Mechanics Transport Communications
Academic journal http://www.mtc-aj.com
ISSN 1312-3823 (print)
ISSN 2367-6620 (online)
volume 19, issue 2, 2021
article № 2176

OPTIMIZATION OF SPEED REDUCER, PRESSURE VESSEL AND HELICAL SPRING BY USING GRASSHOPPER ALGORITHM

Branislav Milenković[1] , Mladen Krstić[2]
bmilenkovic92@gmail.com, mladenkrstic994@gmail.com

[1]Mathematical Institute, Serbian Academy of Sciences and Arts, Belgrade, Serbia
[2]Leoni Wiring System, Kraljevo, Serbia

Keywords: *grasshopper, optimization, speed reducer, pressure vessel, helical spring*

Abstract: *The optimization plays very important roles especially in mechanical engineering, civil engineering, railway engineering, traffic engineering, computer engineering, chemical engineering and electrical engineering. In the past, many researchers have considered the problem of speed reducer, pressure vessel and helical spring optimization by using Genetic Algorithm (GA), Grey Wolf Optimizer (GWO), Harmony Search Algorithm (HSA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Water Cycle Algorithm (WCA), Ant Lion Optimizer (ALO), Differential Evolution (DE), Firefly Algorithm (FA), Bat Algorithm (BA), Whale Optimization Algorithm (WOA) and other optimization algorithms. The goal of this paper is to perform speed reducer, pressure vessel and helical spring parameter optimization by using the Grasshopper Optimization Algorithm (further referred to as GOA). The pseudo code for this algorithm was written using the Matlab R2019a software suite. At the end of this paper, results and conclusions are presented.*

I. INTRODUCTION

In the field of mathematical optimizations, there are many techniques, such as linear programming, nonlinear programming, dynamic programming. Each of this methods has certain shortcomings. Therefore, to achieve a significant result, a development of a new optimization method, such as those based on heuristics, is of utmost importance. All heuristic methods draw inspiration from either natural phenomena or artificial, abstract constructs.

The goal of each mechanical optimization is to achieve minimum cost with maximum gain. In order to achieve this goal, we depend upon the optimization method that we use. Many problems in the field of engineering have been solved by using different optimization techniques. Examples of these problems are: projecting airplane and cosmic framework in order to achieve minimum weight, frame construction, bridge design, minimum weight constructs in order to achieve maximum earthquake protection, wind protection, other kinds of tension, parameter optimization of machines in order to achieve minimum production cost, pump design, turbine design, heat conductivity problems, design of electrical circuits, etc.

All of the fore mentioned problems can be solved using various optimization methods. To solve optimization problems, heuristic algorithms, such as genetic algorithm, bee colony optimization, taboo search, simulated annealing. All heuristic algorithms draw inspiration from either natural phenomena or artificial, abstract constructs. Such are the examples of simulated annealing, which was discover by Kirkpatrick and al. [1] in 1983, based upon the annealing process, bee colony optimization and ant colony optimizations, which are based upon lives of bees and ants. Holland [2] has proposed a genetic algorithm, based on natural selection and population genetics. Zong Voo Geem i Joong Hoon Kim et al. [3] have presented a harmonic search algorithm, based upon music improvement process from jazz music.

In paper by Zhang et al. [4], a multi-objective problem was solved by the modified PSO algorithm, called Niche PSO. The problem involved mapping virtual networks to substrate networks, in terms of revenue and energy cost. The Niche PSO has shown better results for both objective functions, while having a slightly larger execution time. Manickavelu and Vaidyanathan [5] used the PSO algorithm to make predictions about route route rediscovery during route failures in mobile networks. The network consisted of nodes, whose status was decided upon by fuzzified parameters. This method was tested on a randomized network, while the packet size and node speed were varied. The PSO has shown better results in all the test cases.

In the paper by Long et al. [6], an improved version of GWO with modified augmented Lagrangian was used. The modified augmented Lagrangian is used to remove constraints by integrating them into the objective function. GWO is modified in such a manner so that the global optimum exploration factor is decreasing sub linearly. In this paper, a set of 24 optimization problems was selected as a testbench for GWO. Also, a comparison between the standard and improved GWO was drawn, with the conclusion being that the improved GWO yields better solutions for most problems. For the first 13 problems, a comparison between other p-based optimization algorithms and GWO was drawn. In most cases, an equal or better result was yielded by GWO. In this paper, GOA is used for solving several engineering design problems.

