Received: June 22, 2020. Revised: August 24, 2020.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Cicada Swarm Optimization: A New Method for Optimizing Persistent Problems

Hanan A. R. Akkar[1] **Sameem Abbas Salman**[1]*

[1]*Electrical Engineering Department, University of Technology, Baghdad, Iraq*
* Corresponding author's Email: bsceng2006@yahoo.com

Abstract: This paper proposes a new meta-heuristic swarm optimization algorithm called Cicada Swarm Optimization (CISO) algorithm, which mimics the behavior of bio-inspired swarm optimization methods. The CISO algorithm is tested with 23 benchmark functions and taken two problems engineering design, pressure vessel problem and himmelblau's problem. The performance of CISO algorithm is compared with meta-heuristic well-known and recently proposed algorithms (Cockroach Swarm Optimization (CSO), Grasshopper Optimization algorithm (GOA) and Particle Swarm Optimization (PSO)). The obtained results showed that the proposed algorithm succeeded in improving the test functions and solved engineering design problems that could not be improved by other algorithms according to the chosen parameters and the limits of the research space, also showed that CISO has a faster convergence with the minimum number of iterations and also have an accurate calculation efficiency Satisfactory compared to other optimization algorithms.

Keywords: Cicada swarm optimization (CISO), Cockroach swarm optimization (CSO), Grasshopper optimization algorithm (GOA), Particle swarm optimization (PSO).

1. Introduction

1.1 Motivation

Meta-heuristics are a type of approximate methods, that is generally designed to optimize difficult combinatorial optimization problems, in which exact and classical heuristics methods have unable to be efficient and effective. They provide common frameworks that permit for generating new hybrids by merging different algorithms derived from heuristics models such as; AI biologically inspired models, NN systems, and statistical mechanics [1]. Meta-heuristics have been extremely successful in obtaining optimal or close to optimal solutions for many applications of optimizations in multiple areas better than other classical exact and heuristics. Dislike other optimization methods, meta-heuristics algorithms permit the optimization of big size problem by obtaining suitable results in practical periods. Meta-heuristics have taken more popularity in the last few decades [2]. They have been used in many practical applications due to their high performance and success to solve many complicated optimization problems. Applications of meta-heuristic divided into many different fields; some of these fields include machine learning, topology and parameters optimization; data mining; and system modelling [3]. To design meta-heuristic optimization algorithm, three concepts must be considered; (1) Parameters of the problem, the problem can be divided into continuous or discrete depending on the parameters. (2) Constraints of variables, Optimization problems can be classified into constrained and unconstrained ones based on the type of constraints. (3) The cost function of a given problem, the problem can be divided into single-objective and multi-objective problems [4]. Based on the above three points, we need to select the optimization algorithm according to the parameter type, constraint and target number. The development of optimization algorithms is relatively mature at present, and many excellent optimization algorithms have been applied in various fields [5].

In this study, a new nature-inspired algorithm, called Cicada Swarm Optimization (CISO), is proposed to compete with the current optimization algorithms. The main inspiration of the proposed algorithm is the navigating mechanism of Cicada insect in nature and behavior of searching to discovered a best location. The paper first proposes the mathematical model of path Cicada. An optimization algorithm is then proposed using the mathematical model to solve optimization problems in different fields. Although many optimization algorithms face with setting of multiple control parameters, the lack of control parameters is the important strong point of CISO.

1.2 Related works

In the recent years, an increasing number of the meta-heuristic methods have been introduced. The researchers divided the meta-heuristic algorithms into three types, which are based on the principles of biological evolution, population and physical phenomena [6-9]. The evolutionary approach is inspired by the concept of natural evolution. The population-based optimization algorithm is mainly inspired by the social behavior of animal groups, while the physical phenomenon based method mainly imitates the physical rules of the universe. These methods can be summarized into: Particle Swarm Optimization (PSO) [10]; which is inspired by the social behaviors of animals, such as birds and fishes , Genetic algorithm (GA) [11];which is derived from the genetic law and reproduction is indeed based on the Darwin’s theory, Ant Colony Optimization (ACO) [12]; which uses the seeking behavior of the ants , Firefly Algorithm (FA) [13]; which is modelled by observation of the flicker fireflies ,Charged System Search (CSS) [14]; based on some principles from physics and mechanics which each agent is a Charged Particle, Artificial Chemical Reaction Optimization Algorithm (ACROA) [15]; based on chemical reactions possess, Ray Optimization (RO) [16] ; in which each factor is considered as a beam of light and moves in the search space to find the optimum point, Colliding Bodies Optimization (CBO) [17]; which is based on one-dimensional collisions between bodies , Crow Search Algorithm (CSA) [18]; which works based on intelligent behaviors of crows , Kidney-inspired Algorithm (KA) [19]; which uses the kidney process in the human body, Optimal Foraging Algorithm (OFA) [20]; which is inspired by the animal Behavioral Ecology Theory ,Grasshopper Optimization Algorithm (GOA) [21];which is mimics the behavior of Grasshopper insect ,and Rain optimization algorithm (ROA)[22]; which is inspired by the raindrops.

Since none of the algorithms mentioned or others claim that they work to improve all kinds of problems i.e., linear and/or non-linear, constrained and/or non-constrained problems, there are still many opportunities to explore new innovative methods.

The rest of the paper is structured as follows. Section 2 describes the CISO swarm optimization algorithm developed in this study. Section 3 tests the performance of the algorithm on the unimodal functions, multimodal functions and fixed-dimension multimodal functions. Section 4 Results and Discussion. Section 5 applies the CISO algorithm to the Engineering Design problems to further test the performance of the algorithm. Section 6 Conclusions.

2. Cicada swarm optimization method (CISO)

Insects have a shifting chemical sensory system that senses various environmental stimuli and guides their behaviors [23]. The antennae of insects are important chemical receptors. They mainly play olfactory and tactile effects, and some even have an auditory function. They can help insects communicate, find the opposite sex, find food and choose spawning sites. People often use this property of insects to release substances with specific volatile odors to attract or evade insects harmful to plants [24]. The cicada is characterized by two antennae, sometimes up to two times the length of its body. This kind of antenna has two basic functions: one is to explore the surrounding environment. For example, when encountering an obstacle, the feeler can perceive its size, shape, and hardness. The second is to capture the smell of food or find potential mates by swinging the body’s antenna. When a higher concentration of odor is detected on one side of the antenna, the insect will rotate in the same direction, otherwise, it will turn to the other side. According to this simple principle, cicada can effectively find food [25-26]. With the continuous deepening of the experiment, we found that the performance of the CSO algorithm in dealing with high-dimensional functions is not very satisfactory, and the iterative result is very dependent on the initial position of the cockroach. In other words, the choice of an initial position greatly affects the efficiency and effectiveness of optimization. Inspired by the swarm optimization algorithm, we have made further improvements to the CSO algorithm by expanding an individual to a group. The cicada Algorithm can automatically realize the optimization process without knowing the specific form of the function and gradient information. The major advantage of the CISO is the lesser complexity involved in its design

Received: June 22, 2020. Revised: August 24, 2020.

