An Application Of Dingo Optimization Algorithm (DOA) For Solving Continuous Engineering Problems

Branislav Milenković
Junior Research Assistant
Mathematical Institute of SASA

Đorđe Jovanović
Junior Research Assistant
Mathematical Institute of SASA

Mladen Krstić
Junior Research Assistant
University of Kragujevac
Faculty of Mechanical and Civil Engineering

In the current research, problems in engineering are becoming more and more prominent. One of the classes of engineering problems in engineering design problems, where a set of variables is calibrated in order for the optimization function to have a minimal or maximal value. This function considers energy efficiency, cost efficiency, and production efficiency in engineering design. One of the ways such problems are solved is metaheuristics. In this paper, we demonstrate how Dingo Optimization Algorithm (DOA) can be used to solve certain optimization problems in mechanical engineering. Firstly, a brief review of the DOA and its biological inspiration is given, along with the most important formulae. The pseudo-code for this algorithm was written using MATLAB R2020a software suite. Dingo Optimization Algorithm (DOA) was used to optimize engineering problems, such as pressure vessel optimization, stepped cantilever beam, car side-impact, and cone clutch optimization. The results presented in this paper show that the DOA can produce relevant results in engineering design problems.

Keywords: *optimization; dingo; engineering; metaheuristic.*

1. INTRODUCTION

Most engineering problems can be formulated as optimization problems. Different methods have been studied in mathematical programming, operations research, etc. Since 2000, quite a few novel metaheuristics have been developed to solve optimization problems better. Today, 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 Optimization (PSO), Grey Wolf Optimizer (GWO), Simulated Annealing (SA), etc. The differential evolution algorithm (DE) was applied by Gašić, Abderazek for solving structural optimization problems [1]. In the paper by Khalifeh et al. [2], Harris Hawks' Optimization (HHO) was used to optimize a practical problem of a city in Iran called Homashahr, where the problem was minimizing the cost of the water distribution network. The solution yielded by the algorithm was compared to the current cost of the network, and the algorithm optimized that cost.

In computer science, a particular class of problems exists called NP-complete problems or nondeterministic polynomial problems. The main characteristic of such problems is that the solution space is large to the extent that searching it systematically and in detail demands temporal and spacial computer resources in plethora. This means that a given problem' solution may be verified in polynomial time, but whether or not there is a solution cannot be determined. One method of solving such problems is using metaheuristics. Metaheuristics represent a semi-randomized approach to tackling the solution space.

There are two parts to each metaheuristic algorithm: exploration and exploitation. Since the search space is large, it must be explored using a randomized best-effort approach, which is done during prospecting. This performs "large hops" in solution space but does not pursue the best solution in detail. After a part of the solution, space is chosen, it is the job of exploitation to systematically search all possibilities in the vicinity. Metaheuristic algorithms may use one solution to solve a certain problem (called s-based or single solution metaheuristics) or a population of solutions (called p-based or population-based). S-based transforms a single solution to solve a problem, while the p-based uses a population of solutions to search the solution space and converge to the best member of the population. S-based metaheuristics conserve computer memory, but special considerations should be taken in exploring the neighborhood of a solution and the solution space. P-based metaheuristics cover the solution space better while lacking the flexibility and the memory efficiency of s-based metaheuristics. Typical representatives of the s-based metaheuristics are Local Search, Simulated Annealing, and Variable Neighborhood Search. In contrast, the typical representatives of the p-based Meta-heuristics are Whale Optimization Algorithm, Genetic Algorithm, and Ant Colony Optimization.

Metaheuristic algorithms have been applied to many engineering problems, namely in the field of

Received: November 2021, Accepted: March 2022
Correspondence to: Branislav Milenković
Mathematical Institute of SASA,
Kneza Mihaila 36, 10000 Belgrade, Serbia
Email: bmilenkovic92@gmail.com
doi:10.5937/fme2201331M

engineering design [3,4], industry scheduling [5,6], and machining [7].

The whale optimization algorithm [8] (WOA) mi–mics the behavior of humpback whales. This algo–rithm was used to solve various optimization problems, such as the knapsack problem [9] and the traveling salesman problem [10]. Other search agents are moving towards the prey, which is the current best solution, with varying strategies. In the exploration phase, each whale moves towards a random whale to further explore the search space, while in the exploitation phase, the whales encircle the prey and move towards it.

The swarm intelligence of ants inspires ant colony optimization [11] (ACO). The main characteristic of this algorithm is that each time a search agent solves the search space, he leaves a trail of pheromones behind. In time, this trail lingers, while if more and more ants go through the same route, the trail is strengthened. This pheromone trail may represent a part of the solution or the solution space and defines the motion of search agents in the next operation, with the parts of the solution where the pheromone trail is the strongest having the highest probability of being selected. This algorithm was used to solve, for instance, the traveling salesman problem [12] and engineering design problems [13].

