Journal of Mathematics and Statistics Studies
ISSN: 2709-4200
DOI: 10.32996/jmss
Journal Homepage: www.al-kindipublisher.com/index.php/jmss

JMSS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

A Comparative Study of Metaheuristic Optimization Algorithms for Solving Engineering Design Problems

Ghulam Ali Sabery[1] ✉ Ghulam Hassan Danishyar[2] and Ghulam Sarwar Mubarez[3]
[12]Balkh University, Assistant Professor, Faculty of Engineering, Balkh, Afghanistan
[3]Balkh University, Assistant Professor, Faculty of Science, Balkh, Afghanistan
Corresponding Author: Ghulam Ali Sabery, **E-mail**: gh.alisabery@gmail.com

| ABSTRACT
Metaheuristic optimization algorithms (Nature-Inspired Optimization Algorithms) are a class of algorithms that mimic the behavior of natural systems such as evolution process, swarm intelligence, human activity and physical phenomena to find the optimal solution. Since the introduction of meta-heuristic optimization algorithms, they have shown their profound impact in solving the high-scale and non-differentiable engineering problems. This paper presents a comparative study of the most widely used nature-inspired optimization algorithms for solving engineering classical design problems, which are considered challenging. The teen metaheuristic algorithms employed in this study are, namely, Artificial Bee Colony (ABC), Ant Colony Optimization (ACO), Biogeography Based Optimization Algorithm (BBO), Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES), Cuckoo Search algorithm (CS), Differential Evolution (DE), Genetic Algorithm (GA), Grey Wolf Optimizer (GWO), Gravitational Search Algorithm (GSA) and Particle Swarm Optimization (PSO). The efficiency of these algorithms is evaluated on teen popular engineering classical design problems using the solution quality and convergence analysis, which verify the applicability of these algorithms to engineering classical constrained design problems. The experimental results demonstrated that all the algorithms provide a competitive solution.

| KEYWORDS

Optimization, Nature Inspired algorithm, Meta-Heuristic Algorithm, Engineering Design Problem, Population Based Algorithm.

| ARTICLE INFORMATION

ACCEPTED: 20 October 2023 **PUBLISHED:** 04 November 2023 **DOI:** 10.32996/jmss.2023.4.4.6

1. Introduction

Optimization refers to the process of finding the best possible solution for a problem or achieving the highest level of performance within given constraints. It involves maximizing or minimizing an objective function by adjusting variables or parameters. Optimization is paramount in many applications, such as engineering, economics, computer science, business activities and industrial designs. The optimization algorithms aim to find the optimal solution that maximises efficiency, effectiveness, profitability and minimises energy consumption and costs. Due to incompetency of classical optimization algorithms in solving real world optimization problems, which are non- differentiable, large –scale and highly non-linear, there is a need to develop robust, efficient and problems characteristics free computational algorithms, that can solve problems numerically irrespective of their particular characteristics. Drawing inspiration from nature to develop computationally efficient algorithms is one way to deal with real-world optimization problems and this kind of algorithm is called Nature-Inspired Optimization Algorithms (Meta-Heuristic Algorithms). Meta-Heuristic algorithms typically start with an initial solution and iteratively improve it by exploring the search space using a set of heuristic rules. These rules guide the search process towards the promising regions of the search space, allowing the algorithm to escape local optima and find globally optimal or near-optimal solutions [Yang, 2020]. Rechenberg and Schwefel introduced the first meta-heuristic optimization algorithm in the early 1960s. They developed the (1+1)-ES algorithm which is inspired by Darwinian evolution theory [Bansal, 2019]. Since then, meta-heuristic algorithms have attracted a lot of attention from researchers worldwide and many optimization algorithms have been developed. The most popular meta-heuristic algorithms are such as Evolutionary Strategy (ES) [Rechenberg, 1997], Genetic Algorithm (GA) [Holland, 1993] and Differential Evolution (DE) [Storn, 1997] inspired from Darwinian theory, Particle Swarm Optimization (PSO) [Kennedy, 1995] inspired from foraging behavior of birds flocking or schooling of fish, Ant Colony Optimization Algorithm (ACO) [Dorigo, 2004] inspired from foraging behavior of ant, Artificial Bee Colony algorithm (ABC) [Karaboga, 2005] inspired from foraging behavior of honey bee, Spider Monkey Optimization algorithm (SMO) [Bansal, 2014] inspired from foraging behavior of spider monkey, Grey Wolf Optimizer (GWO) inspired by foraging behavior of grey wolfs [Mirjalili, 2014], Biogeography-Based Optimization (BBO) [Simon, 2008] inspired from emigration/immigration of individuals from one islands to another, Firefly Algorithm (FA) [Yang, 2010, Yang, 2009], Cuckoo Search (CS) algorithm [Yang, 2010, Yang, 2009], Cuckoo Optimization Algorithm (COA) [Rajabioun, 2011], Gravitational Search Algorithm (GSA) [Rashedi, 2009],Teaching-Learning Based Optimization algorithm (TLBO) [Rao, 2012] inspired from the influence of teacher on learners and so on. Recently a few mathematically inspired Meta- heuristics optimization algorithms have also been developed. These algorithms adopt geometric, trigonometric and analytical functions in their search equations to direct the solution toward a promising area of search space. There are some of them, such as Sine-Cosine Optimization algorithm (SCA) [Mirjalili, 2016], Spherical Search Optimizer (SSO) [Zhao, 2020], The Arithmetic Optimization Algorithm (AOA) [Abualigah, 2021], Stochastic Fractal Search (SFS) [Salimi, 2015] and Tangent Search Algorithm (TSA) [Layeb, 2022].

Engineers solve problems by creating new products, systems, or environments. Before creating something, it is very important to make the mathematical formulation of the problem, then try to find the parameters of the problem to maximize efficiency, effectiveness, profitability, and minimize the energy consumption and costs. On many occasions, engineering optimization design problems involve a variety of decision variables and complex structured objectives, and constraints. The traditional optimization techniques often face difficulty in solving such optimization problems in their original form. Therefore, this work presents a comparative study of the metaheuristic optimization algorithms for solving engineering classical design problems. This work is done in order to recognize the best algorithms for solving a particular engineering design problem. The remaining of this paper is organized as follows: Second section describes the study's methodology. The third section briefly introduces the metaheuristic optimization algorithms used in this paper. The fourth section illustrates the engineering design problems including objective function and their constraints. The fifth section provides the experimental result and solution analyses of each optimization algorithm. Finally, Section six, summarizes the work which is done in this paper and explains future perspectives and suggestions regarding this research.

2. Methodology

This paper presents a comparative study of meta-heuristic optimization algorithms for solving engineering classical design problems. This work aims to introduce the best algorithm for each engineering problem. The work is done in three steps as follows: Firstly, fifteen well known metaheuristic optimization algorithms, namely, Artificial Bee Colony (ABC), Ant Colony Optimization (ACO), Biogeography Based Optimization Algorithm (BBO), Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES), Cuckoo Search algorithm (CS), Differential Evolution (DE), Genetic Algorithm (GA), Grey Wolf Optimizer (GWO), Gravitational Search Algorithm (GSA) and Particle Swarm Optimization (PSO) opted from among metaheuristic algorithms in the literature, to camper their performance in solving engineering design problems. Although several improved versions of the opted algorithms are in the literature, this study is done by their standard versions. Then, all metaheuristic algorithms are briefly introduced. Secondly, the teen widely used engineering design problems, namely, speed reducer design, tension/compression spring design, pressure vessel design, Three-bar truss design, gear train design, cantilever beam design, I-beam vertical deflection design, tubular column design, piston lever design and welded beam design are selected to evaluate the performance of optimization algorithms. Thirdly, the algorithms and problems are coded in the MATLAB program. Finally, after 10 independent runs of the program, the average of each problem's best solution and cost are recorded in the tables. In order to visualize and compare the performance of the investigated algorithms, the obtained best cost is plotted for each problem, and for the ease of readability, the best solution obtained among the algorithms is highlighted in boldface. The detailed background of the experiments is given in Table 1 below.

