optimization. For example, a modified mutation operator

(MMO) performs a local search method according to the neighborhood best solution and best personal solution (as performed in PSO). Since all the solutions are produced around their neighborhood, the topology of these solutions has the most extensive influence on the algorithm's performance. In general, four types of typologies can be used named stars, wheels, circles, and random edges. In this modification of DE, circles topology has been applied to enhance its performance when applied to the tension/compression string design problem and the economic dispatch problem in the power system. The results of MDE are compared with traditional DE and PSO. These results showed that MDE is better than other methods.

Bai et al. [163] introduced a hybrid DE evolution estimation of distribution algorithm (IHDE-EDA) to solve mixed-integer nonlinear programming (MINLP) and nonlinear programming (NLP) problems. IHDE-EDA combines the information extracted from the annealing mechanism-embedded EDA and the DE technique. In general, the solution is mutated either using the traditional mutated DE operator or the annealing mechanism for EDA. This leads to improve the global searching behavior of IHDE-EDA and increase the convergence rate. To assess IHDE-EDA, it has been applied to solve the industrial-size scheduling of two-pipeline crude oil blending problem. The comparison results with other methods established the performance of IHDE-EDA.

In [164], a hybrid between DE and cuckoo search (CS) has been developed to solve constrained engineering problems. In this algorithm named CSDE, the population is divided into two parts, and one of them is updated using DE, and another is updated using CS. To validate CSDE, it has been applied to solve several optimization problems, including six constrained engineering, ten constrained, and 30 unconstrained benchmark functions. In addition, it has been compared with 12 state-of-the-art algorithms, and it has been observed that high performance of CSDE among the tested problems.

Parouha [165] introduced a 'Memory-based DE (MBDE)' with two 'swarm operators' is proposed. The suggested method is compared with several well-known approaches, and its viability is evaluated on constrained and unconstrained engineering design issues. The introduced algorithm's competence was demonstrated by numerical, mathematical, and graphical analyses.

Aliniya et al. [166] presented a crossover-based imperialist competitive algorithm (CB-ICA). As compared to ICA, the suggested algorithm has three differences. The consistent division crossover and levy modification approach is employed to increase exploration potential in the adaptation and substitution steps. Moreover, using a constant crossover during the improvement phase with relevant information sharing guides to an optimal solution improving convergence speed. The introduced technique can also be employed to solve problems in the real world by using restriction handling technology. Therefore, CB-ICA was examined with different well-known methods and achieved satisfactory results in five engineering design problems.

Balakrishnan [167] discussed the implementation of an evaluative system for a course with high PBL and TBL features, which ensures that students are not only subjected to all learning outcomes when dealing with a project but are tested individually to prove their skills. This framework's primary approach is an evaluative tool known as an 'open-ended design test,' which uses a 1:1 mapping of the team and individual learning, with scaffolding in the course structures to help. Moreover, it describes how this method was used to design two courses at the University of Manitoba, the results and feedback to the course layout, and recommendations for future courses or programs.

Samma et al. [168] proposed an improved optimizer that includes a Q-learning algorithm with SA, called QLSA. The Q-learning method, in general, is integrated into SA to improve its efficiency by adjusting its variables at execution time. The key feature of this method is the employment of a reward/penalty method to monitor the highest-rated model parameters, such as the annealing factor and mutation rate. A seven restricted engineering design issues were utilized to test the efficiency of the suggested QLSA algorithm. The results show that QLSA exceeds other studied algorithms substantially (i.e., confidence level 95 percent).

Nadimi et al. [169] introduced a multi-trial vector-based differential evolution. The technique has been implemented based on the standard MTV method. MTV merges various exploration techniques as vector test manufacturers (TVPs). The suggested MTDE technique has been tested using the CEC 2018 benchmark package, which involves multiple engineering design issues and includes hybrid, multimodal, unimodal, and composition functions. The findings demonstrate that the MTDE algorithms perform better and high accuracy of the optimization process.

Sadeeq et al. [170] presented a hybrid Bird Mating Optimizer (BMO) with the DE method. BMO has been considered a novel method with several weaknesses, including slow convergence, weak solution consistency, and trapping in local optima. A new BMO-DE algorithm is introduced to overcome these shortcomings. BMO-DE was employed to overcome several engineering design optimization problems. Experimental findings indicate that the suggested technique could suggest better solutions compared with other modern meta-heuristic methods that solve such situations.

3.14 Other optimization algorithms

In this paper [171], the Artificial Atom Algorithm proposed to solve the three-bar truss problem, which is a famous engineering problem. Compared with several algorithms such as Cuckoo search, Mine Blast algorithm, Bat algorithm, and Cricket Algorithm shows an outperformance.

In this paper [172], the authors introduce a method to solve multi-objective problems based on an artificial algae algorithm called MO-AAA. This method used the crowding distance and selected a non-dominated sorting approach to maintain the variety among the optimal solutions. To validate this algorithm, MO-AAA was implemented in different engineering problems. The results demonstrate an efficient performance and ease of implementation.

In this paper [173], a multiple criteria optimization approach is introduced. This study hybridizes the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) and the design of experiment (DoE) to identify an objective function to solve a specific engineering problem. Then, the polynomial fitting to exploratory data is in various linear regression analysis fashion. Then, the regression function is combined within the criteria constraints and mathematical pattern to find the optimal criteria set. The proposed approach is applied in two different examples and shows a great performance in solving engineering problems.

In this paper [174], Chagwiza et al. present a method to solve engineering and networking problems called GHMMAS based on a combination of the Grotschel-Holland and Max-Min Ant System method. GHMMAS shows an efficient performance in solving higher-order polynomial problems. In this paper [175], an enhanced version of the Time evolutionary optimization algorithm is proposed, named ETEO. ETEO modifies the time evolutionary part and implemented population clustering. The proposed approach shows a superior performance among other algorithms.

In this paper [176], to solve various engineering problems, a model based on the Chaos-enhanced Stochastic Fractal Search (CFS) method has been proposed. The proposed approach uses to solve the Welded Beam design, Tension/compression spring, and Pressure Vessel Design. The CFS model outperforms in terms of convergence rate and accuracy regarding the standard Stochastic Fractal search algorithm.

In this paper [177], Bilel, Najlawi, et al. presented an improved model based on the colonial competitive algorithm(CCA) to solve different engineering problems. The proposed approach is called MOCCA, which refers to multi-objective problems. The CCA algorithm combined the Pareto concept and the adaptation operators. The performance of MOCCA was tested on multimodal and unimodal functions. The comparative analysis with different algorithms uses three metrics, i.e., solution diversity, convergence, and robustness. MOCCA shows an outperformance comparing with different algorithms implemented in the same multi-objective problems.

In this paper [178], a new method is based on the Learning Automata and modified Butterfly optimization algorithm (BOA) named LABOA. Switch probability is changed by learning automata to make a balance between global and local search. To validate the performance of the LABOA, different benchmark functions are used. The simulation result compared with different algorithms (e.g., BSO, ABC, BA, CLPSO) and LABOA demonstrates an outperformance among different algorithms.

In this paper [179], Li, Guocheng, et al. introduce an enhanced version of BOA combining with the cross-entropy (CE) algorithm called BOA-CE. BOA-CE uses the co-evolution method to balance exploration and exploitation. The proposed algorithm is validated based on 19 benchmarks. BOA-CE provides competitive results in enhancing exploration and exploitation, local optima avoidance, and convergence time.

In this paper [180], a new meta-heuristic method is proposed to solve the continuous nonlinear optimization. The algorithm is inspired by black widow spiders and namely the Black Widow Optimization Algorithm (BWO). This algorithm consists of an independent stage called cannibalism, which achieves a quick convergence. The proposed algorithm shows competitive results such as fast convergence time and reaching fitness value.

In this paper [181], an improved Equilibrium Optimization Algorithm, namely, IEOA. Furthermore, IEOA hybridizes with the recycling method for setting the power distribution grid, with the best allocation of shared generators. The effectiveness of the presented approach was tested on 23 benchmark functions. To examine this approach, verification on a large-scale distributed system with a deferent of control variables is proposed on a 137-bus system. IEOA compared with a different approach such as genetic algorithm, refined genetic, harmony search, refined genetic, firefly optimization algorithms, and fireworks. The result demonstrates that IEOA is adequate and robust compared with other algorithms.

In this paper [182], a new meta-heuristic approach is proposed and called Chaos Game Optimization (CGO). The algorithm build based on the chaos theory. To validate the CGO algorithm, the results were compared with different meta-heuristic algorithms. CGO shows superior performance in most cases.

In this paper [183], a new algorithm is proposed, namely, symbiotic organisms search (SOS), which is inspired by the symbiotic mutual impact system used by organisms to survive and proliferate within the ecosystem. To demonstrate the performance of SOS, four different constrained multi-objective problems were used to prove the stability and precision of multi-objective systems (MOSOS). The algorithm shows an outperformance comparing with different algorithms.

In this paper [91], Abualigah, et al. suggested a novel meta-heuristic approach inspired by four mathematical arithmetic operators including addition, subtraction, multiplication, and division. The proposed algorithm is called Arithmetic Optimization Algorithm (AOA). The performance of AOA validates 29 benchmark functions and different engineering problems. The result reveals that AOA outperforms other algorithms.

In this paper [184], a novel meta-heuristic approach is introduced called Henry gas solubility optimization (HGSO), which imitates the behavior ruled by Henry’s law. Henry’s law is a gas low measuring the dissolved gas in a specific volume and temperature. HGSO mimics the gathering gas behavior to achieve the balance between exploration and exploitation. The HGSO performance shows a competitive result comparing to several algorithms such as PSO, GWO, and WOA.

