# Implementation of Harris Hawks Optimization (HHO) Algorithm to Solve Engineering Problems

*BRANISLAV N. MILENKOVIĆ*, Serbian Academy of Sciences and Arts,
Mathematical Institute, Belgrade

*Original scientific paper*
*UDC: 519.863:004.021*
*DOI: 10.5937/tehnika2104439M*

*Recently, optimization techniques have become very important and popular in different engineering applications. In this paper we demonstrate how Harris Hawks Optimization (HHO) algorithm can be used to solve certain optimization problems in engineering. In the second part, biological fundamentals, as well as method explanation are given. Afterwards, the HHO algorithm and its' applicability is explained in detail. The pseudo code for this algorithm was written using MATLAB R2019a software suite. Harris Hawks Optimization (HHO) algorithm was used for optimization of engineering problems, such as: speed reducer optimization, pressure vessel optimization, cantilever beam optimization and tension/compression spring optimization. The statistical results and comparisons show that the HHO algorithm provides very promising and competitive results compared to others metaheuristic algorithms.*

**Key words:** *Harris hawks optimization algorithm, , optimization, engineering problems, metaheuristic*

## 1. INTRODUCTION

In the past two decades, a set of very efficient methods emerged, with application in solving hard practical optimization problems. These methods are called metaheuristic algorithms, and are often inspired by natural phenomena, since they mimic nature's tried and tested methods. Another name for this class of algorithms is nature-inspired or biological metaheuristic algorithms.

A surge in popularity of these algorithms draws attention of both engineering professionals and industry. One of the reasons for this is that algorithms of this class are adaptable and efficient. Albeit simple in design, they are very successful in solving very complex optimization problems. Metaheuristic algorithms constitute an important part of contemporary global optimization algorithms, artificial intelligence, and informatics in general.

Many metaheuristic algorithms carry a trait that they reach convergence towards a global optimum for the problem after a fairly small number of iterations. Since these problems are complex in nature, determining the global optimum analytically is impossible.

In this case, an algorithm's value for solving a parti-cular problem lies in comparing it to existing solu-tions found in literature.

The most famous biologically inspired metaheuristic algorithms are: Differential evolution (DE), Genetic algorithm (GA), Bat algorithm (BA), Grey Wolf Optimizer (GWO), Ant Colony Optimization (ACO), Grasshopper Optimization Algorithm (GOA), Simulated Annealing (SA), Lion Optimi-zation Algorithm (LOA), etc.

In this paper the Harris hawks' algorithm, one of the biological metaheuristic algorithms created by Heidari[1], will be demonstrated. After the algorithm was published in 2019, it drew researchers' attention, and it was implemented and improved in many areas of expertise

In the paper that introduces Harris Hawk Optimization algorithm, by Heidari et al. [1], a benchmark set of problems was used to show validity of algorithm use. This benchmark set covers three main groups of benchmark landscapes: unimodal (UM), multimodal (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

Author's address: Branislav Milenković, Serbian Academy of Sciences and Arts, Mathematical Institute, Belgrade, Kneza Mihaila 35

e-mail: bmilenkovic@gmail.com

Paper received: 10.02.2021.

Paper accepted: 01.07.2021.

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.

In paper by Houssein et al. [2], a hybrid meta-heuristic algorithm called CHHO-CS is introduced, and it combines Harris Hawks Optimizer (HHO) with two operators: cuckoo search (CS) and chaotic maps. This algorithm is used as a part of the pipeline which optimizes drug design and discovery in chemo-informatics. The role of CS is to control the main position vectors of the HHO algorithm to maintain the balance between exploitation and exploration phases, while the chaotic maps are used to update the control energy parameters to avoid falling into local optimum and premature convergence. This algorithm is then used to optimize parameters for Support Vector Machines. The extensive experimental and statistical analyses exhibit that the suggested CHHO–CS method accomplished much-preferred trade-off solutions over the competitor algorithms including the HHO, CS, particle swarm optimization, moth-flame optimization, grey wolf optimizer, Salp swarm algorithm, and sine–cosine algorithm surfaced in the literature.