In the paper that introduces Harris Hawk Optimization algorithm, by Heidari et al. [7], a benchmark set of problems was used to show validity of algo-rithm use. This benchmark set covers three main groups of benchmark landscapes: unimodal (UM), multi-modal (MM), and composition (CM). The UM functions with unique global best can reveal the exploitative (intensification) capacities of different optimizers, while the MM functions can disclose the exploration (diversification) and LO (Local Optimum) avoidance potentials of algorithms. HHO algorithm was also compared to other p-based metaheuristic algorithms, such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Biogeography-based optimization (BBO), Flower Pollination Algorithm (FPA), Grey Wolf Optimizer (GWO), Bat Algorithm, Firefly Algorithm (FA), Cuckoo Search (CS), Moth-Flame Optimization (MFO), Teaching-Learning-Based Optimization (TLBO), Differential Evolution (DE). In almost all test cases, the HHO algorithm yielded better results than other algorithms.

The first problem to be solved is speed reducer optimization, having the goal of minimizing reducer weight in accordance with bending stress constraints of gear teeth, surface stresses, transverse deflections of shafts and stresses in shafts. This problem was first analyzed and solved by Coello using GA [8].

The second engineering problem that will be considered in this paper is optimization of a pressure vessel. This problem was first analysed and suggested by Sandgren [9]. The goal of this optimization is overall cost reduction including costs of material, montage and welding costs.

The last problem [10] consists of minimization of helical spring weight subject to constrains on minimum deflection, shear stress, surge frequency, limits on the outside diameter and design variables. The design variables are: coil diameter D, wire diameter d and number of active coils N.

II. GRASSHOPPER OPTIMIZATION ALGORITHM

Grasshopper optimization algorithm (GOA) is a P-type optimization algorithm, defined for the first time in paper [11]. Like most P-type optimization algorithms, such as Ant Colony Optimization, Bee Colony, Artificial Immune Systems, GOA also has its roots in describing patterns found in nature. This algorithm draws inspiration from swarm behavior of grasshoppers. Grasshopper's life cycle has three phases: larva, nymph, and adult. The grasshopper's movement is increasing as it goes through these phases.

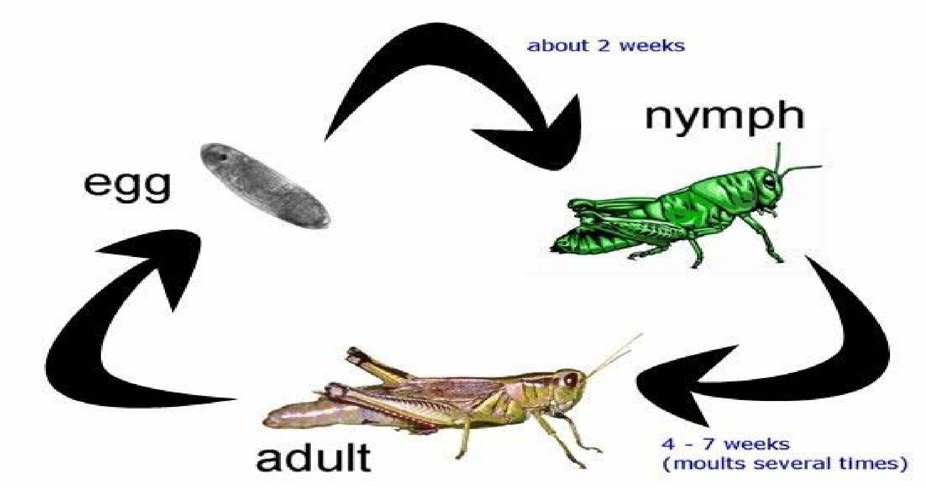

Fig. 1 Grasshopper's life cycle

An optimization problem is defined as follows: the swarm consists of *n* grasshoppers, randomly initialized at the beginning of the algorithm, a maximum number of iterations *L*, while the problem is represented in a *d*-dimensional search space. The algorithm has three phases:

1. Initialization of algorithm parameters
2. Population initialization and initial *fitness* value calculation
3. Loop of *L* iterations
    a. Using the movement function, recalculate each grasshopper's position in search space
    b. Search the best solution

The mathematical model used to simulate grasshopper movement is given by the following equations:

$$X_i = S_i + G_i + A_i$$

$$S_i = \sum_{\substack{j=1 \\ j \neq 1}}^{N} s\left(d_{ij}\right) \cdot \hat{d}_{ij}$$

$$s(r) = f \cdot e^{-\frac{r}{l}} - e^{-r}$$

$$G_i = -g \cdot \hat{e}_g$$

$$A_i = u \cdot \hat{e}_w$$

Where X_i is the grasshopper position, S_i are the social interaction forces, G_i the gravitational force, A_i wind advection and *s(r)* represents the social force between two grasshoppers.