In this paper [185], the authors analyzed the performance of several meta-heuristic algorithms in solving engineering problems. The study cover spotted hyena optimizer (SHO), GWO, bat algorithm (BA), gravitational search algorithm (GSA), WOA, MFO, ACO, and PSO. The experimental results depict that the SHO and GOW algorithm provides the best performance among other algorithms.

3.15 Overview of the studied papers

From observing the behavior of these MH techniques, it has been noticed that the evolutionary-based methods (EBMs) have high efficiency over the other MH techniques. These can be observed from Figs. 8, 9 and 10 that depict the importance of MH and engineering optimization and their influence on different fields from 2008 to 2021. So, this motivated us to present a review for the use of MH in engineering optimization problems. Summary of some studied methods is given in Table 1.

4 Results analysis and discussion

In this section, the results of several engineering optimization problems are given. We have taken some of the most common methods applied to solve various engineering optimization problems.

The results of the selected optimization methods for solving the welded beam problem are given in Table 2. The comparative optimization algorithms (i.e., SIMPLEX [189], APPROX [189], RO [190], DAVID [189], GSA [186], MVO [186], WOA [187], CPSO [191], CSCA [192], HS [193], OBSCA [194], GA [195], and AOA [91]) are used to compare the benchmark problem. It is clear that some algorithms got promising results, such as RO got 1.735344, MVO got 1.72645, OBSCA got 1.722315, and AOA got the best results (i.e., 1.7164). The competitiveness is high in solving this problem due to the results of the presented algorithms. Thus, a new algorithm can be proposed to induce different and better results than the existing algorithms.

The results of the selected optimization methods for solving the tension/compression spring problem are given in Table 3. The comparative optimization algorithms (i.e., RO [190], WOA [187], GSA [67], ES [198], OBSCA [194], MVO [186], CSCA [192], CPSO [191], PSO [191], CC [199], CC [199], HS [200], GA [201], and AOA [91]) are used to compare this benchmark problem. The results in general for this problem are almost similar with a small difference. AOA got the best results in solving the tension/compression spring problem compared with other well-known algorithms. Almost all the other algorithms got the same results in solving this problem, which means that this problem needs an effective new optimization algorithm to get better results. Competitiveness exists in solving this problem due to the variation in the results obtained from the state-of-the-art algorithms. Thus, a new algorithm can be proposed to obtain different results compared with the existing algorithms.

The results of the selected optimization methods for solving the pressure vessel problem are given in Table 4. The comparative optimization algorithms (i.e., MVO [186], HPSO [202], WOA [187], GSA [67], CSCA [192], ES [198], CPSO [191], ACO [203], PSO-SCA [204], HS [200], GA [201], and AOA [91]) are used to compare this benchmark problem. It is clear from the results achieved in solving this problem that the outputs have apparent discrepancies, supporting the need for an effective optimization method to solve this problem efficiently. The best method from the given in Table is AOA, it got the best optimal cost (6048.7844), followed by HPSO got 6059.7410, ACO got 6059.7258, PSO-SCA got 6059.71433, and GA got 6059.94634. Compared to the comparison table’s methods, the AOA method has proven its ability to solve this problem efficiently. This method can be developed in the future to obtain more accurate results.

The results of the selected optimization methods for solving the 3-bar truss problem are given in Table 5. The comparative optimization algorithms including CS [206], Tsa [207], DEDS [208], Ray and Sain [209], MBA [210],

Fig. 8 Publications of optimization methods

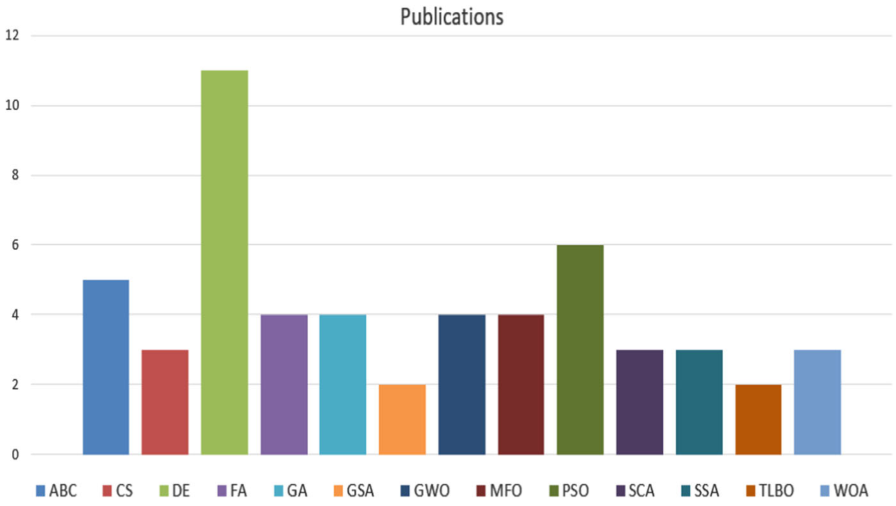
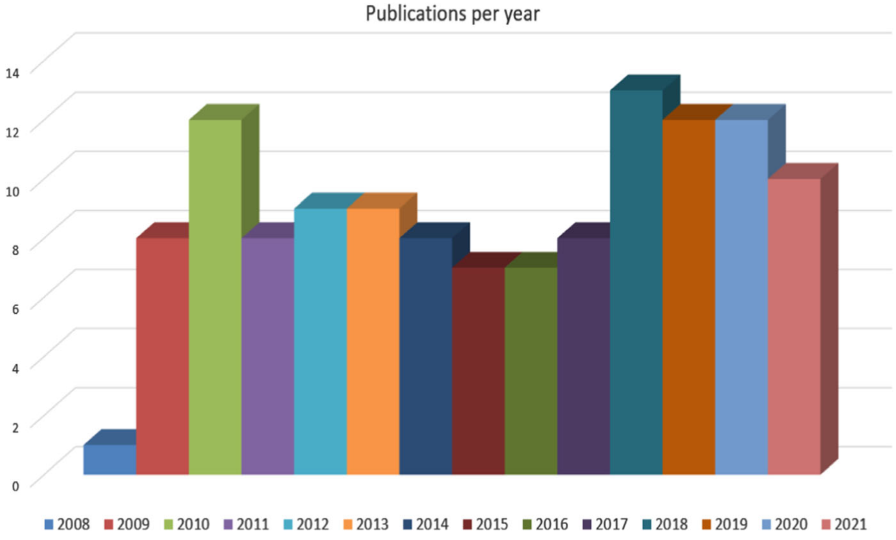


Fig. 9 Engineering optimization over years from 2008 to 2021



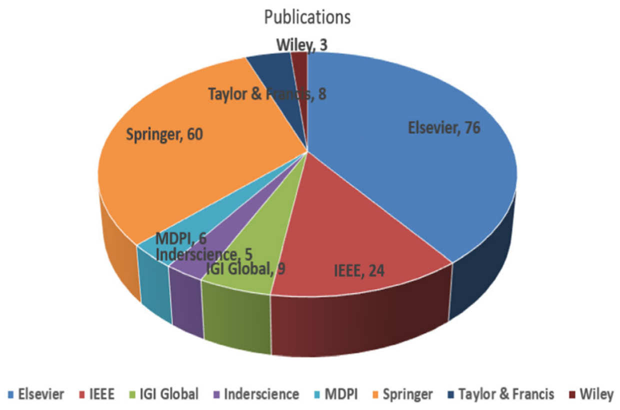


Fig. 10 Publications per publisher

SSA [50], PSO-DE [204], PSO-DE [204], AOA [91], Artificial Atom Algorithm (A3) [171], and enhanced time evolutionary optimization (ETEO) [175]. are used to analyze this benchmark problem. The results listed for solving this problem using different optimization methods indicate that the results are very close. This indicates the need to develop a new and innovative way to obtain strong and distinct results to solve this problem due to the obtained results in solving this problem. Tsa got the best results in solving the 3-bar truss problem, and it got the best optimal weight (263.68) over the comparative methods.

The results of the selected optimization methods for solving the pressure vessel problem are given in Table 6. The comparative optimization algorithms (i.e., CS [206], GWO [53], AAO [211], SCA [212], MFO [213], APSO [214], WSA [215], PSO-DE [204], LGSI2 [215], LGSI4 [216], FA [217], and AOA [91]) are used to this benchmark problem. There are apparent differences in the results obtained from the methods that solve this problem. Some methods got better results than others, but it is clear that there is still a need for better approaches to solve this problem and obtain more accurate results for the variables in this problem. LGSI2, LGSI4, and AOA obtained the best

Table 1 Summary of some studied methods

Reference	Method	Problem	Publisher	Year
[67]	Gravitational Search Algorithm	Cantilever beam design Three-bar truss Gear train design Welded beam design	Elsevier	2009
[53]	Gray Wolf Optimizer	Tension/ compression spring design Welded beam design Pressure vessel design	Elsevier	2014
[186]	Multi-verse Optimizer	Welded beam design Gear train design Three-bar truss Cantilever beam design	Springer	2016
[187]	Whale Optimization Algorithm	Tension/ compression spring design Welded beam design Pressure vessel design	Elsevier	2016
[115]	Artificial Bee Colony	WBD, DPV, SRD-11 and MWTCS	Springer	2016
[188]	Spotted Hyena Optimizer	25-bar truss design Multiple disk clutch brake design	IEEE	2017
[120]	Cuckoo Search Algorithm	Gear Train Design Pressure vessel design Three-bar truss design	IEEE	2017
[171]	Artificial Atom Algorithm	Three Bar Truss	IEEE	2018
[172]	Artificial Algae	Welded beam Spring design Speed reducer problem	Springer	2018
[175]	Time Evolutionary Optimization	Three-bar truss design Pressure vessel design Speed reducer design Tension/compression spring design Welded beam design	Taylor & Francis	2018
[177]	Colonial Competitive Algorithm	Speed reducer design Disc brake design	Springer	2019
[179]	Butterfly Optimization Algorithm	Tension/compression spring design Welded beam design	MDPI	2019
[104]	Particle Swarm Optimization	Three-bar truss design Welded beam Belleville disc spring	SAGA	2019
[107]	Salp Swarm Algorithm	Three-bar truss design Pressure vessel design Welded beam design Tension/compression spring design Speed reducer design	Springer	2020
[112]	Teaching–learning-based Optimization	Tension/comparison spring design Welded beam design Pressure vessel design Speed reducer design	Elsevier	2020
[95]	Differential Evolution	Rolling element bearing Car side design Heater exchanger	Elsevier	2020

