Marine Predators' Algorithm: Application in Applied Mechanics

BRANISLAV N. MILENKOVIĆ, University Union - Nikola Tesla,
Faculty of Applied Science, Niš

MLADEN T. KRSTIĆ, University of Kragujevac,
Faculty of Mechanical and Civil Engineering, Kraljevo

Original scientific paper
UDC: 531.1/3

519.863:004.021

DOI: 10.5937/tehnika2105613M

In this paper we will demonstrate how Marine Predators Algorithm (MPA for short) can be used for solving certain optimization problems in applied mechanics. In the first part, biological fundamentals, as well as method explanation are given. Afterwards, MPA algorithm and its' applicability is explained in detail. The pseudo code for this algorithm was written using Matlab R2019a software suite. This algorithm can be used for optimization of engineering problems, such as: pressure vessel optimization, cantilever beam 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.

Key words: Marine predators' algorithm, optimization, applied mechanics

1. INTRODUCTION

A metaheuristic can be defined as a high-level algorithmic framework or approach that can be specialized to solve optimization problems. It consists of an iterative process, which guides other heuristics in a search for feasible solutions. Almost all of the metaheuristic algorithms have been conceptualized very recently, are simple in design, and their basic characteristics stem from, directly or indirectly, observing systems found in nature. Due to the simplicity of metaheuristic algorithms, they are easy to program in any programming language, since the code for most of these algorithms takes up to 100 lines of code. Such simple algorithms can be applied to a wide range of optimization problems with minimal code adjustments.

The most famous biologically inspired metaheuristic algorithms are: Whale Optimization Algorithm (WOA), Genetic Algorithm (GA), Ant Colony Optimization (ACO), Differential Evolution (DE), Lion Optimization Algorithm (LOA), Grasshopper Optimization Algorithm (GOA), Sine Cosine Algorithm (SCA), Bat Algorithm (BA), Particle Swarm Optimi-

zation (PSO), Grey Wolf Optimizer (GWO), Simulated Annealing (SA), etc.

In this paper the Marine predators algorithm (MPA), one of the biological metaheuristic algorithms created by Faramarzi [1], will be demonstrated. After the algorithm was published in 2019, it drew research attention, and it was implemented and improved in many areas of expertise.

In paper by Abdel-Basset et al. [2], four modified MPA algorithms adapted for multi-objective optimization problems was demonstrated. The first version is a basic modification of the MPA for multi-objective optimization problems. This version of the algorithm is called multi-objective MPA, or MMPA for short. The second version, a strategy called dominance strategy-based exploration-exploitation (DSSE) was incorporated into the first version, putting together dominance of the solutions with the exploration and exploitation phases. The solutions with high dominance undergo an exploitation phase, while others undergo the exploration phase. This algorithm is called M-MMPA for short. The third version adds Gaussian-based mutation to M-MMPA in order to guide exploration and exploitation strategies. The fourth version uses the Nelder-Mead simplex method with M-MMPA at the start of the optimization process to construct a front of the non-dominated solutions that will help M-MMPA to find more good solutions. This version is called M-MMPA-NMM for short. These four versions are tested on a large set of theoretical

Author's address: Branislav Milenković, University Union - Nikola Tesla, Faculty of Applied Science, Niš, Beogradska 10

e-mail: bmilenkovic@mi.sanu.ac.rs

Paper received: 30.08.2021.

Paper accepted: 21.09.2021.

and practical problems. For all the cases, the proposed algorithm and its variants are shown to be superior to a number of well-known multi-objective optimization algorithm.

In the paper by Soliman et al. [3], a MPA model was used to model triple-diode photovoltaic (TDPV) models using a new objective function. A numerical analysis was performed and compared with real-life photovoltaic (PV) panels. The MPA algorithm was successfully utilized to minimize the objective function, obtaining the nine-parameters of the TDPV model of the PV module. This is important, because these parameters cannot be obtained from the data-sheet. The algorithm has shown to successfully optimize the design parameters of two marketable PV modules.

In paper by Ramezani et al. [4], a developed version of the MPA is tested on standard test functions as well as real world optimization problem based on PID control applied to a DC motor. This version of the algorithm is based on the opposition-based learning method, chaos map, self-adaptive of population and switching between exploration and exploitation phases. This algorithm was compared with the original algorithm, as well as several other optimization algorithms. Eventually, the simulation results proved that the suggested algorithm has better results compared with other algorithms for the studied case studies.

