Mechanics Transport Communications

ISSN 1312-3823 (print)

ISSN 2367-6620 (online)

volume 21, issue 2, 2023

Academic journal http://www.mtc-aj.com article № 2366

APPLICATION OF BELUGA WHALE ALGORITHM FOR CLASSICAL ENGINEERING PROBLEMS

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

[1]Faculty of Applied Sciences, Niš,
[2]Leoni Wiring System, Kraljevo,
SERBIA

Key words: *Beluga whale, Algorithm, Optimization, Engineering*

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 Beluga Whale Algorithm can be used to solve certain optimization problems in mechanical engineering. Firstly, a brief review of the Beluga Whale 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 R2022a software suite. The Beluga Whale Algorithm was used for the optimization of engineering problems, such as: 3D beam optimization, multiple-disk clutch brake and cantilever beam optimization. The results presented in this paper show that the Beluga Whale Algorithm can produce relevant results in the field of engineering design problems.*

INTRODUCTION

In the scientific field of algorithms and optimization, there is a class of problems, called NP-hard problems. The solutions to these are very hard to obtain using deterministic approaches whilst verifying the value of the solution has low algorithmic complexity. Examples of these problems are community detection, traveling salesman problem, maximum satisfiability problem, knapsack problem, etc.

One of the reasons why the problems of this nature are hard to solve is the size of the so-called solution space. Given the set of input variables, as well as their possible values, the solution represents a set of all possible solutions for the problem in question. NP-hard problems possess such a vast solution space that using deterministic methods to search through it would consume computer resources and time inefficiently.

One class of algorithms used to solve NP-hard problems are metaheuristic algorithms. This class of algorithm consists of a stochastic component to search through the solution space as best as possible in the amount of time given, and a deterministic component to find the highest possible quality solution given by this stochastic search. The metaheuristic algorithms do not guarantee an optimal solution to NP-hard problems, but they do guarantee that, within a given time span, they output solutions of substantial quality.

Metaheuristic algorithms are divided into two categories: s-based and p-based. S-based (also called single-solution-based metaheuristic algorithms) use a single solution during the course of the algorithm, using fine-grained deterministic procedures to improve that solution. Advantages of such approach are memory efficiency and fine-grained solution improvement, while the disadvantage is lack of breadth in solution space exploration. Examples of s-based metaheuristic algorithms are: variable neighborhood search (VNS), simmulated annealing (SA), hill climbing (HC), tabu search (TS) etc. On the other hand, p-based algorithms (also called population-based metaheuristic algorithms) use a pool of solutions to explore the solution phase, which ultimately converges to the best solution in the pool. What is important to be said is that none of these solutions are improved by a fine-grained procedure, as it is the case with s-based algorithms. Although they explore the solution space better than s-based algorithms, they use more memory. Examples of p-based metaheuristic algorithms are: grasshopper optimization (GA), bat optimization (BA), marine predator optimization (MPA), etc.

One class of optimization problems is engineering design problems, where a set of variables, along with the corresponding optimization function, determines the design of a mechanical part. This design is optimized in regards to certain requirements, such as cost and mass. In this paper, the beluga whale optimization (BWO) will be used to solve a set of engineering design problems. In Chapter 1, the details of the optimization algorithm will be given. In Chapter 2, the set of engineering problems solved will be presented, along with a comparison with results of state-of-the-art algorithms for the same problem. In Chapter 3, a conclusion will be given.

BELUGA WHALE ALGORITHM

Beluga whale optimization algorithm, first proposed in [1], simulates the living habits of beluga whales in the ocean. Beluga whales gather in groups from 2 to 25 members, and the adult members are of pure white color (figure 1). As almost all metaheuristic algorithms, this algorithm contains exploration and exploitation phases. What is characteristic for this algorithm is that the transition between exploration and exploitation is smooth, and it is given by the formula:

$$B_f = B_0\left(1 - T/2T_{\max}\right) \tag{1}$$

where t is the current iteration, T is the maximum iteration number, and B0 is a random number in the interval (0, 1). When the balance factor Bf > 0.5, it corresponds to the exploitation stage. The balance factor Bf ≤ 0.5 corresponds to the exploitation stage. With the increasing number of iterations, the probability of the exploitation phase increases, while the probability of the exploration phase decreases.

