Application of Grasshopper Algorithm for Solving Optimization Problems in Engineering

Branislav Milenković[1*], Radovan Bulatović[2], Ivana Atanasovska[1]

[1]Mathematical Institute of Serbian Academy of Sciences and Arts, Belgrade (Serbia)
[2]Faculty of Mechanical and Civil Engineering Kraljevo, University of Kragujevac, Kraljevo (Serbia)

Today, there is increasing use of optimization methods in order to achieve acceptable performance. In this paper we will demonstrate how Grasshopper Optimization Algorithm (GOA for short) can be used for solving certain optimization problems in engineering. In the first part, biological fundamentals, as well as method explanation are given. Afterwards, the GOA algorithm and its' applicability is explained in detail. The pseudo code for this algorithm was written using Matlab R2018a software suite. This algorithm can be used for optimization of engineering problems, such as: helical spring optimization, car side impact optimization, cone clutch optimization and speed reducer optimization. In the end, all the results for the fore mentioned problems, as well as a result comparison with other methods are shown.

Keywords: Optimization, Grasshopper algorithm, Helical spring, Car side impact, Cone clutch, Speed reducer

1. INTRODUCTION

In the past two decades, a set of methods that proved to be highly efficient in solving difficult optimization problems appeared. These methods are named metaheuristic algorithms and are oft-times inspired by natural phenomena, since they mimic behaviors and patterns found in nature. For this reason, this class of algorithms is also called naturally-inspired or biological metaheuristic algorithms.

The surge in popularity of these algorithms gathers great attention from engineers and industry professionals. One of the reasons of this popularity is their adaptability and efficiency. Also, albeit simple in nature, these algorithms solve complex optimization problems with ease. Metaheuristic algorithms comprise an important part of contemporary optimization algorithms, artificial intelligence and computer science.

Many metaheuristic algorithms have the trait that they attain global optimum convergence in a fairly small number of iterations. Algorithms that belong to this class are: Differential Evolution algorithm, Genetic algorithm, Bat algorithm, Grey Wolf Optimization algorithm, Firefly algorithm, Particle Swarm Optimization algorithm, and many others.

Many of authors for the application in engineering design optimization have used different metaheuristic algorithms which are nature-inspired.

Differential Evolution algorithm (DE) was applied by Gašić, Abderazek for solving structural optimization problems[1-2]. In the paper by Gašić et al. [1], the trapezoidal cross section of the truck crane boom was optimized by using Lagrange's multipliers and the Evolution Algorithm (DE) methods. The results of these two methods were compared to the numerical example for an existing solution. The results have shown that Evolution algorithm gives better solutions for the existing problem. In the paper by Hammoudi and Djeddou [2], a modified version of DE, called Composite Differential Evolution (CoDE) was presented and used to optimize the dimensions of a helical spring. The results have proven to be better than those obtained by the Firefly Algorithm, Particle Swarm Optimization and Genetic Algorithm.

Grasshopper Optimization Algorithm (GOA) has been applied by: Saremi [3], Jovanović and Milenković [4] for solving constrained engineering optimization problems. In [5], authors used a improved grasshopper algorithm to solve a pressure vessel design problem.

In paper by Zhang et al. [6], 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 [7] 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. [8], 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

*Corresponding author: Branislav Milenković, Kneza Mihaila 36, Belgrade, bmilenkovic@mi.sanu.ac.rs

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.

The first problem [9] consists of minimization of 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.

The second problem is automobile side impact optimization problem, with the aim of minimizing total vehicle weight, using eleven design variables. This problem was first subjected in a paper by Gu [10].

The third engineering problem that will be considered in this paper is optimization of a cone clutch. The goal of this optimization is to minimize clutch volume. This example was defined in [11].

The last 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[12].

2. GRASSHOPPER OPTIMIZATION ALGORITHM (GOA)

Grasshoppers (Figure 1) can be the most noticeable and damaging insects to yards and fields. They also are among those most difficult to control, since they are highly mobile. For many reasons, grasshopper populations fluctuate greatly from year to year, and may cause serious damage during periodic outbreaks.

Figure 1:Real grasshopper

Grasshopper Optimization Algorithm (GOA) is a p-type optimization algorithm inspired by movement of grasshopper swarms [21].The grasshopper life cycle has three phases: larva, nymph and adult, with the grasshopper movement becoming quicker with each phase. The life cycle of grasshopper is shown in Figure 2.