This paper consists of five sections. In section 1 brief introduction is given, section 2 is devoted to detail description of Marine Predators Algorithm (MPA). In section 3, the description of optimization models is given and section 4 discusses experimental results for several engineering in applied mechanics. Finally, in section 5, conclusions based on experimental results are discussed.

2. MARINE PREDATORS' ALGORITHM

This nature-inspired algorithm is based on the survival of the fittest theory, where predators choose an optimal strategy in order to have higher chances of encountering prey. The foraging process of most species is modeled as a random walk, where the next position of the animal is determined by a set of probabilities.

Random walk is defined as a sequence of steps on some mathematical space, where the steps are determined by a mathematical distribution. The two random walks used by this algorithm are the Brownian motion and Levy flight.

Brownian motion is a type of random walk whose step sizes are determined from the normal distribution with a mean 0 and variance 1. [5]

$$f_B(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma^2} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \quad (1)$$

Levy flight is type of random walk whose step sizes are determined from a probability function defined by Levy distribution:

$$L(x_j) \approx |x_j|^{1-\alpha} \quad (2)$$

Where x_j is the flight length, α having the value between 1 and 2 is the power-law exponent [6]. In Figure 1, the distributions, 2D trajectories and 3D trajectories for the Brownian motion and Levy flight were shown. What is important to note is the greater range of values in the Brownian motion distribution. This means that the movement of search agents will be greater when the Brownian motion is used, which is convenient for the exploration phase of the MPA, while the smaller movement of search agents using the Levy flight is convenient for the exploitation phase of the MPA.

Figure 1 - Levy flight (a) distribution, b) 2D trajectory, c) 3D trajectory vs Brownian motion (d) distribution, e) 2D trajectory, and f) 3D trajectory [1]

In the first trial, the solutions are uniformly distributed over the search space. As explained later, this population will be used to construct both the Prey and the Elite part of the population.

$$X_0 = X_{\min} + rand(X_{\max} - X_{\min}) \quad (3)$$

Where the terms with the $X_{\min}$ and $X_{\max}$ subscripts design the lower and upper bounds for the variables, and $rand$ is a uniform random vector in the range of 0 to 1.

The population in this algorithm is split into two parts, Elite and Prey, represented by two matrices containing the values of all the dimensions of the problem.

First, the Prey part of the population is initialized, represented by the matrix shown below.

$$\text{Prey} = \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,d} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,d} \\ X_{3,1} & X_{3,2} & \cdots & X_{3,d} \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ X_{n,1} & X_{n,2} & \cdots & X_{n,d} \end{bmatrix}_{n \times d} \quad (4)$$

Where n is the population number, and d is the number of dimensions of the problem. Since predators chase the prey in real life, the positions of members in this population are used to update the positions of the predator population.

The predator population is constructed as such: the best member of the Prey population is the top predator, and that solution is replicated n times in order to construct the Elite matrix. The predator matrix is shown below:

$$\text{Elite} = \begin{bmatrix} X_{1,1}^I & X_{1,2}^I & \cdots & X_{1,d}^I \\ X_{2,1}^I & X_{2,2}^I & \cdots & X_{2,d}^I \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ X_{n,1}^I & X_{n,2}^I & \cdots & X_{n,d}^I \end{bmatrix}_{n \times d} \quad (5)$$

This matrix is of shape $n \times d$, n being the population size and d being the number of dimensions. $\vec{X}^I$ is the top predator vector, the one that is used to initialize the Elite matrix.

MPA optimization process is divided into three main phases of optimization considering different velocity ratio and at the same time mimicking the entire life of a predator and prey:

- high velocity ratio on when prey is moving faster than predator
- unit velocity ratio or when both prey and predator are moving at almost the same pace
- low velocity ratio when predator is moving faster than prey

These three phases are determined by the first, second and third equal part of the iteration count. Essentially, the first phase represents the exploration

phase, since the prey moves as fast as possible from the predator, moving through the search space with great speed. The second phase represents a transition to the exploitation phase, since the prey population moves at a slower pace, while the predator population speeds up. The third phase represents the exploitation phase, since only the predator population moves at a slow pace, converging to the best solution.

