The Seybold REPORT
ISSN 1533-9211
DOI: 10.5281/zenodo.10807204

A MODIFIED HARRIS HAWKS OPTIMIZATION ALGORITHM FOR SOLVING CONSTRAINED ENGINEERING OPTIMIZATION PROBLEMS

RABIE MOSAAD RABIE [1]*, HEGAZI ZAHER [2], NAGLAA RAGAA SAIED [3] and HEBA SAYED [4]

[1,3,4] Department of Operations Research, Faculty of Graduate Studies for Statistical Research, Cairo University, Cairo, Egypt.
[2] Department of Statistics, Faculty of Graduate Studies for Statistical Research, Cairo University, Cairo, Egypt.
Email: [1]12422020556647@pg.cu.edu.eg, [2]hgsabry@cu.edu.eg, [3]naglaa777subkiii@yahoo.com, [4]hmhmdss@yahoo.com

Abstract

Metaheuristics optimization has gained popularity in recent years for its effectiveness in solving real-world problems, such as engineering design. These techniques are especially helpful in solving nonlinear, non-convex, non-differentiable, high-dimensional, NP-hard, and discrete search space problems that are difficult to solve with traditional optimization techniques. In this study, a modified Harris Hawks Optimization (MHHO) algorithm is proposed using a mutation-selection strategy and crossover operator to global optimization problems. It can control the balance between exploration and exploitation in the search process. This flexibility allows the algorithm to adapt to different optimization problems and search landscapes, potentially improving its performance in finding optimal or near-optimal solutions. The proposed method has been tested on a variety of constrained structural engineering design problems and compared with well-known metaheuristic algorithms. The results from systematic experiments demonstrated that the MHHO algorithm provided more reliable solutions than other well-known algorithms. Furthermore, the experimental findings show that MHHO outperformed other metaheuristic algorithms in terms of optimization performance.

Keywords: Optimization, Meta-heuristics, Structural Engineering Design, Harris Hawks Algorithm, Global Optimization Problems.

1. INTRODUCTION

Optimization is the process of identifying the best solution among all possible options to maximize or minimize the output. With the rise in problem complexity over recent years, there has been a need for new optimization techniques that can handle these challenges effectively. Optimization problem-solving methods can be categorized into two groups: deterministic and random approaches.

Deterministic methods are effective for linear, continuous, differentiable, and convex optimization problems. However, they struggle with nonlinear, non-convex, non-differentiable, high-dimensional, NP-hard problems and discrete search spaces - all common features of real-world optimization problems.

Stochastic algorithms, particularly metaheuristic algorithms, have been developed to address these challenges (E. Houssein et al., 2021; Sergeyev et al., 2018).

DOI: 10.5281/zenodo.10807204

Metaheuristic algorithms use random search in the problem space and rely on random operators to provide suitable solutions to optimization problems. However, there is no guarantee that the solution obtained from these methods will be the best or global optimal.

This has led researchers to develop numerous metaheuristic algorithms to improve solutions. Over recent decades, various types of methods have been developed to solve constrained engineering problems.

Two prominent categories of these methods are mathematical and metaheuristic methods (Rao, 2009). Mathematical methods use the gradient of the objective function and constraints of the problem to find the optimal solution.

However, these methods are sensitive to the initial starting point and may not be suitable for complex optimization problems or cases where gradients cannot be calculated easily.

Metaheuristic algorithms follow a general process as shown in Figure 1. The algorithm steps represent the unique operators of each algorithm, which generate new solutions to optimization problems. These operators refer to the optimal process of a particular phenomenon that these algorithms have imitated.

According to the type of basic phenomena, metaheuristic algorithms can be classified into four main categories:

(1) Evolutionary,

(2) Swarm intelligence,

(3) physics-based, and

(4) human-based algorithms.