and in its ability to solve the optimization problem in less time and high accuracy. When using CISO to optimize nonlinear systems, a simple two-step building procedure is employed as follows: (i) model the searching behavior; (ii) formulate the behavior of detecting. In this algorithm, the position of cicada at time $\boldsymbol{t}$ ($\boldsymbol{t} = 1,2,\dots$) is denoted as $\boldsymbol{x^t}$, denote the concentration of odor at position $\boldsymbol{x}$ to be $\boldsymbol{f}\ (\boldsymbol{x})$ known as a fitness function, where the maximum (or minimum) value corresponds to the point of odor source. Each cicada represents a potential solution to the optimization problem, and each cicada corresponds to a fitness value determined by the fitness function. Similar to the particle swarm algorithm, the cicada also share information, but the distance and direction of the cicada are determined by their speed and the intensity of the information to be detected by their two antennae. We borrowed the mathematical idea to CISO from two methods, particle swarm algorithm, and cockroach swarm algorithm. There is a population of $\boldsymbol{n}$ cicada represented as $\boldsymbol{X} = (\boldsymbol{X_1}, \boldsymbol{X_2}, \dots, \boldsymbol{X_n})$ in an S-dimensional search space, where the $\boldsymbol{i}$ th cicada represents an S-dimensional vector $\boldsymbol{X_i} = (\boldsymbol{x_{i1}}, \boldsymbol{x_{i2}}, \dots, \boldsymbol{x_{iS}})^{\boldsymbol{T}}$,represents the position of the $\boldsymbol{i}$ th cicada in the S-dimensional search space, and also represents a potential solution to the problem. According to the target function, the fitness value of each cicada position can be calculated. The speed of the $\boldsymbol{i}$ the cicada is expressed as $\boldsymbol{V_i} = (\boldsymbol{V_{i1}}, \boldsymbol{V_{i2}}, \dots, \boldsymbol{V_{iS}})^{\boldsymbol{T}}$ .The individual extremity of the cicada are represented as $\boldsymbol{P_i} = (\boldsymbol{P_{i1}}, \boldsymbol{P_{i2}}, \dots, \boldsymbol{P_{iS}})^{\boldsymbol{T}}$,and the group extreme value of the population is represented As $\boldsymbol{P_g} = (\boldsymbol{P_{g1}}, \boldsymbol{P_{g2}}, \dots, \boldsymbol{P_{gS}})^{\boldsymbol{T}}$ .The mathematical model for simulating its behavior is as follows:

$$X_{is}^{k+1} = X_{is}^{k} + \lambda V_{is}^{k} + (1-\lambda)\xi_{is}^{k} \tag{1}$$

Where $\boldsymbol{s = 1,2,\dots,S}$; $\boldsymbol{i = 1,2,\dots,n}$; $\boldsymbol{k}$ is the current number of iterations. $\boldsymbol{V_{is}}$ is expressed as the speed of cicada, and $\boldsymbol{\xi_{is}^k}$ represents the increase in cicada position movement. $\boldsymbol{\lambda}$ is a positive constants. Then the speed formula is written as:

$$\begin{aligned} V_{is}^{k+1} = wV_{is}^{k} + c_1 r_1\left(P_{is}^{k} - X_{is}^{k}\right) \\ +c_2 r_2\left(P_{gs}^{k} - X_{gs}^{k}\right) \end{aligned} \tag{2}$$

Where $\boldsymbol{c_1}, \boldsymbol{c_2}$ are two positive constants, and $\boldsymbol{r_1}, \boldsymbol{r_2}$ are two random functions in the range [0,1]. $\boldsymbol{w}$ is the inertia weight. In the standard PSO algorithm, $\boldsymbol{w}$ is a fixed constant, but with the gradual improvement of the algorithm, many scholars have proposed a changing inertia factor strategy. This paper adopts the strategy of decreasing inertia weight, and the formula is as follows:

$$w = w_{max} - \frac{w_{max} - w_{min}}{K} * k \tag{3}$$

Where $\boldsymbol{w_{min}}$ and $\boldsymbol{w_{max}}$ respectively represent the minimum and maximum value of $\boldsymbol{w}$. $\boldsymbol{K}$ and $\boldsymbol{k}$ are the current number of iterations and the maximum number of iterations. In this paper, the maximum value of $\boldsymbol{w}$ is set to 0.9, and the minimum value is set to 0.4, so that the algorithm can search a larger range at the beginning of evolution and find the optimal solution area as quickly as possible. As $\boldsymbol{w}$ gradually decreases, the cicada speed decreases and then enters local search. The $\boldsymbol{\xi}$ function, which defines the incremental function, is calculated as follows:

$$\xi_{is}^{k+1} = \delta^{k} * V_{is}^{k} * sign(f\left(X_{rs}^{k}\right) - f\left(X_{ls}^{k}\right)) \tag{4}$$

In this step, we find a high dimension mechanism formula to detect the behaviors of cicada. $\boldsymbol{\delta}$ Indicates step size. $\boldsymbol{sign}(.)$ Represents a sign function. The search behaviors of the right antenna and the left antenna are respectively expressed as:

$$X_{rs}^{k+1} = X_{rs}^{k} + V_{is}^{k} * d/2 \tag{5}$$

$$X_{ls}^{k+1} = X_{ls}^{k} + V_{is}^{k} * d/2 \tag{6}$$

Where $\boldsymbol{d}$ represents the distance between the two antennae of cicada, can calculate by Eq. (7).

$$d^t = {\delta^t}/{c_2} \tag{7}$$

Where $\boldsymbol{\delta}$ represents the step size of cicada, can be found by Eq. (8).

$$\delta^t = c_1\delta^{t-1} + \delta^o \quad or \quad \delta^t = eta * \delta^{t-1} \tag{8}$$

Where $\boldsymbol{c_1}$, $\boldsymbol{c_2}$ and $\boldsymbol{eta}$ are constants to be adjusted by designers, we recommend eta's value is 0.95.

Fig. 1 shows the trajectories of the Cicada swarm in three-dimensional space. To represent the search path more visually, we used a small population size and showed the location change process of 20 iterations in 3D space. Because factors such as step length and inertial weight coefficient are decreasing in the iterative process, the algorithm will not converge to the target point too quickly, thus avoiding the group falling into the local optimum greatly. The CISO algorithm first initializes a set of random solutions. At each iteration, the search agent

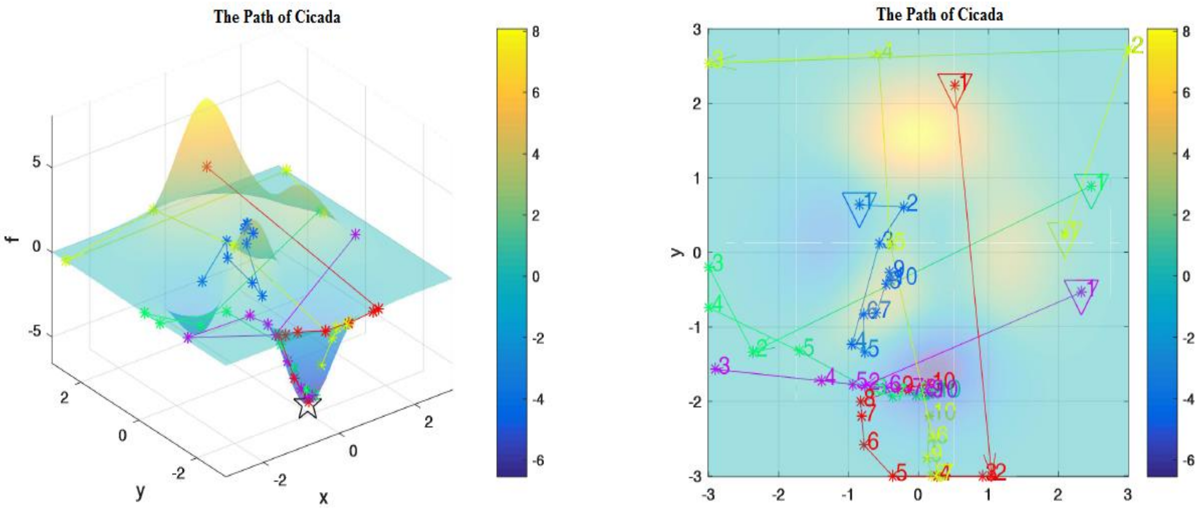


Figure. 1 Cicada search path in 3D space

Table 1. CISO detailed pseudo code

```
Procedure
Initialize the swarm $X_i (i = 1,2, \dots, n)$
Initialize population speed $V$
Set step speed $\delta$ , speed boundary $V_{max}$ and $V_{min}$ ,population size, sizepop and maximum number of iterations $K$
Calculate the fitness of each search agent
While( $k < K$)
    Set inertia weight $w$ using Eq. (3)
    Update $d$ using Eq. (7)
    For each search agent
      Calculate $f(X_{rs})$ and $f(X_{ls})$ using Eq. (5) and Eq. (6)
          Update the incremental function $\xi$ by the Eq. (4)
          Update the speed formula $V$ by the Eq. (2)
          Update the position of the current search agent by the Eq. (1)
    End for
    Calculate the fitness of each search agent $f\ (x)$
    Record and store the location of each search agent
    For each search agent
If $f(x) < f_{pbest}$
            $f_{pbest} = f\ (x)$
          End if
        If $f(x) < f_{gbest}$
            $f_{gbest} = f\ (x)$
        End if
        End for
Update $x$ if there is a better solution
Update step factor $\delta$ by the Eq. (8)
End while
Return $x_{best}, f_{best}$
```

