INNOVATIVE MECHANICAL ENGINEERING **ISSN 2812-9229 (Online)**
University of Niš, Faculty of Mechanical Engineering
VOL. 1, NO 2, 2022, PP. 106-115

Original scientific paper*

MECHANICAL ENGINEERING DESIGN OPTIMIZATION USING THE HONEY BADGER ALGORITHM

Đorđe Jovanović[1], Branislav Milenković[2], Mladen Krstić[3]

[1]Mathematical Institute of SASA, Belgrade, Serbia

[2]Faculty of Applied Sciences, Union Nikola Tesla University, Niš, Serbia

[3]Faculty of Mechanical and Civil Engineering, University of Kragujevac, Kraljevo, Serbia

Abstract*. Engineering problems are some of the currently most prominent research issues. One of the classes of engineering problems are engineering design problems, where a set of variables is calibrated for the optimization function to have a minimal or maximal value. This function often considers energy efficiency, cost efficiency, production efficiency, etc., in engineering design. One of the ways in which such problems are solved is the application of metaheuristics. This paper demonstrates how the Honey Badger Algorithm can be used to solve certain optimization problems in mechanical engineering. Firstly, a brief review of the Honey Badger Algorithm, as well as its biological inspiration, is given along with the most important formulae. The pseudo code for this algorithm was written using the MATLAB R2020a software suite. The Honey Badger Algorithm was used for the optimization of engineering problems, such as: pressure vessel optimization, 3D beam optimization, multiple-disk clutch brake and cantilever beam optimization. The results presented in this paper show that the Honey Badger Algorithm can produce relevant results in the field of engineering design problems.*

Key words: *Honey badger, Algorithm, Optimization, Engineering design, Metaheuristics*

1. INTRODUCTION

Machine design is a field of research where a mechanical part is designed to fulfill certain criteria, whilst keeping in mind its physical limitations. By extracting all the necessary variables, we can define a problem using mathematical programming. The key

***Received: August 10, 2022 / Accepted October 31, 2022.**
Corresponding author: Branislav Milenković
Faculty of Applied Sciences, University Union Nikola Tesla, Dušana Popovića 22a
E-mail: bmilenkovic92@gmail.com

© 2022 by Faculty of Mechanical Engineering, University of Niš, Serbia

elements of this problem are variables, conditions, and a fitness or goal function. Variables represent key features of a certain mechanical part, while conditions represent limitations to the mechanical design. The fitness or goal function is a criterion that should be minimized or maximized (e.g., minimizing the cost by minimizing the surface, or maximizing the force an element imposes).

Problems of mathematical programming are sometimes solved by using metaheuristics, a class of stochastic-deterministic algorithms designed for solving complex NP-hard problems. Metaheuristics are divided into two categories: single-solution (s-based), where a single solution to the problem is transformed in order to find a better solution, and population (p-based), where a population of solutions is used to explore the search space and converge to the best solution. Many p-based algorithms were used for solving machine design problems and will be described in what follows.

The Marine Predator Algorithm (MPA) [1] is based upon the predator-prey model which exists in nature, dividing the population into these two parts for this algorithm. First, solutions are randomized to create the initial population of prey, used to determine the movement of the population, while the predator population is made by replicating the best solution of the prey population. In each phase, the predator population moves faster with each iteration, while the prey population slows down its movement. This process happens in three equally long phases of the algorithm.

The Dingo Optimization Algorithm (DOA) [2] is based upon the hunting behavior of dingoes. The population is split into three roles: best search agent (alpha), second best search agent (beta), and other search agents (dingoes), with each role moving independently through the search space. An interesting characteristic of this algorithm is that the encircling phase guarantees that the prey will be surrounded from each part of the solution space.

The Tunicate Swarm Algorithm (TSA) [3] is inspired by small sea creatures, the tunicates, and their swarm behavior. This algorithm features only one phase, where the movement decreases as iterations pass. The main characteristic of this algorithm is that, during movement, conflicts among the search agents are resolved, so that the search agents do not end up in the same space, and the tunicates move first towards the best neighbor, and, ultimately, towards the best search agent.