The conceptual model of grasshopper interaction and comfort zone using the s function is shown in Figure 2.

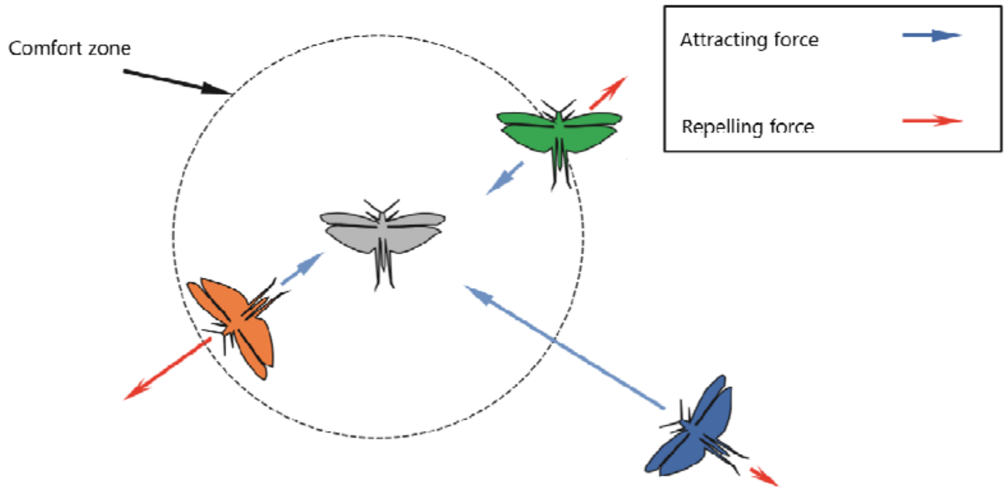

Fig. 2 Primitive correctional patterns between members in grasshopper swarm

The equation for grasshopper movement can be modified as such. Since both the gravitational force and the wind advection have essentially the same mathematical form, they can be represented by one term, $\overline{T_d}$. As the problem space has *d* dimensions, and each one of them has their upper and lower bounds *ubd* and *lbd* by substituting into the movement equation, we get the following form:

$$X_i^d = c \cdot \left(\sum_{\substack{j=1 \\ j \neq i}}^{N} c \cdot \frac{u \cdot b_d - l \cdot b_d}{2} \cdot s\left(\left| x_j^d - x_i^d \right| \right) \cdot \frac{x_j - x_i}{d_{ij}} \right) + \overline{T_d}$$

The complete pseudo-code for the algorithm is given below.

```
Swarm initialization Xi (i = 1, 2, …, n)
Initialize cmax, cmin, and maximum number of iterations
Calculate the fitness value for each grasshopper
T = current best search agent
while (current iteration < maximum number of iterations)
Update c
for each search agent
Normalize the distances between grasshoppers as to fit the interval
Update the current search agent position
Reposition the search agent if it goes out of bounds
end for
Update T if there is a better solution
Increment the iteration counter
end while
return T
```

III. PROBLEM FORMULATION

In this section, for each of the optimization problems we describe the basis of the problem, goal function, algorithm parameters, as well as conditions that are to be met. Every step of this process was done using the MATLAB R2019a software suite.

3.1. Optimization of the speed reducer

The goal of speed reducer optimization is minimizing the reducer weight whilst fulfilling all the defined constraints**.**

In Figure 3 a schematic view of speed reducer is shown.