Table 1. Experimental background details.

Name	**Setting**
System Manufacturer	Acer
Processor	AMD A4-7210 (2.2GHz) APU with AMD Radeon R3 Graphics, 1800 Mhz, 4 Core(s), 4 Logical Processors (s)
HDD	1000GB
RAM	4GB
Operation System	Windows 10, x64-Based PC
Language	MATLAB 2014a

3. Meta-Heuristic Algorithms

This section briefly describes the nature inspired optimization algorithms (meta-heuristic algorithm) used in this study for solving engineering problems.

3.1 Genetic Algorithm (GA)

Genetic algorithm (GA) is the most popular and widely used computational technique inspired by natural selection and genetics principles. John Holland and his collaborators developed it in the 1960s and 1970s [Holland, 1992]. It is used to solve complex optimization problems with the goal of finding the best solution among a large set of possible solutions. The genetic optimization algorithm then applies several genetic operators to evolve the population over multiple generations. These operators include selection, crossover, and mutation. Selection involves choosing the fittest individuals based on their fitness function, which evaluates how well each individual solves the problem. Crossover combines two parent chromosomes to create offspring by exchanging genetic material between them. Mutation introduces small random changes in the offspring's chromosomes to maintain diversity in the population. After applying these operators, a new population is created and evaluated using the fitness function. This process continues for several generations until a termination condition is met.

3.2 Differential Evolution

Differential Evolution (DE) is a nature inspired optimization algorithm that was introduced by Storn and Price in 1997s [Storn, 1997]. DE operates on a population of candidate solutions called individuals or vectors. Each individual represents a potential solution to the optimization problem. The algorithm iteratively improves the population by applying mutation, crossover, and selection operations. The mutation operation creates new trial individuals by perturbing the existing individuals in the population. This perturbation is achieved by adding a scaled difference vector between randomly selected individuals to another individual. This process introduces exploration into the search space. The crossover operation combines information from the trial individuals with the original individuals to create offspring. It determines which components of the trial individual will be inherited by the offspring and which components will be inherited from the original individual. This process allows for exploitation of promising solutions. The selection operation compares each offspring with its corresponding original individual and selects the better one based on their fitness values. The selected offspring replaces its corresponding original individual if it has higher fitness. DE continues this iterative process until a termination criterion is met.

3.3 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a computational technique inspired by the social behavior of bird flocking or fish schooling. It was developed by Kennedy and Eberhart in 1995s [Kennedy, 1995]. In PSO, a group of particles move through a search space to find the optimal solution to a given problem. Each particle represents a potential solution and has its own position and velocity in the search space. The movement of particles is influenced by their own best-known position (pbest) and the best-known position among all particles in the swarm (gbest). The pbest represents the best solution each particle finds, while the gbest represents the global best solution found by any particle in the swarm. At each iteration, particles update their velocities based on their current positions, pbest, and gbest. The new velocity determines how far and in which direction a particle should move. Then, particles update their positions based on their new velocities. The key idea behind PSO is that particles communicate with each other through their pbest and gbest information. This allows them to explore different regions of the search space efficiently. Particles converge towards an optimal solution over time by continuously updating their positions and velocities based on this information exchange.

3.4 Artificial Bee Colony

The Artificial Bee Colony (ABC) optimization algorithm is a metaheuristic algorithm inspired by the foraging behavior of honey bees. It was proposed by Karaboga in 2005s as a simple and efficient optimization technique for solving complex optimization problems [Karaboga, 2005]. In ABC, the search process is modeled based on the behavior of three types of bees: employed bees, onlooker bees, and scout bees. The employed bees explore the search space by visiting food sources (solutions) and evaluating their quality using an objective function. They then share information about their food sources with onlooker bees through a process called waggle dance. Onlooker bees select food sources based on their quality information and perform local search around those solutions to improve them further. This process allows the algorithm to exploit promising regions in the search space. Scout bees are responsible for introducing diversity into the population. If an employed bee exhausts its limit of trials without finding a better solution, it becomes a Scout bee and randomly explores new solutions in the search space. The quality of each solution is evaluated using an objective function, and the algorithm iteratively updates the population by employing different strategies like exploitation and exploration.

3.5 *Biogeography-based optimization algorithm*

Biogeography-Based Optimization (BBO) algorithm is some nature-inspired optimization techniques that draw inspiration from the biogeography. It was proposed by Dan Simon in 2008s [Simon, 2008]. Biogeography is the study of the distribution of species and ecosystems across different geographical regions. BBO algorithms simulate the process of migration/immigration and evolution of species in order to solve optimization problems. The basic idea behind these algorithms is to represent potential solutions to an optimization problem as "habitats" and use migration/immigration and evolution operators to search for the best solution. The main feature of BBO is migration (crossover), mutation and selection.

3.6 Covariance Matrix Adaption Evolution Strategy

Covariance Matrix Adaptation Evolution Strategy (CMA-ES) is a stochastic optimization algorithm used for solving complex optimization problems. It was proposed by Nikolaus Hansen, Ostermeier Andreas, and Gawelczyk Andreas in 1995s [Hansen, 1995]. It is a derivative-free, population-based algorithm belonging to the evolutionary algorithms family. CMA-ES employs a covariance matrix to model the distribution of candidate solutions in the search space. This matrix is adapted over generations to guide the search towards promising regions. CMA-ES utilizes a combination of global exploration and local exploitation strategies. It explores the search space by generating new candidate solutions from the current distribution and evaluates their fitness values. The algorithm then updates the distribution parameters based on the success of these new solutions, favoring those with better fitness. One key feature of CMA-ES is its ability to handle ill-conditioned or high-dimensional problems by adapting its step sizes and covariance matrix accordingly. This adaptability allows it to efficiently navigate complex landscapes with varying degrees of difficulty.

3.7 Grey Wolf Optimizer

The Grey Wolf Optimizer (GWO) is a nature-inspired metaheuristic algorithm that mimics the hunting behavior of grey wolves in a pack. Seyedali Mirjalili proposed it in 2013s and has gained popularity due to its simplicity and effectiveness in solving optimization problems [Mirjalili, 2014]. In GWO, a population of candidate solutions, represented as grey wolves, is iteratively updated to search for the optimal solution. The algorithm imitates the wolves' social hierarchy and hunting behavior to balance exploration and exploitation during the search process. The GWO algorithm consists of four main steps: initialization, updating the position of alpha, beta, and delta wolves, updating the position of other wolves, and boundary handling.

3.8 Cuckoo Search Algorithm

Cuckoo Search Algorithm (CS) is a nature-inspired optimization algorithm that was developed by Xin-She Yang and Suash Deb in 2009s [Yang, 2010]. It is inspired by the breeding behavior of cuckoo birds, specifically their brood parasitism strategy. In nature, some species of cuckoo birds lay their eggs in the nests of other bird species, tricking them into incubating and raising their young. The host birds may eventually recognize and discard the foreign eggs from their nests. This behavior has led to an evolutionary arms race between cuckoos and host birds, where cuckoos continuously adapt their egg-laying strategies to increase their chances of successful reproduction. The Cuckoo Search Algorithm mimics this behavior by using a population of virtual cuckoos to search for optimal solutions in optimization problems. Each cuckoo represents a potential solution, and they lay eggs (new solutions) in different nests (search locations). The quality of each solution is evaluated using an objective function, which determines its fitness. CS incorporates several key components such as random walk steps, Levy flights, and a global discovery rate parameter that controls the balance between exploration and exploitation.

3.9 Gravitational Search Algorithm

The Gravitational Search Algorithm (GSA) is a metaheuristic optimization algorithm inspired by the law of gravity and the behavior of celestial bodies in space. It was proposed by Rashedi, Nezamabadi-pour, and Saryazdi in 2009s [Rashedi, 2009]. GSA mimics the gravitational forces between celestial bodies to search for optimal solutions in a given problem space. In this algorithm, each potential solution is represented as a celestial body, and their positions are updated based on the gravitational forces exerted by other bodies. The GSA starts with an initial population of celestial bodies randomly distributed in the search space. The fitness value of each body represents its quality as a solution to the problem being solved.