In paper by Khalifeh et al. [3], HHO was used to optimize the water distribution network of Homashahr, a city in Iran, with the goal of minimizing the cost of such network. The solution that was yielded by the algorithm was compared to the current cost of the network, and the algorithm did optimize that cost.

In paper by Du et al. [4], Multi-objective Harris Hawks Optimization (MOHHO for short) was used as a part of the pipeline that tries to predict air pollution using time series. MOHHO was in this case used to optimize the parameters of Elaboration likelihood mo-del (ELM for short), which is used to perform time series prediction. In this paper, four multi-objective functions were used for optimization, while the Inverted Generational Distance (IGD for short) was used as a performance metric. By comparing the resu-lts obtained by using MOHHO with those obtained by MOGOA (Multi-Objective Grasshopper Optimiza-ti-on Algorithm), MOPSO (Multi-Objective Particle Swarm Optimization) and MSSA (Multi-objective Salp Swarm Algorithm), it was concluded that MOHHO has shown best results for 3 out of 4 multi-objective functions.

In this paper, the Harris hawks' algorithm is used to solve the following engineering problems.

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[5].

The second problem is pressure vessel optimization. The goal of this problem is to minimize material, welding and shaping costs. This problem was first suggested by Sandgren[6].

The third engineering problem is cantilever beam optmization, where minimal weight that fulfills the constraints is sought after. Gandomi has solved this problem using the Cuckoo Search Algorithm (CSA) [7].

The fourth problem to be solved is spring opmization, first formulated by Arora [8], with the goal of minimizing spring weight in accordance to constraints on minimum deflection, shear stress, surge frequency, limits of the other diameter and other design varible.

## 2. HARISS HAWKS ALGORITHM

The Harris' hawk (Parabuteo unicinctus) is a well-known bird of prey that survives in somewhat steady groups found in southern half of Arizona, USA. Based on Louis Lefebvre's research on avian „IQ", this type of bird is one of the most intelligent species of birds found in nature. The key feature of Harris hawk's behavior is that they hunt in groups, which are able to efficiently trace, encircle, flush out, and attack the prey. The main tactic of Harris' hawks to capture a prey is „surprise pounce", which is also known as „seven kills" strategy. This interesting hunting strategy is comprised of several hawks attacking the prey simultaneously from different locations. The hunt may be rapidly completed in one dive, or it may take multiple quick dives during several minutes. The main idea of this tactic is to confuse and drive the prey to exhaustion, increasing its vulnerability.

Figure 1 shows the behavior of a Harris hawk during hunting.

*Figure 1 - Behavior of a Harris hawk during hun-ting*

This algorithm consists of exploration phase, transition to exploitation phase, and exploitation phase.

In the exploration phase, the hawks perch randomly on some locations, waiting to detect the prey (designated as „rabbit"), which is the current best solution, based on two strategies, each of which having equal chance q to be selected. q is a random variable having the value between 0 and 1, used for a coin toss in this phase of the algorithm. If this variable' s value is less than 0.5, the first strategy is applied. Otherwise, the second strategy is applied. The two fore mentioned strategies are:

$$X\left(t+1\right)=\begin{cases} X_{rand}\left(t\right)-r_1\left|X_{rand}\left(t\right)-2r_2X\left(t\right)\right|, q\geq 0.5 \\ \left(X_{rabbit}\left(t\right)-X_m\left(t\right)\right)-r_3\left(LB+r_4\left(UB-LB\right)\right), q\leq 0.5 \end{cases} \qquad (1)$$

Where $X\left(t+1\right)$ is hawk position in iteration $t+1, X_{rand}\left(t\right)$ is the position of a randomly selected hawk, $X\left(t\right)$ is the hawk's current position, $X_{rabbit}\left(t\right)$ is the current position of the rabbit, $X_m\left(t\right)$ is the current average position of the hawks. The constants $r_1, r_2, r_3, r_4$ are random number inside (0,1) which are updated in each iteration, while UB and LB represent upper and lower bounds of variables. The first strategy generates solutions based on a random position, and the current position of the selected hawk, while the second strategy generates solutions based on current best solution, mean position of all hawks, and the lower and upper bounds for each variable.