It is important to add some remarks in terms of labels used. $\vec{stepsize}_i$ represents the step that the i -th member of a population makes in the given iteration. $\vec{R}_B$ represents a vector of random numbers generated using the Brownian motion random walk, the $\vec{R}_L$ represents a vector of random numbers generated using the Levy flight random walk, while $\vec{R}$ represents a vector of random numbers generated using the uniform distribution in the range $[0, 1]$. Parameter P is equal to 0.5, while parameter CF is the adaptive factor, controlling the step size of predator movement during a given iteration (the equation given later in text). $\vec{Elite}_i$ and $\vec{Prey}_i$ are the i -th member of the Elite and Prey populations, respectively

Figure 2 - The three Marine Predators Algorithm optimization phases [1]

In high velocity ratio or when predator is moving faster than prey. This scenario happens in the initial iterations of optimization, where the exploration matters. In the case when $v \geq 10$, where v is the velocity ratio of prey to predator, the best strategy for predator is not moving at all, while the prey moves from the predator in random fashion, where the randomness is defined by the Normal distribution representing the Brownian motion. The equation for prey movement is shown below:

$$\begin{aligned} \vec{stepsize}_i &= \vec{R}_B \otimes (\vec{Elite}_i - \vec{R}_B \otimes \vec{Prey}_i) \\ \vec{Prey}_i &= \vec{Prey}_i + P * \vec{R} \otimes \vec{stepsize}_i \end{aligned} \quad (6)$$

In unit velocity ratio or when both prey and predator are moving at the same pace. It mimics that both of them are looking for their prey. This phase represents slowly transitioning to exploitation. In this phase, both exploration and exploitation matters. Consequently, half of the population is designated for exploration and the other half for exploitations. In this phase, prey is responsible for exploitation and predator for exploration. Based on this rule, in the unit velocity ratio ($v \approx 1$), if prey moves in Levy, the best strategy for predator is Brownian. Thus, this study considers prey moves in Levy while predator moves in Brownian. The movement equations for Elite and Prey in this phase of the algorithm are:

$$\begin{aligned}\overrightarrow{stepsize}_i &= \overrightarrow{R}_L \otimes (\overrightarrow{Elite}_i - \overrightarrow{R}_L \otimes \overrightarrow{Prey}_i) \\ \overrightarrow{Prey}_i &= \overrightarrow{Prey}_i + P * \overrightarrow{R} \otimes \overrightarrow{stepsize}_i\end{aligned}\quad (7)$$

$$\begin{aligned}\overrightarrow{stepsize}_i &= \overrightarrow{R}_B \otimes (\overrightarrow{R}_B \otimes \overrightarrow{Elite}_i - \overrightarrow{Prey}_i) \\ \overrightarrow{Elite}_i &= \overrightarrow{Elite}_i + P * \overrightarrow{CF} \otimes \overrightarrow{stepsize}_i\end{aligned}\quad (8)$$

$$CF = (1 - \frac{iter}{maxiter})^{2*iter/maxiter}$$

Where iter is the current iteration, and maxiter is the maximum number of iterations.

In low velocity ratio or when predator is moving faster than prey. This scenario happens in the last phase of the optimization process which is mostly associated with high exploitation capability. In low velocity ratio ($v = 0.1$) the best strategy for predator is Levy. The movement equation for Elite in this phase of the algorithm is:

$$\begin{aligned}\overrightarrow{stepsize}_i &= \overrightarrow{R}_L \otimes (\overrightarrow{R}_L \otimes \overrightarrow{Elite}_i - \overrightarrow{Prey}_i) \\ \overrightarrow{Elite}_i &= \overrightarrow{Elite}_i + P * \overrightarrow{CF} \otimes \overrightarrow{stepsize}_i\end{aligned}\quad (9)$$

Another point which causes a behavioral change in marine predators is environmental issues such as the eddy formation or Fish Aggregating Devices (FADs) effects. The FADs are considered as local optima and their effect as trapping in these points in search space. Consideration of these longer jumps during simulation avoids stagnation in local optima. The FAD effect is mathematically presented as:

$$\overrightarrow{Prey}_i = \begin{cases} \overrightarrow{Prey}_i + CF[\overrightarrow{X}_{min} + \overrightarrow{R} \otimes (\overrightarrow{X}_{max} - \overrightarrow{X}_{min})] \otimes \overrightarrow{U}, & \text{if } r \leq FADs \\ \overrightarrow{Prey}_i + [FADs(1-r) + r](\overrightarrow{Prey}_{r1} - \overrightarrow{Prey}_{r2}), & \text{if } r > FADs \end{cases} \quad (10)$$