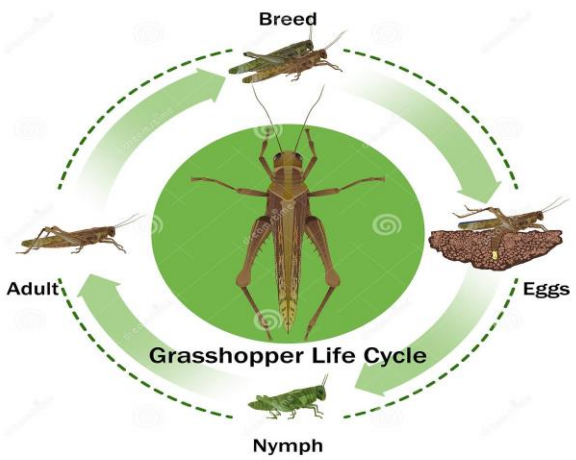

Figure 2:Life cycle of grasshopper

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

$$X_i = S_i + G_i + A_i \quad (1)$$

$$S_i = \sum_{\substack{j=1 \\ j\neq 1}}^{N} s\left(d_{ij}\right)\cdot \hat{d}_{ij} \quad (2)$$

$$s\left(r\right) = f\cdot e^{-\frac{r}{l}} - e^{-r} \quad (3)$$

$$G_i = -g\cdot \hat{e}_g \quad (4)$$

$$A_i = u\cdot \hat{e}_w \quad (5)$$

Where X_i is the grasshopper position, S_i are the social interaction forces, G_i the gravitational force, and A_i wind advection. s(r) represents the social force between two grasshoppers. 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, ub_d and lb_d, by substituting into the movement equation, we get the following form:

$$X_i^d = c\cdot\left(\sum_{\substack{j=1 \\ j\neq i}}^{N} c\cdot\frac{u\cdot b_d - l\cdot b_d}{2}\cdot s\left(\left|x_j^d - x_i^d\right|\right)\cdot\frac{x_j - x_i}{d_{ij}}\right) + \overline{T_d} \quad (6)$$

The algorithm consists of three phases:

1. Constant initialization
2. Initial population creation and calculating their fitness values
3. Main loop, having *L* iterations

a) Recalculating the parameter values for all grasshoppers, using the movement equation
b) Search for the best solution