Evolutionary algorithms are motivated by natural evolution. Swarm intelligence algorithms model the natural behavior of animals in teamwork such as foraging and hunting. Physical phenomena and laws of science inspire physics-based algorithms. Finally, human-based algorithms mimic various optimal behaviors of humans in different condition (Trojovský & Dehghani, 2023). Some popular and novel metaheuristic algorithms are presented in Table 1

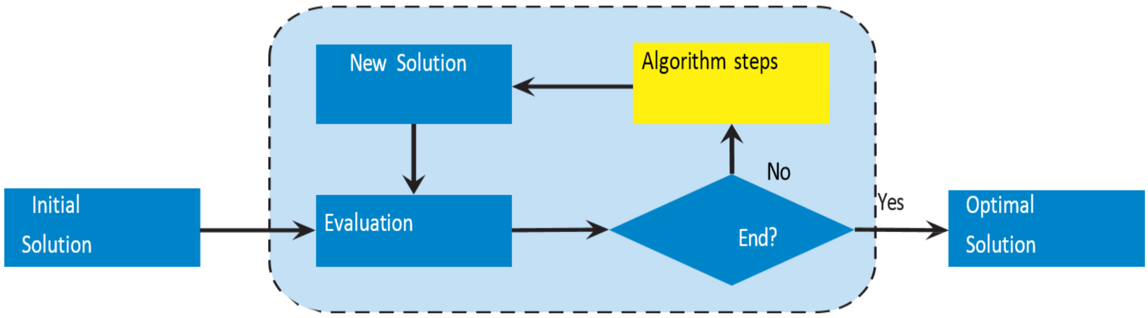

Figure 1: The General Structure of Optimization Algorithms

ISSN 1533-9211

DOI: 10.5281/zenodo.10807204

Table 1: List of Some Popular and New Metaheuristic Algorithms

Algorithm	Ref	Algorithm	Ref
Genetic Algorithms (GA)	(Holland, 1992)	Multi-Verse Optimizer (MVO)	(Mirjalili et al., 2016)
Particle Swarm Optimization (PSO)	(Kennedy & Eberhart, 1995)	(Harris Hawks Optimization) HHO	(Heidari et al., 2019)
Antlion Optimizer (ALO)	(Mirjalili, 2015)	Wild Horse Optimizer (WHO)	(Naruei & Keynia, 2022)
Aquila Optimizer (AO)	(Abualigah et al., 2021)	Dynamic Cat Swarm Optimization Algorithm (DCSO)	(Ahmed et al., 2021)
Grey Wolf Optimizer (GWO)	(Mirjalili et al., 2014)	Whale Optimization Algorithm (WOA)	(Mirjalili & Lewis, 2016)
Dingo Optimization Algorithm (DOA)	(Peraza-Vázquez et al., 2021)	War Strategy Optimization (WSO)	(Braik et al., 2022)

Harris Hawks optimization (HHO) is a new metaheuristic optimization algorithm inspired by the cooperative behavior and foraging patterns of Harris Hawks. HHO exhibits simplicity of implementation, a high level of exploration and exploitation, and requires a small number of controlling parameters (Heidari et al., 2019). In this paper, we propose a modified version of the Harris Hawks optimization algorithm (MHHO) using a mutation-selection approach.

We evaluate the MHHO by applying it to four engineering design problems, highlighting its effectiveness in real-world applications. Comparative experiments involving basic HHO and several well-known metaheuristic algorithms demonstrate the superior performance of the proposed MHHO algorithm.

2. HARRIS HAWKS OPTIMIZATION (HHO)

Harris Hawks Optimization (HHO) is motivated by the remarkable cooperative foraging behavior observed in Harris' hawks. These hawks exhibit a diverse range of chasing patterns in response to the dynamic environment and the evasive strategies employed by their prey. These agile switching activities effectively confuse the prey, while the cooperative strategies employed by the hawks assist in pursuing and eventually exhausting the detected prey, rendering it more vulnerable.

The HHO algorithm specifically emulates the process of hunting prey observed in hawks. It encompasses two distinct phases: the exploration phase and the exploitation phase. Each phase mimics different behaviors exhibited by hawks during the predation process. As with other swarm intelligence algorithms. Figure 2 provides a detailed illustration of the exploratory and exploitative phases of HHO (Heidari et al., 2019).