3.10 Ant Colony Optimization Algorithm

Ant Colony Optimization (ACO) is a metaheuristic algorithm inspired by the behavior of ants searching for food. It was first introduced by Marco Dorigo in the early 1990s [Dorigo, 2004]. ACO is primarily used to solve combinatorial optimization problems, such as the traveling salesman or vehicle routing problems. The algorithm mimics the behavior of real ants, which communicate with each other through pheromone trails to find the shortest (best) path between their nest and a food source. In ACO, artificial ants are used to explore the solution space of a given problem.

4. Engineering Design Problems

This section briefly describes the teen classical engineering design problems used in this study.

4.1 Problem 1. Speed Reducer Design Problem

Figure 1 illustrate the speed reducer design problem. The speed reducer design problem is one of the benchmark structural engineering problems. This problem contains seven decision variables namely, face width, x_1, module of teeth, x_2, number of teeth on pinion, x_3, length of first shaft between bearings, x_4, length of second shaft between bearings x_5, diameter of first shaft, x_6, and diameter of second shaft, x_7. The objective of the problem is to minimize the total weight of the speed reducer. There are nine constraints in the problem. The best obtained solutions of the problem by teen optimization algorithms are presented in Table 2 and the convergence graph of the objective function, $f(\vec{x})$, is plotted in figure 11. The mathematical model of the problem is given as follows:

$$f(\vec{x}) = 0.7854x_1{x_2}^2(3.3333{x_3}^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.477(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2)$$

Subject to

$$g_1(\vec{x}) = -x_1{x_2}^2x_3 + 27 \le 0, g_2(\vec{x}) = -x_1{x_2}^2{x_3}^2 + 397.5 \le 0, g_3(\vec{x}) = -\frac{x_2x_3x_6^3}{x_4^3} + 1.93 \le 0,$$

$$g_4(\vec{x}) = -\frac{x_2x_3x_7^3}{x_5^3} + 1.93 \le 0, g_5(\vec{x}) = 10x_6^{-3}\sqrt{(745x_4/x_2x_3)^2 + 157{,}5 \times 10^6} - 1100 \le 0,$$

$$g_6(\vec{x}) = 10x_7^{-3}\sqrt{(745x_4/x_2x_3)^2 + 1.69 \times 10^6} - 850 \le 0, g_7(\vec{x}) = x_2x_3 - 40 \le 0, g_8(\vec{x}) = -\frac{x_1}{x_2} + 5 \le 0,$$

$$g_9(\vec{x}) = \frac{x_1}{x_2} - 12 \le 0, g_{10}(\vec{x}) = 1.5x_6 - x_4 + 1.9 \le 0, g_{11}(\vec{x}) = 1.1x_7 - x_5 + 1.9 \le 0.$$

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

4.2 Problem 2. Tension/Compression Spring Design

Figure 2 illustrate the tension/compression spring design problem. The objective of the problem is to minimize the weight of tension/compression spring subject to given constraints. In this problem, there are three decision variables: mean coil diameter, D, number of active coils, N, and wire diameter, d. The best obtained solutions of the problem by teen optimization algorithms are presented in Table 3 and the convergence graph of the objective function, $f(\vec{x})$ is plotted in figure 12. The mathematical model of the problem is given as follows:

Consider $[x_1, x_2, x_3] = [d, D, N]$

$f(\vec{x}) = (x_3 + 2)x_1^2x_2$

Subject to

$$g_1(\vec{x}) = 1 - \frac{x_2^3x_3}{71785x_1^4} \le 0, g_2(\vec{x}) = \frac{4x_2^2 - x_1x_2}{12566(x_1^3x_2 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \le 0,$$

$$g_3(\vec{x}) = 1 - \frac{140.45x_1}{x_2^2x_3} \le 0, g_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \le 0.$$

Variable range $0.05 \le x_1 \le 2, 0.25 \le x_2 \le 1.30, 2 \le x_3 \le 15$

4.3 Problem 3. Pressure Vessel Design Problem

Figure 3 illustrate the pressure vessel design problem. The purpose of the pressure vessel design problem is to minimize the welding, the material, and forming cost, $f(\vec{x})$. There are four decision variables in the problem, namely, thickness of the head, T_h, thickness of the shell, T_s, length of the cylindrical section without considering the head, L and the inner radius, R and containing four constraints. The best obtained solutions of the problem by teen optimization algorithms are presented in Table 4 and the convergence graph of the objective function, $f(\vec{x})$, is plotted in figure 13. The mathematical model of the problem is given as follows:

Consider $[x_1, x_2, x_3, x_4] = [T_s, T_h, R, L]$

$$f(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

Subject to

$$g_1(\vec{x}) = -x_1 + 0.0193x_3 \le 0, g_2(\vec{x}) = -x_2 + 0.00954x_3 \le 0, g_3(\vec{x}) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0,$$

$$g_4(\vec{x}) = x_4 - 240 \le 0,$$

Variable range $0 \le x_1, x_2 \le 99, 10 \le x_3, x_4 \le 200.$

4.4 Problem 4. Three-Bar Truss Design Problem

Figure 4 illustrate the three bar truss design problem. This problem's objective is to minimise the structure's weight, subject to given constraints. The constraints here are deflection, stress and buckling constraints. The best obtained solutions of the problem by teen optimization algorithms are presented in Table 5 and convergence graph of the objective function, $f(\vec{x})$, is plotted in figure 14. The mathematical model of the problem is given as follows:

Consider $[x_1, x_2] = [A_1, A_2]$
$Minimize\ f(\vec{x}) = \left(2\sqrt{2}x_1 + x_2\right) \times l$
Subject to

$$g_1(\vec{x}) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \le 0, g_2(\vec{x}) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \le 0, g_3(\vec{x}) = \frac{1}{\sqrt{2}x_1 + x_1}P - \sigma \le 0,$$

Variable range $0. \le x_1, x_2 \le 1.$ Where $l = 100cm$ and $P = \sigma = 2\text{KN}/cm^2$

4.5 Problem 5. Gear Train Design Problem

Figure 5 illustrate the gear train design problem. The objective of this problem is to design a gear train that the gear ratio should be close to $\frac{1}{6.931}$ in order to minimize the farming cost. The teeth of the gears namely, n_A, n_B, n_C and n_D are four decision variables of the problem. The best obtained solutions of the problem by teen optimization algorithms are presented in Table 5 and convergence graph of the objective function, $f(\vec{x})$, is plotted in figure 15. The mathematical model of the problem is given as follows:

$Min\ f\left((\vec{x})\right) = \left(\frac{1}{6.931} - \frac{x_3x_2}{x_1x_4}\right)^2$, Subject to $12 \le\ x_1,\ x_2, x_3, x_4 \le 60$

4.6 Problem 6. Cantilever Beam Design Problem

Figure 6 illustrate the cantilever beam design problem. The problem includes five hollow elements with square-shaped cross-sections, so there is a total of five structural variables, because the thickness is constant. The objective of the cantilever beam design problem is to minimize the weight of the beam. The best obtained solutions of the problem by teen optimization algorithms are presented in Table 7 and convergence graph of the objective function, $f(\vec{x})$, is plotted in figure 16. The mathematical model of the problem is given as follows:

$Minimize\ f(\vec{x}) = 0.0624(x_1 + x_2 + x_3 + x_4 + x_5)$

Subject to; $g_1(\vec{x}) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \le 0.$ Variables range; $0.01 \le x_i \le 100, i = 1,2,3,4,5$