The complete pseudocode 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
```

3. EXPERIMENTAL ENGINEERING EXAMPLES FOR OPTIMIZATION

This chapter will present certain examples of engineering problems, such as: optimization of helical spring, car side impact, cone clutch and speed reducer. The basis of the problem, the objective function, variable parameters that should be found as well as the constraints that should be respected will be shown. Then the results obtained by the GOA method (Chapter 2) will be presented and they will be compared to the optimum results for these four examples obtained and published so far. Analysis and obtaining of results by GOA were performed in the code written in Matlab R2018a.

Results of the Grasshopper optimization algorithm (GOA) will be compared to results obtained by the modified ant colony algorithm (MACA), genetic algorithm (GA), cuckoo search (CS), improved cuckoo search (ICS), water cycle algorithm (WCA), whale optimization algorithm (WOA), grey wolf optimization (GWO), moth flame optimization (MFO), firefly algorithm (FA) depending of solutions found in literature.

In Figure 3, a schematic view of helical spring, along with all the project variables, is shown.

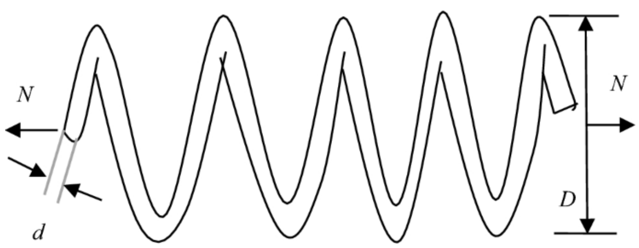

Figure 3: Schematic view of helical spring with variable parameters

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.

There are three variables that should be optimized:

- the wire diameter (d)
- the coil diameter (D)
- the number of active coils (N)

Goal function to be minimized is defined as:

$$f(x) = (x_3 + 2)\, x_2 x_1^2$$

subject to the following constraints:

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

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

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

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

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

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

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

The car (Figure 4) is exposed to a side impact on the foundation of the European Enhanced Vehicle-Safety Committee (EEVC) procedures. The aim is to minimise the total weight of the car using eleven mixed variables.

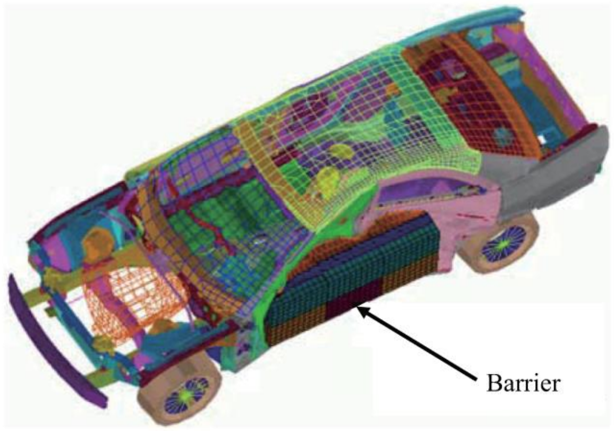

Figure 4: Finite element model utilized in the car side impact problem

There are eleven variables that should be optimized:

- the thickness of the B-Pillar inner (x_1)

- the thickness of the B-Pillar reinforcement (x_2)
- the thickness of the floor side inner (x_3)
- the thickness of the cross members (x_4)
- the thickness of the door beam (x_5)
- the thickness of the door belt line reinforcement (x_6)
- the thickness of the roof rail (x_7)
- the thickness of the materials of B-pillar inner (x_8)
- the thickness of the materials of floor side inner (x_9)
- barrier height (x_{10}) and
- hitting position (x_{11})

The problem is reduced to minimization of the function:

$$f(x)=1.98+4.90x_1+6.67x_2+6.98x_3+4.01x_4+1.78x_5+2.73x_7$$

subject to:

$$g_1(x)=F_a \le 1(kN)$$
$$g_2(x):=VC_u \le 0.32(m/s)$$
$$g_3(x):=VC_m \le 0.32(m/s)$$
$$g_4(x):=VC_1 \le 0.32(m/s)$$
$$g_5(x):=\Delta_{ur} \le 32(mm)$$
$$g_6(x):=\Delta_{mr} \le 32(mm)$$
$$g_7(x):=\Delta_{lr} \le 32(mm)$$
$$g_8(x):=F_p \le 4(kN)$$
$$g_9(x):=V_{MBP} \le 9.9(mm/ms)$$
$$g_{10}(x):=V_{FD} \le 15.7(mm/ms)$$
$$0.5 \le x_1,x_2,x_3,x_4,x_5,x_6,x_7 \le 1.5$$
$$-30 \le x_{10},x_{11} \le 30$$
$$x_8,x_9 \in \{0.192, 0.345\}$$

Variables $F_a, VC_u, VC_m, VC_1, \Delta_{ur}, \Delta_{mr}, \Delta_{lr}, F_p, V_{MBP}, V_{FD}$ are mathematical described in the paper [13].

The cone clutch problem (Figure 5) must be designed for minimum volume coupling to two constraints.

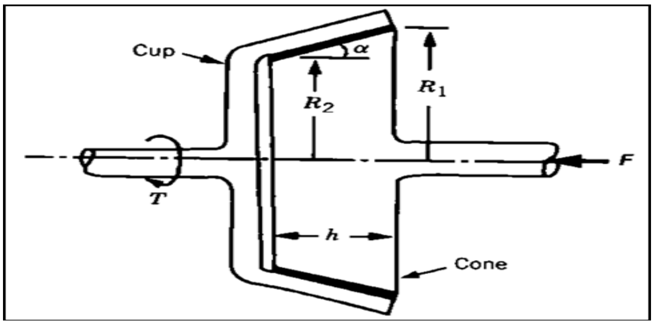

Figure 5: Schematic view of cone clutch with variable parameters

Problem variables are:

- inner radius of the clutch $R_1 \equiv x_1$ and