ISSN 1533-9211
DOI: 10.5281/zenodo.10807204

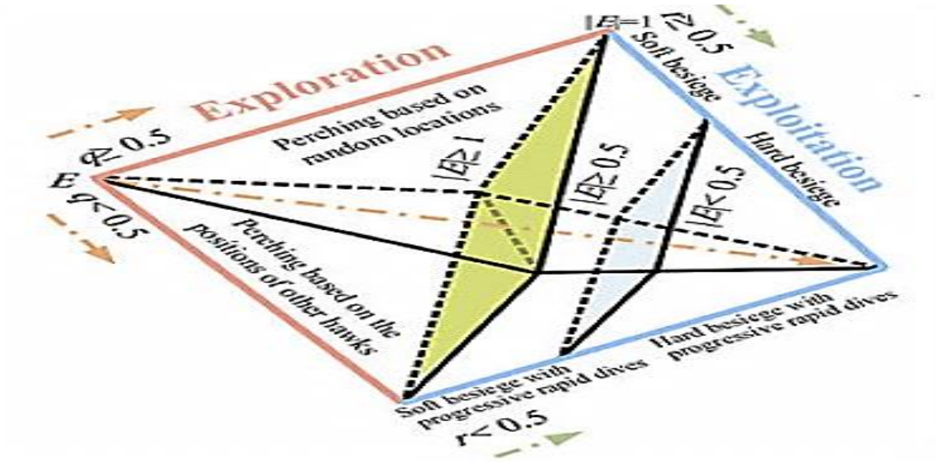

Figure 2: The Phases of Harris Hawks Algorithm

2.1 Exploration Phase

Harris's hawks typically perch on random locations and monitor the desert to spot prey.

Two perching strategies are used, based on the positions of other family members and the prey, selected randomly according to q value. q Is an equal chance for each strategy during exploration. If$q < 0.5$, Harris's hawks perch near family members and prey.

Otherwise, they perch on random tall trees (within their home range). The updated position during exploration is modeled using equation (1):

$$X_i(t+1) = \begin{cases} X_{rand}(t) - r_1 |X_{rand}(t) - 2r_2 X_i(t)|, & q \geq 0.5 \\ X_{best}(t) - X_M(t) - r_3(LB + r_4(UB - LB)), & q < 0.5 \end{cases} \quad (1)$$

$X_i(t)$ and $X_{best}(t)$ represent the position of the i^{th} hawk and rabbit at the t^{th} iteration, respectively. q is a random number between 0 and 1, which denotes an equal chance for each perching strategy in the exploration phase.

If $q < 0.5$, Harris's hawks perch based on the positions of other family members (to be close enough to them when attacking) and the rabbit.

Otherwise, they perch on random tall trees (random locations within their home range).

The updated position during exploration is modeled using equation (1), with, $r_1; r_2; r_3; r_4$ as random numbers inside (0; 1), which are updated in each iteration, with r_3 being a scaling coefficient to further increase randomness when r_4 takes values close to 1 and similar distribution patterns may occur. LB and UB Denote the lower and upper bounds of variables. $X_{rand}(t)$ Refers to a randomly selected hawk from the current population, and

$$X_M(t) = \frac{1}{N}\sum_1^N X_i(t) \quad (2)$$

This represents the typical position of the present population of hawks, N being the total number of hawks, and each location is within the group's home range (LB and UB).

The Seybold Report
ISSN 1533-9211
DOI: 10.5281/zenodo.10807204

2.2 Transition from Exploration to Exploitation Phase

The Harris Hawks Optimization (HHO) algorithm incorporates a transition mechanism that switches from an exploration phase to an exploitation phase, depending on the prey's escaping energy. In this algorithm, the prey's energy is represented as gradually decreasing during its escape behavior.

$$E = 2E_0(1 - \frac{t}{T}) \quad (3)$$

The prey's escaping energy, denoted as E and initialized asE_0, determines the exploration or exploitation phase in the HHO algorithm. If$|E| \geq 1$, the algorithm is in the exploration phase, while $|E| < 1$ indicates the exploitation phase.