The Firefly Algorithm (FA) [4] was inspired by the behavior of a population of fireflies. Firstly, a population of fireflies is generated at random. Then, based on the quality of the solution, each firefly emits a light around itself. The better the solution, the stronger the light. As the iterations pass, the fireflies converge towards the best solution.

The Aquila Optimization algorithm (AO) [5] was inspired by a common bird of prey of the same name. The main characteristic of this algorithm is that it is split into two phases: exploration (in the first two-thirds of iterations) and exploitation (the final third of iterations). In each of these two phases, the Aquila may move in an expanded or narrowed manner, determined by a random number variable. This division of phases and types of movement in each phase is this algorithm’s most prominent feature.

The Ant Colony Optimization (ACO) [6] uses the movement of ants in a colony as inspiration. The problem is divided into parts of a trail, while the whole solution represents a trail which the ants follow. At first, the ants might follow any trail in order to construct the solution, but they leave a pheromone trail behind themselves, making the next batch of ants follow that trail with higher probability. Higher quality solutions leave a stronger trail, bringing the algorithm to convergence.

The Reptile Search Algorithm (RSA) [7] is based upon the hunting behavior of crocodiles. This algorithm is divided into two phases: encircling (exploration) and hunting (exploitation). In the exploration phase, the crocodiles use two different strategies: high walking and belly walking. This is done in order to explore the solution space as much as possible.

The Honey Badger Algorithm (HBA) [8] has shown ability in solving parameter optimization problems. In [9], the HBA was used to optimize parameters of a neural network which is used to identify the model of proton-exchange membrane fuel cells. In fuel cell design [10], the HBA was used to determine the parameter of the proton exchange membrane fuel cell (PEMFC) and has shown promising results on three datasets. In the field of photovoltaic cells, the HBA was used to accurately identify the single diode model (SDM), dual diode model (DDM), and three diode model (TDM) parameters of solar photovoltaic cells [11]. Since the core of engineering design problems is parameter optimization, the effectiveness of HBA in solving such problems will be presented in this paper.

This paper consists of five sections. In section 1, a brief introduction to the field of metaheuristics, engineering design problems and used algorithms is given. In section 2, the Honey Badger Algorithm (HBA) is described in detail. In section 3, optimization models are discussed, and section 4 compares experimental results for the selected set of data. In the last section, conclusions based on experimental results are discussed.

2. HONEY BADGER ALGORITHM

Honey badger is a mammal with black and white fluffy fur, known for its fearless nature, often found in the semi-deserts and rainforests of Africa, Southwest Asia, and the Indian subcontinent. This dog size (60 to 77 cm body length and 7 to 13 kg body weight) fearless forager preys on sixty different species including dangerous snakes.

The Honey Badger Algorithm (HBA) is inspired by the foraging behavior of honey badgers (Fig. 1). There are two ways in which the honey badger searches for food: by using its own sense of smell, or by following the honeyguide bird, known for easily locating honey. The algorithm is split into two phases: digging phase and honey phase.

Fig. 1 Honey badger attacks python

The main characteristics of this algorithm are the following: using randomization for initialization, two phases that are randomly determined, and Cardioid motion in the exploration phase. The i-th badger in the following text is denoted by x_i, while the current best solution is denoted by x_{prey}. A general note is needed about variables, and that is that all the variables that are named as r_x, where x is a number, are random numbers in the interval [0, 1], which have uniform distribution.

In the initialization phase, the position of each badger is determined by:

$$x_i = lb_i + r_1 \times (ub_i - lb_i) \tag{1}$$

where lb_i and ub_i represent the lower and the upper bounds for the variables, respectively.