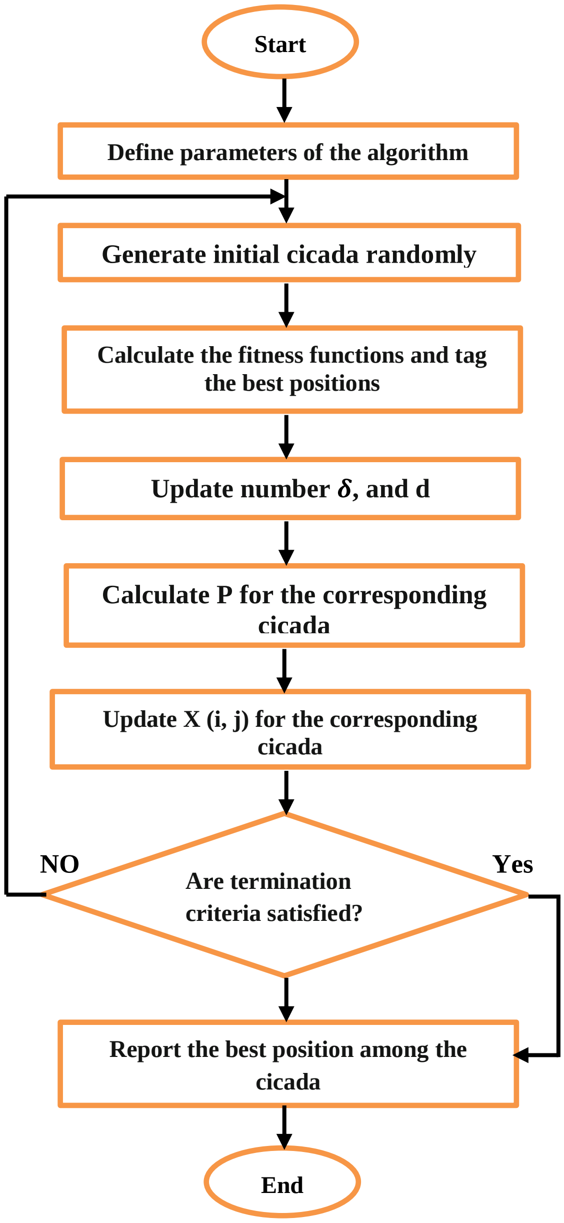

Start
Define parameters of the algorithm
Generate initial cicada randomly
Calculate the fitness functions and tag the best positions
Update number δ, and d
Calculate P for the corresponding cicada
Update X (i, j) for the corresponding cicada
NO
Are termination criteria satisfied?
Yes
Report the best position among the cicada
End

Figure. 2 CISO algorithm flowchart

Received: June 22, 2020. Revised: August 24, 2020.

Table 2. Characteristics of the CISO algorithm

General algorithm	Cicada Swarm optimization
Decision variable	Cicada positions in each dimension
Solution	Cicada positions
Old solution	Old positions of Cicada
New solution	New positions of Cicada
Best solution	Positions of target
Fitness function	Distance between Cicada and target
Initial solution	Random positions of Cicada
Process of generating new solution	Flying in a spiral path toward a target

updates its location based on its own search mechanism and the best solution currently available. The combination of these two parts can not only accelerate the population's iteration speed, but also reduce the probability of the population falling into the local optimum, which is more stable when dealing with high-dimensional problems. Table 1 illustrates the detailed CISO algorithm pseudo-code, Table 2 illustrates the Characteristics of the CISO algorithm, while Fig. 2 shows the proposed algorithm flowchart.

In theory, the CISO algorithm includes exploration and exploitation ability, so it belongs to global optimization. Furthermore, the linear combination of speed and Cicada search enhances the rapidity and accuracy of population optimization and makes the algorithm more stable. In the next section, we will examine the performance of the proposed algorithm through a set of mathematical functions.

3. Tests performance of algorithm

In the optimization field, a set of mathematical functions with optimal solutions is usually used to test the performance of different optimization algorithms quantitatively and the test functions should be diverse so that the conclusions are not too one-sided. In this paper, three groups of test functions with different characteristics are used to benchmark the performance of the proposed algorithm which are unimodal functions, multimodal functions and fixed-dimension multimodal functions [27-30].The specific form of the function is given in Tables (3)-(5), where Dim represents the dimension of the function, Range represents the range of independent variables, that is, the range of population, and f_{min} represents the minimum value of the function.

Fig. 3. shows the two-dimensional versions of the unimodal function, multimodal function and fixed-dimension multimodal function respectively. The unimodal test function has only one global optimal solution, which is helpful to find the global optimal solution in the search space, and it can test the convergence speed and efficiency of the algorithm well. From Fig. 4, it can also be seen that the multimodal function, and the fixed-dimension multimodal test function have multiple locally optimal solutions, which can be used to test the algorithm to avoid the performance of the local optimal solution, and the fixed-dimension multimodal function compared with unimodal test function is more challenging. In the part of qualitative analysis, six typical test functions are provided, including optimal trajectory map, contour map and convergence curve of the search path. In the quantitative analysis part, 50 search agents were used, the maximum number of iterations was set to 1000, and each test function was run 30 times to generate statistical results. Quantitative evaluation was performed using the mean, standard deviation, and program performance time of three performance indicators. Statistical results are reported in Table 6. CISO was compared with PSO [10], CSO [31], and GOA [21].

4. Results and discussion

The numerical analysis has been included testing of benchmark functions, statistical analysis of multi-objective functions, implementing to design problems and implementing of engineering application. In the next subsection, we discussed these problems respectively. Also, all methods have been coded in MATLAB 2018 software, and PC with Intel i7 CPU, 8 GB RAM and Nvidia GTX950 hardware has been used for all simulations. Note that, all methods have been simulated in same conditions (same population size and number of iterations).

4.1 Qualitative results

In this paper, six unimodal, multimodal and fixed-dimension multimodal functions are selected to observe the CISO algorithm's optimization behavior. In order to express the optimization trajectory more intuitively, we use five search agents. Fig. 4 shows the optimal trace of each test function, the contour map of the search path, and the convergence curves. The optimal trajectory gives the best cicada optimization route. Since the initial position of the cicada is randomly generated, the optimal trajectory may be different when reproducing the result. The contour map of the search path can more intuitively display the cicada's trajectory, and connecting the same z-values on the x, y plane makes it easier to observe cicada movements. The convergence curve shows the function value of the best solution obtained so far. From Fig. 4 it can be seen that cicada gradually move to the best point and eventually gather around the global best point. This phenomenon can be observed in unimodal, multimodal, and fixed-dimension multimodal functions. The results show that the CISO algorithm has a good balance between exploration and exploitation capabilities to promote the cicada to move to the global optimum. In addition, in order to more clearly represent the trajectory of the cicada, some of the function images are processed. Such as $f10$, this paper selects the opposite form and can more intuitively observe the optimal trajectory. The CISO algorithm of the cicada self-optimization mechanism has been added, which can more intelligently avoid local optimums. During the optimization process, we found that some cicada always move quickly toward the maximum value, and then reach the maximum value and then perform normal iterations. This mechanism makes the cicada cleverly avoid the local optimum during the optimization process. For unimodal and multimodal functions, the advantage of the self-optimization mechanism is even more pronounced. Fig. 5 provides a convergence curve to further prove that this mechanism can improve the search results. The convergence curve clearly shows the descending behavior of all test functions. Observe that the CISO search agent suddenly changes during the early stage of the optimization process, and then gradually converges. this behavior ensures that the algorithm quickly converges to the optimal point to reduce the iteration time.