4.7 Problem 7. One Beam Vertical Deflection Problem

Figure 7 illustrate the I-beam vertical deflection problem. The objective is to minimize the deflection of the beam. There are four decision variable in the problem namely, flange width, b, flange thickness, t_f, beam height, h, and web thickness, t_w. The best obtained solutions of the problem by fifteen algorithms are presented in Table 8 and convergence graph of the objective function, $f(\vec{x})$, is plotted in figure 17. The mathematical model of the problem is given as follows:

Consider $[x_1, x_2, x_3, x_4] = \left[b, h, t_w, t_f\right]$, $Minimize\ f(\vec{x}) = \frac{5000}{T_1+T_2+T_3}$

$$T_1 = \frac{x_3(x_2 - 2x_4)^3}{12}, T_2 = \frac{x_1x_4^3}{6}, T_3 = 2x_1x_4\left(\frac{x_2 - x_4}{2}\right)^2$$

Subject to

$$g_1(\vec{x}) = 2x_2x_4 + x_3(x_1 - 2x_4) - 300 \le 0, g_2(\vec{x}) = \frac{18 \times 10^4 x_1}{L_1 + L_2} + \frac{15 \times 10^3 x_2}{L_3 + L_4} - 6 \le 0,$$

$$L_1 = x_1x_3 - 2x_4^3, L_2 = 2x_2x_4(4x_4^2 + 3x_1^2 - 2x_4),\ L_3 = x_1 - 2x_3^3x_4, L_4 = 2x_2^3x_4,$$

Variable range $10 \le x_1 \le 50, 10 \le x_2 \le 80, 0.9 \le x_1 \le 5, 0.9 \le x_1 \le 5,$

4.8 Problem 8. Tubular Column Design Problem

Figure 8 illustrate the tubular column design problem. A tubular column is a structural element that consists of a hollow cylinder made of metal, concrete, or other material. It is commonly used in construction to support beams and other building elements. It can be used in structural design of bridges and other structures. The objective of the problem is to minimize the cost of building the column, using decision variable d, the mean diameter of the column, and t, the thickness of the column. The best obtained solutions of the problem by teen optimization algorithms are presented in Table 9 and convergence graph of the objective function, $f(\vec{x})$, is plotted in figure 18. The final formulation for the cantilever beam design problem is shown as follows:

Consider $[x_1, x_2] = [d, t]$

$Minimize\ f(\vec{x}) = 9.82x_1x_2 + 2x_1$
Subject to
$g_1(\vec{x}) = 1.59 - x_1x_2 \leq 0, g_2(\vec{x}) = 47.4 - x_1x_2(x_1^2 + x_2^2) \leq 0, g_3(\vec{x}) = \dfrac{2}{x_1} - 1 \leq 0, g_4(\vec{x}) = \dfrac{x_1}{14} - 1 \leq 0,$
$g_5(\vec{x}) = \frac{x_1}{8} - 1 \leq 0$, Variables range; $2 \leq x_1 \leq 14, 0.2 \leq x_2 \leq 0.8$

4.9 Problem 9. Piston Lever Design Problem

Figure 9 illustrate the piston lever design problem. The objective of the problem is to locate the piston components, H, B, D, and X by minimizing the oil volume when the lever of the piston is lifted up from 0º to 45º as shown in Figure 9. The best obtained solutions of the problem by teen optimization algorithms are presented in Table 10 and convergence graph of the objective function, $f(\vec{x})$, is plotted in figure 19. The mathematical model of the problem is given as follows:

Consider $[x_1, x_2, x_3, x_4] = [H, B, D, X]$

$f(\vec{x}) = \dfrac{1}{4}\pi {x_3}^2(L_2 - L_1)$
Subject to
$g_1(\vec{x}) = QLcos\theta - RF \leq 0 \quad at\ \theta = 45^o, g_2(\vec{x}) = Q(L - x_4) - M_{max} \leq 0, g_3(\vec{x}) = 1.2(L_2 - L_1) - L_1 \leq 0,$
$g_4(\vec{x}) = D/2 - B \leq 0.$
Where
$R = \dfrac{|x_4(x_4sin\theta + x_1) + x_1(x_2 - x_4cos\theta)|}{\sqrt{(x_4 - x_2)^2 + {x_1}^2}}, F = \dfrac{\pi PD^2}{4}, L_1 = \sqrt{(x_4 - x_2)^2 + {x_1}^2},\ L_2 = \sqrt{(x_4sin45 + x_1)^2 + (x_2 - x_4cos45)^2}$

Where the lever is L = 240 in, the pay load is P = 10,000 lbs, the maximum allowable bending = moment of the lever is 6 max M = 1.8 9 10 lbs in, and the oil pressure is given as 1,500 psi.

4.10 Problem 10. Welded Beam Design Problem

Figure 10 illustrated the Welded beam design problem. The objective of welded beam design problem is to minimize the fabrication cost by determining the optimal value of four variables namely, length of attached part of bar, l, thickness of weld, h, the height of the bar, t, and thickness of the bar, b. The best obtained solutions of the problem by teen optimization algorithms are presented in Table 11 and convergence graph of the objective function, $f(\vec{x})$, is plotted in figure 20. The mathematical model of the problem is given as follows:

Consider $[x_1, x_2, x_3, x_4] = [h, l, t, b]$

$Minimize\ f(\vec{x}) = 1.1047x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$

Subject to

$g_1(\vec{x}) = \tau(\vec{x}) - \tau_{max} \leq 0, g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{max} \leq 0, g_3(\vec{x}) = \delta(\vec{x}) - \delta_{max} \leq 0, g_4(\vec{x}) = x_1 - x_4 \leq 0,$
$g_5(\vec{x}) = P - P_c(\vec{x}) \leq 0, g_6(\vec{x}) = 0.125 - x_1 \leq 0, g_7(\vec{x}) = 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0,$

Variable range $0.1 \leq x_1, x_4 \leq 2, 0.1 \leq x_2, x_3 \leq 10.$ where,

$$\tau(\vec{x}) = \sqrt{\left(\frac{P}{\sqrt{2x_1x_2}}\right)^2 + 2\frac{P}{\sqrt{2x_1x_2}}\frac{MR}{J}\frac{x_2}{2R} + \left(\frac{MR}{J}\right)^2}, M = P\left(L + \frac{x_2}{2}\right), R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2},$$

$$P_C(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2x_4^2}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right), J = 2\left\{\sqrt{2x_1x_2}\left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}, \sigma(\vec{x}) = \frac{6PL}{x_4x_3^2}, \delta(\vec{x}) = \frac{6PL^3}{Ex_3^2x_4}, P = 6000lb$$

$L = 14\ in., \delta_{max} = 0.25in., E = 30 \times 10^6 psi, \tau_{max} = 13600\ psi, \sigma_{max} = 30000\ psi, G = 12 \times 10^6 psi$

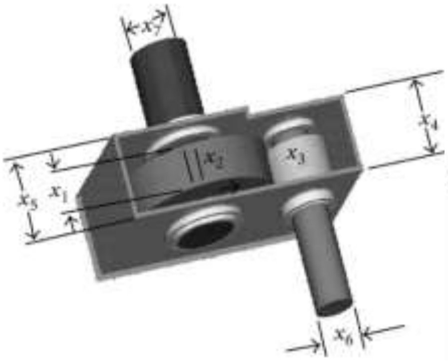


Figure 1. Speed reducer design.