Bats' echolocation inspires the Bat optimization algorithm [9] (BA). The spatial orientation of bats works so that a bat finds out about its surroundings based on the echo his shriek returns. The bats let out a shriek of random wavelength in a random search direction. Based on the average loudness of bats and the chosen direction, the bats move towards the prey, indicated by the best solution. This algorithm was used to solve engineering problems [14,15], as well as distribution generation problems [16].

A genetic algorithm [17] is a class of algorithms based on natural selection. First, a population is created, where each solution consists of parts called genes. A portion of the population is selected at the end of each iteration, based on a higher fitness value (goal function), to be further modified in the next iteration. Then, the selection is mutated using various operators by selecting a part of the solution and changing it somehow. This algorithm was applied in structural design [18] and fuel cell design [19] problems.

The sine cosine algorithm [20] bases the movement of the population on sine and cosine functions. The initial population is created randomly, and it moves towards or from the best solution using sine and cosine-based mathematical models. Several random and adaptive variables are present in the algorithm to better explore the solution space and achieve convergence. This algorithm has been used in solving a wide range of optimization problems [21,22].

Simulated annealing [23] is inspired by the annealing process in metallurgy. The main problem in optimization is getting stuck in local optima. This problem is particularly present in single-solution metaheuristics. In order to avoid this, the solution space must be explored in a certain manner. Simulated annealing does this by accepting worse solutions than the current best with a probability. As time or iterations pass, the probability of accepting a worse solution is lower, therefore focusing on the exploitation part of the algorithm. Simulated annealing was used in engineering problems [24,25].

The Grey wolf optimization [26] (GWO) draws inspiration from wolf packs. In the grey wolf hierarchy, there are alpha, beta, delta, and omega members. The wolves hunt the prey (best solution), each in their own manner. The alpha, beta, and delta are the second, third and fourth best search agents, while the omega wolves are other search agents. The main characteristic of this algorithm is that the alpha, beta, and delta wolves move directly towards the prey. In contrast, all other wolves move with a mean vector of movement vectors of alpha, beta, and delta wolves. This algorithm solved heat distribution [27] and mechanical design problems [28].

In this paper, the Dingo Optimization Algorithm (DOA) is used to solve selected engineering problems. The first problem is the optimization of a pressure vessel, where the main goal is to reduce the material, montage, and welding costs. This problem was solved by Differential Evolution (DE) algorithm in [29]. The second engineering problem is stepped cantilever beam optimization, where the goal is to minimize the weight under a set of constraints. Gandomi used the cuckoo Search Algorithm (CSA) to solve this problem [30].

The third problem is the automobile side-impact optimization problem, intending to minimize total vehicle weight using eleven design variables. This problem was first subjected in a paper by Gu [31]. The fourth and last engineering problem is cone coupling optimization. Better cone couplings have less volume, which is the optimal value for this problem. This example was defined in [32].

This paper consists of five sections. In section 1 brief introduction to the field of metaheuristics, engineering design problems, and used algorithms is given. In section 2, the Dingo Optimization Algorithm (DOA) is described in detail. In section 3, optimization models are discussed, and section 4 compares experimental results for the selected set of engineering problems in applied mechanics. Finally, in section 5, conclusions based on experimental results are discussed.

2. DINGO OPTIMIZATION ALGORITHM

Dingoes are mammals that live predominantly in Australia and represent an ancient lineage of dogs [33]. The name "dingo" stems from the Dharug language used by the Indigenous Australians of the Sydney area. The word "dingo" stands for "dog" in their language. The taxonomy of this species is unclear since the scientific community cannot agree whether dingoes are a subspecies of dogs or wolves or an entire species on their own. There are four taxonomical terms for the dingo: Canis familiaris, Canis familiaris dingo, Canis dingo, and Canis lupus dingo. The main characteristic of dingoes that inspired this algorithm is the social hierarchy. When dingos hunt, they either hunt alone or in cooperative packs. The typical diet consists of small game such as rabbits, rodents, birds, and lizards. This species is not a carnivorous one since dingos can also eat fruits and plants. When near human settlements, they may scavenge for food, a common occurrence in their Asian range. In Figure 1, a typical representation of the hunting behavior of dingos is given.