Based on escaping energy of the prey, this algorithm switches between exploration and exploit-ation phases. This energy is modeled as:

$$E=2E_0\left(1-\frac{t}{T}\right) \qquad (2)$$

where E indicates the escaping energy of the prey, T is the maximum number of iterations, and $E_0$ is the initial state of its energy. When $\left|E\right|\geq 1$, this algorithm is in exploration phase, while for $\left|E\right|<1$, the algorithm is in exploitation phase.

In the exploitation phase, the hawks perform the surprise pounce described earlier, by attacking the intended prey. What can happen in this phase is that the prey has enough energy to escape the dive, meaning that the hawks will have to perform several dives on it. This is modeled by the variable r, and the prey has equal chances for escaping and not escaping the dive. Whatever the prey does, the hawks will perform a hard ($\left(\left|E\right|<0.5\right)$) or soft ($\left(\left|E\right|\geq 0.5\right)$) besiege.

In the case of soft besiege, when $r\geq 0.5$ and $\left|E\right|\geq 0.5$, the prey tries to escape the hawks, yet unsuccessfully.

This is modeled by the following equ-ations:

$$X\left(t+1\right)=\Delta X\left(t\right)-E\left|JX_{rabbit}\left(t\right)-X\left(t\right)\right| \qquad (3)$$

$$\Delta X\left(t\right)=X_{rabbit}-X\left(t\right) \qquad (4)$$

Where $\Delta X\left(t\right)$ is the difference between the position vector of the rabbit and the current position $t\,J=2\left(1-r_5\right)$ is the random jump strength of the rabbit while trying to run away, and r$_5$ is a random variable inside (0,1).

In order to simulate the motion of the rabbit, the value $J$ changes randomly in each iteration.

In the case of hard besiege, when $r\geq 0.5$ and , $\left|E\right|<0.5$ the rabbit does not have enough energy to escape, and the hawks move in for the attack. The positions in this case are updated using the following equation:

$$X\left(t+1\right)=X_{rabbit}\left(t\right)-E\left|\Delta X\left(t\right)\right| \qquad (5)$$

In the case of soft besiege with progressive rapid dives, when $\left|E\right|<0.5$ and $r\geq 0.5$, the rabbit has enough energy to escape the attack, while the hawks perform a soft besiege.

In order to model the movement of the escaping rabbit, the levy flight concept is used in this algorithm. In order to per-form a soft besiege, the hawks decide their next move based on equation:

$$Y=X_{rabbit}\left(t\right)-E\left|JX_{rabbit}\left(t\right)-X\left(t\right)\right| \qquad (6)$$

This result is then compared to the previous dive, to see if this dive will be good or not. If the dive is worse, the hawks perform random rapid dives when approaching the rabbit, based on the following equation:

$$Z=Y+S\,x\,LF\left(D\right) \qquad (7)$$

where D is the dimension of problem and S is a random vector by size $1\,x\,D$ and $LF$ is the levy flight function, which is calculated using the following equation:

$$LF(x)=0.01\frac{u\,x\,\sigma}{|\nu|^{\frac{1}{\beta}}},\sigma=\left(\frac{\Gamma(1+\beta)\,x\sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right)x\,\beta\,x\,2^{\frac{\beta+1}{2}}}\right)^{\frac{1}{\beta}} \tag{8}$$

The constants $u,\nu$ are random values inside (0,1), while $\beta$ is the default constant set to 1.5. Based on previous discussion, the final strategy for updating hawk position is given by:

$$X(t+1)=\begin{cases}Y, & if\ F(Y)<F(X(t))\\ Z, & if\ F(Z)<F(X(t))\end{cases} \tag{9}$$

In the case of hard besiege with progressive rapid dives, when $|E|<0.5$ i $r<0.5$,the rabbit does not have enough energy to escape, and the hard besiege is performed by the hawks. This time, the hawks try to decrease the distance to their average location with the escaping prey. This is modeled by the following equations:

$$X(t+1)=\begin{cases}Y, & if\ F(Y)<F(X(t))\\ Z, & if\ F(Z)<F(X(t))\end{cases} \tag{10}$$