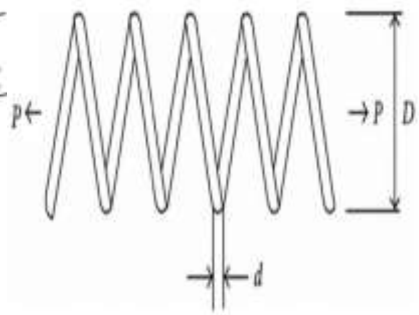


Figure 2. Tension/compression spring design

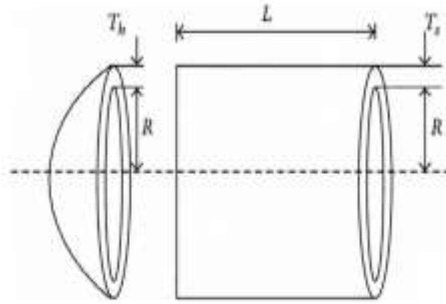


Figure 3. Pressure vessel design

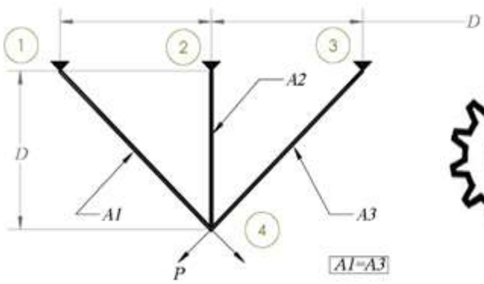


Figure 4. Tree bar design.

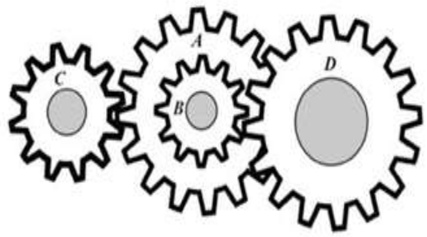


Figure 5. Gear train design

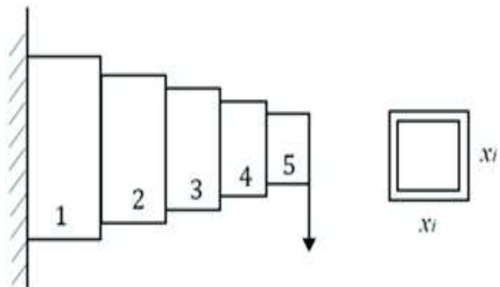


Figure 6. Cantilever beam design

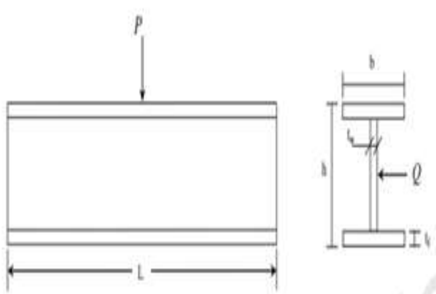


Figure 7. One beam design

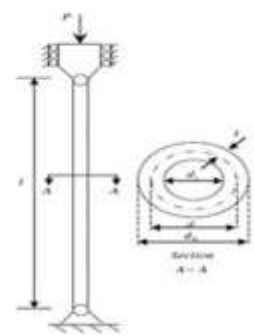


Figure 8. Tubular Column Design

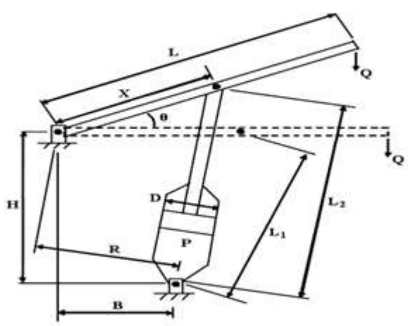


Figure 9. Piston Lever Design

5. Experimental Results

Engineering design problems are mostly constrained. Two classes of constraints are involved in defining the feasible solutions during the design process: equality and inequality constraints. For optimizing constrained engineering design problems, a constraint handling method must be integrated to the optimization algorithms. There are several methods of constraints handling in the literature: special operators, penalty functions, separation of objective functions and constraints, hybrid methods and repair algorithms. Since finding a good constraints handling method for the selected nature inspired optimization algorithms is out of the scope of this paper, the simplest method called death penalty is used in the experiments. In all experiments in this section, the maximum number of iteration is 500 and population size is

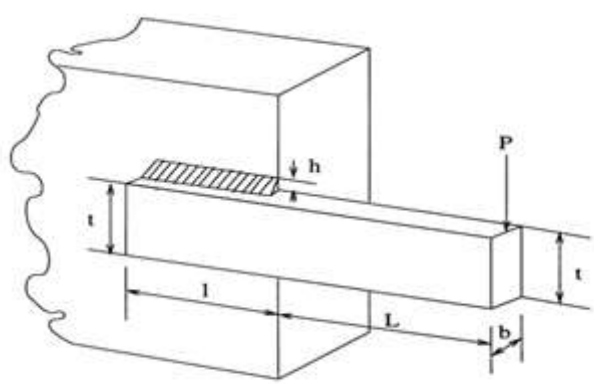


Figure 10. Welded beam design

taken. Since the main objective of solving an engineering design problem is to achieve the global optimum with the least possible computational cost, this section only presents the best obtained solution stored in the tables. In order to conduct the experiments, each algorithm 10 times independently runs carried out to find the optimal solution. Other specific control parameters of the algorithms are presented as follows:

ABC: Limit = (population size × problem dimension)/2.
ACO: Sample size =40, Intensification Factor =0.5 and Deviation-Distance Ratio =1.
BBO: Emigration Rates are the same as original form of the algorithm, Habitat Keep Rate =0.4, Habitat Keep Size = round (Habitat Keep Rate× Population size), alpha =0.9 and mutation rate is 0.15.
CMA-ES: Lambda= (4+round (3×log(problem dimension))) ×5, mu =lambda/2, and alpha_mu =2.
CS: Discovery rate of alien eggs =0.25 and beta =3/2.
DE: Crossover rate =0.5.
GA: Uniform Crossover, Mutation rate =0.4, pc =beta =1 and sigma =1.6
GSA: ElitistCheck=1, Rpower=1, alpha=20, G0=100 and Final Percentage =2.
GWO: a =2, Coefficient Vector $\vec{A}$ =2×a× $rand(0,1) - a$ and $\vec{A}$ =2× $rand(0,1)$
PSO: Inertia weight =1, Dumping ratio of the inertia =0.99 and Acceleration Coefficients =2.

Table 2. The optimum table of speed reducer design problem.

Values	Meta-Heuristic Optimization Algorithms									
	ABC	**ACO**	**BBO**	**CMA-ES**	**CS**	**DE**	**GA**	**GSA**	**GWO**	**PSO**
x_1	3.46903	3.5	3.5061	3.60000	3.50000	3.5	3.50614	3.5	3.50383	3.58
x_2	0.738422	0.7	0.70095	0.70000	0.70000	0.7	0.7	0.7	0.70001	0.7
x_3	22.1127	17	17.3288	17.02269	17.0000	17	17	19.2815	17.00109	17
x_4	7.73173	7.3	7.90396	7.66286	7.30015	7.3	7.3	7.75102	7.66704	7.9
x_5	7.90261	7.7153	7.96752	8.22044	7.71567	7.7153	7.72769	7.88221	8.07049	8.20653
x_6	3.67473	3.3505	3.47351	3.48091	3.35057	3.3505	3.35698	3.34987	3.36105	3.3517
x_7	5.38887	5.2867	5.30899	5.44752	5.28670	5.2867	5.29133	5.28644	5.28903	5.35081
$f(\vec{x})$	**2994.425**	**2994.425**	3119.142	3197.373	2994.454	**2994.425**	3001.72	3436.664	3011.428	3084.798

Table 3. The optimum table of tension/compression spring design problem.

Values	Meta-Heuristic Optimization Algorithms									
	ABC	**ACO**	**BBO**	**CMA-ES**	**CS**	**DE**	**GA**	**GSA**	**GWO**	**PSO**
x_1	0.19154	0.05787	0.063632	0.057280	0.05167	0.05173	0.0715	0.05344	0.05303	0.05311
x_2	0.91193	0.53449	0.726039	0.473600	0.35643	0.3574	0.97562	0.39654	0.39452	0.40773
x_3	10.64974	6.10533	8.905250	12.29424	11.31581	11.25386	2.05605	10.23385	10.43686	11.67213
$f(\vec{x})$	0.013431	0.013617	0.031074	0.022630	0.012669	**0.012666**	0.019314	0.01350	0.012892	0.013425

Table 4. The optimum table of the pressure vessel design problem.