4.2 Quantitative results

The above discussion proves that the proposed algorithm can solve the optimization problem, but pure qualitative test can not prove the superiority of the algorithm. This section raises the dimensions of test functions other than fixed dimensions to 30 dimensions and gives quantified results. Table 6 gives the experimental results of the test function. As shown in Table 6, when dealing with the unimodal functions, the processing speed of CISO is comparable to that of PSO, but it is obviously better than CSO and GOA algorithm. In addition, compared with the other three algorithms, CISO algorithm is more stable in performance. Adding the cicada search mechanism in the process of optimization makes the algorithm have better global optimization performance, accelerates the convergence speed of the algorithm, and effectively avoids the phenomenon of "premature". When dealing with multimode functions, CISO algorithm shows good performance again. Because multimodal functions have multiple local optimal solutions, the results can be directed to show that CISO algorithm is effective and efficient in avoiding local optimal solutions. For the fixed-dimension multimodal functions, the proposed algorithm gives very competitive results. The CISO algorithm has the ability to balance the exploration and exploitation of the individual and can solve more challenging problems.

Convergence curves of CISO, CSO, GOA and PSO are compared in Fig. 5 for all of the test functions. The figure shows that CISO has good processing ability for unimodal functions, multimodal functions and fixed-dimension functions, and the processing process is very stable. Especially when solving more complex fixed-dimension functions, CISO shows more obvious advantage than other algorithms. It can be seen that CISO is enough competitive with other state of the art metaheuristic algorithms. As a summary, the results of this section revealed different characteristics of the proposed CISO algorithm. Efficient and stable search capabilities benefit from cicada-specific optimization features. The increase in the exploration function of the left and right must greatly improve the stability of the search, making the exploration and exploitation capabilities more balanced, and the CISO can handle better for high-dimensional and more complex problems. Overall, the success rate of the CISO algorithm seems to be higher in solving challenging problems. In the next sections, CISO performance is validated on more challenging multi-objective issues. The summery results of proposed method show that Functions F1 to F7 are unimodal. The mean results of 30 times of the algorithm's independent running, are shown in Table 6. These results show that the proposed CISO has a better performance in all F1 to F7 functions than other algorithms. In multimodal functions of F8 to F13, by increasing the function dimensions, the number of local responses is increased exponentially. Therefore, arriving at the minimum response of these functions is hardly possible. In these types of functions, arriving at a response close to the ideal response represents the algorithm's high power in passing through the local wrong responses. Results gained from assessing F8 to F13 after 30 times running of CISO and other algorithms are shown in Table 6. In all these functions, CISO shows a better performance.

Table 3. Description of unimodal benchmark functions

Function	**Dim**	**Range**	$\boldsymbol{f_{min}}$
$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	$[-100,100]$	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	$[-10,10]$	0
$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j \right)^2$	30	$[-100,100]$	0
$f_4(x) = max_i\{\lvert x_i \rvert, 1 \le i \le n\}$	30	$[-100,100]$	0
$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-30,30]$	0
$f_6(x) = \sum_{i=1}^{n} ([\, x_i + 0.5])^2$	30	$[-100,100]$	0
$f_7(x) = \sum_{i=1}^{n} ix^4 + random[0,1]$	30	$[-1.28,1.28]$	0

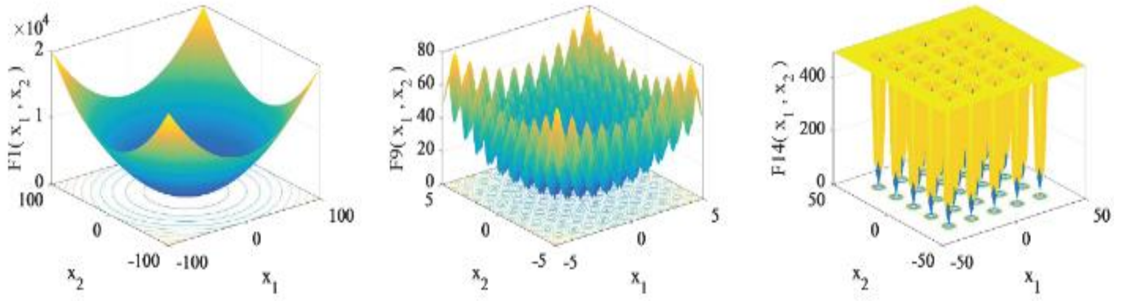
Table 4. Description of multimodal benchmark functions

Function	**Dim**	**Range**	$\boldsymbol{f_{min}}$
$f_8(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert}\,)$	30	$[-500,500]$	0
$f_9(x) = \sum_{i=1}^{n} [\, x_i^2 - 10\cos(2\pi x_i) + 10]$	30	$[-5.12,5.12]$	0
$f_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20$	30	$[-30,30]$	0
$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$	30	$[-600,600]$	0
$f_{12}(x) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10sin^2(\pi\, y_i + 1)] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(\, x_i, 10,100,4)$ $y_i = 1 + \frac{x_i + 1}{4}, u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m x_i > a \\ -a < x_i < a \\ k(-x_i - a)^m x_i > -a \end{cases}$	30	$[-50,50]$	0
$f_{13}(x) = 0.1\left\{sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2[1 + sin^2(\,3\pi x_i + 1)] + (x_n - 1)^2[1 + sin^2(2\pi x_n)]\right\} + \sum_{i=1}^{n} u(\, x_i, 5,100,4)$	30	$[-50,50]$	0

Functions F14 to F23 have both fixed-dimensions and low local responses. Results obtained from 30 times running of CISO and other algorithm, are shown in Table 6. These results represent the suitable performance of CISO in relation to other algorithms.

Table 5. Description of fixed-dimension multimodal benchmark functions

Function	**Dim**	**Range**	f_{min}
$f_{14}(x) = (\frac{1}{500} + \sum_{j=1}^{25} (j + \sum_{i=1}^{2} (x_i - a_{ij})^6)^{-1})^{-1}$	2	[−65,65]	0.998
$f_{15}(x) = \sum_{i=1}^{11} [a_i - \frac{x_1(b_i^2 + b_i x_2}{b_i^2 + b_i x_3 + x_4}]^2$	4	[−5,5]	0
$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[−5,5]	−1.03
$f_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10\left(1 - \frac{1}{8\pi}\right) cosx_1 + 10$	2	[−5,5]	0.398
$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$ $\times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[−2,2]	3
$f_{19}(x) = \sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{3} a_{ij}(x_i - p_{ij})^2)$	3	[1,3]	-3.86
$f_{20}(x) = \sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij}(x_i - p_{ij})^2)$	6	[0,1]	−3.32
$f_{21}(x) = -\sum_{i=1}^{5} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	−10.15
$f_{22}(x) = -\sum_{i=1}^{7} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	−10.42
$f_{23}(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	−10.56



(a) (b) (c)

Figure. 3 2D version functions: (a) unimodal function, (b) multimodal function, and (c) fixed-dimension function

5. CISO algorithm for engineering design problems

In this section, the better illustrate superiority and competitiveness of the CISO algorithm in solving optimization problems. The CISO method was applied to solve two well-known constrained engineering design problems, including Himmelblau's problem, and the pressure vessel design. The results of proposed method are compared with the results of other algorithms. As seen in the results, the CISO algorithm has found optimum solutions within a lower number of analyses, particularly in engineering problems optimization applications. Finally, the CISO method has an interesting algorithm and is an effective and reliable method in terms of efficiency.