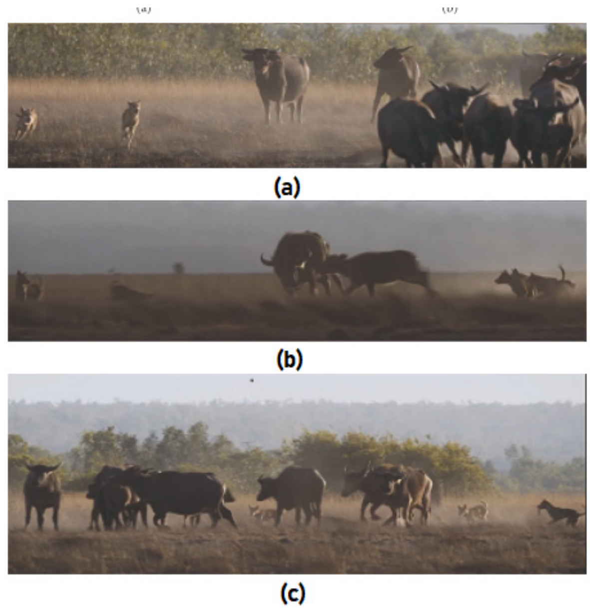

Figure 1. Hunting behavior of dingoes: (a) chase and approach the prey, (b) encircle and harass the prey, (c) attack and hunt down the prey

The pack is led by its most dominant member, male or female. Their responsibility is to make decisions for the pack, select sleeping places, and lead the hunt. The second level in the hierarchy is the beta dingo, which is the mediator between the alpha and the rest of the pack. Also, if the alpha dies, the beta takes over his place. Other pack members are the subordinates, who follow alphas and betas. Dingoes have highly developed communication, which exchanges information about the pack, a form of greeting, contesting dominance, etc. Eight sound classes with 19 sound types have been identified, most of which are barking, howling, and other sounds. Dingoes are said to be intelligent predators. This is reflected in the behavior that dingoes try to find the weak spot of the animal they're hunting and exploit it. Usually, they prey on young, vulnerable, and wounded animals. They feed on small prey, such as fish, insects, crabs, frogs, rabbits, and large prey, such as kangaroos, cattle, water buffalo, and feral horses. Hunting strategy is characterized in their phases as follows, as shown in Figure 1:

- Chasing and approaching
- Encircling and harassing
- Attack

Two main parts of this algorithm are exploration and exploitation. Exploration (represented in Figure 1 (a), also called the encircling phase) serves to explore the problem space as broadly as possible. In contrast, exploitation (represented in Figure 1 (b), also called the attack phase) serves to converge to a current best solution in the final iterations of the algorithm. The best search agent is the goal or aim prey, while the other search agents change their strategies to get closer to prey whilst exploring the search space to their fullest. In the encircling phase, dingoes move according to equations (1)-(5):

$$\overrightarrow{D_d} = \left|\vec{A}\cdot\vec{P}_p(x) - \vec{P}(i)\right| \tag{1}$$

$$\vec{P}(i+1) = \vec{P}_p(i) - \vec{B}\cdot\vec{D}(d) \tag{2}$$

$$\vec{A} = 2\cdot\overrightarrow{a_1} \tag{3}$$

$$\vec{B} = 2\vec{b}\cdot\overrightarrow{a_2} - \vec{b} \tag{4}$$

$$\vec{b} = 3 - \left(I * \left(\frac{3}{I_{\max}}\right)\right) \tag{5}$$

Here, vectors a1 and a2 represent vectors of random variables in [0,1]. In these formulae, vector D represents the distance vector, while the vector P represents the position vector. The subscript d represents the dingos, while the subscript p represents the prey. Here, the prey refers to the best search agent, while the dingos refer to all other search agents. Vectors $\vec{A}\ \vec{B}$ determine the portion of the solution space around the prey to which other dingoes converge. $\vec{B}$ It is used to determine whether the prey is moving away from the search agent or the prey is chased by the pack, indicated by values less than -1 in the first case and greater than 1 in the second case.
In the hunting phase, a common supposition in the case of this class of biologically inspired algorithms is that the members of the pack (other search agents) have a good intuition about where the prey (the best search agent) is located. In this phase, the alpha and beta are the two best solutions in the dingo pack, shown in the formulae by their subscripts, while other dingoes follow them, updating their positions accordingly. The movement equations are (6)-(11), given below:

$$\vec{D}_\alpha = \left|\vec{A}_1\cdot\vec{P}_\alpha - \vec{P}\right| \tag{6}$$

$$\vec{D}_\beta = \left|\vec{A}_2\cdot\vec{P}_\beta - \vec{P}\right| \tag{7}$$

$$\vec{D}_o = \left|\vec{A}_3\cdot\vec{P}_o - \vec{P}\right| \tag{8}$$

$$\vec{P}_1 = \left|\vec{P}_\alpha - \vec{B}\cdot\vec{D}_\alpha\right| \tag{9}$$

$$\vec{P}_2 = \left|\vec{P}_\beta - \vec{B}\cdot\vec{D}_\beta\right| \tag{10}$$

$$\vec{P}_3 = \left|\vec{P}_o - \vec{B}\cdot\vec{D}_0\right| \tag{11}$$

The intensities of each dingo are calculated as follows:

$$\vec{I}_\alpha = \log\left(\frac{1}{F_\alpha - (1E-100)} + 1\right) \tag{12}$$

$$\vec{I}_\beta = \log\left(\frac{1}{F_\beta - (1E-100)} + 1\right) \tag{13}$$

$$\vec{I}_o = \log\left(\frac{1}{F_o - (1E-100)} + 1\right) \tag{14}$$

If the position is not updated, the hunt is over, and dingoes proceed to attack the prey. The value $\vec{b}$ is linearly decreased with a number of iterations. The values $\overrightarrow{D_\alpha}$ are in the range [-3b, 3b]. Therefore, as ite–rations pass, this range is shortened, and the dingoes slowly come to a stop.