Values	Meta-Heuristic Optimization Algorithms									
	ABC	**ACO**	**BBO**	**CMA-ES**	**CS**	**DE**	**GA**	**GSA**	**GWO**	**PSO**
x_1	53.41444	17.53227	31.9342	15.08084	12.96178	13.02424	16.1962	16.14815	13.07634	16.92374
x_2	43.32319	8.93109	20.7027	8.3705	7.19033	6.96844	8.07215	12.13964	6.93799	8.36021
x_3	59.82609	56.60267	56.6552	47.1189	42.39246	42.0984	52.3517	50.14964	42.41433	53.53823
x_4	118.5264	60.88875	57.9356	136.5882	173.5198	176.6366	86.10287	98.29682	173.1477	86.7682
$f(\vec{x})$	6742.043	6814.896	16651.33	6932.2	6070.141	**6059.714**	6531.356	7671.275	6064.998	6735.203

Table 5. The optimum table of the three bar truss design problem.

Values	Meta-Heuristic Optimization Algorithms									
	ABC	ACO	BBO	CMA-ES	CS	DE	GA	GSA	GWO	PSO
x_1	0.83105	0.78868	0.793722	0.79276	0.7887	0.7887	0.811	0.77102	0.78844	0.80979
x_2	0.43663	0.40822	0.42455	0.39977	0.40821	0.4082	0.35268	0.4636	0.40899	0.36747
$f(\vec{x})$	264.2724	263.8962	266.9535	264.204	**263.8958**	**263.8958**	264.6586	264.4393	263.9034	265.7909

Table 6. The optimum table of the gear train design problem.

Values	Meta-Heuristic Optimization Algorithms									
	ABC	ACO	BBO	CMA-ES	CS	DE	GA	GSA	GWO	PSO
x_1	45.50281	44.97903	52.7511	54.86008	49.14665	53.99039	52.018	47.69817	46.24571	47.88687
x_2	24.01581	18.83844	20.7243	24.57884	16.39411	19.9608	18.72379	19.21913	18.85618	21.38847
x_3	18.36112	16.92246	17.4072	22.23511	19.42264	22.36668	20.1361	17.85425	16.3491	21.06099
x_4	47.28579	47.35928	46.4695	54.01053	41.79137	53.2693	46.82071	48.98712	46.58534	52.80312
$f(\vec{x})$	2.3E-11	1.96E-11	4.08E-17	2.38E-07	3.27E-15	1.07E-20	7.23E-13	2.48E-27	6.18E-12	**7.6E-30**

Table 7. The optimum table of the cantilever beam design problem.

Values	Meta-Heuristic Optimization Algorithms									
	ABC	ACO	BBO	CMA-ES	CS	DE	GA	GSA	GWO	PSO
x_1	34.96363	6.01636	5.91823	6.01885	6.03426	6.01674	6.01712	6.17858	6.018	6.02401
x_2	31.17536	5.30695	5.2822	5.30526	5.30905	5.30859	5.33971	5.26544	5.30569	5.30195
x_3	22.11757	4.49385	4.44764	4.49214	4.47877	4.49481	4.50328	4.37435	4.49906	4.49522
x_4	28.57672	3.50303	4.7632	3.49830	3.50031	3.50124	3.46815	4.4468	3.50025	3.50038
x_5	24.21287	2.15364	3.34591	2.16086	2.15321	2.15228	2.18155	2.29922	2.15225	2.153
$f(\vec{x})$	1.40472	**1.34**	1.48245	1.34006	1.34008	**1.34**	1.34221	1.40801	1.34006	1.34002

Table 8. The optimum table of the one beam vertical deflection design problem.

Values	Meta-Heuristic Optimization Algorithms									
	ABC	ACO	BBO	CMA-ES	CS	DE	GA	GSA	GWO	PSO
x_1	56.24547	80	77.94950	79.66980	79.99999	80	80	52.93364	80	80
x_2	36.79195	50	39.78700	43.74490	49.99498	50	44.10551	33.96276	49.99662	50
x_3	1.96862	0.9	1.123057	0.900000	5.80986	0.9	0.90001	0.90198	5.81	0.9
x_4	2.60887	2.34137	2.836650	2.805180	2.32201	2.3218	2.71691	3.82224	2.32181	2.3218
$f(\vec{x})$	0.015098	0.01308	2.059639	0.116947	0.013074	**0.013074**	0.013196	58.78099	0.013075	0.013074

Table 9. The optimum table of the tubular column design problem.

Values	Meta-Heuristic Optimization Algorithms									
	ABC	ACO	BBO	CMA-ES	CS	DE	GA	GSA	GWO	PSO
x_1	5.75393	5.4522	5.39201	5.60713	5.4522	5.4522	5.47185	5.462	5.45276	5.4522
x_2	0.36262	0.2916	0.328956	0.28538	0.2916	0.2916	0.29842	0.2913	0.29165	0.2916
$f(\vec{x})$	26.63162	**26.4864**	27.8482	26.81243	26.4864	**26.4864**	26.91682	26.51721	26.49097	**26.4864**

Table 10. The optimum table of the piston lever design problem.

Values	Meta-Heuristic Optimization Algorithms									
	ABC	**ACO**	**BBO**	**CMA-ES**	**CS**	**DE**	**GA**	**GSA**	**GWO**	**PSO**
x_1	183.9049	0.05	174.625	150.035	0.05	0.05	209.5217	225.993	149.8804	300.02
x_2	287.1188	2.0415	249.4162	159.493	2.04165	1.04575	413.5279	282.8183	151.424	300.8166
x_3	29.86139	4.083	4.42546	3.90891	4.08313	4.10015	2.92495	11.93954	3.52209	2.95986
x_4	91.60922	120	64.8717	98.7006	119.9994	120	68.15038	73.55425	102.0025	84
cost	8.43801	**8.4127**	520.4436	165.7549	8.4137	104.9726	284.3971	11456.59	56.23043	103.8487

Table 11. The optimum table of welded beam design problem.

Values	Meta-Heuristic Optimization Algorithms									
	ABC	**ACO**	**BBO**	**CMA-ES**	**CS**	**DE**	**GA**	**GSA**	**GWO**	**PSO**
x_1	0.70030	0.31749	0.504049	0.24914	0.205	0.2057	0.50505	0.53483	0.20517	0.20627
x_2	5.06083	5.17563	3.80350	7.23077	7.11726	7.0924	3.92521	3.87374	7.1101	6.95803
x_3	5.18349	7.33520	6.07009	8.49776	9.06279	9.0366	5.82759	5.27035	9.05081	9.31923
x_4	0.81911	0.31749	0.605837	0.28008	0.20594	0.2057	0.53369	0.61507	0.20583	0.21077
Cost	3.00981	2.68963	3.88370	2.84074	2.22647	**2.2182**	3.58493	3.94203	2.22264	2.28834