As it is with each p-type algorithm, a smooth transition from exploration to exploitation must be performed with each iteration. This is done by using the formula:

$$\alpha = C \times exp\left(\frac{-t}{t_{max}}\right) \tag{2}$$

where C is a constant greater or equal to 1, with the default being 2. T represents the current iteration, while t_{max} represents the maximum number of iterations

Then, the intensity I is calculated for each badger:

$$I_i = r_2 \times \frac{S}{4\pi d_i^2} \tag{3}$$

where S is the source strength or concentration strength, and d_i is the distance from the i-th badger to the prey. The intensity is related to the concentration strength of the prey and the distance between it and the i-th badger.

Because this shape has two directions in which the badgers can move, for each badger the direction is determined by using the flag F:

$$F = \begin{cases} 1 & if r_6 \leq 0.5 \\ -1 & else, \end{cases} \tag{4}$$

Then, the position of each badger is updated according to:

$$x_{new} = x_{prey} + F \times \beta \times I \times x_{prey} +$$
$$+F \times r_3 \times \alpha \times d_i \times |cos(2\pi r_4) \times [1 - cos(2\pi r_5)]| \tag{5}$$

where β is a parameter larger than one, representing the ability to get food (the default value is set to 5), and d_i is the distance between the prey and the i-th badger.

In the honey phase, the badgers seek help from the guide bird to determine where the honey is located, and their position is updated according to:

$$x_{new} = x_{prey} + F \times r_7 \times \alpha \times d_i \tag{6}$$

After all the badgers move, the prey is determined again, and this algorithm is repeated until the stopping criterion (processor time or number of iterations) is met.

3. OPTIMIZATION PROBLEMS

In this section, each optimization problem is described in detail, namely: fitness or goal function, the practical basis for the problem, which parameter it consists of, and which conditions are required of the variables. Every step of this process was done using the MATLAB R2020a software suite. In each example, the fitness function is denoted by f(x), while the i-th constraint is represented by $g_i(x)$.

The primary concern for designing a pressure vessel (Fig. 2) is to reduce the costs of material, montage and welding. The problem takes into account the variables presented in Fig. 2: shell thickness (x_1), head thickness (x_2, T_s), shell radius (x_3, R), and shell length (x_4).

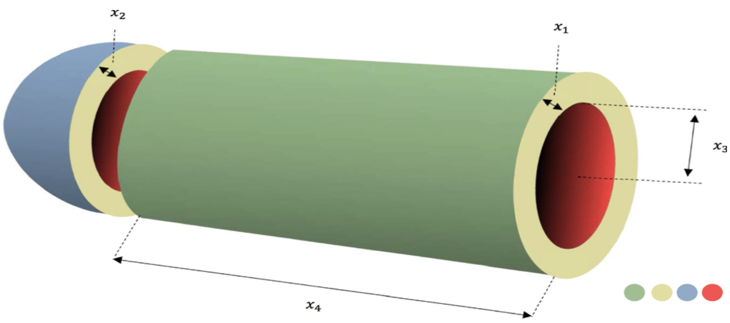

Fig. 2 Pressure vessel design problem

The mathematical formulation constraints of this problem are described in Eqs. (7-13)

$$f(x) = 0{,}6224x_1x_3x_4 + 1{,}7781x_2x_3^2 + 3{,}1661x_1^2x_4 + 19{,}84x_1^2x_3 \quad (7)$$

$$g_1(x) = -x_1 + 0{,}0193x_3 \leq 0; \quad (8)$$

$$g_2(x) = -x_2 + 0{,}00954x_3 \leq 0; \quad (9)$$

$$g_3(x) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0; \quad (10)$$

$$g_4(x) = x_4 - 240 \leq 0; \quad (11)$$

$$0 \leq x_i \leq 100; \quad i = 1,2; \quad (12)$$

$$10 \leq x_i \leq 200; \quad i = 3,4; \quad (13)$$

The second problem consists of minimizing cross-section heights of all elements of a cantilever beam, which is shown in Fig. 3. A vertical shift of point A is defined in advance, having a specified upper limit. The beam is under continual load (q_1, q_2) on the horizontal parts of the beam, as well as horizontal force F, which affects the vertical part of the beam.