Where FADs = 0.2 is the probability of FADs effect on the optimization process. $\overrightarrow{U}$ is the binary vector with arrays including zero and one. This is constructed by generating a random vector in [0,1] and changing its array to zero if the array is less than 0.2 and one if it is greater than 0.2. r is the uniform

random number in [0,1]. $\overrightarrow{X}_{min}$ and $\overrightarrow{X}_{max}$ are the vectors containing the lower and upper bounds of the dimensions. r_1 and r_2 subscripts denote random indexes of prey matrix.

Based on the highlighted points, marine predators have a good memory in reminding the place where they have been successful in foraging. This capability simulated by memory saving in MPA. After updating the Prey and implementing FADs effect, this matrix is evaluated for fitness to update the Elite. The fitness of each solution of the current iteration is compared to its equivalent in prior iteration, and the current one replaces the solution if it is more fitted.

The pseudo-code for the algorithm is given below:

```

Initialize search agents (Prey) populations
i=1,...,n While termination criteria are not met
Calculate the fitness and construct the Elite matrix
If Iter<Max_Iter/3
    Update prey based on Eq. 6 in paper [1]
Else if Max_Iter/3<Iter<2*Max_Iter/3
    For the first half of the populations
    (i=1,...,n/2)
        Update prey based on Eq. 7 in paper [1]
    For the other half of the populations
    (i=n/2,...,n)
        Update prey based on Eq. 8 in paper [1]
Else if Iter>2*Max_Iter/3
    Update prey based on Eq. 9 in paper [1]
End (if)
Accomplish memory saving and Elite update
Applying FADs effect and update based on Eq. 10
in paper [1]
End while

```

3. OPTIMIZATION MODELS

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.

Figure 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).

Goal function to be minimized is defined as:

$$f(x) = 0,7854x_1x_2^2(3,3333x_3^2 + 14,933x_3 - 43,0934) - 1,508x_1(x_6^2 + x_7^2) + 7,4777(x_6^3 + x_7^3) + 0,7854(x_4x_6^2 + x_5x_7^2) \quad (11)$$

Whilst the conditions to be met are:

$$g_1(x) = \frac{27}{x_1x_2^2x_3} - 1 \leq 0 \quad (12)$$

$$g_2(x) = \frac{397,5}{x_1x_2^2x_3^2} - 1 \leq 0 \quad (13)$$

$$g_3(x) = \frac{1,93x_5^3}{x_2x_3x_7^4} - 1 \leq 0 \quad (14)$$

$$g_4(x) = \frac{1,93x_5^3}{x_2x_3x_7^4} - 1 \leq 0 \quad (15)$$

$$g_5(x) = \frac{\left(\left(\frac{475x_4}{x_2x_3}\right)^2 + 16,9 \times 10^6\right)^{1/2}}{110x_6^3} - 1 \leq 0 \quad (16)$$

$$g_6(x) = \frac{\left(\left(\frac{475x_4}{x_2x_3}\right)^2 + 157,5 \times 10^6\right)^{1/2}}{85x_7^3} - 1 \leq 0 \quad (17)$$

$$g_7(x) = \frac{x_2x_3}{40} - 1 \leq 0 \quad (18)$$

$$g_8(x) = \frac{5x_2}{x_1} - 1 \leq 0 \quad (19)$$

$$g_9(x) = \frac{x_1}{12x_2} - 1 \leq 0 \quad (20)$$

$$g_{10}(x) = \frac{15x_6 + 1,9}{x_4} - 1 \leq 0 \quad (21)$$

$$g_{11}(x) = \frac{1,1x_7 + 1,9}{x_5} - 1 \leq 0 \quad (22)$$

$$2,6 \leq x_1 \leq 3,6; 0,7 \leq x_2 \leq 0,8; 17 \leq x_3 \leq 28$$

$$7,3 \leq x_4 \leq 8,3; 7,3 \leq x_5 \leq 8,3; 2,9 \leq x_6 \leq 3,9$$

$$5,0 \leq x_7 \leq 5,5 \quad (22)$$

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).