Table 1 (continued)

Reference	Method	Problem	Publisher	Year
		Corrugated bulkheads design		
		Tension-Compression Spring		
		Multiple disk clutch brake		
[181]	Equilibrium Optimization	137-bus system	Elsevier	2021
		69-bus test system		
[68]	Aquila Optimizer	Welded beam design	Elsevier	2021
		3-bar truss design		
		Pressure vessel design		
		Tension/compression spring design		
		Speed reducer		
		Disc clutch brake		
		Cantilever beam design		
[91]	Arithmetic Optimization Algorithm	Welded beam design	Elsevier	2021
		Pressure vessel design		
		Tension/compression spring design		
		Speed reducer		
		3-bar truss design		

results, and it is clear that the ability of AOA is good in solving real-world engineering problems; it got 2997.9157. In future work, it is possible to develop a smart way to obtain better results.

The results of the selected optimization methods for solving the pressure vessel problem are given in Table 7. The comparative optimization algorithms (i.e., ALO [218], CS [206], MMA [219], GCA_I [219], SOS [220], GCA_II [219], MFO [213], and AOA [91]) are used to compare benchmark problems for the analysis of the MCAOA results. There are few and unclear differences in the results obtained from the methods that solve this problem. Some methods achieved better results than others at low rates, but it is clear that there is still a need for better approaches to solve this problem and obtain more accurate results for the variables in this problem. The results indicate that this problem needs a compelling optimization method to get results different from the comparative results. This problem is considered more complex than other similar problems. The AOA method obtained the best results (1.339074), and almost all the remaining methods got the same results in solving the pressure vessel problem. It is possible to develop an innovative approach for obtaining better results in future work to be clear of improvement compared to previous results.

The results of the selected optimization methods for solving the pressure vessel problem are given in Table 7. The comparative optimization algorithms (i.e., MVO [222], CMVO [222], MFO [223], NSGA-II [224], TLBO [75], WCA [56], and AOA [91]) are utilized to compare this benchmark problem. It is also clear from the results to solve this problem that there are few and insufficient differences based on the results taken from the methods that solve this problem. Some methods obtained better results than others, with meager and insufficient rates. However, it is clear that there is still a need for better approaches to solve this problem and obtain more accurate results for the variables in this problem. The results indicate that this problem needs a powerful way to obtain results different from the comparative results. This problem is considered more complex than other similar problems. It is possible to produce a new innovative optimization method to obtain better results and improve compared to previous work results (Table 8).

The results mentioned in this research included the outputs of the essential research in engineering problems. Therefore, these results will be a vital reference for researchers in this field to prepare future comparisons between the algorithms used to solve such problems. Besides, researchers may benefit from explaining the problems mentioned in this research to prepare their future research. Moreover, based on the results, it is possible to identify areas that need more significant improvement than mentioned.

Table 2 Results of the selected optimization methods for solving the welded beam problem

Algorithm	Parameters values				Optimal
	h	*l*	*t*	*b*	Cost
NDE [158]	0.20572963	3.470488	9.03662	0.205729	1.72485
MTDE [169]	0.205730	3.470489	9.036624	0.205730	1.724852
CSDE [164]	0.2057	3.470	9.037	0.2057	1.7250
SIMPLEX [189]	0.2792	5.6256	7.7512	0.2796	2.5307
APPROX [189]	0.2444	6.2189	8.2915	0.2444	2.3815
RO [190]	0.203687	3.528467	9.004233	0.207241	1.735344
DAVID [189]	0.2434	6.2552	8.2915	0.2444	2.3841
GSA [67]	0.182129	3.856979	10.000	0.202376	1.87995
MVO [186]	0.205463	3.473193	9.044502	0.205695	1.72645
WOA [187]	0.205396	3.484293	9.037426	0.206276	1.730499
CPSO [191]	0.202369	3.544214	9.04821	0.205723	1.72802
CSCA [192]	0.203137	3.542998	9.033498	0.206179	1.733461
HS [193]	0.2442	6.2231	8.2915	0.2400	2.3807
OBSCA [194]	0.230824	3.069152	8.988479	0.208795	1.722315
GA [195]	0.2489	6.1730	8.1789	0.2533	2.4300
AOA [91]	0.194475	2.57092	10.000	0.201827	1.7164
GWO [53]	0.205676	3.478377	9.03681	0.205778	1.72624
PSO [191]	0.202369	3.544214	9.048210	0.205723	1.731485
BA [196]	0.2015	3.562	9.0414	0.2057	1.7312
AO [68]	0.1631	3.3652	9.0202	0.2067	1.6566
CSA [197]	0. 205730	3.470489	9.036624	0.205730	1.724852
BOA-CE [179]	0.205730	3.470481	9.036611	0.205730	1.724854
ETEO	0.205729	3.470494	9.036626	0.205729	1.724853
DAPSO-GA [104]	0.205728318	2.994714573	9.036612	0.205730	1.6600473

5 Discussion

Each paper in this survey produced results that show how continuous algorithms can be helpful in optimization problems. The vast number of published works supports this by adapting the various optimization algorithms for combinatorial engineering problems like a welded beam, tension/compression spring, pressure vessel, 3-bar truss, speed reducer, cantilever beam, and multiple disc clutch brake problem.

While most optimization algorithms in the literature are for continuous domains, they can also solve discrete problems. This seems to be a promising area for analysis, either introducing novel analytical methods or adapting them to particular discrete/combinatorial problems.

While nearly all of the optimization algorithms listed above were created to solve continuous optimization problems, a few were later modified to solve discrete domain problems. The elements within a discrete domain—such as integers or binary digits—accept only distinct, isolated values, unlike the components inside a continuous field, which have the property of changing smoothly. The discrete domain is described by dealing with finite or infinite countable sets. Discrete domain problems have binary and combinatorial implementations.

The most recent works and advances in the advancement and verification of optimization algorithms in solving various famous engineering problems were reviewed in this article. The collected works show that there are many factors to remember for both traditional and improved methods: first, efficiency. According to various reports, the success of optimization methods is one of the reasons for interest in them over other rivals.

The optimization methods produce high-quality results under the same conditions as other methods due to various factors such as dynamic modules, greedy filtering, and multi-phase searching laws. One of the critical reasons that optimization approaches are used in tracked articles is because of this. The need for performance enhancement is the second factor. The author of most variants states that the optimizers' stability needs to be increased to achieve higher levels of convergence and lower stagnation problems. Both population-based approaches must meet this criterion to achieve a more stable equilibrium between local and global search inclinations.

Table 3 Results of the selected optimization methods for solving the tension/compression spring problem

Algorithm	Parameters values			Optimal
	d	*D*	*N*	Weight
ETEO	0.0516991	0.3569643	11.274546	0.012665
NDE [158]	0.0516890	0.356717	11.2889687	0.0126652
BMO-DE [170]	0.05171	0.35651	11.21432	0.012597
MTDE [169]	0.051689	0.356718	11.288965	0.012665
CSDE [164]	0.05168	0.3565	11.30	0.0126997
DE [147]	0.051689	0.35671	11.2889	0.012665
RO [190]	0.051370	0.349096	11.76279	0.0126788
WOA [187]	0.051207	0.345215	12.004032	0.0126763
GSA [67]	0.050276	0.323680	13.525410	0.0127022
ES [198]	0.051643	0.355360	11.397926	0.012698
OBSCA [194]	0.05230	0.31728	12.54854	0.012625
MVO [186]	0.05251	0.37602	10.33513	0.012790
CSCA [192]	0.051609	0.354714	11.410831	0.0126702
CPSO [191]	0.051728	0.357644	11.244543	0.0126747
PSO [191]	0.051728	0.357644	11.244543	0.0126747
CC [199]	70.050000	0.315900	14.250000	0.0128334
BA [196]	0.051690	0.356730	11.288500	0.012670
CSA [197]	0.051689	0.356717	11.289012	0.012665
SSA [50]	0.051207	0.345215	12.004032	0.012676
BOA-CE [179]	0.051618	0.355004	11.390144	0.012665
AO [68]	0.0502439	0.35262	10.5425	0.011165
HS [200]	0.051154	0.349871	12.076432	0.0126706
GA [201]	0.051480	0.351661	11.632201	0.01270478
AOA [91]	0.0500	0.349809	11.8637	0.012124
GWO [53]	0.051690	0.356737	11.288850	0.012666
DAPSO-GA [104]	0.050	0.3744328	8.5465733	0.0098724562

The majority of experiments have improved the equilibrium between the optimization methods' exploratory and exploitative propensities. Until now, the most often upgraded features in literature have been the accuracy of findings and convergence speed. As a result, we saw in the papers we looked at that the authors used various optimization techniques and variants to solve various new problems and datasets. According to the no free lunch (NFL) hypothesis, writers better understand how they adapt algorithm functionality and how some of their improved operations of each process add to the final effects' effectiveness than before 2017. As a result, they may look at various facets of optimization methods' success in dealing with various real-world problems.

Finally, the results of optimization techniques with a deeper evolutionary foundation, such as co-evolutionary technologies, multi-population techniques, memetic methods, and concurrent computation, can be improved further. Such a foundation will also help harmonize global and local patterns, resulting in enhanced optimization process variants.