A. Pressure Vessel Design Problem

As shown in Fig. 6, two hemispheres cover the ends of the cylinder to form a pressure vessel. The goal is to minimize the total cost including material costs, welding costs and melding costs [32]:

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

where x_1 is the thickness of the shell (R_s), x_2 is the thickness of the head (R_h), x_3 is the inner radius (r), and x_4 is the length of the section of the cylinder of the container (L). R_s and R_h are integral times of 0.0625, the available thickness of rolled steel plates, r and L are continuous. The constraint function can be stated as follows:

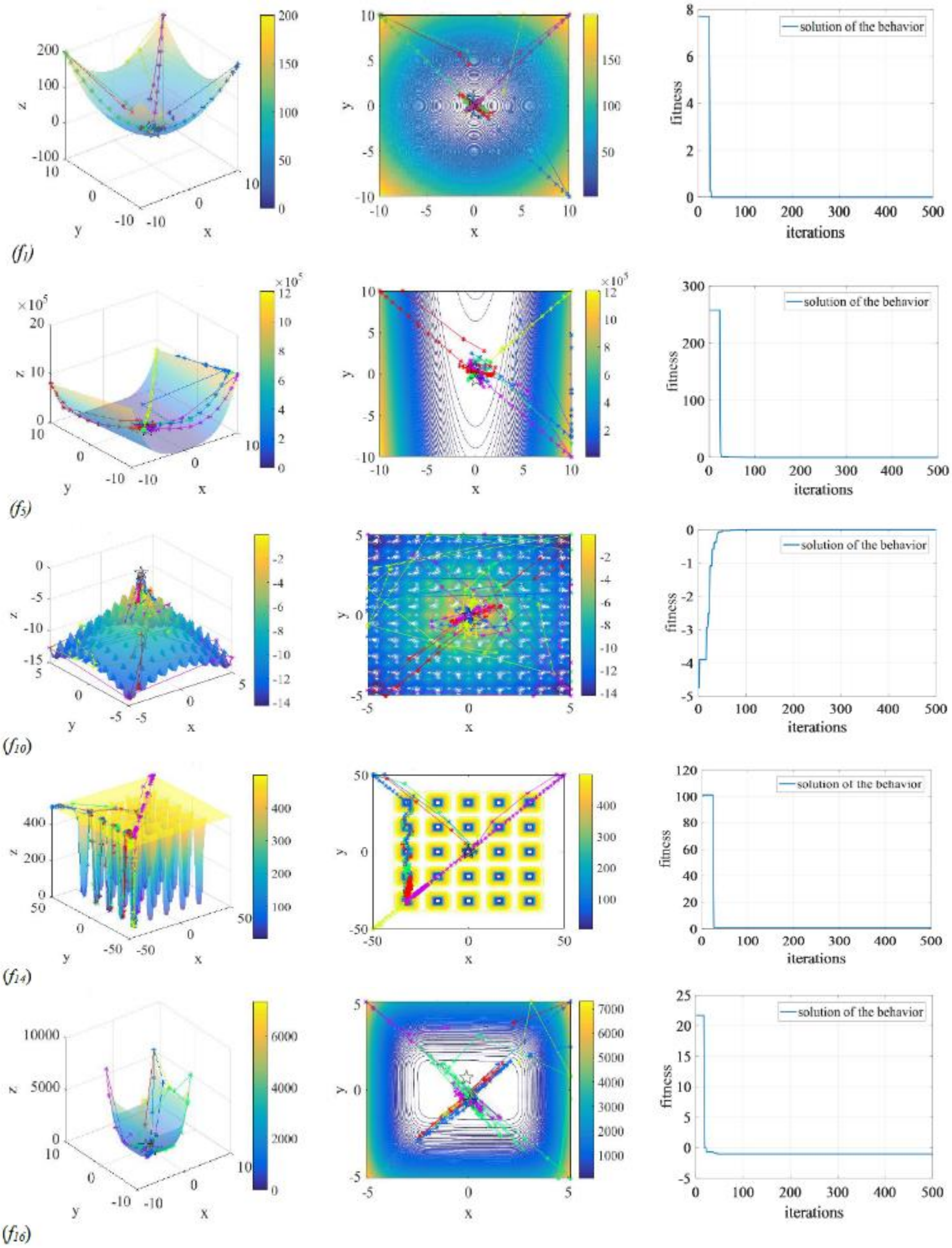


Figure. 4 Behavior of CISO on the 2D benchmark problems

$$
\begin{aligned}
s.t.\ \ & g_1(x) = -x_1 + 0.0193x_3 \leq 0,\\
& g_2(x) = -x_2 + 0.00954x_3 \leq 0,\\
& g_3(x) = -\pi x_3^2 x_4 - \tfrac{4}{3}\pi x_3^3 + 1296000 \leq 0,\\
& g_4(x) = x_4 - 240 \leq 0,\\
& \qquad x_1 \in \{1,2,3,\ldots,99\} \times 0.0625,\\
& \qquad x_2 \in \{1,2,3,\ldots,99\} \times 0.0625,\\
& \qquad x_3 \in [10,200],\\
& \qquad x_4 \in [10,200].
\end{aligned}
$$

Table 7 shows the best results obtained by the CISO algorithm compared to other existing algorithms. The goal of this example is to achieve better results and lower cost (pressure vessel) function. The results show the high performance of the CISO algorithm in approximating the global optimum for this issue. In other words, it's a sufficient search capability, and convergence is quite good. The CISO algorithm iterative process is shown in Fig. 7.

Received: June 22, 2020. Revised: August 24, 2020.

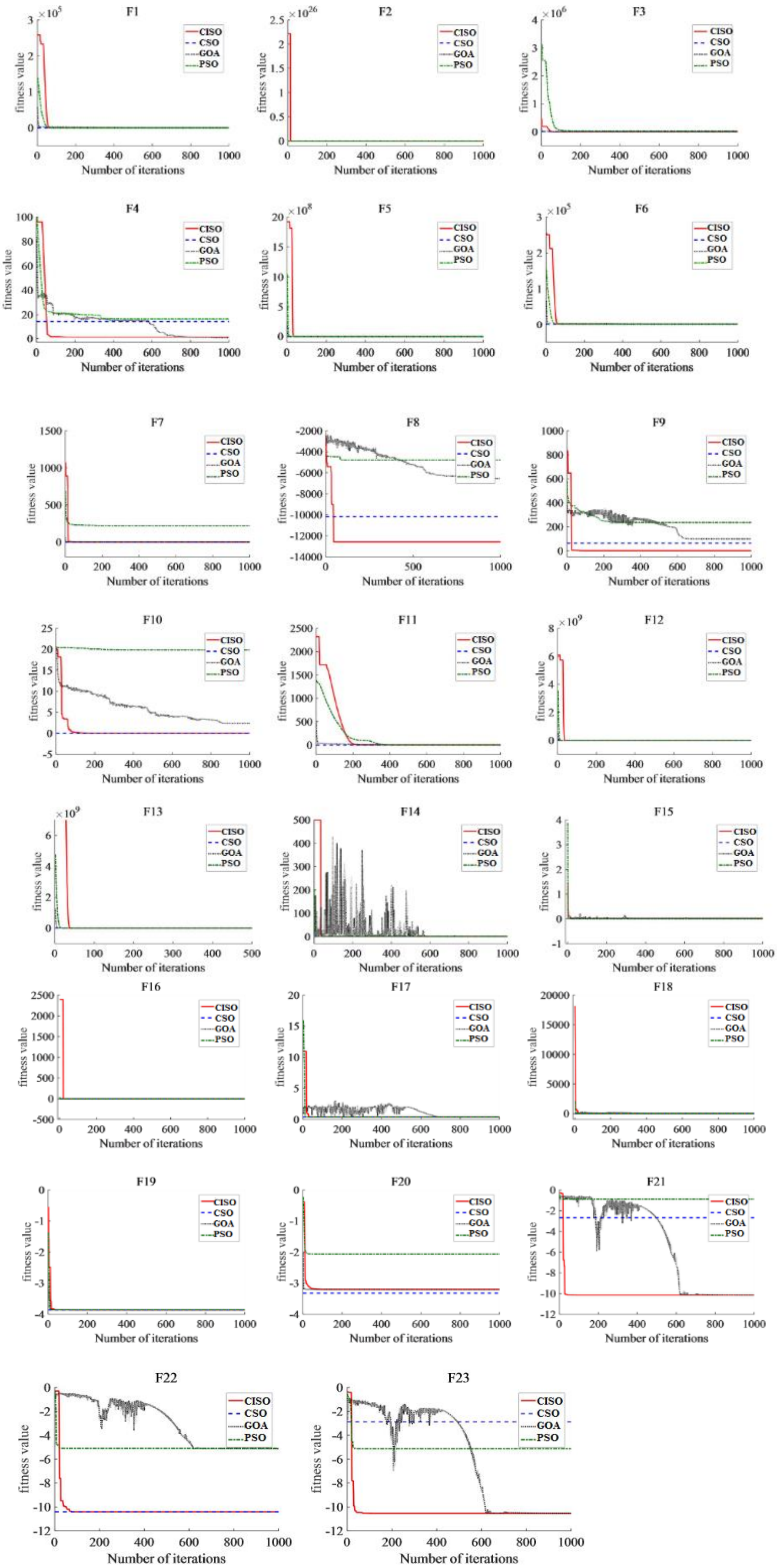


Figure. 5 Comparison of convergence curves of CISO and other optimization algorithms

Received: June 22, 2020. Revised: August 24, 2020.