The pseudo-code for this algorithm is given below:

Input: The population of dingoes D_n (i = 1, 2, …, n)

Output: The best dingo. (This being the case of minimization problem)

(1) Generate initial search agents D_{in}
(2) Initialize the value of $\vec{b}$, $\vec{A}$, and $\vec{B}$
(3) While Termination conditions not reached do
(4) Evaluate each dingo's fitness and intensity cost.
(5) D_α = dingo with the best search
(6) D_β = dingo with the second-best search
(7) D_o = dingoes search results afterward
(8) Iteration1
(9) Repeat
(10) For i=1: D_n do
(11) Renew the latest search agent status
(12) End for
(13) Estimate the fitness and intensity cost of dingoes
(14) Record the value of F_α, F_β and F_o
(15) Record the value of $\vec{b}$, $\vec{A}$, and $\vec{B}$
(16) Iteration = Iteration + 1
(17) Check if Iteration >= Stopping criteria
(18) Output

End while

3. PROBLEM FORMULATION

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. In each example, the fitness function is denoted by f(x), while the i-th constraint is represented by $g_i(x)$. Every step of this process was done using the MATLAB R2020a software suite.

3.1 Example 1

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

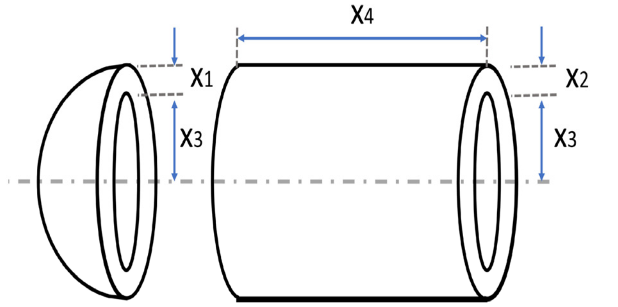

Figure 2. Pressure vessel design problem

The mathematical formulation constraints of this problem are described in Eqs. (15) to (21):

$$f(x) = 0.622 x_1 x_3 x_4 + 1.7781 x_2 x_3^2 + 3.1661 x_1^2 x_4 + 19.84 x_1^2 x_3 \quad (15)$$

$$g_1(x) = -x_1 + 0.0193 x_3 \le 0 \quad (16)$$

$$g_2(x) = -x_2 + 0.00594 x_3 \le 0 \quad (17)$$

$$g_3(x) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3 + 1296000 \le 0 \quad (18)$$

$$g_4(x) = x_4 - 240 \le 0 \quad (19)$$

The considered variable ranges are described in Eq. (20) to (21):

$$0 \le x_i \le 100; \; i = 1, 2; \quad (20)$$

$$10 \le x_i \le 200; \; i = 3, 4; \quad (21)$$

3.2 Example 2

A cantilever beam (Figure 3) 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. As seen in Figure 3, the beam consists of five hollow box-shaped bearings and square-shaped frames. The lengths of the five bearings are this problem's variables. Due to the fact that the beams' basis is a square for each beam, the values for h_i and b_i are the same and are the five variables used in this model.

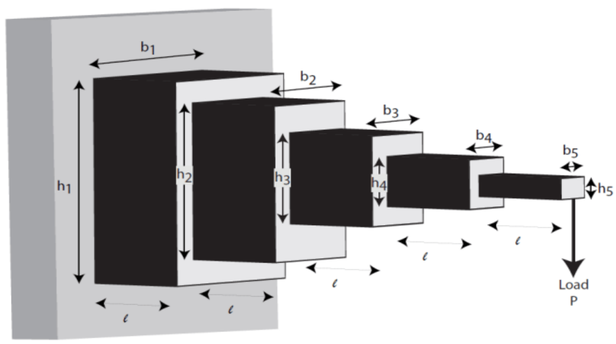

Figure 3. Stepped cantilever beam design problem

The mathematical formulation constraints of this problem are described in Eqs. (22) to (24) :

$$f(x) = 0,6224(x_1 + x_2 + x_3 + x_4 + x_5) \quad (22)$$

$$g(x) = \frac{61}{x_1^3} + \frac{27}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \le 0 \quad (23)$$

The considered variable ranges are described in Eq (24).

$$0.01 \le x_1, x_2, x_3, x_4, x_5 \le 100 \quad (24)$$

3.3 Example 3

The automobile (Figure 4) is exposed to a side impact, taking into consideration the procedures of the EEVC, which is short for the European Enhanced Vehicle-Safety Committee. The aim is to minimize the car's total weight using eleven mixed variables.