2.3 Exploitation Phase

During the exploitation phase, the HHO algorithm employs four distinct chasing and attack strategies based on the prey's escaping energy and the hawks' chasing behavior. The parameter r is used to select a chasing strategy depending on whether the prey successfully escapes $(r < 0.5)$ *or not* $(r \geq 0.5)$ before an attack.

i. Soft Besiege

When the probability of escape $(r \geq 0.5)$ and the escaping energy $(|E| \geq 0.5)$, the prey still possesses sufficient energy and attempts to escape. In response, the Harris' hawks softly surround the prey to deplete its remaining energy before launching an attack. The behavior of the Harris' hawks in this phase is modeled as follows:

$$X(t+1) = \Delta X(t) - E|JX_{best}(t) - X(t)| \quad (4)$$

$$\Delta X(t) = X_{best}(t) - X(t) \quad (5)$$

$$J = 2(1 - r_5) \quad (6)$$

$\Delta X(t)$ Is used to indicate the difference between the prey's present position and the previous position, while J is used to describe the prey's ability to jump randomly.

ii. Hard Besiege

When the probability of escape $(r \geq 0.5)$ and the escaping energy $(|E| < 0.5)$, the prey's energy is low, and the Harris' hawks readily encircle it before launching an attack. The positions of the prey and the hawks are updated using the following equations:

$$X(t+1) = X_{best}(t) - E|\Delta X(t)| \quad (7)$$

iii. Soft Besiege with Progressive Rapid Dives

When the prey has enough energy to successfully escape $(|E| \geq 0.5)$ and$(r < 0.5)$, the Harris' hawks perform a soft besiege with several rapid dives around the prey to progressively correct its position and direction. This behavior is modeled using the following equations:

$$Y = X_{best}(t) - E|JX_{best}(t) - X(t)| \quad (8)$$

The Seybold REPORT
ISSN 1533-9211
DOI: 10.5281/zenodo.10807204

$$Z = Y + S \times LF(D) \tag{9}$$

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

Where S is a random vector. The next position will be chosen based on the best position between Y and Z.

iv. Hard Besiege with Progressive Rapid Dives

When (|E| < 0.5) and (r < 0.5), indicating that the prey has insufficient energy to escape, the hawks perform a hard besiege by decreasing the distance between their average position and the prey. This behavior is modeled using the following equations:

$$Y = X_{best}(t) - E|JX\_best\ (t) - X_M(t)| \tag{11}$$

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

$$X(t+1) = \begin{cases} Y & if\ F(Y) < F(X(t)) \\ Z & if\ F(Z) < F(X(t)) \end{cases} \tag{13}$$

The new iteration will be chosen based on the best position between Y and Z (Wang et al., 2021).

3. THE PROPOSED HARRIS HAWKS OPTIMIZATION (MHHO)

In this study, a modified version of the Harris Hawks Optimizer (MHHO) is proposed. To incorporate mutation and crossover operator into the algorithm. a mutation operator is introduced that randomly modifies the position of some hawks in the population at each iteration. This can be done by randomly selecting a subset of hawks and adding a small random perturbation to their positions. Let X_a , X_b and $X_c$be the best three hawks positions based on fitness function value, respectively. Then, the new mutation position vector $X(\text{mut})$ for ith hawk can be defined as:

$$X(mut) = X(t+1) + 2 * \left(1 - \frac{t}{t_{max}}\right) * (2 * rand - 1) * (X_a - X_b - X_c) + (2 * rand - 1)\big(X_a - X(t+1)\big) \tag{14}$$

Where N denotes total number of hawks and all locations are within group's home range$(LB; UB)$. The position vector for next generation $X(new)$ can be obtained through selection process described in Eq.

$$X(new) = \begin{cases} \text{X(mut)} & \big(F(X(\text{mut}) < F(X(t+1))\big) \\ X(t+1) & (\,F(X(\text{mut}) \geq F(X(t+1)) \end{cases} \tag{15}$$

The crossover operation combines the positions of two hawk positions, using crossover Probability CR, the algorithm creates a new solution that shares properties with positions $X(t+1)$ and $X(mut)$.

The Seybold REPORT
ISSN 1533-9211
DOI: 10.5281/zenodo.10807204

The equation for the crossover operation is as follows:

$$X(new) = \begin{cases} X(t+1) & rand > CR \\ \mathrm{X(mut)} & otherwise \end{cases} \quad (16)$$

The crossover function is responsible for performing the crossover operation, if the random number generated is less than the crossover rate, the crossover is applied and the resulting position is assigned to the hawk as the new solution.