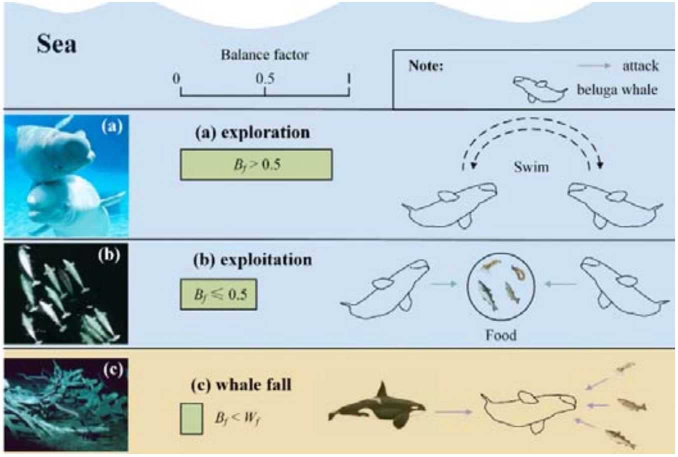

Fig.1. Behaviors of beluga whales

Exploration phase is inspired by the synchronous or mirror behaviors of beluga whale in swimming or diving. This is reflected in the fact that the movement of the j-th dimension of the problem is different and based whether j is even or odd. The formulas for the movement of beluga whales in exploration phase is:

$$\begin{cases} X_{i,j}^{T+1} = X_{i,p_j}^{T} + \left(X_{r,p_1}^{T} - X_{i,p_j}^{T}\right)\left(1+r_1\right)\sin\left(2\pi r_2\right), & j = even \\ X_{i,j}^{T+1} = X_{i,p_j}^{T} + \left(X_{r,p_1}^{T} - X_{i,p_j}^{T}\right)\left(1+r_1\right)\cos\left(2\pi r_2\right), & j = odd \end{cases} \quad (2)$$

is the position of the i-th individual on the j-th dimension during the next iteration. P and P are randomly selected positive integers in [1, dim], and they are not equal. X ,
and X, represent the position of the i-th and r-th individuals under the current iteration, and r1 and r2 are random numbers in (0, 1).

Beluga whales, being social animals which hunt in groups, share information about their location with each other, so that they know what is their position relative to the best member of the population, as well as other individuals. Assuming that beluga whales can use a Lévy flight strategy to capture prey, the specific formula is shown as:

$$X_i^{T+1} = r_3 X_{best}^{T} - r_4 X_i^{T} + C_1 \cdot L_F \cdot \left(X_r^{T} - X_i^{T}\right) \quad (3)$$

$$L_F = 0.05 \times \frac{u \times \sigma}{|v|^{1/\beta}} \quad (4)$$

X and X represent the current positions of the i and r individuals in the current iteration. X is the new position of the i-th individual, X is the best position, and r3 and r4 are random numbers in (0, 1). C1 is a random jump, which measures the intensity of the Lévy flight. L is a random number consistent with the Lévy distribution, calculated by the following formula:

$$\sigma = \left(\frac{\Gamma\left(1+\beta\right) \times \sin\left(\pi\beta/2\right)}{\Gamma\left(\left(1+\beta\right)/2\right) \times \beta \times 2^{(\beta-1)/2}}\right)^{1/\beta} \quad (5)$$

where u and v are random numbers obeying a normal distribution, and β is a constant set to 1.5.

Beluga whales are highly dependent on their group behavior. During their lifetime, they either migrate to another group, or risk dying alone. This is modeled in the algorithm by updating the position of individuals according to step size of whale fall:

$$X_i^{T+1} = r_5 X_i^T - r_6 X_r^T + r_7 X_{step} \qquad (6)$$

where r5, r6, and r7 are random numbers in (0, 1), and Xstep is the step size of whale fall established as:

$$X_{step} = \left(u_b - l_b\right)\exp\left(-C_2 T / T_{\max}\right) \qquad (7)$$

C2 is the step size factor, which is related to the probability of beluga whale fall and population size
(C2 = 2Wf × n). ub and lb are the upper and lower bounds of variables, respectively. The probability of a beluga whale fall is calculated as a linear function, and the formula is as follows:

$$W_f = 0.1 - 0.05 T / T_{\max} \qquad (8)$$

OPTIMIZATION PROBLEMS AND RESULTS

In this section, each optimization problem is described in detail, namely: fitness or goal function, the practical basis for the problem, which parameter it is consisted of, and which conditions are required of the variables. Every step of this process was done using the MATLAB R2022a software suite. In each example, the fitness function is denoted by f(x), while the i-th constraint is represented by $g_i(x)$.