Eleven variables should be optimized: the thickness of the B-Pillar inner (x_1), the thickness of the B-Pillar reinforcement (x_2), the thickness of the floor side inner (x_3), the thickness of the cross members (x_4), the thickness of the door beam (x_5), the thickness of the door beltline reinforcement (x_6), the thickness of the roof rail (x_7), the thickness of the materials of B-pillar inner (x_8), the thickness of the materials of floor side inner (x_9), barrier height (x_{10}) and hitting position (x_{11}).

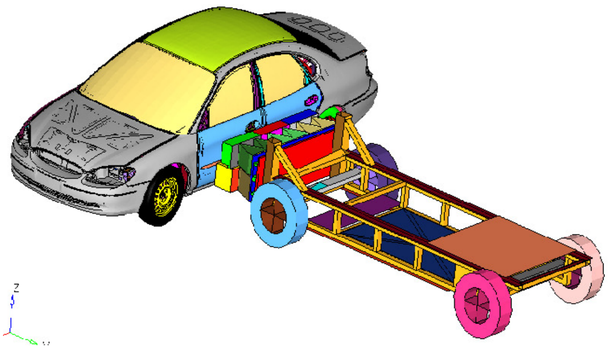

Figure 4. Car model for side impact design problem

The mathematical formulation constraints of this problem are described in Eqs.(25) to (38) :

$$f(x) = 1.98 + 4.90x_1 + 6.67x_2 + 6.98x_3 + 4.01x_1 + 1.78x_5 + 2.73x_7 \quad (25)$$

subject to

$$g_1(x) = F_a \le 1(kN) \quad (26)$$

$$g_2(x) := VC_u \le 0.32(m/s) \quad (27)$$

$$g_3(x) := VC_m \le 0.32(m/s) \quad (28)$$

$$g_4(x) := VC_1 \le 0.32(m/s) \quad (29)$$

$$g_5(x) := \Delta_{ur} \le 32(mm) \quad (30)$$

$$g_6(x) := \Delta_{mr} \le 32(mm) \quad (31)$$

$$g_7(x) := \Delta_{lr} \le 32(mm) \quad (32)$$

$$g_8(x) := F_p \le 4(kN) \quad (33)$$

$$g_9(x) := V_{MBP} \le 9.9(mm/ms) \quad (34)$$

$$g_{10}(x) := V_{FD} \le 15.7(mm/ms) \quad (35)$$

$$0.5 \le x_1, x_2, x_3, x_4, x_5, x_6, x_7 \le 1.5 \quad (36)$$

$$-30 \le x_{10}, x_{11} \le 30 \quad (37)$$

$$x_8, x_9 \in \{0.192, 0.345\} \quad (38)$$

Variables $F_a, VC_u, VC_m, VC_1, \Delta_{ur}, \Delta_{mr}, \Delta_{lr}, F_p, V_{MBP}, V_{FD}$ are mathematically described in the paper [29].

3.4 Example 4

The cone clutch (Figure 5) problem must be designed to minimize the volume coupling and be subjected to two constraints. Problem variables in the case of this prob–lem are the inner radius x_1 and the outer radius x_2 of the coupling.

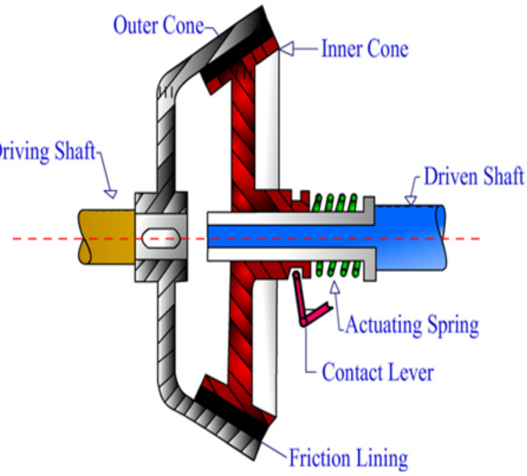

Figure 5. Cone clutch design problem

The goal function in the case of this problem is:

$$f(x) = \left(x_1^3 - x_2^3\right) \quad (39)$$

while the conditions to be met are:

$$g_1(x) = \frac{x_1}{x_2} \ge 2 \quad (40)$$

$$g_2(x) = \frac{\left(x_1^2 + x_1x_2 + x_2^2\right)}{\left(x_1 + x_2\right)} \ge 5 \quad (41)$$

$$1 \le x_1, x_2 \le 10 \quad (42)$$

4. RESULTS AND DISCUSSION

This section gives the results obtained by using DOA on a set of selected engineering problems.