Where Y and Z are given by the following equations:

$$Y=X_{rabbit}(t)-E\left|JX_{rabbit}(t)-X_m(t)\right| \tag{11}$$

$$Z=Y+S\ x\ LF(D) \tag{12}$$

The pseudocode for the algorithm is given below:

```
Pseudo-code of HHO algorithm

Inputs: The population size N and maximum number of iterations T
Outputs: The location of rabbit and its fitness value

Initialize the random population $X_i$ (i = 1 , 2 , . . . , N)
while (stopping condition is not met) do
        Calculate the fitness values of hawks
        Set X rabbit as the location of rabbit (best location)
        for (each hawk ($X_i$ )) do
              Update the initial energy $E_0$ and jump strength J
              $E_0$ =2rand()-1, J=2(1-rand())
              Update E
              if (|E|≥ 1) then
                 # exploration phase
                 Update positions using the formula for exploration phase
              if (|E| < 1) then
                 # exploitation phase
                 if (r ≥0.5 and |E|≥ 0 . 5 ) then
                    # soft besiege        Update positions using the formula for soft besiege
                 else if (r ≥0.5 and |E| < 0 . 5 ) then
                    # hard besiege
                    Update positions using the formula for hard besiege
                 else if (r < 0 . 5 and |E|≥ 0 . 5 ) then
                    # soft besiege with progressive rapid dives
                    Update positions using the formula for soft besiege with progressive rapid dives
                 else if (r < 0 . 5 and |E| < 0 . 5 ) then
                    # hard besiege with progressive rapid dives
                    Update positions using the formula for hard besiege with progressive rapid dives
Return $X_{rabbit}$
```

## 3. EXPERIMENTAL ENGINEERING EXAMPLES FOR OPTIMIZATION

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 2 a schematic view of speed reducer is shown.

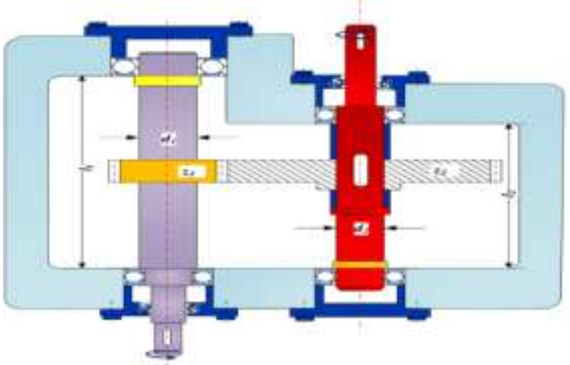

*Figure 2 - Schematic view of the speed reducer with variable parameters*

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 ($x_7$).

Goal function to be minimized is defined as:

$$f(x)=0,7854x_1x_2^2\left(3,3333x_3^2+14,933x_3-43,0934\right)-1,508x_1\left(x_6^2+x_7^2\right)+ \\ +7,4777\left(x_6^3+x_7^3\right)+0,7854\left(x_4x_6^2+x_5x_7^2\right) \tag{13}$$

Whilst the conditions to be met are:

$$g_1(x) = \frac{27}{x_1 x_2^2 x_3} - 1 \le 0 \qquad (14)$$

$$g_2(x) = \frac{397.5}{x_1 x_2^2 x_3^2} - 1 \le 0 \qquad (15)$$

$$g_3(x) = \frac{1.93 x_5^3}{x_2 x_3 x_7^4} - 1 \le 0 \qquad (16)$$

$$g_4(x) = \frac{1.93 x_5^3}{x_2 x_3 x_7^4} - 1 \le 0 \qquad (17)$$

$$g_5(x) = \frac{\left(\left(\frac{475 x_4}{x_2 x_3}\right)^2 + 16.9 \times 10^6\right)^{1/2}}{110 x_6^3} - 1 \le 0 \qquad (18)$$

$$g_6(x) = \frac{\left(\left(\frac{475 x_4}{x_2 x_3}\right)^2 + 157.5 \times 10^6\right)^{1/2}}{85 x_7^3} - 1 \le 0 \qquad (19)$$