- outer radius of the clutch $R_2 \equiv x_2$.

Goal function to be minimized is defined as:

$$f(X)=\left(x_1^3-x_2^3\right)$$

Whilst the conditions to be met are:

$$g_1(X)=\frac{x_1}{x_2}\ge 2$$
$$g_2(X)=\frac{\left(x_1^2+x_1x_2+x_2^2\right)}{\left(x_1+x_2\right)}\ge 5$$
$$X=\left(x_1,x_2\right)$$
$$1\le x_1,\ x_2 \le 10$$

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

In Figure 6 a schematic view of speed reducer is shown.

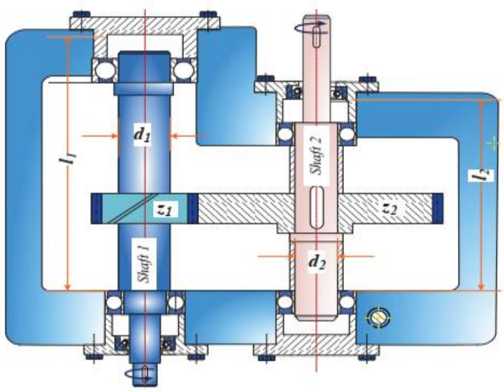

Figure 6: Schematic view of speed reducer with variable parameters

There are seven variables:

- the width between the shafts (b)
- the module of the teeth (m)
- the number of teeth in the pinion (z)
- the length of the first shaft between the bearings (l_1)
- the length of the second shaft between the bearings (l_2)
- the diameter of the first shaft (d_1)

- the diameter of the second shaf (d_2)

The problem can be expressed as minimization of the function:

$$f(x)=0,7854x_1x_2^2\left(3,3333x_3^2+14,933x_3-43,0934\right)-1,508x_1\left(x_6^2+x_7^2\right)+$$
$$+7,4777\left(x_6^3+x_7^3\right)+0,7854\left(x_4x_6^2+x_5x_7^2\right)$$

subject to the following constraints:

$$g_1(x)=\frac{27}{x_1x_2^2x_3}-1\le 0$$

$$g_2(x)=\frac{397.5}{x_1x_2^2x_3^2}-1\le 0$$

$$g_3(x)=\frac{1.93x_5^3}{x_2x_3x_7^4}-1\le 0$$

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

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

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

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

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

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

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

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

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

Experimental research was performed for the helical spring problem, and the results of GOA algorithm, along with the results for MACA, FSO, and WCA algorithms, are given in Table 1.

Table 1. *Comparison of results for the helical spring*

Variables	MACA[14]	ICS[15]	WCA[16]	**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 MACA give a slightly worse result.

A detailed display of the results obtained by GOA and a comparison with several results obtained by other methods, for the problem of car side impact, are shown in Table 2.

Table 2. *Comparison of results for the car side impact*

Variables	MFO[17]	GWO[18]	WOA[19]	**GOA**
x_1	0.5	0.5	0.5	**0.5**
x_2	1.116	1.14	1.108	**1.115**
x_3	0.5	0.5	0.534	**0.5**
x_4	1.301	1.268	1.305	**1.303**
x_5	0.5	0.5	0.5	**0.5**
x_6	1.5	1.5	1.473	**1.5**
x_7	0.5	0.5	0.5	**0.5**
x_8	0.345	0.345	0.345	**0.345**
x_9	0.345	0.192	0.192	**0.286**
x_{10}	-19.530	-20.605	-19.699	**-19.715**
x_{11}	0.0000	0.5	3.481	**0.320**
f(x)	22.842	22.878	23.042	**22.878**

In the case of car side impact optimization, GOA gives the same result as GWO, WOA give a slightly worse result, while MFO gave better results.

For the cone clutch design problem, the results shown in Table 3, along with the results obtained by FA, CS and GWO methods.

Table 3. *Comparison of results for the cone clutch*

Variables	FA[20]	CS[20]	GWO[18]	**GOA**
x_1	4.2987	4.2858	4.286	**4.2861**
x_2	2.1405	2.1428	2.142	**2.143**
f(x)	69.6278	68.887	68.893	**68.8948**

In this case, the GOA gives the near same result as the CS and GWO algorithm, while FA gave worse results.

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 4.

Table 4. *Comparison of results for the speed reducer*

Variables	ICS[15]	GWO[18]	WCA[16]	**GOA**
x_1	3.499	3.502	3.5	**3.5**
x_2	0.700	0.7	0.7	**0.7**
x_3	17	17	17	**17**
x_4	7.300	7.333	7.3	**7.3**
x_5	7.800	7.8	7.715	**7.8**
x_6	3.350	3.351	3.350	**3.35022**
x_7	5.287	5.288	5.286	**5.28762**
$f(x)$	2997.058	2998.299	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.

4. CONCLUSION