In the case of the pressure vessel problem, what has been 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 pressure vessel

Variables	GWO[30]	GOA[31]	WCA[32]	**DOA**
x_1	0.8220	0.8736	0.7781	**0.7781**
x_2	0.4060	0.4318	0.3846	**0.3846**
x_3	42.6020	45.2666	40.3196	**40.3196**
x_4	170.4840	199.9998	200.0000	**200.0000**
$f(x)$	5964.5000	7666.1258	5888.3327	**5885.3313**

As the results show, DOA has shown better results than current literature.

In the case of the cantilever beam design problem, the results are presented in Table 2. The results from the literature, where the ALO, MMA, and GOA methods are used for this problem, are to be found in the same table.

Table 2. Comparison of results for the cantilever beam

Variables	ALO[33]	GOA[31]	MMA[34]	**DOA**
x_1	6.0180	6.0110	6.0100	**6.0174**
x_2	5.3110	5.3120	5.3000	**4.9686**
x_3	4.4880	4.4830	4.4900	**4.4797**
x_4	3.4970	3.5020	3.4900	**3.4002**
x_5	2.1580	2.1630	2.1500	**2.0840**
f(x)	1.3390	1.3390	1.3400	**1.3030**

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

For the problem of a car side impact, the results are displayed in Table 3, along with results obtained by MFO, GOA, and WOA methods.

Table 3. Comparison of results for the car side impact

Variables	MFO[35]	GOA[31]	WOA[3]	**DOA**
x_1	0.5000	0.5000	0.5000	**0.5000**
x_2	1.1160	1.1150	1.1080	**1.1130**
x_3	0.5000	0.5000	0.5340	**0.5000**
x_4	1.3010	1.3030	1.3050	**1.3070**
x_5	0.5000	0.5000	0.5000	**0.5000**
x_6	1.5000	1.5000	1.4730	**1.5000**
x_7	0.5000	0.5000	0.5000	**0.5000**
x_8	0.3450	0.3450	0.3450	**0.3440**
x_9	0.3450	0.2860	0.1920	**0.2850**
x_{10}	-19.5300	-19.7150	-19.6990	**-20.1530**
x_{11}	0.0000	0.3200	3.4810	**0.0510**
f(x)	22.8420	22.8434	23.0420	**22.8430**

In the case of car side impact optimization, DOA gives the same result as MFO and GOA, while WOA gives slightly worse results.

The DOA, GWO, GOA, and WCA algorithms were found in the literature for the case of the cone clutch problem. The result comparison is given in Table 4.

Table 4. Comparison of results for the cone clutch

Variables	GWO[30]	CS[36]	GOA [31]	**DOA**
x_1	4.2860	4.2850	4.2860	**4.2857**
x_2	2.1420	2.1420	2.1430	**2.1429**
f(x)	68.8930	68.8870	68.8940	**68.8776**

As can be seen from Table 4, the DOA gives better results than all three algorithms.

As can be seen from comparing the results to the solutions in the literature, the DOA has shown to obtain optimal or near-optimal solutions in the case of given engineering problems. This indicates that the DOA has the potential for solving this class of optimization problems.

5. CONCLUSION

This paper describes the DOA algorithm and applies it to a selected set of engineering problems. This set comprises pressure vessel, cantilever beam, car side-impact, and cone clutch design problems, which are described in detail, and highlighted by figures, goal function, and constraints descriptions.

The chosen input parameters are 100 search agents and 500 iterations of the algorithm. This is because, as was discovered during the research, increasing the values of these input parameters did not yield better solutions.

In the case of pressure vessel optimization, the DOA gives better results than the methods it was compared to. In the case of the other three optimization problems, namely: cantilever beam, cone clutch, and car side impact problem, the DOA yielded near-optimal solutions. The main downside to this algorithm, as is with all p-based metaheuristic algorithms, is that it is computationally intensive. Therefore, the DOA should be tested on problems with more dimensions. Since the results of this paper have been compared to the results found in the literature, the DOA yielded optimal or near-optimal results. This means that the algorithm should be tested to solve other mechanical design problems, which could represent the basis for future work. The results are shown in this paper, considering that with this amount of search agents and iteration count, near-optimal or better results, compared to the current literature, were obtained, paving the way for testing this algorithm further in the area of mechanical design. The main downside to this algorithm, as is with all p-based metaheuristic algorithms, is that it is computationally intensive, which should hinder problems with a large number of dimensions. The algorithm's parameters should be tuned for the problems where the optimal solution was not found with the help of metaheuristic algorithms. Some of the problems could potentially have better solutions outside the defined lower and upper bounds, leading to further research. Because metaheuristic algorithms use stochastics to move through the solution space, different random number generators and varying seed values should be tested to find better solutions for optimization problems.