The goal function to be minimized is defined as:

$$f(X) = 0.8x_1 + x_2 + 0.8x_3, \quad (14)$$

while the conditions to be met are:

$$u_A(X) = \left[\frac{11.2480 \cdot 10^{-3}}{x_1^3} + \frac{3.539\overline{9} \cdot 10^{-3}}{x_2^3} + \frac{0.3840 \cdot 10^{-3}}{x_3^3}\right] \leq 0.05[m],$$
$$0.1 \leq x_1 \leq 0.9,$$
$$0.1 \leq x_2 \leq 0.9, \qquad (15)$$
$$0.1 \leq x_3 \leq 0.9,$$

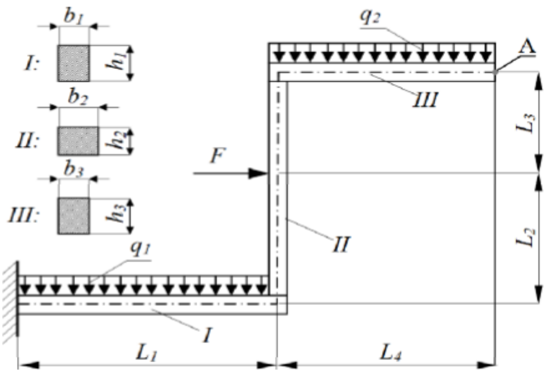

Fig. 3 3D beam design problem

The objectives of the problem are to minimize the mass of the brake. The disc brake optimization model has four variables (as shown in Fig. 4) – inner radius of the discs (x_1), outer radius of the discs (x_2), engaging force (x_3) and number of the friction surfaces (x_4).

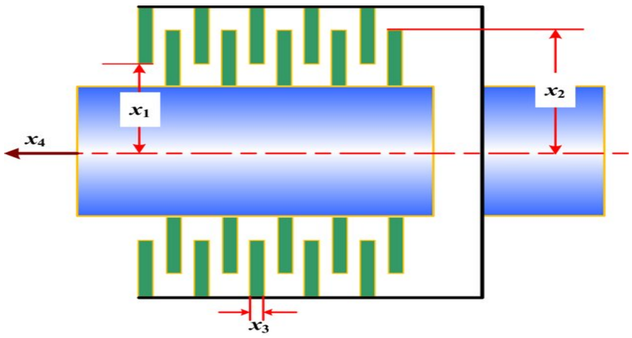

Fig. 4 Multiple-disk clutch brake design problem

The objective functions and constraints of the disc brake design optimization are defined as follows:

$$f_1(x) = 4.9\times10^{-5}\left(x_2^2 - x_1^2\right)\left(x_4 - 1\right), \tag{16}$$

$$g_1(x) = \left(x_2 - x_1\right) - 20 \geq 0. \tag{17}$$

$$g_2(x) = 30 - 2.5\left(x_4 + 1\right) \geq 0. \tag{18}$$

$$g_3(x) = 0.4 - \frac{x_3}{3.14\left(x_2^2 - x_1^2\right)} \geq 0. \tag{19}$$

$$g_4(x) = 1 - \frac{2.22\times10^{-3} x_3\left(x_2^3 - x_1^3\right)}{\left(x_2^2 - x_1^2\right)^2} \geq 0. \tag{20}$$

$$g_5(x) = \frac{2.66\times10^{-2} x_3 x_4\left(x_2^3 - x_1^3\right)}{\left(x_2^2 - x_1^2\right)} - 900 \geq 0. \tag{21}$$

$$55 \leq x_1 \leq 80,$$
$$75 \leq x_2 \leq 110,$$
$$1000 \leq x_3 \leq 3000,$$
$$2 \leq x_4 \leq 20. \tag{22}$$

A cantilever beam (Fig. 5) is an important element in mechanical engineering, whose design is to be handled with utmost care. Minimization of said beam weight represents the main goal in design. The lengths of the five bearings are this problem's variables.