This paper deals with the GOA optimization algorithm and applies it to a few engineering design examples: speed reducer, helical spring, side impact of a car, and cone clutch.

In the introduction, a brief overview of literature was given. In Section 2, the inspiration, the mathematical model, and the pseudo code for the GOA algorithm was presented. In Section 3, the engineering design examples, which are common benchmarks for optimization methods, were solved by using the GOA algorithm. For this algorithm, 30 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.

This algorithm was used to obtain optimal or near-optimal results, as shown in the examples. Further developing of this algorithm can be used to redefine and upgrade this method, as to gain better results.

ACKNOWLEDGEMENT

This work was supported by the Serbian Ministry of Education, Science and Technological Development through Mathematical Institute of the Serbian Academy of Sciences and Arts.

REFERENCES

[1] M.Gašić,M.Savković,R.Bulatović, “Optimization of Trapezoidal Cross Section of the Truck Crane Boom by Lagrange Multipliers and by Differential Evolution Algorithm (DE)”, Journal of Mechanical Engineering 57(2011)4, 304-312.

[2] A. Hammoudi, F. Djeddou, “Composite Differential Evolution Algorithm: Mixed Variable Structural Optimization Problem”, Journal of Chemistry and Materials Research Vol. 3, 2015, 3–8.

[3] Shahrzad S, Seyedali Mirjalili and Andrew Lewis, “Grasshopper Optimisation Algorithm: Theory and Application” Advances in Engineering Software, Volume 105, March 2017, pp 30-47.

[4] Dj.Jovanović,B.Milenković,M.Krstić,“Application of Grasshopper Algorithm in Mechanical Engineering”, YOURS 2020, pp.1-6.

[5] S.Guo,J.Wang,W.Xie,M.Guo,L.Zhu, “Improved Grasshopper Algorithm Based on Gravity Search Operator and Pigeon Colony Landmark Operator”,IEEE Access Vol 8, February 2020,pp 1-22.

[6] 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.

[7] 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.

[8] X. L. S. C. J. J. W. Z. Wen Long, "A modified augmented Lagrangian with improved grey wolf optimization to constrained optimization problems," Neural Computing & Applications.

[9] Arora JS. Introduction to optimum design. New York: *McGraw-Hill*; 1989.

[10] Gu, L., Yang, R., Tho, C., Makowskit, M., Faruquet, O. and Y.Li, Y. (2001) ‘Optimisation and robustness for crashworthiness of side impact’, International Journal of Vehicle Design, Vol. 26, No. 4, pp.348–360.

[11] R.V.J..Savsani, Mechanical Design Optimization Using Advanced Optimization Techniques, *Springer-Verlag* London, 2012.

[12] Carlos A. Coello Coello: Constraint-handling in genetic algorithms through the use of dominance based tournament selection, Advanced Engineering Informatics, 16 (2002):193–203.

[13] T.Kim,M.Cho,S.Shin, “Constrained Mixed Variable Design Optimization Based on Particle Swarm Optimizer with a Diversity Classifier for Cyclically Neighboring Subpopulations” *Mathematics* 2020, *8*, 2016,pp.1-29.

[14] V.Grković,R.Bulatović, Modified Ant Colony Algorithm for Solving Engineering Optimization Problems, IMK-14-Research and Development 18(2012)4.

[15] 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.

[16] 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.

[17] Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl Based Syst 89:228–249.

[18] 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.

[19] S.Mirjalili, A.Lewis, ,"The Whale Optimization Algorithm" Advances in Engineering Software, Volume 95, March 2016, pp 51-67.

[20] G.Miodragović,Advanced bio-inspired algorithms development for solving optimization problems in applied mechanics, doctoral thesis, Faculty of Mechanical and Civil Engineering Kraljevo, University of Kragujevac (2015).

[21] Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper optimisation algorithm: theory and application. /Advances in Engineering Software/, /105/, 30-47.