Figure 11: Speed Reducer Design

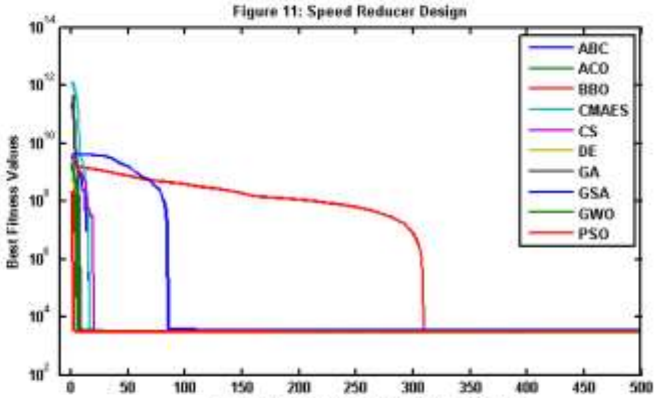


Figure 12: Tension/Compression Spring Design

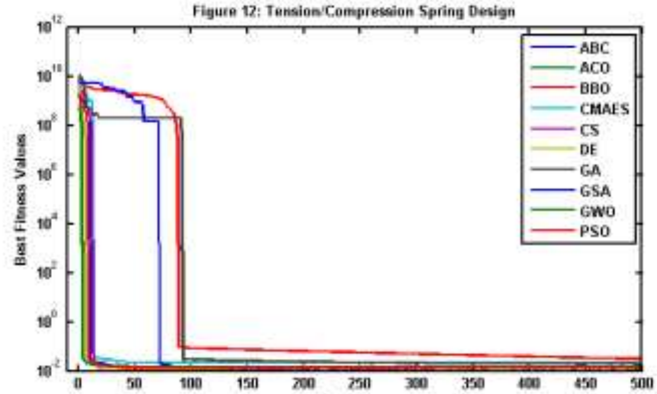


Figure 13: Pressure Vessel Design Problem

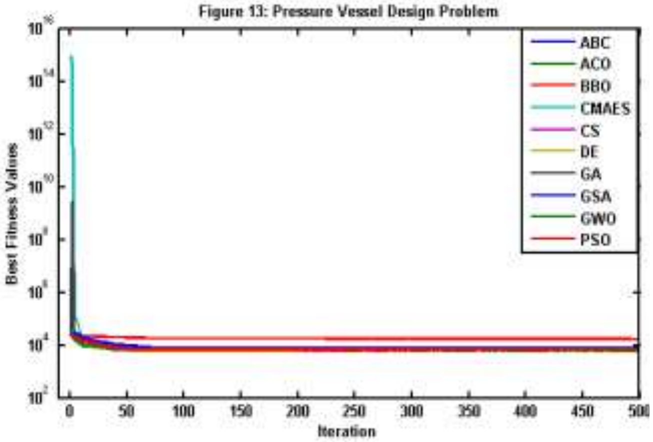
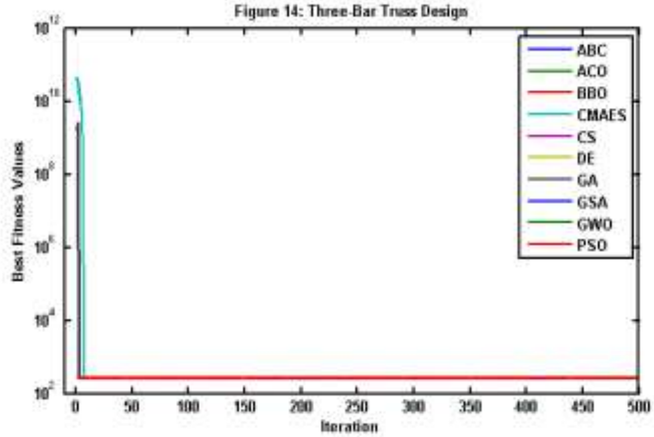
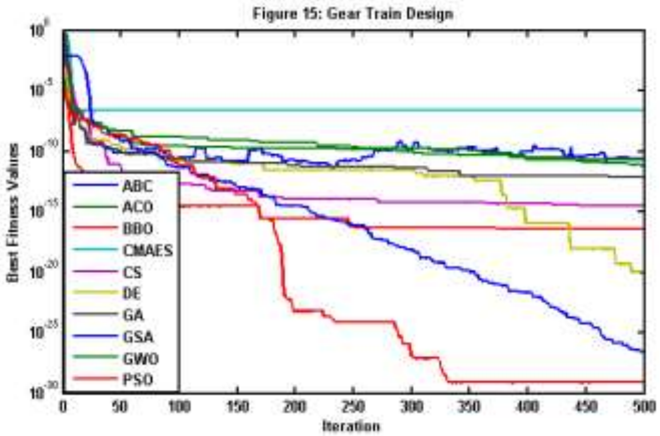
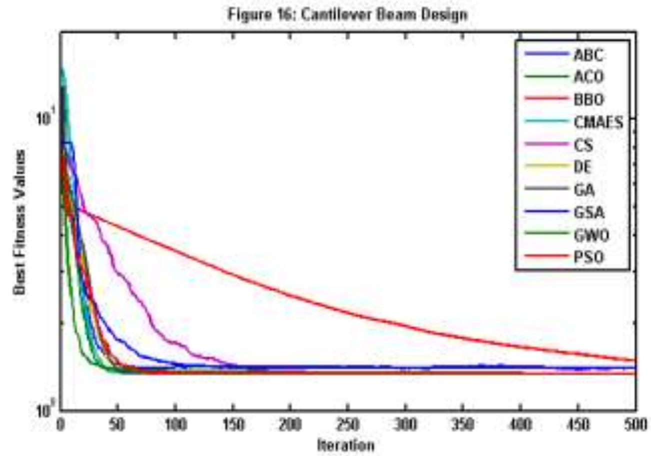
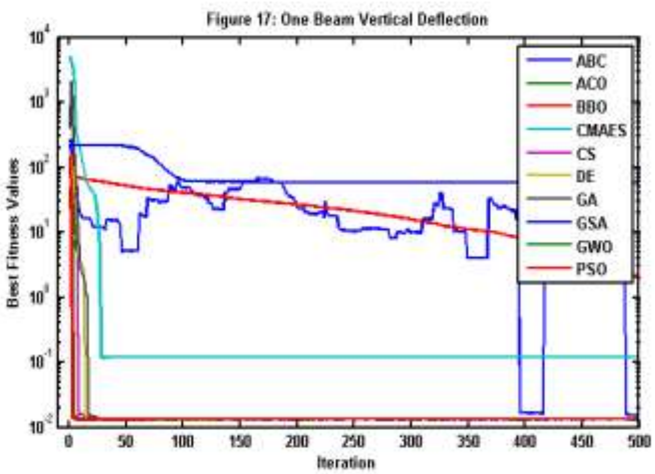