6 Conclusions and future works

Engineering designs are famous problems in various research domains, such as science, industry, medicine, and engineering. Many optimization algorithms have been used in the literature to handle engineering design problems, such as meta-heuristic optimization methods. It is one of the most comment optimization search algorithms used to solve different problems. However, the algorithm's achievement rapidly reduces with the increase in the problem's dimensional and difficulty. Various versions of the meta-heuristic optimization methods have been proposed to address engineering design problems in the literature efficiently.

In this paper, a comprehensive survey is proposed, which reviews the meta-heuristic optimization methods that have been employed to solve engineering design problems. We used differential, evolution, engineering, design, and problems as search keywords to collect the related works. The state-of-the-art methods are shown in detail over various categories, including basic, modified,

Table 4 Results of the selected optimization methods for solving the pressure vessel problem

Algorithm	Parameters values				Optimal cost
	T_s	T_h	R	L	
NDE [158]	0.81250	0.43750	42.09844	176.63659	6059.714335
BMO-DE [170]	0.81250	0.43750	42.09844	168.73659	5882.77
MTDE [169]	0.778169	0.384649	40.319619	200	5885.3328
CSDE [164]	0.8125	0.4375	0.4210	176.6	6060
DE [147]	0.8125	0.4375	42.0984	176.6365	6059.71433
Branch-bound [205]	1.125	0.625	48.97	106.72	7982.5
MVO [186]	0.8125	0.4375	42.090738	176.73869	6060.8066
HPSO [202]	0.8125	0.4375	42.0984	176.6366	6059.7143
WOA [187]	0.812500	0.437500	42.0982699	176.638998	6059.7410
GSA [67]	1.125	0.625	55.9886598	84.4542025	8538.8359
CSCA [192]	0.8125	0.4375	42.098411	176.63769	6059.7340
ES [198]	0.8125	0.4375	42.098087	176.640518	6059.74560
CPSO [191]	0.8125	0.4375	42.091266	176.7465	6061.0777
ACO [203]	0.812500	0.437500	42.098353	176.637751	6059.7258
PSO-SCA [204]	0.8125	0.4375	42.098446	176.6366	6059.71433
HS [200]	1.125000	0.625000	58.29015	43.69268	7197.730
GA [201]	0.81250	0.43750	42.097398	176.65405	6059.94634
AOA [91]	0.8303737	0.4162057	42.75127	169.3454	6048.7844
AO [68]	1.0540	0.182806	59.6219	38.8050	5949.2258
GWO [53]	0.8125	0.4345	42.089181	176.758731	6051.5639
PSO [191]	0.8125	0.4375	42.091266	176.746500	6061.0777
BA [196]	0.8125	0.4375	42.0984456	176.636596	6059.7143
BOA-CE [179]	0.8125	0.4375	42.0984456	176.6365958	6059.7143
ETEO	0.77816	0.38464	40.31962	199.9999	5885.3341
DAPSO-GA [104]	0.8125	0.4375	42.0984	176.6366	6059.7143
CS [120]	0.7782	0.3831	40.3231	199.9525	5880.87

Table 5 Results of the selected optimization methods for solving the 3-bar truss problem

Algorithm	Parameters values		Optimal weight
	x_1	x_2	
NDE [158]	0.7886753	0.408247767	263.89584
MTDE [169]	0.788675	0.408248	263.895800
CSDE [164]	0.7886	0.4082	263.9
CS [206]	0.78867	0.40902	263.9716
Tsa [207]	0.788	0.408	263.68
DEDS [208]	0.78867513	0.40824828	263.89584
Ray and Sain [209]	0.795	0.395	264.3
MBA [210]	0.7885650	0.4085597	263.89585
SSA [50]	0.78866541	0.408275784	263.89584
PSO-DE [204]	0.7886751	0.4082482	263.89584
AOA [91]	0.79369	0.39426	263.9154
AO [68]	0.7926	0.3966	263.8684
A3 [171]	0.7887354	0.408078	263.895880
ETEO [175]	0.788675	0.408246	263.895843
CS [120]	0.78867	0.40825	263.8958
DAPSO-GA [104]	0.7886769887	0.4082430493	263.89584368

Table 6 Results of the selected optimization methods for solving the speed reducer problem

Algorithm	Parameters values							Optimal weight
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
ETEO	3.499999	0.700000	17	7.300000	7.715319	3.350214	5.286654	2994.471067
NDE [158]	3.500	0.70000	17.0000	7.3000	7.7153199	3.350214	5.286654	2994.47106
CSDE [164]	3.241	0.700	1.700	7.30	7.715	3.350	5.287	2893.0
AO [68]	3.5021	0.7000	17.0000	7.3099	7.7476	3.3641	5.2994	3007.7328
CS [206]	3.5015	0.7000	17	7.6050	7.8181	3.3520	5.2875	3000.9810
GWO [53]	3.501	0.7	17	7.3	7.811013	3.350704	5.287411	2997.81965
AAO [211]	3.4999	0.7	17	7.3	7.8	3.3502	5.2877	2997.058
SCA [212]	3.521	0.7	17	8.3	7.923351	3.355911	5.300734	3026.83772
MFO [213]	3.5	0.7	17	7.3	7.8	3.350215	5.286683	2996.348165
APSO [214]	3.501313	0.7	18	8.127814	8.042121	3.352446	5.287076	3187.630486
WSA [215]	3.500	0.7	17	7.3	7.8	3.350215	5.286683	2996.348225
PSO-DE [204]	3.5	0.7	17	7.3	7.8	3.35021	5.28668	2996.3481
LGSI2 [215]	3.5	0.7	17	7.3	7.8	3.350215	5.286683	2996.348166
LGSI4 [216]	3.501	0.7	17	7.3	7.8	3.350214	5.286683	2996.348205
FA [217]	3.507495	0.7001	17	7.719674	8.080854	3.351512	5.287051	3010.137492
AOA [91]	3.50384	0.7	17	7.3	7.72933	3.35649	5.2867	2997.9157
DAPSO-GA [104]	3.5	0.7	17	7.3	7.71531911	3.350214	5.28665	2994.47107

Table 7 Results of the selected optimization methods for solving the cantilever beam problem

Algorithm	Parameters values					Optimal weight
	x_1	x_2	x_3	x_4	x_5	
AO [68]	5.8881	5.5451	4.3798	3.5973	2.1026	1.3390
CSDE [164]	6.018	5.308	4.496	3.501	2.152	1.340
ALO [218]	6.01812	5.31142	4.48836	3.49751	2.158329	1.33995
CS [206]	6.0089	5.3049	4.5023	3.5077	2.1504	1.3399
MMA [219]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
GCA_I [219]	6.0100	5.30400	4.4900	3.4980	2.1500	1.3400
SOS [220]	6.01878	5.30344	4.49587	3.49896	2.15564	1.33996
GCA_II [219]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
MFO [213]	5.9830	5.3167	4.4973	3.5136	2.1616	1.33998
SMA [221]	6.017757	5.310892	4.493758	3.501106	2.150159	1.33996
AOA [91]	5.88901	5.5399	4.38001	3.6025123	2.103258	1.339074

and hybrid meta-heuristic optimization methods. Moreover, we present the results of the most common state-of-the-art methods in that domain to determine which version of meta-heuristic optimization methods performs more beneficial in solving the considered problems.

Finally, we provide remarkable future research directions for the potential methods and conclude. The meta-heuristic optimization methods can be further modified using another optimization procedure from the recently published algorithms. Several optimization algorithms have been proposed recently, such as Aquila Optimizer, Arithmetic Optimization Algorithm, and Henry Gas Solubility Optimization. Many other engineering problems can be solved using the versions of the meta-heuristic optimization methods, such as multi-objective engineering problems, unconstrained engineering problems, a proportional-integral-derivative controller (PID), Constrained Truss Optimisation Problems, the power loss of large-scale photovoltaic array, power flow problem, parameter estimation of photovoltaic, a cam-follower mechanism, scheduling in cloud computing environments, etc.

This work reports the main vital topics in the engineering and artificial intelligence field. It presents all the published literature methods related to the differential

Table 8 Results of the selected optimization methods for solving the multiple disc clutch brake problem

Algorithm	Parameters values					Optimal weight
	x_1	x_2	x_3	x_4	x_5	
AO [68]	78.4228	98.5674	1	846.8894	2.5294	0.30835
MVO [222]	70	90	1	910	3	0.313656
CMVO [222]	70	90	1	910	3	0.313656
MFO [223]	70	90	1	910	3	0.313656
NSGA-II [224]	70	90	1.5	1000	3	0.470400
TLBO [75]	70	90	1	810	3	0.313657
WCA [56]	70	90	1	910	3	0.313656
AOA [91]	65.9105	90	1	1000	2.42904	0.315588
SHO [188]	69.99	90.00	1.00	999.92	2.31	0.257
DAPSO-GA [104]	70	90	1.0	1000	3	0.31365661

evolution methods in solving various engineering design problems (i.e., welded beam design problem, tension/compression spring design problem, pressure vessel design problem, 3-bar truss design problem, speed reducer design problem, cantilever beam design problem, and multiple disc clutch brake problem). Future research can depend on this review to explore the meta-heuristic optimization methods and engineering design problems. The researchers can find the starting point for future researchers also easier by explore the given studies.