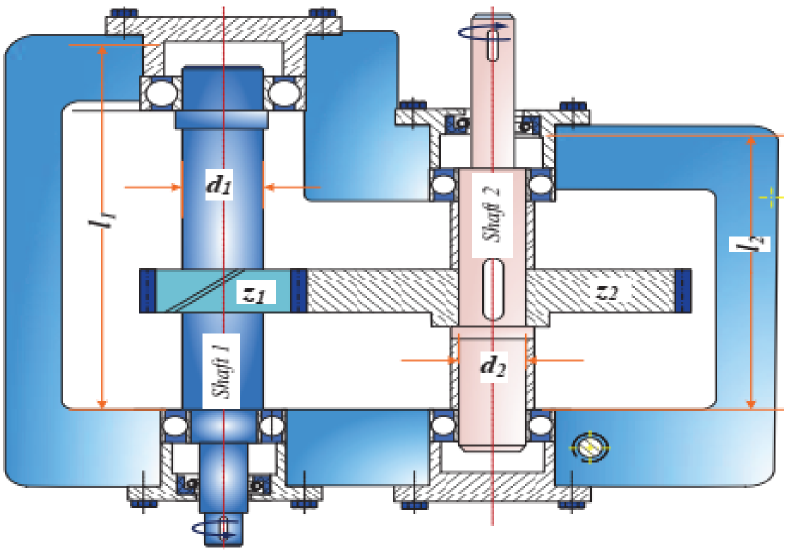

Fig. 3 Schematic view of the speed reducer with variable parameters

Project variables for the speed reducer problem are: the width between the shafts (x_1), the module of the teeth (x_2), the number of teeth in the pinion (x_3), the length of the first shaft between the bearings (x_4), the length of the second shaft between the bearings (x_5), the diameter of the first shaft (x_6) and the diameter of the second shaft (x_7).

The problem can be mathematically formulated as follows:

Minimize:

$$f(x) = 0,7854 x_1 x_2^2 \left(3,3333 x_3^2 + 14,933 x_3 - 43,0934\right) - 1,508 x_1 \left(x_6^2 + x_7^2\right) + $$
$$+ 7,4777 \left(x_6^3 + x_7^3\right) + 0,7854 \left(x_4 x_6^2 + x_5 x_7^2\right)$$

Subject to:

$$g_1(x) = \frac{27}{x_1 x_2^2 x_3} - 1 \le 0$$

$$g_2(x) = \frac{397.5}{x_1 x_2^2 x_3^2} - 1 \le 0$$

$$g_3(x) = \frac{1.93 x_5^3}{x_2 x_3 x_7^4} - 1 \le 0$$

$$g_4(x) = \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \le 0$$

$$g_5(x) = \frac{\left(\left(\frac{475x_4}{x_2x_3}\right)^2 + 16.9\times10^6\right)^{1/2}}{110x_6^3} - 1 \le 0$$

$$g_6(x) = \frac{\left(\left(\frac{475x_4}{x_2x_3}\right)^2 + 157.5\times10^6\right)^{1/2}}{85x_7^3} - 1 \le 0$$

$$g_7(x) = \frac{x_2x_3}{40} - 1 \le 0$$

$$g_8(x) = \frac{5x_2}{x_1} - 1 \le 0$$

$$g_9(x) = \frac{x_1}{12x_2} - 1 \le 0$$

$$g_{10}(x) = \frac{15x_6 + 1.9}{x_4} - 1 \le 0$$

$$g_{11}(x) = \frac{1.1x_7 + 1.9}{x_5} - 1 \le 0$$

$2{,}6 \le x_1 \le 3{,}6$; $0{,}7 \le x_2 \le 0{,}8$; $17 \le x_3 \le 28$
$7{,}3 \le x_4 \le 8{,}3$; $7{,}3 \le x_5 \le 8{,}3$; $2{,}9 \le x_6 \le 3{,}9$
$5{,}0 \le x_7 \le 5{,}5$

3.2. Optimization of the pressure vessel

Second problem is optimization of a pressure vessel (figure 4), which consists of reducing costs of material, montage and welding costs. Four variables are defined for this problem: radius of the shell (x_1), length of the shell (x_2), thickness of the shell (x_3) and thickness of the dish end (x_4).

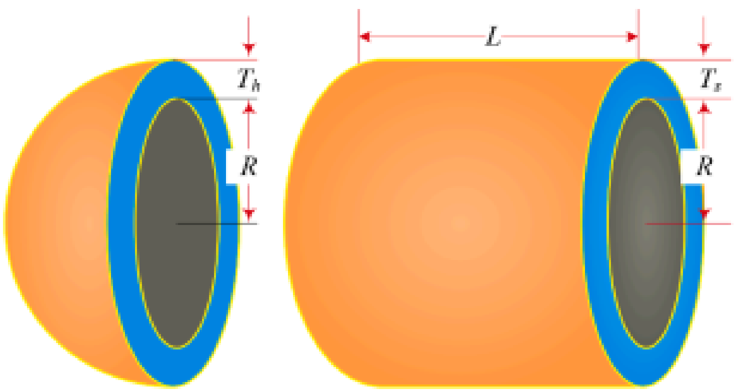

Fig. 4 Schematic view of the pressure vessel with variable parameters

The problem can be mathematically formulated as follows:

Minimize:

$$f(x) = 0,6224x_1x_3x_4 + 1,7781x_2x_3^2 + 3,1661x_1^2x_4 + 19,84x_1^2x_3$$

Subject to:

$$g_1(x) = -x_1 + 0,0193x_3 \le 0;$$

$$g_2(x) = -x_2 + 0,00954x_3 \le 0;$$

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

$$g_4(x) = x_4 - 240 \le 0;$$

$$0 \le x_i \le 100; \quad i = 1,2;$$

$$10 \le x_i \le 200; \quad i = 3,4;$$

3.3. Optimization of the helical spring

This problem consists of three continual vaiables, two linear constraints, and five nonlinear constraints, given in inequality form. The goal of this optimization is minimizing the weight of the spring.

In Figure 5, a schematic view of helical spring, along with all the project variables, is shown.

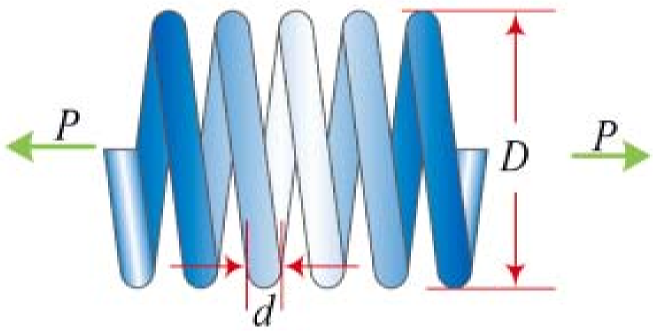

Fig. 5 Schematic view of the helical spring with variable parameters

The problem can be mathematically formulated as follows:

Minimize:

$$f(x) = (x_3 + 2)x_2x_1^2$$

Subject to:

$$g_1(x) = 1 - \frac{x_2^3 x_3}{71785x_1^4} \le 0;$$

$$g_2(x) = \frac{4x_2^2 - x_1 x_2}{12566\left(x_2 x_1^3 - x_1^4\right)} + \frac{1}{5108 x_1^2} - 1 \le 0;$$

$$g_3(x) = 1 - \frac{140,45 x_1}{x_2^2 x_3} \le 0;$$

$$g_4(x) = \frac{x_1 + x_2}{1,5} \le 0;$$

$$0,05 \le x_1 \le 2;$$

$$0,25 \le x_2 \le 1,3;$$

$$2 \le x_3 \le 15;$$

IV. EXPERIMENTAL RESULT AND DISCUSSION

In this section, the results obtained by using GOA algorithm on previously defined engineering problems is given.

Results of the Grasshopper Optimization Algorithm (GOA) will be compared to results obtained by the Grey Wolf Optimizer (GWO), Improved Cuckoo Search (ICS) and Water Cycle Algorithm (WCA) depending of solutions found in literature.

A detailed display of the results obtained by GOA and a comparison with several results obtained by other methods, for the problem of speed reducer, are shown in Table 1.

Table 1. Comparison of results for the speed reducer

Variables	GWO[12]	ICS[13]	WCA[14]	**GOA**
x_1	3.499	3.499	3.5	**3.5**
x_2	0.699	0.7	0.7	**0.7**
x_3	17	17	17	**17**
x_4	8.051	7.3	7.3	**7.3**
x_5	8.084	7.8	7.715	**7.8**
x_6	3.351	3.350	3.350	**3.35022**
x_7	5.286	5.287	5.286	**5.28762**
f(x)	3009.669	2997.058	2994.471	**2996.9641**

In the case of the speed reducer problem, the GOA algorithm gave better results than ICS and GWO,while WCA results are better.

For the pressure vessel problem, the expected value for the goal function is 5885.3327, with the results shown in Table 2.