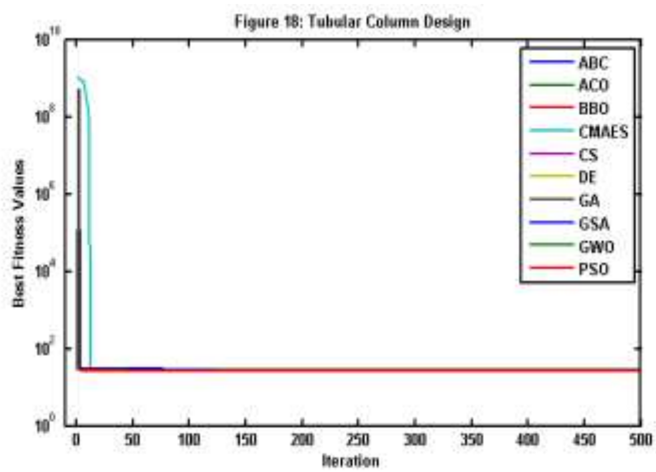
Figure 14: Three-Bar Truss Design

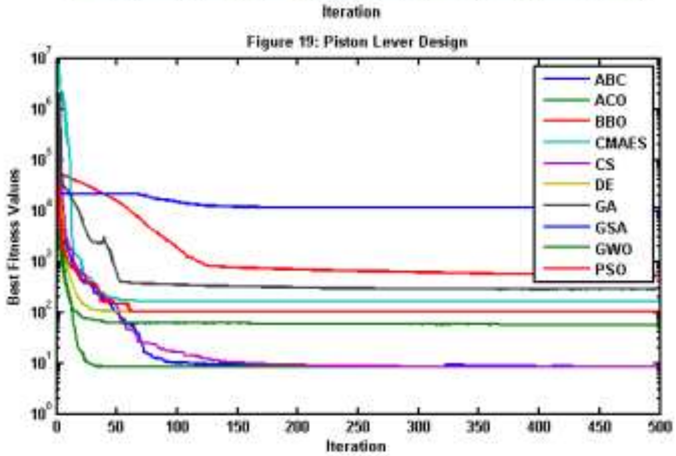


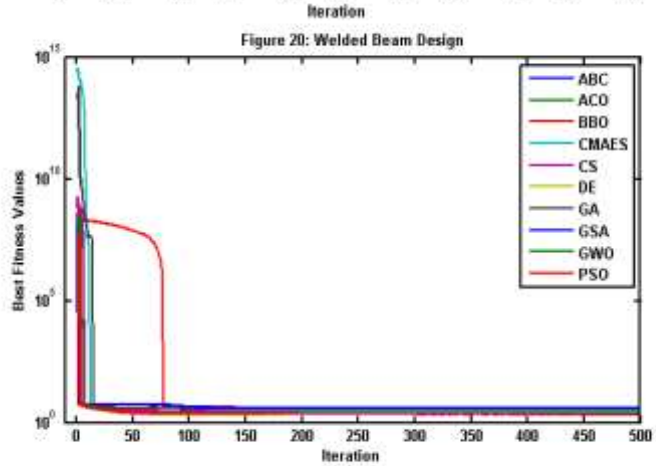












6. Findings

This work presents a comparative study of teen well-known metaheuristic algorithm's performance on teen engineering design problems. Regarding the speed reducer design problem, DE, ACO and ABC provided the best solution, but GSA provided the worst. Regarding the tension/compression spring design problem, all the algorithms are very competitive in the sense of solution quality. Regarding the pressure vessel design problem, DE provided the best solution, CS provided a very competitive answer, and GSA provided the worst answer. On the three-bar truss design problem, DE and CS provided the same better answer, while the rest of the algorithms provided very close answers to the best answer. PSO provided the best answer to the gear train design problem, while, on the cantilever beam design problem, DE and ACO provided the best solution. DE also provided the best solution on the one beam vertical deflection design problem, welded beam design problem and tubular column design problem. Finally, ACO provided the best solution on the welded beam design problem. Overall comparison between the performances of the algorithms demonstrated that all the algorithms provide a competitive answer. The ranking of the algorithms in terms of providing the best solution for each problem is presented in table 12 in below.

Table 12. The ranking table of algorithms in solving engineering design problems

Ranking	Engineering Design Problems									
	Prob 1	**Prob 2**	**Prob 3**	**Prob 4**	**Prob 5**	**Prob 6**	**Prob 7**	**Prob 8**	**Prob 9**	**Prob 10**
1	ABC	DE	DE	DE	PSO	ACO	PSO	DE	ACO	DE
2	ACO	CS	GWO	CS	GSA	DE	DE	PSO	CS	GWO
3	DE	GWO	CS	ACO	DE	PSO	CS	CS	ABC	CS
4	CS	PSO	GA	GWO	BBO	CMAES	GWO	ACO	GWO	PSO
5	GA	ABC	PSO	CMAES	CS	GWO	ACO	GWO	PSO	ACO
6	GWO	GSA	ABC	ABC	GA	CS	GA	GSA	DE	CMAES
7	PSO	ACO	ACO	GSA	GWO	GA	ABC	ABC	CMAES	ABC
8	BBO	GA	CMAES	GA	ACO	ABC	CMAES	CMAES	GA	GA
9	CMAES	CMAES	GSA	PSO	ABC	GSA	BBO	GA	BBO	BBO
10	GSA	BBO	BBO	BBO	CMAES	BBO	GSA	BBO	GSA	GSA

7. Conclusion

This work presents a comparative study on the performance of the well-known metaheuristic optimization algorithms: ABC, ACO, BBO, CS, CMAES, DE, GA, GSA, GWO and PSO in solving the teen real structural design problem. The teen problems are, namely, speed reducer design, tension/compression spring design, pressure vessel design, Three-bar truss design, gear train design, cantilever beam design, I-beam vertical deflection design, tubular column design, piston lever design and welded beam design. Although the best solutions for all the problems are provided by ABC, DE, ACO, PSO and CS, other algorithms have also shown a competitive answer. The algorithms, namely, BBO, GSA and CMAES provided all the worst solutions. The study demonstrates that the DE is much better than the other algorithms in terms of solution quality. Parameter settings also have effects in performance of the algorithms, of course for different parameter settings one probably finds different results. Finally, from the comparative study of algorithms, it can be concluded that in solving engineering design problems, it will be important to choose a suitable metaheuristic algorithm in order to achieve the best possible answer.

Funding: This research received no external funding.
Conflicts of Interest: The authors declare no conflict of interest.
Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

[1] Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., & Gandomi, A. H. (2021). The arithmetic optimization algorithm. Computer methods in applied mechanics and engineering, 376, 113609.
[2] Bansal, J. C., Singh, P. K., & Pal, N. R. (Eds.). (2019). Evolutionary and swarm intelligence algorithms (Vol. 779). Cham: Springer.
[3] Bansal, J. C., Sharma, H., Jadon, S. S., & Clerc, M. (2014). Spider monkey optimization algorithm for numerical optimization. Memetic computing, 6, 31-47.
[4] Corne, D., Dorigo, M., Glover, F., Dasgupta, D., Moscato, P., Poli, R., & Price, K. V. (Eds.). (1999). *New ideas in optimization*. McGraw-Hill Ltd., UK.
[5] Dorigo, M., & Stützle, T. (2004). Ant Colony Optimization, Bradford Publisher.
[6] Hansen, N., Ostermeier, A., & Gawelczyk, A. (1995, July). On the Adaptation of Arbitrary Normal Mutation Distributions in Evolution Strategies: The Generating Set Adaptation. In ICGA (pp. 57-64).
[7] Holland, J. H. (1992). Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press.
[8] Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In Proceedings of ICNN'95-international conference on neural networks (Vol. 4, pp. 1942-1948). IEEE.
[9] Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization (Vol. 200, pp. 1-10). Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.
[10] Layeb, A. (2022). Tangent search algorithm for solving optimization problems. *Neural Computing and Applications*, *34*(11), 8853-8884.
[11] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. Advances in engineering software, 69, 46-61.
[12] Mirjalili, S. (2016). SCA: a sine cosine algorithm for solving optimization problems. Knowledge-based systems, 96, 120-133.
[13] Rechenberg, I. (1973). Evolutionsstrategie. Optimierung technischer Systeme nach Prinzipien derbiologischen Evolution.
[14] Rajabioun, R. (2011). Cuckoo optimization algorithm. Applied soft computing, 11(8), 5508-5518.
[15] Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: a gravitational search algorithm. Information sciences, 179(13), 2232-2248.

[16] Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2012). Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. *Information sciences, 183*(1), 1-15.
[17] Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization, 11*, 341-359.
[18] Simon, D. (2008). Biogeography-based optimization. IEEE transactions on evolutionary computation, 12(6), 702-713.
[19] Salimi, H. (2015). Stochastic fractal search: a powerful metaheuristic algorithm. Knowledge-based systems, 75, 1-18.
[20] Yang, X. S. (2020). Nature-inspired optimization algorithms: Challenges and open problems. *Journal of Computational Science, 46*, 101104.
[21] Yang, X. S. (2010). Firefly algorithm, stochastic test functions and design optimisation. *International journal of bio-inspired computation, 2*(2), 78-84.
[22] Yang, X. S., & Deb, S. (2009, December). Cuckoo search via Lévy flights. In 2009 World congress on nature & biologically inspired computing (NaBIC) (pp. 210-214). Ieee.
[23] Yang, X. S., & Deb, S. (2010). Engineering optimisation by cuckoo search. *International Journal of Mathematical Modelling and Numerical Optimisation, 1*(4), 330-343.
[24] Yang, X. S., & Deb, S. (2009, December). Cuckoo search via Lévy flights. In 2009 World congress on nature & biologically inspired computing (NaBIC) (pp. 210-214). Ieee.
[25] Zhao, J., Tang, D., Liu, Z., Cai, Y., & Dong, S. (2020). Spherical search optimizer: a simple yet efficient meta-heuristic approach. *Neural Computing and Applications, 32*, 9777-9808.