Declaration

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Zheng R, Jia H, Abualigah L, Liu Q, Wang S (2021) Deep ensemble of slime mold algorithm and arithmetic optimization algorithm for global optimization. Processes 9(10):1774
2. Oliva D, Abd El Aziz M, Hassanien AE (2017) Parameter estimation of photovoltaic cells using an improved chaotic whale optimization algorithm. Appl Energy 200:141–154
3. Ewees AA, AbdElaziz M, Houssein EH (2018) Improved grasshopper optimization algorithm using opposition-based learning. Expert Syst Appl 112:156–172
4. Elsheikh AH, Sharshir SW, AbdElaziz M, Kabeel A, Guilan W, Haiou Z (2019) Modeling of solar energy systems using artificial neural network: a comprehensive review. Sol Energy 180:622–639
5. Al-Qaness MA, Ewees AA, Fan H, Abd El Aziz M (2020) Optimization method for forecasting confirmed cases of covid-19 in China. J Clin Med 9(3):674
6. AbdElaziz M, Xiong S, Jayasena K, Li L (2019) Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution. Knowl-Based Syst 169:39–52
7. Alresheedi SS, Lu S, AbdElaziz M, Ewees AA (2019) Improved multiobjective salp swarm optimization for virtual machine placement in cloud computing. HCIS 9(1):1–24
8. Abualigah L, Diabat A (2020) A novel hybrid antlion optimization algorithm for multi-objective task scheduling problems in cloud computing environments. Cluster Comput, pp 1–19
9. Attiya I, Abd Elaziz M, Xiong S (2020) Job scheduling in cloud computing using a modified harris hawks optimization and simulated annealing algorithm. Comput Intell Neurosci
10. Abd Elaziz M, Attiya I (2020) An improved henry gas solubility optimization algorithm for task scheduling in cloud computing. Art Intell Rev, pp 1–39
11. Pan I, AbdElaziz M, Bhattacharyya S (2020) Swarm intelligence for cloud computing. CRC Press, Cambridge
12. Abd Elaziz M, Elsheikh AH, Oliva D, Abualigah L, Lu S, Ewees AA (2021) Advanced metaheuristic techniques for mechanical design problems. Arch Comput Methods Eng, pp 1–22
13. Abd El Aziz M, Ewees AA, Hassanien AE (2017) Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation. Expert Syst Appl 83:242–256
14. AbdElaziz M, Oliva D, Xiong S (2017) An improved opposition-based sine cosine algorithm for global optimization. Expert Syst Appl 90:484–500
15. Abd El Aziz M, Hassanien AE (2018) Modified cuckoo search algorithm with rough sets for feature selection. Neural Comput Appl 29(4):925–934
16. Barshandeh S, Piri F, Sangani SR (2020) Hmpa: an innovative hybrid multi-population algorithm based on artificial ecosystem-based and harris hawks optimization algorithms for engineering problems. Eng Comput, pp 1–45
17. Elsheikh AH, Saba AI, AbdElaziz M, Lu S, Shanmugan S, Muthuramalingam T, Kumar R, Mosleh AO, Essa F, Shehabeldeen TA (2021) Deep learning-based forecasting model for covid-19 outbreak in saudi arabia. Process Saf Environ Prot 149:223–233
18. Dahou A, Xiong S, Zhou J, Elaziz MA (2019) Multi-channel embedding convolutional neural network model for arabic sentiment classification. ACM Trans Asian Low-Resour Language Inf Process 18(4):1–23
19. Dahou A, Elaziz MA, Zhou J, Xiong S (2019) Arabic sentiment classification using convolutional neural network and differential evolution algorithm, Comput Intell Neurosci
20. Abualigah LM, Khader AT (2017) Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering. J Supercomput 73(11):4773–4795

21. Abualigah LM, Khader AT, Hanandeh ES (2018) A new feature selection method to improve the document clustering using particle swarm optimization algorithm. J Comput Sci 25:456–466
22. Abualigah LMQ (2019) Feature selection and enhanced krill herd algorithm for text document clustering. Springer, New York
23. Abualigah LM, Khader AT, Hanandeh ES (2018) Hybrid clustering analysis using improved krill herd algorithm. Appl Intell 48(11):4047–4071
24. Abualigah LM, Khader AT, Hanandeh ES (2018) A combination of objective functions and hybrid krill herd algorithm for text document clustering analysis. Eng Appl Artif Intell 73:111–125
25. Abualigah LM, Khader AT, Hanandeh ES, Gandomi AH (2017) A novel hybridization strategy for krill herd algorithm applied to clustering techniques. Appl Soft Comput 60:423–435
26. Yousri D, AbdElaziz M, Oliva D, Abualigah L, Al-qaness MA, Ewees AA (2020) Reliable applied objective for identifying simple and detailed photovoltaic models using modern metaheuristics: Comparative study. Energy Convers Manage 223:113279
27. Zhang Y, Jin Z, Mirjalili S (2020) Generalized normal distribution optimization and its applications in parameter extraction of photovoltaic models. Energy Convers Manage 224:113301
28. Xiong G, Zhang J, Shi D, He Y (2018) Parameter extraction of solar photovoltaic models using an improved whale optimization algorithm. Energy Convers Manage 174:388–405
29. Li S, Gong W, Yan X, Hu C, Bai D, Wang L, Gao L (2019) Parameter extraction of photovoltaic models using an improved teaching-learning-based optimization. Energy Convers Manage 186:293–305
30. Yousri D, AbdElaziz M, Abualigah L, Oliva D, Al-Qaness MA, Ewees AA (2021) Covid-19 x-ray images classification based on enhanced fractional-order cuckoo search optimizer using heavy-tailed distributions. Appl Soft Comput 101:107052
31. Shao Z, Sumari NS, Portnov A, Ujoh F, Musakwa W, Mandela PJ (2021) Urban sprawl and its impact on sustainable urban development: a combination of remote sensing and social media data. Geo-spatial Inf Sci 24(2):241–255
32. Shao Z, Wu W, Li D (2021) Spatio-temporal-spectral observation model for urban remote sensing. Geo-spatial Inf Sci, pp 1–15
33. Abualigah L, Diabat A (2020) Advances in sine cosine algorithm: a comprehensive survey, Art Intell Rev, pp 1–42
34. Abualigah L (2020) Group search optimizer: a nature-inspired meta-heuristic optimization algorithm with its results, variants, and applications. Neural Comput Appl, pp 1–24
35. Abualigah L, Diabat A (2020) A comprehensive survey of the grasshopper optimization algorithm: results, variants, and applications. Neural Comput Appl, pp 1–24
36. Meraihi Y, Gabis AB, Ramdane-Cherif A, Acheli D (2020) A comprehensive survey of crow search algorithm and its applications. Art Intell Rev, pp 1–48
37. Abualigah L, Diabat A, Geem ZW (2020) A comprehensive survey of the harmony search algorithm in clustering applications. Appl Sci 10(11):3827
38. Abualigah L, Shehab M, Alshinwan M, Alabool H (2019) Salp swarm algorithm: a comprehensive survey. Neural Comput Appl, pp 1–21
39. Abualigah L, Shehab M, Alshinwan M, Mirjalili S, Abd Elaziz M (2020) Ant lion optimizer: a comprehensive survey of its variants and applications. Arch Comput Methods Eng, pp 1–20
40. Hassan MH, Kamel S, Abualigah L, Eid A (2021) Development and application of slime mould algorithm for optimal economic emission dispatch. Expert Syst Appl 182:115205
41. Abualigah L, Diabat A, Elaziz MA (2021) (Improved slime mould algorithm by opposition-based learning and levy flight distribution for global optimization and advances in real-world engineering problems. J Ambient Intell Human Comput, pp 1–40
42. Houssein EH, Dirar M, Abualigah L, Mohamed WM (2011) An efficient equilibrium optimizer with support vector regression for stock market prediction. Neural Comput Appl, pp 1–36
43. Wang S, Liu Q, Liu Y, Jia H, Abualigah L, Zheng R, Wu D (2021) A hybrid SSA and SMA with mutation opposition-based learning for constrained engineering problems. Comput Intell Neurosci
44. Alshinwan M, Abualigah L, Shehab M, Abd Elaziz M, Khasawneh AM, Alabool H, Al Hamad H (2021) Dragonfly algorithm: a comprehensive survey of its results, variants, and applications. Multimed Tools Appl, pp 1–38
45. Abualigah L, Abd Elaziz M, Hussien AG, Alsalibi B, Jalali SMJ, Gandomi AH (2020) Lightning search algorithm: a comprehensive survey. Appl Intell, pp 1–24
46. Abualigah L, Gandomi AH, Elaziz MA, Hamad HA, Omari M, Alshinwan M, Khasawneh AM (2021) Advances in meta-heuristic optimization algorithms in big data text clustering. Electronics 10(2):101
47. Abualigah L, Gandomi AH, Elaziz MA, Hussien AG, Khasawneh AM, Alshinwan M, Houssein EH (2020) Nature-inspired optimization algorithms for text document clustering-a comprehensive analysis. Algorithms 13(12):345
48. Samuel P, Subbaiyan S, Balusamy B, Doraikannan S, Gandomi AH (2021) A technical survey on intelligent optimization grouping algorithms for finite state automata in deep packet inspection. Arch Comput Methods Eng 28(3):1371–1396
49. Sharma M, Kaur P (2020) A comprehensive analysis of nature-inspired meta-heuristic techniques for feature selection problem. Arch Comput Methods Eng, pp 1–25
50. 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
51. Jain M, Singh V, Rani A (2018) A novel nature-inspired algorithm for optimization: Squirrel search algorithm. Swarm Evol Comput
52. Gandomi AH, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simul 17(12):4831–4845
53. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
54. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory, in: Micro Machine and Human Science, 1995. MHS'95. In: Proceedings of the sixth international symposium on, IEEE, pp 39–43
55. Kashani AR, Chiong R, Mirjalili S, Gandomi AH (2020) Particle swarm optimization variants for solving geotechnical problems: review and comparative analysis. Arch Comput Methods Eng, pp 1–57
56. 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 110:151–166
57. Bayraktar Z, Komurcu M, Werner DH (2010) Wind driven optimization (wdo): a novel nature-inspired optimization algorithm and its application to electromagnetics In: Antennas and propagation society international symposium (APSURSI), 2010 IEEE, pp 1–4
58. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680