ACKNOWLEDGMENT

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

REFERENCES

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

[2] 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.

[3] Boško P. Rašuo, Aleksandar Č. Bengin, Optimization of Wind Farm Layout, FME Transactions (2010) 38, 107-114

[4] Som Nath Pandey, Jogendra Kumar, Sasmit Yadav, Rajesh Kumar Verma, A Combined Approach of Nature-Inspired Firefly Algorithm and Weighted Principal Component Analysis in Machining of In–conel X-750, FME Transactions (2020) 48, 439-446

[5] Mário Leite, Telmo Pinto, Cláudio Alves, A Real-Time Optimization Algorithm for the Integrated Planning and Scheduling Problem Towards the Context of Industry 4.0, FME Transactions (2019) 47, 775-781

[6] Filipe Alves et al. A Human Centred Hybrid MAS and Meta-Heuristics Based System for Simultaneously Supporting Scheduling and Plant Layout Adjustment, FME Transactions (2019) 47, 699-710

[7] R.K. Ghadai, K. Kalita, X-Z. Gao, Symbolic Regression Metamodel Based Multi-Response Optimization of EDM Process, FME Transactions (2020) 48, 404-410

[8] Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. Adv. Eng. Softw. 2016, 95, 51–67.

[9] Li, Y.; He, Y.; Liu, X.; Guo, X.; Li, Z. A novel discrete whale optimization algorithm for solving knapsack problems. Appl. Intell. 2020, prepublish.

[10] Jin Zhang, Li Hong and Qing Liu, An Improved Whale Optimization Algorithm for the Traveling Salesman Problem, Symmetry, 2020.

[11] M. Dorigo, M. Birattari and T. Stutzle, "Ant colony optimization," in IEEE Computational Intelligence Magazine, vol. 1, no. 4, pp. 28-39, Nov. 2006, doi: 10.1109/MCI.2006.329691.

[12] Dorigo, M., Gambardella, L.M., Ant Colonies for the Traveling Salesman Problem, BioSystems, 43:73-81, 1997.

[13] B. Chandra Mohan, R. Baskaran, A survey: Ant Colony Optimization based recent research and implementation on several engineering domain, Expert Systems with Applications, 2012.

[14] Yang, X. and Hossein Gandomi, A. (2012), "Bat algorithm: a novel approach for global engineering optimization", Engineering Computations, Vol. 29 No.5, pp.464-483.

[15] T. Jayabarathi, T. Raghunathan, A. H. Gandomi, The Bat Algorithm, Variants and Some Practical Engineering Applications: A Review, Nature-Inspired Algorithms and Applied Optimization pp 313-330, Springer, 2017.

[16] T. Yuvaraj, K. R. DevabalajiK. Ravi, Optimal Allocation of DG in the Radial Distribution Network Using Bat Optimization Algorithm, Ad–vances in Power Systems and Energy Management pp 563-569, Springer, 2017.

[17] Davis, L., *Handbook of genetic algorithms*, Van Nostrand Reinhold, New York, 1991.

[18] W.M. Jenkins, Towards structural optimization via the genetic algorithm, Computers & Structures, Volume 40, Issue 5, 1991.

[19] Markku Ohenoja, Kauko Leiviskä, Validation of genetic algorithm results in a fuel cell model, International Journal of Hydrogen Energy, Volume 35, Issue 22, 2010.

[20] Seyedali Mirjalili, SCA: A Sine Cosine Algorithm for solving optimization problems, Knowledge-Based Systems, Volume 96, 2016.

[21] Abualigah, L., Diabat, A. Advances in Sine Cosine Algorithm: A comprehensive survey. Artif Intell Rev 54, 2567–2608 (2021).

[22] Abdel-Fattah Attia, Ragab A. El Sehiemy, Hany M. Hasanien, Optimal power flow solution in power systems using a novel Sine-Cosine algorithm, International Journal of Electrical Power & Energy Systems, Volume 99, 2018.

[23] Press, WH; Teukolsky, SA; Vetterling, WT; Flannery, BP (2007). "Section 10.12. Simulated Annealing Methods". Numerical Recipes: The Art of Scientific Computing (3rd ed.). New York: Cambridge University Press. ISBN 978-0-521-88068-8.

[24] Nazmul Siddique and Hojjat Adeli, Simulated Annealing, Its Variants and Engineering Appli–cations, International Journal on Artificial Intelli–gence Tools, Vol. 25, No. 06, 1630001 (2016).

[25] Shun-Fa Hwang, Rong-Song He, Improving real-parameter genetic algorithm with simulated annealing for engineering problems, Advances in Engineering Software, Volume 37, Issue 6, 2006.

[26] Seyedali Mirjalili, Seyed Mohammad Mirjalili, Andrew Lewis, Grey Wolf Optimizer, Advances in Engineering Software, Volume 69, 2014.