Table 6. The Comparison of optimization results obtained for the (unimodal, multimodal, and fixed-dimension multimodal) functions

F	CISO			PSO			CSO			GOA		
	ave	*std*	*aver_time (sec)*	*ave*	*std*	*aver_time (sec)*	*ave*	*std*	*aver_time (sec)*	*ave*	*std*	*aver_time (sec)*
F_1	0	9.36E-12	0.5153	0	0	0.4597	0.0025	0.0017	3.7335	0.4004	0.3342	14.5615
F_2	1.02E-04	3.92E-05	0.6217	1.33	3.4575	0.5099	0.008	0.0068	4.9162	1.3612	2.0519	19.638
F_3	0	3.31E-11	0.875	1.67	9.1287	0.636	7.66	2.34	0.608	0	0	2.983
F_4	3.55E-06	1.07E-08	0.547	0	0	0.459	5.772	4.817	3.6927	2.50E-03	1.21E-03	17.5846
F_5	0.0035	1.41E-02	0.725	0.153	3.418	0.5247	6.782	3.267	5.287	3.172	1.649	18.137
F_6	0	0	0.5081	0	0	0.4591	0.0007	0.0011	3.821	0	0	19.509
F_7	5.17E-04	4.47E-03	0.6382	2.98E-03	0.0003	0.5219	0.0019	0.009	2.9028	0.0737	0.1023	13.681
F_8	-4.79	7.34	0.7912	-1.40	8.5742	0.532	-8.97	3.73506	3.802	-1.74	1.832	2.9743
F_9	0.04311	0.9305	0.5215	5.178	2.641	0.5881	7.974	8.7574	3.117	5.302	2.9227	12.5213
F_{10}	0.0197	0.4181	0.6285	3.961	8.4257	0.5253	0.073	0.963	3.7441	0.6931	1.358	2.7189
F_{11}	0.0126	0.0849	0.7103	0.1384	0.0926	0.8518	0.725	0.1001	4.7637	0.1227	0.638	2.158
F_{12}	7.02E-06	3.84E-05	0.8427	0	0	09052	6.1241	5.0446	4.1286	0.1594	0.8141	3.5571
F_{13}	0.0011	0.0034	0.6192	0	0	0.9123	5.765	9.12744	1.5871	0.4825	0.9743	2.8164
F_{14}	0.0998	1.54E-06	1.0257	0.4116	0	3.1104	0.7917	0	2.6241	0.8119	0	8.4716
F_{15}	0.0015	0.0036	0.586	0.0368	0.0117	0.8159	0.0394	0.0781	1.6981	0.035	0.06125	6.5118
F_{16}	-2.1036	6.71E-6	0.9115	0	-1.6882	0	0.9741	-1.0338	0	1.2447	-1.0855	0
F_{17}	0.03971	0	0.6615	0.0928	0.4418	0.5779	0.3975	0	1.2171	0.5519	0	9.2551
F_{18}	0.3118	1.08E-10	0.3894	0.7925	0	0.9774	1.6671	2.457	2.6972	3.179	5.2887	10.3014
F_{19}	-3.8609	0.0034	0.8683	-2.3971	0.1247	0.6522	-1.661	0	1.5924	-0.9275	0.1411	12.6881
F_{20}	-3.1256	0.00891	0.7501	-2.819	0.5567	0.6541	-2.1901	0.5567	0.6478	-1.7089	0.6905	1.894
F_{21}	-9.8164	0.8362	0.5519	-1.0902	1.6894	0.9072	-5.9724	3.37309	1.9346	-2.6099	4.9887	13.5689
F_{22}	-10.0513	1.3381	0.6845	-1.0196	0.7063	1.2563	-3.8974	3.5088	2.1298	-6.9885	3.7405	13.259
F_{23}	-9.1069	1.0111	0.9185	-2.784	0.6276	1.3545	-5.7112	3.5424	2.4851	-5.9841	3.0727	17.2574

Received: June 22, 2020. Revised: August 24, 2020.

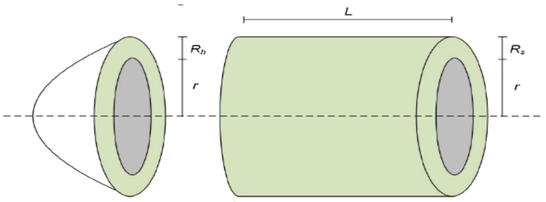


Figure. 6 Pressure vessels

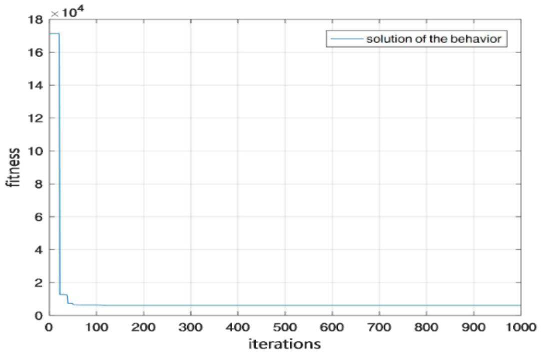


Figure. 7 Iteration process for pressure vessel design

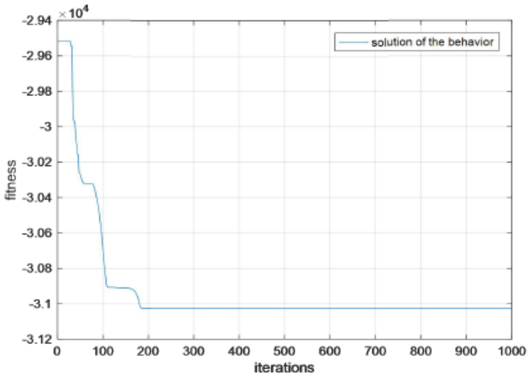


Figure. 8 Iteration process for himmelblau's optimization

B. *Himmelblau's optimization problem*

This problem is proposed by Himmelblau [44] and is a common function for nonlinear constrained optimization problems. It is widely used in the optimization field. It consists of five variables, three equality constraints and six inequality constraints. The specific forms are as follows:

$$
\begin{aligned}
&minimize\ f(x) = 5.35785x_3^2 + 0.83568x_1x_5 \\
&\qquad + 37.29329x_1 - 40792.141 \\
&s.t.\ \ g_1(x) = 85.3344 + 0.005685x_2x_5 \\
&\qquad + 0.00026x_1x_4 - 0.0022x_3x_5, \\
&g_2(x) = 80.51249 + 0.007131x_2x_5 + \\
&\qquad 0.00299x_1x_2 + 0.002181x_3^2, \\
&g_3(x) = 9.30096 + 0.004702x_3x_5 + \\
&\qquad 0.001254x_1x_3 + 0.001908x_3x_4, \\
&0 \le g_1(x) \le 92, \\
&90 \le g_2(x) \le 110, \\
&20 \le g_3(x) \le 25, \\
&78 \le x_1 \le 102, \\
&33 \le x_2 \le 45, \\
&27 \le x_3 \le 45, \\
&27 \le x_4 \le 45, \\
&27 \le x_5 \le 45.
\end{aligned}
$$

Several researchers have employed different algorithms to solve this problem, such as SSA [45], RW - GWO [46], hybrid GSA – GA [47], AQPSO [48], fb_TSA [49], and DGWO [50]. Table 8 shows the best results obtained from the CISO method compared to the previously mentioned methods. The CISO algorithm iterative process is shown in Fig. 8.