59. Safaldin M, Otair M, Abualigah L (2020) Improved binary gray wolf optimizer and svm for intrusion detection system in wireless sensor networks. J Ambient Intell Human Comput, pp 1–18
60. Ewees AA, Abualigah L, Yousri D, Algamal ZY, Al-qaness MA, Ibrahim RA, Abd Elaziz M (2021) Improved slime mould algorithm based on firefly algorithm for feature selection: A case study on qsar model. Eng Comput, pp 1–15
61. Şahin CB, Dinler ÖB, Abualigah L (2021) Prediction of software vulnerability based deep symbiotic genetic algorithms: phenotyping of dominant-features. Appl Intell, pp 1–17
62. Ezugwu AE, Prayogo D (2020) Symbiotic organisms search algorithm: theory, recent advances and applications. Exp Syst Appl
63. Shah-Hosseini H (2009) The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm. Int J Bio-inspir Comput 1(1–2):71–79
64. Krishnanand K, Ghose D (2006) Glowworm swarm based optimization algorithm for multimodal functions with collective robotics applications. Multiag Grid Syst 2(3):209–222
65. Alsalibi B, Abualigah L, Khader AT (2020) A novel bat algorithm with dynamic membrane structure for optimization problems. Appl Intell, pp 1–26
66. Črepinšek M, Liu S-H, Mernik L (2012) A note on teaching-learning-based optimization algorithm. Inf Sci 212:79–93
67. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) Gsa: a gravitational search algorithm. Inf Sci 179(13):2232–2248
68. Abualigah L, Yousri D, Abd Elaziz M, Ewees AA, Al-qaness MA, Gandomi AH (2021) Aquila optimizer: a novel meta-heuristic optimization algorithm. Comput Ind Eng 107250
69. Lam AY, Li VO (2010) Chemical-reaction-inspired metaheuristic for optimization. IEEE Trans Evol Comput 14(3):381–399
70. Birbil Şİ, Fang S-C (2003) An electromagnetism-like mechanism for global optimization. J Global Optim 25(3):263–282
71. Ibrahim RA, Abualigah L, Ewees AA, Al-Qaness MA, Yousri D, Alshathri S, AbdElaziz M (2021) An electric fish-based arithmetic optimization algorithm for feature selection. Entropy 23(9):1189
72. Wang S, Jia H, Abualigah L, Liu Q, Zheng R (2021) An improved hybrid aquila optimizer and harris hawks algorithm for solving industrial engineering optimization problems. Processes 9(9):1551
73. Gul F, Mir I, Abualigah L, Sumari P (2021) Multi-robot space exploration: An augmented arithmetic approach. IEEE Access 9:107738–107750
74. Dai C, Zhu Y, Chen W (2006) Seeker optimization algorithm. In: International conference on computational and information science. Springer, pp 167–176
75. Rao RV, Savsani VJ, Vakharia D (2011) Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des 43(3):303–315
76. Moghdani R, Salimifard K (2018) Volleyball premier league algorithm. Appl Soft Comput 64:161–185
77. Moosavi SHS, Bardsiri VK (2019) Poor and rich optimization algorithm: a new human-based and multi populations algorithm. Eng Appl Artif Intell 86:165–181
78. Khishe M, Mosavi MR (2020) Chimp optimization algorithm. Expert Syst Appl 149:113338
79. Mousavirad SJ, Ebrahimpour-Komleh H (2017) Human mental search: a new population-based metaheuristic optimization algorithm. Appl Intell 47(3):850–887
80. Kashan AH (2009) League championship algorithm: a new algorithm for numerical function optimization. In: International conference of soft computing and pattern recognition. IEEE, pp 43–48
81. Coello Coello CA, Becerra RL (2004) Efficient evolutionary optimization through the use of a cultural algorithm. Eng Optim 36(2):219–236
82. Tzanetos A, Dounias G (2021) Nature inspired optimization algorithms or simply variations of metaheuristics? Artif Intell Rev 54(3):1841–1862
83. Bentéjac C, Csörgő A, Martínez-Muñoz G (2021) A comparative analysis of gradient boosting algorithms. Artif Intell Rev 54(3):1937–1967
84. Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. IEEE Trans Evol Comput 3(2):82–102
85. Holland J (2021) Adaptation in artificial and natural systems. Ann Arbor: The University of Michigan Press
86. Goldberg DE, Holland JH (1988) Genetic algorithms and machine learning. Mach Learn 3(2):95–99
87. Koza JR (1994) Genetic programming as a means for programming computers by natural selection. Stat Comput 4(2):87–112
88. Storn R, Price K (1997) Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11(4):341–359
89. Gandomi AH (2014) Interior search algorithm (isa): a novel approach for global optimization. ISA Trans 53(4):1168–1183
90. Michalewicz Z (1996) Evolution strategies and other methods. In: Genetic algorithms+ data structures= evolution programs. Springer, pp 159–177
91. Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH (2021) The arithmetic optimization algorithm. Comput Methods Appl Mech Eng 376:113609
92. Premkumar M, Jangir P, Kumar BS, Sowmya R, Alhelou HH, Abualigah L, Yildiz AR, Mirjalili S (2021) A new arithmetic optimization algorithm for solving real-world multiobjective cec-2021 constrained optimization problems: diversity analysis and validations, IEEE Access
93. Nadimi-Shahraki MH, Taghian S, Mirjalili S (2021) An improved grey wolf optimizer for solving engineering problems. Expert Syst Appl 166:113917
94. Zamani H, Nadimi-Shahraki MH, Gandomi AH (2021) Qana: Quantum-based avian navigation optimizer algorithm. Eng Appl Artif Intell 104:104314
95. Gandomi AH, Deb K (2020) Implicit constraints handling for efficient search of feasible solutions. Comput Methods Appl Mech Eng 363:112917
96. Fesanghary M, Mahdavi M, Minary-Jolandan M, Alizadeh Y (2008) Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems. Comput Methods Appl Mech Eng 197(33–40):3080–3091
97. Gholizadeh S, Salajegheh E (2009) Optimal design of structures subjected to time history loading by swarm intelligence and an advanced metamodel. Comput Methods Appl Mech Eng 198(37–40):2936–2949
98. Safaldin M, Otair M, Abualigah L (2021) Improved binary gray wolf optimizer and svm for intrusion detection system in wireless sensor networks. J Ambient Intell Humaniz Comput 12(2):1559–1576
99. Kaveh A, Talatahari S, Khodadadi N (2020) Stochastic paint optimizer: theory and application in civil engineering. Eng Comput, pp 1–32
100. Kaveh A, Eslamlou AD, Khodadadi N (2020) Dynamic water strider algorithm for optimal design of skeletal structures. Periodica Polytech Civ Eng 64(3):904–916
101. Liu H, Wang Y, Tu L, Ding G, Hu Y (2019) A modified particle swarm optimization for large-scale numerical optimizations and engineering design problems. J Intell Manuf 30(6):2407–2433