Second problem consists of minimizing cross□section heights of all elements of a cantilever beam, which is shown in Figure 2. 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 horizontal parts of the beam, as well as horizontal force F, which affects the vertical part of the beam.

Goal function to be minimized is defined as:

$$f(X) = 0.8x_1 + x_2 + 0.8x_3, \qquad (9)$$

Whilst the conditions to be met are:

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

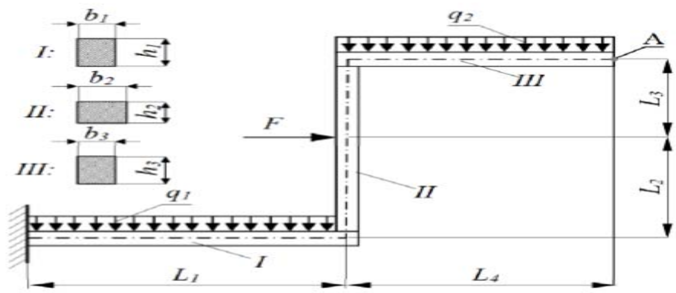

Fig. 2. 3D beam design problem

The objectives of the problem are to minimize the mass of the brake. The disc brake optimization model has four variables (as shown in Figure 3) that are 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).

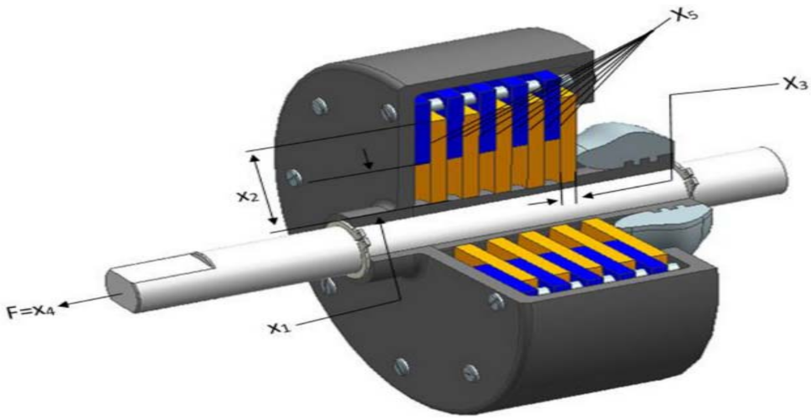

Fig. 3. Multiple-disk clutch brake design problem

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

$$f_1(x) = 4.9\times 10^{-5}\left(x_2^2 - x_1^2\right)\left(x_4 - 1\right), \quad (11)$$

$$g_1(x) = \left(x_2 - x_1\right) - 20 \geq 0. \quad (12)$$

$$g_2(x) = 30 - 2.5\left(x_4 + 1\right) \geq 0. \quad (13)$$

$$g_3(x) = 0.4 - \frac{x_3}{3.14\left(x_2^2 - x_1^2\right)} \geq 0. \quad (14)$$

$$g_4(x) = 1 - \frac{2.22\times 10^{-3} x_3\left(x_2^3 - x_1^3\right)}{\left(x_2^2 - x_1^2\right)^2} \geq 0. \quad (15)$$

$$g_5(x) = \frac{2.66\times 10^{-2} x_3 x_4\left(x_2^3 - x_1^3\right)}{\left(x_2^2 - x_1^2\right)} - 900 \geq 0. \quad (16)$$

$$55 \leq x_1 \leq 80,$$
$$75 \leq x_2 \leq 110,$$
$$1000 \leq x_3 \leq 3000,$$
$$2 \leq x_4 \leq 20. \quad (17)$$

Cantilever beam (Figure 4) is an important element in mechanical engineering, whose design is to be handled with utmost care. Minimization of the said beam's weight represents the main goal in design. The lengths of the five bearings are this problem's variables.