Figure 4 - Schematic view of the speed pressure vessel with variable parameters

Goal function to be minimized is defined as:

$$f(x) = 0,6224x_1x_3x_4 + 1,7781x_2x_3^2 + 3,1661x_1^2x_4 + 19,84x_1^2x_3 \quad (23)$$

Whilst the conditions to be met are:

$$g_1(x) = -x_1 + 0,0193x_3 \leq 0; \quad (24)$$

$$g_2(x) = -x_2 + 0,00954x_3 \leq 0; \quad (25)$$

$$g_3(x) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0; \quad (26)$$

$$g_4(x) = x_4 - 240 \leq 0; \quad (27)$$

$$0 \leq x_i \leq 100; \quad i = 1, 2; \quad (28)$$

$$10 \leq x_i \leq 200; \quad i = 3, 4; \quad (29)$$

3.3 Optimization of the cantilever beam

In this part, we will be solving the popular structural design problem of a cantilever beam (Figure 5). The goal is to minimize the weight of the cantilever beam.

Figure 5 - Schematic view of the cantilever beam with variable parameters

As seen in Figure 5, 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:

$$f(x) = 0,6224(x_1 + x_2 + x_3 + x_4 + x_5), \quad (30)$$

Whilst the only constraint for this problem being:

$$g(x) = \frac{61}{x_1^3} + \frac{27}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0, \quad (31)$$

$$0,01 \leq x_1, x_2, x_3, x_4, x_5 \leq 100, \quad (32)$$

3.4 Optimization of the cone clutch

In Figure 6, a schematic view of cone clutch, along with all the project variables, is shown.

Figure 6 - Schematic view of cone clutch with variable parameters

The cone clutch problem must be designed for minimum volume coupling to two constraints. Problem variables are: inner radius of the coupling $R_1 \equiv x_1$ and outer radius of the coupling $R_2 \equiv x_2$.

Goal function to be minimized is defined as:

$$f(X) = (x_1^3 - x_2^3) \quad (33)$$

Whilst the conditions to be met are:

$$g_1(X) = \frac{x_1}{x_2} \geq 2 \quad (34)$$

$$g_2(X) = \frac{(x_1^2 + x_1 x_2 + x_2^2)}{(x_1 + x_2)} \geq 5 \quad (35)$$

$$1 \leq x_1, x_2 \leq 10 \quad (36)$$

4. RESULTS AND DISCUSSION

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

Results of the Marine predators' algorithm (MPA) will be compared to results obtained by the Grey Wolf Optimizer (GWO), Cuckoo Search (CS), Improved Cuckoo Search (ICS), 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 MPA 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[7]	GOA[8]	ICS[9]	MPA
x1	3.499	3.5	3.5	3.499
x2	0.699	0.7	0.7	0.7
x3	17	17	17	17
x4	8.051	7.3	7.3	7.3
x5	8.084	7.8	7.8	7.715
x6	3.351	3.350	3.350	3.35021
x7	5.286	5.287	5.286	5.28665
f(x)	3009.669	2996.964	2996.2578	2994.470

In the case of this problem, MPA 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	GWO[7]	GOA[8]	WCA[10]	MPA
x1	0.822	0.8736	0.7781	0.778
x2	0.406	0.4318	0.3846	0.384
x3	42.602	45.2666	40.3196	40.319
x4	170.484	199.9998	200	200
f(x)	5964.50	7666.1258	5888.3327	5885.331

As in the case of the speed reducer problem, the MPA 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 obtained by ALO, MMA and GOA methods.

Table 3. Comparison of results for the cantilever beam

Variables	ALO[11]	GOA[8]	MMA[12]	MPA
x_1	6.018	6.011	6.010	5.978
x_2	5.311	5.312	5.300	4.876
x_3	4.488	4.483	4.490	4.466
x_4	3.497	3.502	3.490	3.479
x_5	2.158	2.163	2.150	2.139
$f(x)$	1.339	1.339	1.340	1.303

In this case, the MPA gives near the same result as the MMA algorithm, ALO and GOA.

Experimental research was performed for the cone clutch problem, and the results of MPA algorithm, along with the results for GWO, GOA, and WCA algorithms, are given in Table 4.