[27] N. Jayakumar, S. Subramanian, S. Ganesan, E.B. Elanchezhian, Grey wolf optimization for combined heat and power dispatch with cogeneration systems, International Journal of Electrical Power & Energy Systems, Volume 74, 2016.

[28] Wen Long, Ximing Liang, Shaohong Cai, Jianjun Jiao & Wenzhuan Zhang, A modified augmented Lagrangian with improved grey wolf optimization to constrained optimization problems, Neural Computing and Applications volume 28, pages421–438 (2017).

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

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

[31] Gu L, Yang R. J, Cho C. H, Cho C. H, Makowski M, Faruque M, Li Y, Optimization and robustness for crashworthiness, International Journal Vehile Des 26(4):348-60, 2001.

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

[33] https://en.wikipedia.org/wiki/Dingo

[34] T.Kim,M.Cho,S.Shin,"Constrained Mixed Variable Design Optimization Based on Particle Swarm Optimizer with a Diversity Classifier for Cyclically

Neighboring Subpopulations” Mathematics 2020, 8, 2016,pp.1-29.

[35] B.Milenković, M.Krstić, Đ.Jovanović: Primena al–goritma sivog vuka za rešavanje inženjerskih optimizacionih problema, Tehnika, 2021.,Vol.76, Br.1. str.50-57, ISSN 0040-2176.

[36] Saremi, S., Mirjalili, S., & Lewis, A. (2017). Gras–shopper optimisation algorithm: theory and appli–cation. /Advances in Engineering Software/, /105/, 30-47.

[37] H.Eskandar, et al.: Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems, Computers and Structures,2012.

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

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

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

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

NOMENCLATURE

DOA	Dingo Optimization Algorithm
WOA	Whale Optimization Algorithm
CSA	Cuckoo Search Algorithm
GA	Genetic Algorithm
HHO	Harris Hawks Optimizer
ACO	Ant Colony Optimization
DE	Differential Evolution
LOA	Lion Optimization Algorithm
GOA	Grasshopper Optimization Algorithm
SCA	Sine Cosine Algorithm
BA	Bat Algorithm
PSO	Particle Swarm Optimization
GWO	Grey Wolf Optimizer
SA	Simulated Annealing
VNS	Variable Neighborhood Search
$\vec{A}$	Coefficient vector
$\vec{B}$	Coefficient vector
$\vec{b}$	Linearly decrement from 3 to 0 at every iteration
$\overrightarrow{a_1}$	Random vector in [0,1]
$\overrightarrow{a_2}$	Random vector in [0,1]
$\overrightarrow{D_d}$	Distance between the dingo and prey
$\overrightarrow{P_p}$	Position vector (prey)
$\vec{P}$	Position vector (dingo)
I	Number of iteration
$I_{\max}$	Maximum number of iteration
F_α	Fitness value of alpha dingo
F_β	Fitness value of beta dingo
F_o	Fitness value of other dingoes
$\overrightarrow{I_\alpha}$	Intensity of alpha dingo
$\overrightarrow{I_\beta}$	Intensity of beta dingo
$\overrightarrow{I_o}$	Intensity of other dingoes
D_α	dingo with the best search
D_β	dingo with the second best search
D_o	all dingoes except the best and the second best
f(x)	objective function
$\mathbf{g_i(x)}$	constrains
R_1	inner radius of the coupling
R_2	outer radius of the coupling

ПРИМЕНА ДИНГО ОПТИМИЗАЦИОНОГ АЛГОРИТМА (ДОА) НА РЕШАВАЊЕ КОНТИНУАЛНИХ ПРОБЛЕМА У ИНЖЕЊЕРСТВУ

Б. Миленковић, Ђ. Јовановић, М. Крстић

У скорашњим истраживањима, проблеми у инжењерству се све више истичу.Једна класа проблема у инжењерству се бави проблемима дизајна, код којих се скуп варијабли калибрише, како би оптимизациона функција имала минимум или максимум. Ова функција често узима у обзир енергетску ефикасност, трошкове израде, произ–водње. Један од начина за решавање ових проблема јесте коришћење метахеуристика. У овом раду се користи Динго оптимизациони алгоритам (ДОА) за решавање проблема у машинском инжењерству. Прво је дат кратак приказ ДОА, заједно са био–лошком инспирацијом за њега, као и скуп најваж–нијих формула. Псеудокод овог алгоритма је написан у „MATLAB R2020a“ софтверском алату. Динго оптимизациони алгоритам (ДОА) је употреб–љен за оптимизацију проблема у инжењерству, попут: суда под притиском, конзолне греде, бочног судара и конусног квачила. Резултати овог рада приказују да ДОА може дати релевантне резултате у области проблема пројектовања у инжењерству.