$$g_7(x) = \frac{x_2 x_3}{40} - 1 \le 0 \qquad (20)$$

$$g_8(x) = \frac{5 x_2}{x_1} - 1 \le 0 \qquad (21)$$

$$g_9(x) = \frac{x_1}{12 x_2} - 1 \le 0 \qquad (22)$$

$$g_{10}(x) = \frac{15 x_6 + 1.9}{x_4} - 1 \le 0 \qquad (23)$$

$$g_{11}(x) = \frac{1.1 x_7 + 1.9}{x_5} - 1 \le 0 \qquad (24)$$

$$\begin{gathered} 2,6 \le x_1 \le 3,6 \,;\; 0,7 \le x_2 \le 0,8 \,;\; 17 \le x_3 \le 28 \\ 7,3 \le x_4 \le 8,3 \,;\; 7,3 \le x_5 \le 8,3 \,;\; 2,9 \le x_6 \le 3,9 \\ 5,0 \le x_7 \le 5,5 \end{gathered} \qquad (25)$$

### 3.2. Optimization of the pressure vessel

Second problem is optimization of a pressure vessel (figure 3), which consists of reducing costs of material, montage and welding costs.

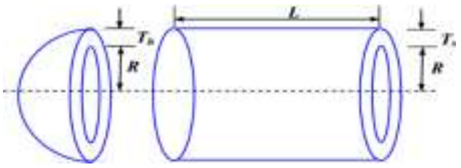

*Figure 3 - Schematic view of the speed pressure vessel with variable parameters*

Four variables are defined for this problem: radius of the shell (x1), length of the shell (x2), thickness of the shell (x3) and thickness of the dish end (x4).Goal function to be minimized is defined as:

$$f(x) = 0,6224 x_1 x_3 x_4 + 1,7781 x_2 x_3^2 + 3,1661 x_1^2 x_4 + 19,84 x_1^2 x_3 \qquad (26)$$

Whilst the conditions to be met are:

$$g_1(x) = -x_1 + 0,0193 x_3 \le 0; \qquad (27)$$

$$g_2(x) = -x_2 + 0,00954 x_3 \le 0; \qquad (28)$$

$$g_3(x) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0; \qquad (29)$$

$$g_4(x) = x_4 - 240 \le 0; \qquad (30)$$

$$0 \le x_i \le 100; \quad i = 1,2; \qquad (31)$$

$$10 \le x_i \le 200; \quad i = 3,4; \qquad (32)$$

### 3.3. Optimization of the cantilever beam

In this part, we will be solving the popular structural design problem of a cantilever beam (Figure 4). The goal is to minimize the weight of the cantilever beam.

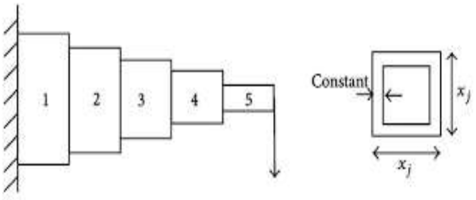

*Figure 4 - Schematic view of the speed cantilever beam with variable parameters*

As seen in Figure 4, the cantilever beam consists of five hollow, box shaped bearings with a square shaped frame. Project variables are lengths of the five squares which make up the cantilever beam.

The goal function is defined by the equation:

$$f(x) = 0,6224(x_1 + x_2 + x_3 + x_4 + x_5), \qquad (33)$$

Whilst the only constraint for this problem being:

$$g(x) = \frac{61}{x_1^3} + \frac{27}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \le 0, \qquad (34)$$

$$0,01 \le x_1, x_2, x_3, x_4, x_5 \le 100, \qquad (35)$$

### 3.4. Optimization of the tension/compression spring

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

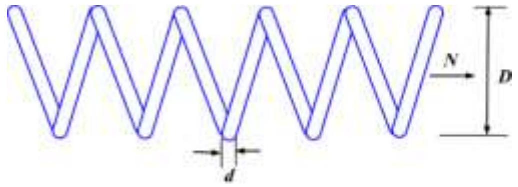

*Figure 5 - Schematic view of the spring with variable parameters*

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.