Table 4. Comparison of results for the cone clutch

Variables	GWO[7]	CS[13]	GOA [8]	MPA
x_1	4.286	4.285	4.286	4.285
x_2	2.142	2.142	2.143	2.142
$f(x)$	68.893	68.887	68.894	68.877

In the case of cone clutch optimization, MPA gives the same result as CS, while GWO and GOA giving a slightly worse result.

5. CONCLUSION

This paper describes using Marine predators' optimization algorithm in order to solve a few engineering problems with a constant number of variables. For this algorithm, 25 search agents and 500 iterations were chosen as input parameters. Larger numbers for search agents and iteration count did not yield better solutions, all the while increasing the time to finish the algorithm. Smaller numbers of search agents and iteration count did speed up the algorithm, yet solutions better than those shown in the paper were not reached. Therefore, we consider these values of input parameters to be optimal for the presented set of problems.

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

This paper focuses on application of optimization algorithms on engineering problems in applied mechanics. 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] Faramarzi A, Heidarinejad M, Mirjalili S, Gandomi A, Marine Predators Algorithm: A nature – inspired metaheuristic, *Expert Systems with Applications* 152, 113377, 2020.
- [2] Basset A, Mohamed R, Mirjalili S, Chakraborty R., Ryan M, An efficient Marine Predators Algorithm for Solving Multi-objective Optimization Problems: Analysis and Validations, *IEEE Access* PP(99):1-1, March 2021.
- [3] Soliman M, Hasanien H, Alkuhayli A, Marine Predators Algorithm for Parameters Identification of Triple-Diode Photovoltaic Models, *IEEE Access* PP(99):1-1, August 2020.
- [4] Ramezani M, Razmjoooy N, A New Improved Model of Marine Predator Algorithm for Optimization Problems, *Arabian Journal for Science and Engineering*, May 2021.
- [5] Einstein A, *Investigations on the theory of the Brownian motion*, 1956.
- [6] Mantegna R, Fast, accurate algorithm for numerical simulation of Levy stable stochastic processes, *Physical Review E*, 49(5) 4677-4683, 1994.
- [7] Milenković B, Krstić M, Jovanović Đ, Primena algoritma sivo vuka za rešavanje inženjerskih optimizacionih problema, *Tehnika*, Vol. 76, Br. 1. str. 50-57, 2021.
- [8] Saremi S, Mirjalili S, Lewis A, Grasshopper optimization algorithm: theory and application, *Advances in Engineering Software* 105 30-47, 2017.
- [9] Bulatović R, Bošković G, Savković M, Gašić M, Improved Cuckoo Search (ICS) algorithm for constrained optimization problems, *Latin American Journal of Solids and Structures*, 11(8) 1349-1362, 2014.

- [10]Eskandar H, Sadollah A, Bahreininejad A, Hamdi M, Water cycle algorithm-A novel metaheuristic optimization method for solving constrained optimization problems, *Computers and Structures*, 110-111 151-166.
- [11] Mirjalili S, The Ant Lion Optimizer, *Advances in Engineering Software* 83, 80-98, 2015.
- [12]Chickermane H, Gea H, Structural optimization using a new local approximation method, *International Journal for Numerical Methods in Engineering*, 39:829-46, 1996.
- [13]Miodragović G, *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.
- [14]<https://www.mathworks.com/matlabcentral/fileexchange/74578-marine-predators-algorithm-mpa>

REZIME

ALGORITAM MORSKIH PREDATORA: PRIMENA U PRIMENJENOJ MEHANICI

U ovom radu ćemo pokazati kako se Algoritam morskih predatora (skraćeno MPA) može koristiti za rešavanje određenih problema optimizacije u primenjenoj mehanici. U prvom delu su date biološke osnove, kao i objašnjenje metoda. Nakon toga, MPA algoritam i njegova primenljivost su detaljno objašnjeni. Pseudo kod za ovaj algoritam napisan je pomoću programskog paketa Matlab R2019a. Ovaj algoritam se može koristiti za optimizaciju inženjerskih problema, kao što su: optimizacija posuda pod pritiskom, optimizacija konzolnih greda, optimizacija konusnog kvačila i optimizacija reduktora brzine. Na kraju su prikazani svi rezultati prethodno navedenih problema, kao i poređenje rezultata sa drugim metodama.

Ključne reči: Algoritam morskih predatora, optimizacija, primenjena mehanika