102. Tam JH, Ong ZC, Ismail Z, Ang BC, Khoo SY (2019) A new hybrid ga- aco- pso algorithm for solving various engineering design problems. Int J Comput Math 96(5):883–919
103. Belkourchia Y, Azrar L, Zeriab E-SM (2019) A hybrid optimization algorithm for solving constrained engineering design problems. In: 2019 5th international conference on optimization and applications (ICOA), IEEE, pp 1–7
104. Zhu H, Hu Y, Zhu W (2019) A dynamic adaptive particle swarm optimization and genetic algorithm for different constrained engineering design optimization problems. Adv Mech Eng 11(3):1687814018824930
105. Fakhouri HN, Hudaib A, Sleit A (2020) Hybrid particle swarm optimization with sine cosine algorithm and nelder-mead simplex for solving engineering design problems. Arab J Sci Eng 45(4):3091–3109
106. Liu Z, Nishi T (2020) Multipopulation ensemble particle swarm optimizer for engineering design problems. Math Problems Eng
107. Abualigah L, Shehab M, Diabat A, Abraham A (2020) Selection scheme sensitivity for a hybrid salp swarm algorithm: analysis and applications. Eng Comput, pp 1–27
108. Dhiman G, Garg M (2020) Mosse: a novel hybrid multi-objective meta-heuristic algorithm for engineering design problems. Soft Comput 24(24):18379–18398
109. Jangir P, Jangir N (2021) Non-dominated sorting whale optimization algorithm (nswoa): a multi-objective optimization algorithm for solving engineering design problems. Glob J Res Eng
110. Chen H, Xu Y, Wang M, Zhao X (2019) A balanced whale optimization algorithm for constrained engineering design problems. Appl Math Model 71:45–59
111. Abdel-Basset M, Mohamed R, Mirjalili S (2021) A novel whale optimization algorithm integrated with nelder-mead simplex for multi-objective optimization problems. Knowl-Based Syst 212:106619
112. Zhang Y, Jin Z, Chen Y (2020) Hybrid teaching-learning-based optimization and neural network algorithm for engineering design optimization problems. Knowl-Based Syst 187:104836
113. Li Z, Zhang X, Qin J, He J (2020) A reformative teaching-learning-based optimization algorithm for solving numerical and engineering design optimization problems. Soft Comput 24(20):15889–15906
114. Sharma TK, Pant M (2015) Improved search mechanism in abc and its application in engineering design problems. J Eng Sci Technol 10(1):111–133
115. Mollinetti MAF, Souza DL, Pereira RL, Yasojima EKK, Teixeira ON(2016) Abc+ es: combining artificial bee colony algorithm and evolution strategies on engineering design problems and benchmark functions. In: International conference on hybrid intelligent systems. Springer, pp 53–66
116. Gebreslassie BH, Diwekar UM (2017) Homogenous multi-agent optimization for process systems engineering problems with a case study of computer aided molecular design. Chem Eng Sci 159:194–206
117. Dhouib S, Dhouib S, Chabchoub H (2016) Enriched artificial bee colony metaheuristic for hierarchical goal programming engineering design problems. Int J Metah 5(3–4):173–192
118. Sharma TK, Abraham A (2020) Artificial bee colony with enhanced food locations for solving mechanical engineering design problems. J Ambient Intell Humaniz Comput 11(1):267–290
119. Mlakar U (2016) Hybrid cuckoo search for constraint engineering design optimization problems. In: Proceedings of StuCoSReC, pp 57–60
120. Pauline O, Sin HC, Sheng DDCV, Kiong SC, Meng OK (2017) Design optimization of structural engineering problems using adaptive cuckoo search algorithm. In: 2017 3rd international conference on control, automation and robotics (ICCAR), IEEE, pp 745–748
121. Pathak VK, Srivastava AK (2020) A novel upgraded bat algorithm based on cuckoo search and sugeno inertia weight for large scale and constrained engineering design optimization problems. Eng Comput, pp 1–28
122. Kasdirin HA, Yahya NM, Tokhi MO (2015) Hybridizing firefly algorithm with invasive weed optimization for engineering design problems. In: 2015 IEEE international conference on evolving and adaptive intelligent systems (EAIS), IEEE, 2015, pp 1–6
123. Francisco RB, Costa MFP, Rocha AMA (2015) A firefly dynamic penalty approach for solving engineering design problems. In: AIP conference proceedings, Vol. 1648, AIP Publishing LLC, 2015, p. 140010
124. Du T-S, Ke X-T, Liao J-G, Shen Y-J (2018) Dslc-foa: improved fruit fly optimization algorithm for application to structural engineering design optimization problems. Appl Math Model 55:314–339
125. Balande U, Shrimankar D (2019) Srifa: stochastic ranking with improved-firefly-algorithm for constrained optimization engineering design problems. Mathematics 7(3):250
126. Sanabria A, Soh B, Dillon T, Chang E (2021) Genetic algorithms for constrained optimisation problems in web engineering design
127. Basak R, Sanyal A, Das A, Ghosh A, Poddar A (2021) Performance analysis of genetic algorithm as a stochastic optimization tool in engineering design problems
128. Yan X, Liu H, Zhu Z, Wu Q (2017) Hybrid genetic algorithm for engineering design problems. Clust Comput 20(1):263–275
129. Xia Y, Liu X, Du G (2018) Solving bi-level optimization problems in engineering design using kriging models. Eng Optim 50(5):856–876
130. Rather SA, Bala PS (2021) Swarm-based chaotic gravitational search algorithm for solving mechanical engineering design problems, World J Eng
131. Rather SA, Bala PS (2019) Hybridization of constriction coefficient-based particle swarm optimization and chaotic gravitational search algorithm for solving engineering design problems. In: International conference on advanced communication and networking, Springer, 2019, pp 95–115
132. Tawhid MA, Savsani V (2019) Multi-objective sine-cosine algorithm (mo-sca) for multi-objective engineering design problems. Neural Comput Appl 31(2):915–929
133. Rizk-Allah RM (2018) Hybridizing sine cosine algorithm with multi-orthogonal search strategy for engineering design problems. J Comput Des Eng 5(2):249–273
134. El-Shorbagy M, Farag M, Mousa A, El-Desoky I (2019) A hybridization of sine cosine algorithm with steady state genetic algorithm for engineering design problems. In: International conference on advanced machine learning technologies and applications. Springer, pp 143–155
135. Kumar V, Kumar D (2017) An astrophysics-inspired grey wolf algorithm for numerical optimization and its application to engineering design problems. Adv Eng Softw 112:231–254
136. Gupta S, Deep K, Moayedi H, Foong LK, Assad A (2020) Sine cosine grey wolf optimizer to solve engineering design problems. Eng Comput, pp 1–27
137. Li Z, Zhou Y, Zhang S, Song J (2016) Lévy-flight moth-flame algorithm for function optimization and engineering design problems. Math Prob Eng
138. Jangir P, Trivedi IN (2018) Non-dominated sorting moth flame optimizer: a novel multi-objective optimization algorithm for solving engineering design problems. Eng Technol Open Access J, pp 1–15

139. Sayed GI, Hassanien AE (2018) A hybrid sa-mfo algorithm for function optimization and engineering design problems. Compl Intell Syst 4(3):195–212
140. Shehab M, Alshawabkah H, Abualigah L, Nagham A-M (2020) Enhanced a hybrid moth-flame optimization algorithm using new selection schemes. Eng Comput, pp 1–26
141. Abualigah L, Diabat A, Sumari P, Gandomi AH (2021) A novel evolutionary arithmetic optimization algorithm for multilevel thresholding segmentation of covid-19 ct images. Processes 9(7):1155
142. Shen A, Li J (2015) A fast differential evolution for constrained optimization problems in engineering design. In: Bio-inspired computing-theories and applications. Springer, pp 362–377
143. Karaboğa D, Ökdem S (2004) A simple and global optimization algorithm for engineering problems: differential evolution algorithm. Turkish J Electric Eng Comput Sci 12(1):53–60
144. Kim H-K, Chong J-K, Park K-Y, Lowther DA (2007) Differential evolution strategy for constrained global optimization and application to practical engineering problems. IEEE Trans Magn 43(4):1565–1568
145. Ali M, Pant M, Abraham A (2009) A modified differential evolution algorithm and its application to engineering problems. In: International conference of soft computing and pattern recognition. IEEE pp 196–201
146. Price W (1983) Global optimization by controlled random search. J Optim Theory Appl 40(3):333–348
147. de Melo VV, Carosio GLC (2012) Evaluating differential evolution with penalty function to solve constrained engineering problems. Expert Syst Appl 39(9):7860–7863
148. Ponsich A, Coello CC (2011) Differential evolution performances for the solution of mixed-integer constrained process engineering problems. Appl Soft Comput 11(1):399–409
149. Xiao J, He J-J, Chen P, Niu Y-Y (2016) An improved dynamic membrane evolutionary algorithm for constrained engineering design problems. Nat Comput 15(4):579–589
150. Ao Y-Y, Chi H-Q et al (2010) An adaptive differential evolution algorithm to solve constrained optimization problems in engineering design. Engineering 2(01):65
151. Ali M, Pant M, Singh V (2010) Two modified differential evolution algorithms and their applications to engineering design problems. World J Model Simul 6(1):72–80
152. Azad MAK, Fernandes EM (2011) Modified differential evolution based on global competitive ranking for engineering design optimization problems. In: International conference on computational science and its applications. Springer, pp 245–260
153. De Melo VV, Carosio GL (2013) Investigating multi-view differential evolution for solving constrained engineering design problems. Expert Syst Appl 40(9):3370–3377
154. Bui T, Pham H, Hasegawa H (2013) Improve self-adaptive control parameters in differential evolution for solving constrained engineering optimization problems. J Comput Sci Technol 7(1):59–74
155. Muangkote N, Photong L, Sukprasert A (2018) Comparative study of constrained handling techniques of constrained differential evolution algorithms applied to constrained optimization problems in mechanical engineering. In 3rd Technology innovation management and engineering science international conference (TIMES-iCON). IEEE, pp 1–5
156. Gong W, Cai Z, Liang D (2014) Engineering optimization by means of an improved constrained differential evolution. Comput Methods Appl Mech Eng 268:884–904
157. Tsai J-T (2015) Improved differential evolution algorithm for nonlinear programming and engineering design problems. Neurocomputing 148:628–640
158. Mohamed AW (2018) A novel differential evolution algorithm for solving constrained engineering optimization problems. J Intell Manuf 29(3):659–692
159. Mohamed AW, Mohamed AK, Elfeky EZ, Saleh M (2019) Enhanced directed differential evolution algorithm for solving constrained engineering optimization problems. Int J Appl Metah Comput 10(1):1–28
160. Kizilay D, Tasgetiren MF, Oztop H, Kandiller L, Suganthan P (2020) A differential evolution algorithm with q-learning for solving engineering design problems. In: IEEE congress on evolutionary computation (CEC). IEEE pp 1–8
161. Sun P, Liu H, Zhang Y, Tu L, Meng Q (2021) An intensify atom search optimization for engineering design problems. Appl Math Model 89:837–859
162. Li J (2009) A hybrid differential evolution method for practical engineering problems. In: 2009 IITA international conference on control, automation and systems engineering (case 2009), IEEE, 2009, pp 54–57
163. Bai L, Wang J, Jiang Y, Huang D (2012) Improved hybrid differential evolution-estimation of distribution algorithm with feasibility rules for nlp/minlp engineering optimization problems. Chin J Chem Eng 20(6):1074–1080
164. Zhang Z, Ding S, Jia W (2019) A hybrid optimization algorithm based on cuckoo search and differential evolution for solving constrained engineering problems. Eng Appl Artif Intell 85:254–268
165. Parouha RP (2018) An efficient differential evolution for engineering design problems. Int J Appl Eng Res 13(12):10845–10854
166. Aliniya Z, Keyvanpour MR (2019) Cb-ica: a crossover-based imperialist competitive algorithm for large-scale problems and engineering design optimization. Neural Comput Appl 31(11):7549–7570
167. Balakrishnan N (2019) Developing a framework to evaluate individual learning in engineering design problems–part 2: assessment of individual learning in team environments. In: Proceedings of the canadian engineering education association (CEEA)
168. Samma H, Mohamad-Saleh J, Suandi SA, Lahasan B (2020) Q-learning-based simulated annealing algorithm for constrained engineering design problems. Neural Comput Appl 32(9):5147–5161
169. Nadimi-Shahraki MH, Taghian S, Mirjalili S, Faris H (2020) Mtde: An effective multi-trial vector-based differential evolution algorithm and its applications for engineering design problems. Appl Soft Comput 97:106761
170. Sadeeq H, Abdulazeez A, Kako N, Abrahim A (2017) A novel hybrid bird mating optimizer with differential evolution for engineering design optimization problems. In: International conference of reliable information and communication technology. Springer, pp 522–534
171. Yildirim AE, Karci A (2018) Application of three bar truss problem among engineering design optimization problems using artificial atom algorithm. In: 2018 international conference on artificial intelligence and data processing (IDAP), IEEE, pp 1–5
172. Tawhid MA, Savsani V (2018) A novel multi-objective optimization algorithm based on artificial algae for multi-objective engineering design problems. Appl Intell 48(10):3762–3781
173. İç YT (2016) Development of a new multi-criteria optimization method for engineering design problems. Res Eng Des 27(4):413–436
174. Chagwiza G, Jones B, Hove-Musekwa S, Mtisi S (2018) A new hybrid matheuristic optimization algorithm for solving design and network engineering problems. Int J Manage Sci Eng Manag 13(1):11–19