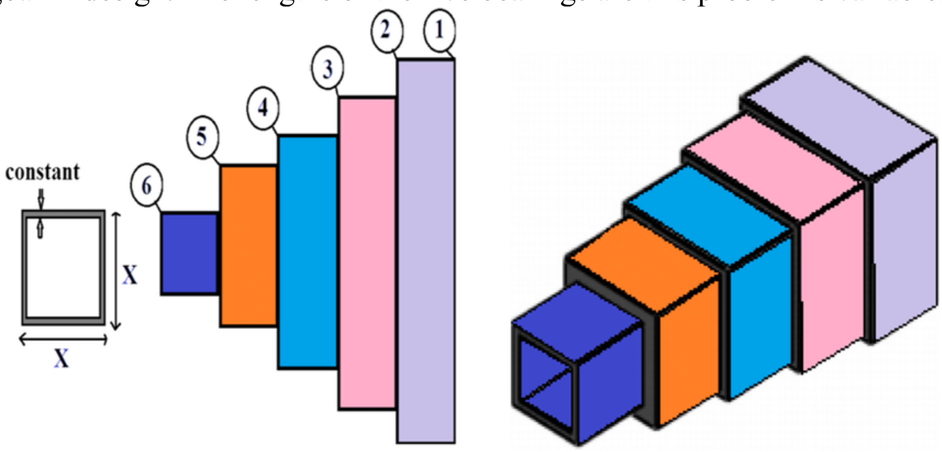

Fig. 4. Cantilever beam

The mathematical formulation constraints of this problem are described in Eqs. (18) to (19) :

$$f(x) = 0,6224(x_1 + x_2 + x_3 + x_4 + x_5), \quad (18)$$

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

The considered variable ranges are described in Eq (20).

$$0,01 \leq x_1, x_2, x_3, x_4, x_5 \leq 100, \quad (20)$$

As the results show, BWO has shown results that are better than current literature.
In Table 1, a comparison of results for design of a 3D beam optimization problem are shown.

Table 1. Comparison of results for the first example (3D beam)

Variables	ANSYS [2]	GOA [3]	BWO
x_1	0,804	0,804	0,793
x_2	0,569	0,569	0,595
x_3	0,345	0,345	0,332
$f(x)$	1,461	1,409	1,495

In Table 2, a comparison of results for design of a cantilever beam optimization problem are shown. In this case, the BWO gives a result comparable to ones described in papers [2] and [3]. ANSYS/Design Optimization-Subproblem Approximation Method gives a better result, yet it violates the condition g1<0.05.

A detailed display of the results obtained by BWA and a comparison with several results obtained by other methods, for the problem of disk brake, are shown in Table 2.

Table 2. Comparison of results for the second example (disk brake)

Variables	PSA [4]	GA [5]	BWO
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 this problem, BWO has given better results than GA and PSA.

In the case of the cantilever beam design problem, the results are presented in Table 3. The results from literature, where the ALO and MMA methods are used for this problem, are to be found in the same table.

Table 3. Comparison of results for the third example (cantilever beam)

Variables	ALO [6]	MMA [7]	BWO
x_1	6,018	6,010	6,297
x_2	5,311	5,300	4,599
x_3	4,488	4,490	4,538
x_4	3,497	3,490	3,476
x_5	2,158	2,150	2,047
$f(x)$	1,339	1,340	1,308

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

CONCLUSION

This paper describes the BWO algorithm, and applies it to a selected set of engineering problems. This set is comprised of: cantilever beam, 3D beams and multiple-disk clutch brake design problems, which are described in detail, and highlighted by figures, goal function and constraints' descriptions.
The input parameters that were chosen are 50 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 case of multiple-disk clutch brake, the BWO gives better results than the methods to which it was compared. In the case of the other two optimization problems, namely: cantilever beam and 3D beam the BWO 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]. C.Zhong, G.Li, Z.Meng., 2022, *Beluga whale optimization: A novel nature-inspired metaheuristic algorithm,* Knowledge-Based Systems, 109215.

[2]. 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.

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

[4]. 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.

[5]. J. L. Marcelin, *Genetic Optimisation of Gear* International Journal of Advanced Manufacturing Technology, vol. 17, no. 12, pp. 910–915, 2001.

[6]. Mirjalili S, *The Ant Lion Optimizer*, Adv Eng Software,;83:80-98, 2015.

[7]. Chickermane H, Gea H. *Structural optimization using a new local approximation method*, Int J Number Methods Eng, 39:829-46, 1996.

[8].https://www.mathworks.com/matlabcentral/fileexchange/112830-beluga-whale-optimization-bwo#:~:text=Discussions%20(0)-,Beluga%20whale%20optimization%20(BWO)%20algorithm%20is%20a%20swarm%2Dbased,phase%2C%20and%20whale%20fall%20phase