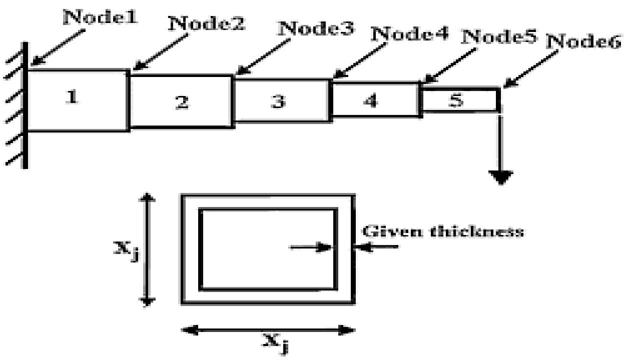

Fig. 5 Cantilever beam

The mathematical formulation constraints of this problem are described in Eqs. (23-24):

$$f(x) = 0,6224\left(x_1 + x_2 + x_3 + x_4 + x_5\right), \tag{23}$$

$$g(x)=\frac{61}{x_1^3}+\frac{27}{x_2^3}+\frac{19}{x_3^3}+\frac{7}{x_4^3}+\frac{1}{x_5^3}-1\leq 0, \tag{24}$$

The considered variable ranges are given in Eq (25).

$$0,01\leq x_1,x_2,x_3,x_4,x_5\leq 100, \tag{25}$$

4. RESULTS

In this section, the results obtained by using HBA on a set of selected engineering problems is given.

In the case of the pressure vessel problem, what is considered a good result for the goal function is 5885.3327, with the results shown in Table 1.

Table 1 Comparison of results for the first example (design of a pressure vessel)

Variables	GWO [12]	WCA [13]	HBA
x_1	0.822	0.778	0.778
x_2	0.406	0.431	0.384
x_3	42.602	40.319	40.319
x_4	170.484	200.000	200.000
$f(x)$	5964.500	5888.332	5885.331

As the table shows, the HBA yields results that are better than the current literature.
In Table 2, a comparison of results for the design of a 3D beam optimization problem is shown.

Table 2 Comparison of results for the second example (3D beam)

Variables	ANSYS [14]	GOA [15]	HBA
x_1	0.804	0.804	0.804
x_2	0.569	0.569	0.569
x_3	0.345	0.345	0.345
$f(x)$	1.461	1.409	1.490

In Table 2, a comparison of results for the design of a cantilever beam optimization problem is shown. In this case, the HBA gives a result comparable to the ones described in papers [14,15]. ANSYS/Design Optimization-Subproblem Approximation Method gives a better result, yet it violates the condition $g_1<0.05$.

A detailed display of the results obtained by the HBA and a comparison with the results obtained by other methods, for the disk brake problem, are shown in Table 3. For this problem, the HBA yields better results than GA and PSA.

Table 3 Comparison of results for the third example (disk brake)

Variables	PSA [16]	GA [17]	HBA
x_1	62.600	65.800	55.000
x_2	83.500	86.100	75.000
x_3	2920.900	2982.400	1000.000
x_4	11.000	10.000	2.000
$f(x)$	1.790	1.660	0.127

In the case of the cantilever beam design problem, the results are presented in Table 4. The results from the literature, where the ALO and MMA methods are used for this problem, are to be found in the same table.

Table 4 Comparison of results for the fourth example (cantilever beam)

Variables	ALO [18]	MMA [19]	HBA
x_1	6.018	6.010	5.965
x_2	5.311	5.300	4.873
x_3	4.488	4.490	4.460
x_4	3.497	3.490	3.476
x_5	2.158	2.150	2.138
$f(x)$	1.339	1.340	1.301

As can be seen from the results, the HBA gives near optimal results, close to the MMA and ALO methods.

4. CONCLUSION

This paper describes the HBA algorithm and applies it to a selected set of engineering problems. This set is comprised of the pressure vessel, cantilever beam, 3D beams and multiple-disk clutch brake design problems, which are described in detail, and highlighted by figures, goal function and descriptions of constraints.