Table 2. Comparison of results for the pressure vessel

Variables	GWO[12]	ICS[13]	WCA[14]	**GOA**
x_1	0.822	0.7389	0.7781	**0.8736**
x_2	0.406	0.3655	0.3846	**0.4318**
x_3	42.602	38.2795	40.3196	**45.2666**
x_4	170.484	230.5087	200	**199.9998**
f(x)	5964.50	5823.3792	5888.3327	**5824.1734**

In the case of the pressure vessel problem, the GOA algorithm gave better results than WCA and GWO,while ICS results are better.

Experimental research was performed for the helical spring problem, and the results of HHO algorithm, along with the results for GWO, ICS, and WCA algorithms, are given in Table 3. The expected value for the goal function in this case is 0.012665.

Table 3. Comparison of results for the spring

Variables	GWO[12]	ICS[13]	WCA[14]	**GOA**
x_1	0.0523	0.0517	0.0516	**0.0524**
x_2	0.3722	0.3570	0.3562	**0.3750**
x_3	10.4141	11.2699	11.3004	**10.3038**
f(x)	0.0128	0.0126	0.0126	**0.0126**

In the case of helical spring optimization, GOA gives the same result as WCA and ICS, while GWO give a slightly worse result.

V. CONCLUSION

This paper describes using Grasshopper optimization algorithm in order to solve a few engineering problems with a constant number of variables. For this algorithm, 50 search agents and 1000 iterations were chosen as input parameters. The mathematical formulation, graphical representation, and the results were shown. The obtained results were compared to latest papers published in SCI list journals.

During the course of the research, it has been noted that increasing search agent and iteration count did not yield better solutions. Therefore, this combination of input parameters was chosen, since it gives minimal execution time.

This paper focuses on application of optimization algorithms on engineering problems. In this field of study, further research is always necessary, due to its applicability and possibility for the improvement of the optimization results.

REFERENCES:

[1] Kirkpatrick S., Gelatt C., and Vecchi M., Optimization of simulated annealing, Science, vol 220, 4598, pp. 671-680, 1983.

[2] Holland J H, Adaptation in Natural and Artificial System, University of Michigan Press, Ann Arbor, 1975.

[3] Zong Woo Geem and Joong Hoon Kim, A new Heuristic Optimization Algorith: Harmony Search, Simulation 76, Vol 2. Pp.60-68, 2001.

[4] Kraiss K. Advanced man-machine interaction (Fundamentals and implementation). Springer Berlin Heidelberg New York, ISBN-10 3-540-30618-8.

[5] P.Zhang, H.Yao,C.Fang and Y.Lui, “Multi-objective enhanced particle swarm optimization in virtual network embedding”, EURASIP Journal on Wireless Communications and Networking (2016) 2016:167.

[6] D.Manickavelu, R.Vaidyanathan, “Particle swarm optimization (PSO)-based node and link lifetime prediction algorithm for route recovery in MANET”, EURASIP Journal on Wireless Communications and Networking2014,2014:107.

[7] Ali Asghar Heidari, Seyedali Mirjalili, Hossam Faris, Ibrahim Aljarah, Majdi Mafarja, Huiling Chen: Harris hawks optimization: Algorithm and applications, Future Generation Computer Systems, March 2019.

[8] Carlos A. Coello Coello: Constraint-handling inge-netic algorithms through the use of dominance ba-sed tournament selection, Advanced Engineering In-formatics, 16, 193–203, 2002.
[9] Sandgren E, Nonlinear integer and discrete progra-mming in mechanical design optimization, Journal of Mechanical Design;112(43):223-9, 1990.
[10] Arora J. S. Introduction to optimum design. New York: McGraw-Hill; 1989.
[11] Abhishek G Neve, Ganesh M Kakandikar and Omkar Kulkarni, Application of Grasshopper Optimization for Constrained and Unconstrained Test Functions, India, 2017.
[12] B.Milenković,M.Krstić,Đ.Jovanović:Primena algoritma sivog vuka za rešavanje inženjerskih optimizacionih problema, Tehnika, 2021.,Vol.76, Br.1. str.50-57, ISSN 0040-2176.
[13] Bulatović, R. R., Bošković, G., Savković, M. M., & Gašić, M. M. (2014). Improved Cuckoo Search (ICS) algorthm for constrained optimization problems. /Latin American Journal of Solids and Structures/, /11/(8), 1349-1362.
[14] H.Eskandar, A. Sadollah, A. Bahreininejad, M.Hamdi: Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems, Computers and Structures, 2012.
[15] https://www.mathworks.com/matlabcentral/fileexchange/61421-grasshopper-optimisation-algorithm-goa