175. Azqandi MS, Delavar M, Arjmand M (2020) An enhanced time evolutionary optimization for solving engineering design problems. Eng Comput 36(2):763–781
176. Rahman TA, Jalil NA, As'arry A, Ahmad RR (2021) Performance evaluation of chaos-enhanced stochastic fractal search algorithm using constrained engineering design problems
177. Bilel N, Mohamed N, Zouhaier A, Lotfi R (2019) An efficient evolutionary algorithm for engineering design problems. Soft Comput 23(15):6197–6213
178. Arora S, Anand P (2018) Learning automata-based butterfly optimization algorithm for engineering design problems. Int J Comput Mater Sci Eng 7(04):1850021
179. Li G, Shuang F, Zhao P, Le C (2019) An improved butterfly optimization algorithm for engineering design problems using the cross-entropy method. Symmetry 11(8):1049
180. Hayyolalam V, Kazem AAP (2020) Black widow optimization algorithm: A novel meta-heuristic approach for solving engineering optimization problems. Eng Appl Artif Intell 87:103249
181. Shaheen A, Elsayed A, El-Sehiemy RA, Abdelaziz AY (2021) Equilibrium optimization algorithm for network reconfiguration and distributed generation allocation in power systems. Appl Soft Comput 98:106867
182. Talatahari S, Azizi M (2020) Optimization of constrained mathematical and engineering design problems using chaos game optimization. Comput Ind Eng 145:106560
183. Ustun D, Carbas S, Toktas A (2021) A symbiotic organisms search algorithm-based design optimization of constrained multi-objective engineering design problems. Eng Comput
184. Hashim FA, Houssein EH, Mabrouk MS, Al-Atabany W, Mirjalili S (2019) Henry gas solubility optimization: a novel physics-based algorithm. Futur Gener Comput Syst 101:646–667
185. Kaleka KK, Kaur A, Kumar V (2020) A conceptual comparison of metaheuristic algorithms and applications to engineering design problems. Int J Intell Inf Database Syst 13(2–4):278–306
186. Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl 27(2):495–513
187. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
188. Dhiman G, Kaur A (2017) Spotted hyena optimizer for solving engineering design problems. In: international conference on machine learning and data science (MLDS). IEEE, pp 114–119
189. Ragsdell K, Phillips D (2021) Optimal design of a class of welded structures using geometric programming
190. Kaveh A, Khayatazad M (2012) A new meta-heuristic method: ray optimization. Comput Struct 112:283–294
191. He Q, Wang L (2007) An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng Appl Artif Intell 20(1):89–99
192. Huang F-Z, Wang L, He Q (2007) An effective co-evolutionary differential evolution for constrained optimization. Appl Math Comput 186(1):340–356
193. Lee KS, Geem ZW (2005) A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput Methods Appl Mech Eng 194(36–38):3902–3933
194. Elaziz MA, Oliva D, Xiong S (2017) An improved opposition-based sine cosine algorithm for global optimization. Expert Syst Appl 90:484–500
195. Deb K (1991) Optimal design of a welded beam via genetic algorithms. AIAA J 29(11):2013–2015
196. Gandomi AH, Yang X-S, Alavi AH, Talatahari S (2013) Bat algorithm for constrained optimization tasks. Neural Comput Appl 22(6):1239–1255
197. Askarzadeh A (2016) A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. Comput Struct 169:1–12
198. Mezura-Montes E, Coello CAC (2008) An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. Int J Gen Syst 37(4):443–473
199. Arora JS (2004) Introduction to optimum design. Elsevier, New York
200. Mahdavi M, Fesanghary M, Damangir E (2007) An improved harmony search algorithm for solving optimization problems. Appl Math Comput 188(2):1567–1579
201. Coello CAC (2000) Use of a self-adaptive penalty approach for engineering optimization problems. Comput Ind 41(2):113–127
202. He Q, Wang L (2007) A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. Appl Math Comput 186(2):1407–1422
203. Kaveh A, Talatahari S (2021) An improved ant colony optimization for constrained engineering design problems. Eng Comput
204. Liu H, Cai Z, Wang Y (2010) Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. Appl Soft Comput 10(2):629–640
205. Sandgren E (2021) Nonlinear integer and discrete programming in mechanical design optimization
206. Gandomi AH, Yang X-S, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 29(1):17–35
207. Tsai J-F (2005) Global optimization of nonlinear fractional programming problems in engineering design. Eng Optim 37(4):399–409
208. Zhang M, Luo W, Wang X (2008) Differential evolution with dynamic stochastic selection for constrained optimization. Inf Sci 178(15):3043–3074
209. Ray T, Saini P (2001) Engineering design optimization using a swarm with an intelligent information sharing among individuals. Eng Optim 33(6):735–748
210. 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 13(5):2592–2612
211. Czerniak JM, Zarzycki H, Ewald D (2017) Aao as a new strategy in modeling and simulation of constructional problems optimization. Simul Model Pract Theory 76:22–33
212. Mirjalili S (2016) Sca: a sine cosine algorithm for solving optimization problems. Knowl-Based Syst 96:120–133
213. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl-Based Syst 89:228–249
214. Guedria NB (2016) Improved accelerated pso algorithm for mechanical engineering optimization problems. Appl Soft Comput 40:455–467
215. Baykasoğlu A, Akpinar Ş (2015) Weighted superposition attraction (wsa): a swarm intelligence algorithm for optimization problems-part 2: Constrained optimization. Appl Soft Comput 37:396–415
216. Brancato V, Calabrese L, Palomba V, Frazzica A, Fullana-Puig M, Solé A, Cabeza LF (2018) Mgso4· 7h2o filled macro cellular foams: An innovative composite sorbent for thermo-chemical energy storage applications for solar buildings. Sol Energy 173:1278–1286
217. Baykasoğlu A, Ozsoydan FB (2015) Adaptive firefly algorithm with chaos for mechanical design optimization problems. Appl Soft Comput 36:152–164
218. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98

219. Chickermane H, Gea H (1996) Structural optimization using a new local approximation method. Int J Numer Meth Eng 39(5):829–846

220. Cheng M-Y, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. Comput Struct 139:98–112

221. 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

222. Sayed GI, Darwish A, Hassanien AE (2018) A new chaotic multi-verse optimization algorithm for solving engineering optimization problems. J Exp Theoret Art Intell 30(2):293–317

223. Bhesdadiya R, Trivedi IN, Jangir P, Jangir N (2018) Moth-flame optimizer method for solving constrained engineering optimization problems. In: Advances in computer and computational sciences. Springer, pp 61–68

224. Deb K, Srinivasan A (2008) Innovization: discovery of innovative design principles through multiobjective evolutionary optimization. In: Multiobjective problem solving from nature. Springer, pp 243–262

Authors and Affiliations

Laith Abualigah[1,2] · **Mohamed Abd Elaziz**[3,8,9,10] · **Ahmad M. Khasawneh**[1] · **Mohammad Alshinwan**[1] · **Rehab Ali Ibrahim**[3] · **Mohammed A. A. Al-qaness**[4] · **Seyedali Mirjalili**[5,6] · **Putra Sumari**[2] · **Amir H. Gandomi**[7]

✉ Laith Abualigah
laythdyabat@aau.edu.jo; Aligah.2020@gmail.com

Mohamed Abd Elaziz
abd_el_aziz_m@yahoo.com

Ahmad M. Khasawneh
a.khasawneh@aau.edu.jo

Mohammad Alshinwan
mohmdsh@aau.edu.jo

Mohammed A. A. Al-qaness
alqaness@yahoo.com

Seyedali Mirjalili
ali.mirjalili@torrens.edu.au

Putra Sumari
putras@usm.my

Amir H. Gandomi
Gandomi@uts.edu.au

[1] Faculty of Computer Sciences and Informatics, Amman Arab University, Amman, Jordan

[2] School of Computer Science, Universiti Sains Malaysia, Penang, Malaysia

[3] Department of Mathematics, Faculty of Science, Zagazig University, Zagazig 44519, Egypt

[4] Faculty of Engineering, Sana'a University, Sana'a 12544, Yemen

[5] Centre for Artificial Intelligence Research and Optimization, Torrens University Australia, 90 Bowen Terrace, Fortitude Valley, QLD 4006, Australia

[6] Yonsei Frontier Lab, Yonsei University, Seoul, Korea

[7] Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia

[8] Artificial Intelligence Research Center (AIRC), Ajman University, Ajman 346, United Arab Emirates

[9] Faculty of Computer Science & Engineering, Galala University, Suze 435611, Egypt

[10] School of Computer Science and Robotics, Tomsk Polytechnic University, Tomsk 634050, Russia