The input parameters chosen are 30 search agents and 1000 iterations of the algorithm. The reason for this is that, as was discovered during the research, increasing the values of these input parameters did not yield better solutions.

In the case of the pressurized vessel optimization and the multiple-disk clutch brake, the HBA provided better results than the methods to which it was compared. In the case of the other two optimization problems, cantilever beam and 3D beam, the HBA yielded near optimal solutions.

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. B.Milenković, Đ.Jovanović, M.Krstić., An application of Dingo Optimization Algorithm (DOA) for solving continuous engineering problems, FME Transactions, 2022, Vol.50, No.2, pp. ISSN 1451-2092 (print), doi:10.5937/fme2201331M
3. S.Kaur, L.Awasthi, A.Sangal, G.Dhiman., Tunicate Swarm Algorithm:A new bio-inspired based metaheuristic paradigm for global optimization, Engineering Applications of Artificial Intelligence 90 (2020).
4. G.Miodragović, M.Bošković., The Application of Firefly Algorithm for Solving Problems of Applied Mechanics, IMK 14-Istraživanje i razvoj 18 (2012), ISBN 0354-6829.
5. L. Abualigah, D.Yousri, M.Elaziz, A.Ewees, M.Al-qaness, A.Gandomi., Aquila Optimizer: A novel meta-heurristic optimization algorithm, Computers and Industrial Engineering, Vol.157, July 2021, 107250.
6. M.Dorigo, M.Birattari, T.Stuzle., Ant Colony Optimization, IEE Computational Intelligence Magazine, December 2006.
7. L.Abualigah, M.Elaziz, P.Sumari, Z.Geem, A.Gandomi., Reptile Search Algorithm (RSA): A nature-inspired metaheuristic optimizer, Expert Systems with Applications 191(11):116158, doi:10.1016/j.eswa.2021.116158
8. Hashim et al, Honey Badger Algorithm: New metaheuristic algorithm for solving optimization problems, Mathematics and Computers in Simulation, Volume 192, February 2022, Pages 84-110
9. Han and Ghadimi, Model identification of proton-exchange membrane fuel cells based on a hybrid convolutional neural network and extreme learning machine optimized by improved honey badger algorithm, Sustainable Energy Technologies and Assessments, Volume 52, Part A, August 2022, 102005
10. Almodfer et al, Improving Parameter Estimation of Fuel Cell Using Honey Badger Optimization Algorithm, Front. Energy Res., Sec. Sustainable Energy Systems and Policies
11. Lei et al, Solar Photovoltaic Cell Parameter Identification Based on Improved Honey Badger Algorithm, Sustainability 2022, 14, 8897
12. Seyedali Mirjalili, Seyed Mohammad Mirjalili, Andrew Lewis, Grey Wolf Optimizer, Advances in Engineering Software, Volume 69, 2014.
13. 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.
14. F.Fedorik, Using optimizations algorithms by designing structures, doctoral thesis, Institute of Structural Mechanics, Faculty of Civil Engineering Brno, University of Technology, (2013), 119-140.
15. Dj. Jovanović, B. Milenković, M. Krstić, Application of Grasshopper Algorithm in Mechanical Engineering, YOURS, pp.1-6, 2020.
16. P. Sabarinath, M. R. Thansekhar, and R. Saravanan, Performance Evaluation of Particle Swarm Optimization Algorithm for optimial design of belt pulley system in Swarm, Evolutionary, and Memetic Computing, vol. 8297 of Lecture Notes in Computer Science, pp. 601–616, Springer, Cham, Switzerland, 2013.
17. J. L. Marcelin, Genetic Optimisation of Gear International Journal of Advanced Manufacturing Technology, vol. 17, no. 12, pp. 910–915, 2001.
18. Mirjalili S, The Ant Lion Optimizer, Adv. Eng. Software, 83:80-98, 2015.
19. Chickermane H, Gea H. Structural optimization using a new local approximation method, Int J Number Methods Eng, 39:829-46, 1996.