6. Conclusion

In this article, a new optimization algorithm, so called the "Cicada swarm optimization" (CISO), is introduced. This algorithm is inspired by the behavior of the movement of cicada insect. In order to evaluate the algorithm, it was examined 23 benchmark test function CISO and three additional optimization algorithms were evaluated, CISO performs well compared to CSO, GOA, and PSO, according to the results. Based on the results achieved for CISO and other mentioned optimization algorithms, it has been

Table 7. Comparisons results for pressure vessel function

Methods	$R_s = (x_1)$	$R_h = (x_2)$	$r = (x_3)$	L= (x_4)	f^*
(PFA) (H.Yapici, 2019) [33]	0.7782	0.3846	40.3196	199.9899	5885.3351
(HHO-SCA) (V.K. Kamboj, 2019) [34]	0.9459	0.4471	46.8513	125.4684	6393.0927
(WOA) (S. Mirjalili and A. Lewis, 2016) [35]	0.8125	0.4375	42.0982	176.6389	6059.7410
(IGMM) (H.Varaee,2017) [36]	0.8125	0.4375	42.0984	176.6365	6059.7143
(LWOA) (Y. Zhou, 2018) [37]	0.7788	0.3853	40.3261	200	5893.3390
(QEA) (M. Kamalinejad, 2019)) [38]	0.7656	0.3789	39.5625	211.1641	5887.3969
(SBO) (M. S. Massoudi, 2020) [39]	0.7781	0.3846	40.3196	199.9999	5885.3326
(CE-CBA) (W. Xiao, 2019) [40]	0.8125	0.4375	42.0998	176.6365	6059.7143
(TEO) (A. Kaveh, and A. Dadras, 2017) [41]	0.7791	0.3852	40.3698	199.3018	5887.5111
(MVO) (S. A. Mirjalili, 2016) [42]	0.8125	0.4375	42.0907	176.7386	6060.8066
(SSOA) (A. Kaveh, 2020) [43]	0.7781	0.3846	40.3201	199.9999	5885.3258
(CISO) Proposed Method	0.7781	0.3846	40.3196	200	**5885.3028**

Table 8. Comparisons results for himmelblau function

Methods	x_1	x_2	x_3	x_4	x_5	f^*
(SSA) (J.Xue and B.Shen, 2020) [45]	78.00	33.00	29.995	45.00	36.775	−30665.537
(RW-GWO) (S.Gupta and K.Deep, 2018) [46]	78.00	33.00	27.078	45.00	44.958	−31024.158
(hybrid GSA-GA) (H.Garg, 2018) [47]	77.96	32.99	27.072	45.00	44.973	−31027.640
(AQPSO) (N.Kumar, 2020) [48]	78.00	33.00	29.996	45.00	36.776	−30665.233
(fb_TSA) (J.Jiang, 2020) [49]	78.01	33.02	27.097	44.982	44.898	−31021.208
(DGWO) (F. Yan, 2019) [50]	78.00	33.00	30.046	45.00	36.710	−30653.30
(CISO) Proposed Method	89.72	35.44	40.377	27.268	33.354	**−32217.431**

demonstrated that CISO can very efficiently manage various kinds of restrictions and provides better solutions. Results obtained from running CISO show that it performs very satisfactorily for all criterion functions. To show the efficiency and robustness of the CISO, two classic engineering problems (design of pressure vessel and Himmelblau's Problem), are solved by the CISO. Results indicate that CISO can find optimal solution with less number of analyses for these problems as opposed to other considered algorithms.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationship that could have appeared to influence the work reported in this paper.

Author Contributions

Conceptualization, Hanan A.R,and Sameem Abbas.; methodology, Hanan A.R.; software, Sameem Abbas.; validation, Hanan A.R,and Sameem Abbas.; formal analysis, Hanan A.R,and Sameem Abbas.; investigation, Hanan A.R,and Sameem Abbas.; resources, Hanan A.R.; data curation, Sameem Abbas.; writing—original draft preparation, Hanan A.R,and Sameem Abbas.; writing—review and editing, Hanan A.R.; visualization,Hanan A.R,and Sameem Abbas.; supervision, Hanan A.R.; project administration, ,Hanan A.R.; funding acquisition, Sameem Abbas.

References

[1] Y. Djenouri, A. Belhadi, and R. Belkebir, "Bees Swarm Optimization Guided by Data Mining Techniques for Document Information Retrieval", *Expert Systems with Applications*, Vol. 94, pp. 126-136, 2018.

[2] K. L. Du and M. Swamy, "Particle Swarm Optimization", *Search and Optimization by Metaheuristics*, ed: Springer, pp. 153-173, 2016.

[3] N. E. Karkalos, A. P. Markopoulos, and J. P. Davim, "Evolutionary-Based Methods", *Computational Methods for Application in Industry 4.0*, ed: Springer, pp. 11-31, 2019.

[4] G. Dhiman and V. Kumar, "Emperor Penguin Optimizer: A Bio-inspired Algorithm for Engineering Problems", *Knowledge-Based Systems*, Vol. 159, pp. 20-50, 2018.

[5] N. S. Jaddi, J. Alvankarian, and S. Abdullah, "Kidney-inspired Algorithm for Optimization Problems", *Communications in Nonlinear Science and Numerical Simulation*, Vol. 42, pp. 358–369, 2017.

[6] O. P. Verma, D. Aggarwal, and T. Patodi, "Opposition and Dimensional Based Modified Firefly Algorithm", *Expert Systems with Applications*, Vol. 44, pp. 168–176, 2016.

[7] M. Yazdani and F. Jolai, "Lion Optimization Algorithm (LOA): A nature-inspired Metaheuristic Algorithm", *Journal of Computational Design and Engineering*, Vol. 3, No. 1, pp.24–36, 2016.

[8] A. Baykasoğlu and Ş. Akpinar, "Weighted Superposition Attraction (WSA): A swarm Intelligence Algorithm for Optimization Problems – Part 1: Unconstrained optimization", *Applied Soft Computing*, Vol. 56, pp. 520–540, 2017.

[9] I. Strumberger, M. Sarac, D. Markovic, and N. Bacanin, "Hybridized Monarch Butterfly Algorithm for Global Optimization Problems", *International Journal of Computers*, Vol. 3, pp. 63–68, 2018.

[10] F. Wang, H. Zhang, K. Li, Z. Lin, J. Yang, and X.-L. Shen, "A hybrid Particle Swarm Optimization Algorithm Using Adaptive Learning Strategy", *Information Sciences*, Vol. 436, pp. 162– 177, 2018.

[11] S. Mirjalili, "Genetic Algorithm", *Evolutionary Algorithms and Neural Networks*, ed: Springer, pp. 43-55, 2019.

[12] F. Min, Z.-H. Zhang, and J. Dong, "Ant colony Optimization with Partial-complete Searching for Attribute Reduction", *Journal of*

Computational Science, Vol. 25, pp. 170–182, 2018.

[13] A. Yelghi and C. Ko¨se, “A modified Firefly Algorithm for Global Minimum Optimization”, *Applied Soft Computing*, Vol. 62, pp. 29-44, 2018.

[14] A. Kaveh and S. Talatahari, “A novel Heuristic Optimization Method: Charged System Search”, *Acta Mechanica*, Vol. 213, pp. 267-289, 2010.

[15] N. Siddique and H. Adeli, “Nature-Inspired Chemical Reaction Optimization Algorithms”, *Cognitive Computation*, Vol. 4, pp. 411–422, 2017.