Goal function to be minimized is defined as:

$$f(x) = (x_3+2)x_2x_1^2 \qquad (36)$$

Whilst the conditions to be met are:

$$g_1(x) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \le 0; \qquad (37)$$

$$g_2(x) = \frac{4x_2^2 - x_1x_2}{12566\left(x_2x_1^3 - x_1^4\right)} + \frac{1}{5108x_1^2} - 1 \le 0; \qquad (38)$$

$$g_3(x) = 1 - \frac{140,45x_1}{x_2^2x_3} \le 0; \qquad (39)$$

$$g_4(x) = \frac{x_1+x_2}{1,5} \le 0; \qquad (40)$$

$$0,05 \le x_1 \le 2; \qquad (41)$$

$$0,25 \le x_2 \le 1,3; \qquad (42)$$

$$2 \le x_3 \le 15; \qquad (43)$$

## 4. RESULTS AND DISCUSSION

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

*Table 1. Comparison of results for the speed reducer*

| Variables | MACA[9] | GOA[12] | WCA[11] | **HHO** |
|---|---|---|---|---|
| $x_1$ | 3.499 | 3.5 | 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 | **3.35021** |
| $x_7$ | 5.286 | 5.287 | 5.286 | **5.28668** |
| *f(x)* | 3009.669 | 2996.964 | 2994.471 | **2995.542** |

Results of the Harris hawks' algorithm (HHO) will be compared to results obtained by the modified ant colony algorithm (MACA), grasshopper optimization algorithm (GOA), water cycle algorithm (WCA), Ant Lion Optimization (ALO), and Method of Moving Asymptotes (MMA), depending of solutions found in literature.

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

In the case of this problem, HHO has given better results than those found in literature.

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 | MACA[9] | GOA[10] | WCA[11] | **HHO** |
|---|---|---|---|---|
| $x_1$ | 0.822 | 0.8736 | 0.7781 | **0.798** |
| $x_2$ | 0.406 | 0.4318 | 0.3846 | **0.393** |
| $x_3$ | 42.602 | 45.2666 | 40.3196 | **41.255** |
| $x_4$ | 170.484 | 199.9998 | 200 | **187.369** |
| *f(x)* | 5964.50 | 7666.1258 | 5888.3327 | **5826.81** |

As in the case of the speed reducer problem, the HHO algorithm gave better results than those found in literature. For the cantilever beam design problem, the results shown in Table 3, along with the results obta-ined by ALO, MMA and GOA methods.

*Table 3. Comparison of results for the cantilever beam*

| Variables | ALO[12] | GOA[12] | MMA[12] | **HHO** |
|---|---|---|---|---|
| $x_1$ | 6.018 | 6.011 | 6.010 | **6.011** |
| $x_2$ | 5.311 | 5.312 | 5.300 | **5.349** |
| $x_3$ | 4.488 | 4.483 | 4.490 | **4.476** |
| $x_4$ | 3.497 | 3.502 | 3.490 | **3.491** |
| $x_5$ | 2.158 | 2.163 | 2.150 | **2.145** |
| *f(x)* | 1.339 | 1.339 | 1.340 | **1.34** |

In this case, the HHO gives the same result as the MMA algorithm, while ALO and GOA gave better results.

Experimental research was performed for the helical spring problem, and the results of HHO algorithm, along with the results for MACA, GOA, and WCA algorithms, are given in Table 4. The expected value for the goal function in this case is 0.012665.

*Table 4. Comparison of results for the spring*

| Variables | MACA[9] | GOA[12] | WCA[11] | **HHO** |
|---|---|---|---|---|
| $x_1$ | 0.0523 | 0.05 | 0.0516 | **0.0524** |
| $x_2$ | 0.3722 | 0.3174 | 0.3562 | **0.3750** |
| $x_3$ | 10.4141 | 4.4584 | 11.3004 | **10.3038** |
| *f(x)* | 0.0128 | 0.0128 | 0.0126 | **0.0126** |

In the case of helical spring optimization, HHO gives the same result as WCA, while MACA and GOA giving a slightly worse result.

## 5. CONCLUSION

Contemporary optimization entails applying modern optimization methods, particularly in the area of modern engineering, where in solving optimization problems often means solving nonlinear goal functions with a large number of constraints and variables. In order to solve optimization problems, one ne-eds to go through several steps.

Firstly, problem para-meters are to be identified. Secondly, the constraints for problem parameters need be acknowledged. Thirdly, optimization problem goals must be resea-rched and considered thoroughly. Lastly, based on the identified parameters, constraints, and goal functions, a convenient optimization method must be chosen.

This paper describes using Harris hawks' 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.

During the course of the research, it has been noted that increasing sea-rch agent and iteration count did not yield better so-lutions. Therefore, this combination of input para-meters was chosen, since it gives minimal execution time.

In the case of speed reducer and pressure vessel optimization, the Harris hawks' algorithm gives better results than other methods found in literature. The results for the other two optimization problems, namely the cantilever beam and helical spring problem, were shown to be near optimal.

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.

## 6. 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] 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.

[2] Essam H. Houssein, Mosa e. Hosney, Mohamed Elhoseny, Diego Oliva, Waleed M. Mohamed, M. Hassaballah: Hybrid Harris hawks optimization with cuckoo search for drug design and discovery in chemoinformatics, *Scientific Reports*, 10:14439, 2020.

[3] Saeid Khalifeh, Saeid Akbarifard, Vahid Khalifeh, Ebrahim Zallaghi: Optimization of water distribution of network systems using the Harris Hawks optimization algorithm (Case study: Homashahr city) *MethodsX*, Vol 7,2020,100948.

[4] Pei Du, Jianzhou Wang, Yan Hao, Tong Niu, Wendong Yang: A novel hybrid model based on multi-objective Harris hawks optimization algorithm for daily PM 2.5 and PM 10 forecasting, *Applied Soft Computing Journal*, Vol 96, 2020, 106620.

[5] Carlos A. Coello Coello: Constraint-handling ingenetic algorithms through the use of dominance based tournament selection, *Advanced Engineering Informatics*, 16, 193–203, 2002.

[6] Sandgren E, Nonlinear integer and discrete programming in mechanical design optimization, *Journal of Mechanical Design*;112(43):223-9, 1990.

[7] A. Gandomi, X. S. Yang, A. H. Alavi, Cucko search algorithm:a metaheuristic approach to solve structural optimization problems, *Springer-Verlag*, London, 2011.

[8] Arora J. S. *Introduction to optimum design*. New York: McGraw-Hill; 1989.

[9] S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Advances in Engineering Software* 69, 46-61, 2014.

[10] Đ. Jovanović, B. Milenković, M. Krstić, Application of Grasshopper Algorithm in Mechanical Engineering, *YOURS 2020*, pp.1-6.

[11] H. Eskandar, A. Sadollah, A. Bahreininejad, M. Hamdi: Water cycle algorithm – A novelmeta-heuristic optimization method for solving constrained engineering optimization problems, *Computers and Structures*, 2012.

[12] Shahrzad S, Seyedali Mirjalili and Andrew Lewis, Grasshopper Optimisation Algorithm: Theory and Application, *Advances in Engineering Software*, Volume105,March, pp.30-47, 2017.

[13] https://en.wikipedia.org/wiki/Harris%27s_hawk

## REZIME

IMPLEMENTACIJA ALGORITMA HARISOVOG JASTREBA ZA REŠAVANJE INŽENJERSKIH PROBLEMA

*U skorije vreme, tehnike optimizacije su postale veoma važne i popularne za primene u inženjerstvu. U ovom radu je demonstriran Algoritam Harrisovog jastreba (Harris Hawks' Optimization, HHO) za rešavanje optimizacionih problema u inženjerstvu. U drugom delu rada, date su biološke osnove, kao i objašenjenje principa rada algoritma. Nakon toga, HHO algoritam, kao i njegova primenjivost je detaljno objašnjena. Pseudo kod je napisan u MATLAB R2019a softverskom paketu. HHO algoritam je potom korišćen za rešavanje inženjerskih problema, poput: optimizacija reduktora, optimizacija suda pod pritiskom, optimizacija konzolnog nosača, i optimizacija zategnute/sabijene opruge. Statistički rezultati, su pokazali da je HHO daje kompetitivno dobre rezultate, u poređenju sa drugim metaheurističkim algoritmima.*

**Ključne reči:** *algoritam Harisovog jastreba, optimizacija, inženjerski problemi, metaheuristike*