[16] A. Kaveh, M. Ilchi Ghazaan, and T. Bakhshpoori, “An improved Ray Optimization Algorithm for Design of Truss Structures”, *Periodica Polytechnica Civil Engineering*, Vol. 57, pp. 97-112,2013.

[17] J. Cheng and W. Zhao, “Chaotic Enhanced Colliding Bodies Optimization Algorithm for Structural Reliability Analysis”, *Advances in Structural Engineering*, pp. 1-16, 2019.

[18] A. Askarzadeh, “A novel Metaheuristic Method for Solving Constrained Engineering Optimization Problems: Crow Search Algorithm”, *Computers & Structures*, Vol. 169, pp. 1-12, 2016.

[19] N. S. Jaddi and S. Abdullah, “Kidney-inspired Algorithm with Reduced Functionality Treatment for Classification and Time Series Prediction”, *PLOS ONE*, Vol. 14, pp. 1-17. 2019.

[20] Y. Fu, W. Zhang, C. Qu, and B. Huang, “Optimal Foraging Algorithm Based on Differential Evolution”, *IEEE Access*, Vol. 8, pp. 19657–19678,2020.

[21] S. Saremi, S. Mirjalili, and A. Lewis, “Grasshopper Optimization Algorithm: Theory and Application”, *Advances in Engineering Software*, Vol. 105, pp. 30–47, 2017.

[22] A. R. Moazzeni and E. Khamehchi, “Rain Optimization Algorithm (ROA): A new Metaheuristic Method for Drilling Optimization Solutions”, *Journal of Petroleum Science and Engineering*, Vol. 105, pp. 1-29, 2020.

[23] D. Mazzi and S. Dorn, “Movement of insect pests in agricultural landscapes”, *Annals of Applied Biology*, Vol. 160, No. 2, pp. 97–113, 2012.

[24] M. Moriyama and H. Numata, “Ecophysiological Responses to Climate Change in Cicadas”, *Physiological Entomology*, pp. 1-12, 2019.

[25] C. Luo and C. Wei, “Stridulatory Sound-Production and Its Function in Females of the Cicada Subpsaltria yangi”, *PLOS ONE*, Vol. 10, pp. 1-13, 2015.

[26] J. Hasan, A. Roy, K. Chatterjee, and P. K. D. V. Yarlagadda, “Mimicking Insect Wings: The Roadmap to Bioinspiration”, *ACS Biomaterials Science & Engineering*, Vol. 5, No. 7, pp. 3139–3160, 2019.

[27] X. Yao, Y. Liu, and G. Lin, “Evolutionary Programming Made Faster”, *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 2, pp. 82-102, 1999.

[28] B. Y. Qu, J. J. Liang, Z. Y. Wang, Q. Chen, and P. N. Suganthan, “Novel Benchmark Functions for Continuous Multimodal Optimization with Comparative Results”, *Swarm and Evolutionary Computation*, Vol. pp. 26, 23–34, 2016.

[29] H. A. Akkar and F. R. Mahdi, “Grass Fibrous Root Optimization Algorithm”, *International Journal of Intelligent Systems and Applications(IJISA)*, Vol. 9, No. 6, pp. 15-23, 2017.

[30] F. Hemasian-Etefagh and F. Safi-Esfahani, “Group-Based Whale Optimization Algorithm”, *Soft Computing*, Vol. 24, pp. 3647–3673, 2020.

[31] I. C. Obagbuwa and A. P. Abidoye, “Binary Cockroach Swarm Optimization for Combinatorial Optimization Problem”, *Algorithms-MDPI*, Vol. 9, No. 3, pp. 59-74.2016.

[32] A. Sadollah, A. Bahreininejad, H. Eskandar, and M. Hamdi, “Mine Blast Algorithm: A new Population Based Algorithm for Solving Constrained Engineering Optimization Problems”, *Applied Soft Computing*, Vol. 13, No. 5, pp. 2592–2612, 2012.

[33] H. Yapici and N. Cetinkaya, “A new Meta - heuristic Optimizer: Pathfinder Algorithm (PFA)”, *Applied Soft Computing*, Vol. 78, pp. 545-568, 2019.

[34] V. K. Kamboj, A. Nandi, A. Bhadoria, and S. Sehgal, “An intensify Harris Hawks Optimizer for Numerical and Engineering Optimization Problems”, *Applied Soft Computing*, Vol. 89, pp. 1-35, 2019.

[35] S. Mirjalili and A. Lewis, “The Whale Optimization Algorithm”, *Advances in Engineering Software*, Vol. 95, pp. 51-67, 2016.

[36] H. Varaee and M. R. Ghasemi, “Engineering Optimization Based on Ideal Gas Molecular Movement Algorithm”, *Engineering with Computers*, Vol. 33 No. 1, pp. 71-93, 2017.

[37] Y. Zhou, Y. Ling, and Q. Luo, “Lévy Flight Trajectory-Based Whale Optimization Algorithm for Engineering Optimization”, *Engineering Computations*, Vol. 35 No. 7, pp. 2406-2428, 2018.

[38] M. Kamalinejad, H. Arzani, and A. Kaveh, “Quantum Evolutionary Algorithm with Rotational Gate and H-gate Updating in Real and Integer Domains for Optimization”, *Acta Mechanica*, Vol. 320, pp. 2937–2961, 2019.

[39] M. S. Massoudi, S. Sarjamei, and M. Esfandi Sarafraz, “Smell Bees Optimization Algorithm for Continuous Engineering Problem”, *Asian Journal of Civil Engineering*, Vol. 21, pp. 925–946, 2020.

[40] W. Xiao, Q. Liu, L. Zhang, K. Li, and L. Wu, “A novel Chaotic Bat Algorithm Based on Catfish Effect for Engineering Optimization Problems”, *Engineering Computations*, Vol. 36, pp. 1744-1763,2019.

[41] A. Kaveh and A. Dadras, “A novel Meta - heuristic Optimization Algorithm: Thermal Exchange Optimization”, *Advances in Engineering Software*, Vol. 110, pp. 69-84, 2017.

[42] S. A. Mirjalili, S. M. Mirjalili, and A. Hatamlou, “Multi-verse Optimizer: A nature-inspired Algorithm for Global Optimization”, *Neural Computing and Applications*, Vol. 27, pp. 495–513, 2016.

[43] A. Kaveh and A. Zaerreza, “Shuffled Shepherd Optimization Method: A new Meta - heuristic Algorithm”, *Engineering Computations*, Vol. 37, No. 7, pp.2357–2389, 2020.

[44] D. M. Himmelblau, “Applied Nonlinear Programming”, McGraw - Hill Companies, New York, 1972.

[45] J. Xue and B. Shen, “A novel Swarm Intelligence Optimization Approach: Sparrow Search Algorithm”, *Systems Science & Control Engineering*, Vol. 8, No. 1, pp. 22–34, 2020.

[46] S. Gupta and K. Deep, “Random Walk Grey Wolf Optimizer for Constrained Engineering Optimization Problems”, *Computational Intelligence*, Vol. 34, No. 4, pp. 1-21, 2018.

[47] H. Garg, “A hybrid GSA-GA Algorithm for Constrained Optimization Problems”, *Information Sciences*, Vol. 478, pp. 499-523, 2018.

[48] N. Kumar, S. K. Mahato, and A. K. Bhunia, “A new QPSO Based Hybrid Algorithm for Constrained Optimization Problems via Tournamenting Process”, *Soft Computing*, Vol. 24, pp. 11365–11379, 2020.

[49] J. Jiang, X. Meng, Y. Chen, C. Qiu,Y. Liu, and K. Li, “Enhancing Tree-Seed Algorithm via Feed-Back Mechanism for Optimizing Continuous Problems”, *Applied Soft Computing*, Vol.92, pp. 1-29, 2020.

[50] F. Yan, J. Xu, and K. Yun, “Dynamically Dimensioned Search Grey Wolf Optimizer Based on Positional Interaction Information”, *Complexity*, Vol. 2019, pp. 1-37, 2019.