The mutation rate determines the intensity of the mutation, and the crossover rate determines the probability of crossover during the optimization process. These parameters allow for greater flexibility and control over the search behavior of the algorithm.

The mutation function has been added to perform the mutation operation, and the crossover function has been added to perform the crossover operation. These functions are called within the main loop of the HHO function based on the given mutation and crossover rates. By adjusting the mutation and crossover rates, you can control the balance between exploration and exploitation in the search process. This flexibility allows the algorithm to adapt to different optimization problems and search landscapes, potentially improving its performance in finding optimal or near-optimal solutions.

Overall, these modifications enhance the algorithm's ability to explore and exploit the search space, potentially leading to improved optimization performance.

4. EXPERIMENT RESULTS AND DISCUSSION

The MHHO algorithm performance is assessed using four benchmark problems encompassing various engineering fields. Each problem is independently executed 30 times with the MHHO algorithm, and the outcomes are compared against counterpart algorithms from existing literature. For the performance evaluation, the input parameters are set as follows: $N = 50$, t_{max}= 1000, and $\beta = 1.5$. To obtain the best, worst, average optimal values, and standard deviation for each algorithm, the aforementioned algorithms are executed 30 times. The best optimal value is denoted as '*Best*,' the worst optimal value as '*Worst*,' the average optimal value as '*Average*,' and the standard deviation as '*Std*.' The best values are indicated in bold. By conducting this rigorous evaluation, we can effectively compare the performance of the MHHO algorithm with other algorithms on the selected benchmark problems.

4.1 Tension/Compression Spring Design Optimization Problem

In practical applications, the design of tension/compression springs presents a challenge in an effort to minimize their weight. A schematic of this design is displayed in Figure 3 (Arora, 2011). The problem formulation for tension/compression spring design involves finding the optimal parameters that will achieve minimum weight while maintaining stability and reliability.

ISSN 1533-9211
DOI: 10.5281/zenodo.10807204

The tension/compression spring problem is modeled using the following equations:

Consider $X = [x_1, x_2, x_3] = [d, D, P]$.

Minimize

$$f(X) = (x_3 + 2)x_2 x_1^2$$

Subject to:

$$g_1(X) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0,$$

$$g_2(X) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \leq 0,$$

$$g_3(X) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0,$$

$$g_4(X) = \frac{x_1 + x_2}{1.5} - 1 \leq 0,$$

Variable range:

$$0.05 \leq x_1 \leq 2, 0.25 \leq x_2 \leq 1.3, 2 \leq x_3 \leq 15.$$

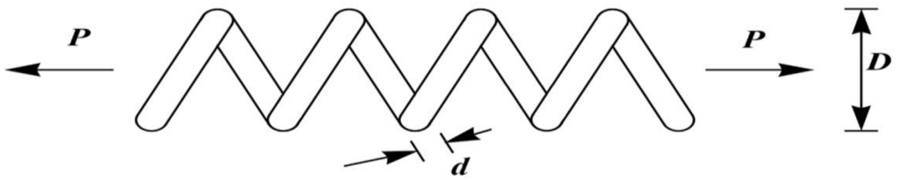

Figure 3: Tension/Compression Spring

Table 2: Comparison Results for the Tension/Compression Spring Design Problem

Algorithm	Optimum variables			Optimum cost
	d	D	P	
MHHO	0.051643	0.3556	11.3548	0.012665
HHO	0.052524	0.377148	10.1845	0.012678
ALO	0.052211	0.369398	10.5825	0.01267
AO	0.05	0.312028	15	0.013261
DOA	0.051689	0.356719	11.2889	0.012665
GWO	0.051473	0.351456	11.6085	0.012672
MVO	0.05	0.3123	14.7356	0.013066
WHO	0.051692	0.356797	11.2843	0.012665
WOA	0.052026	0.364868	10.8267	0.012667
WSO	0.052108	0.366878	10.7173	0.012668
DCSO	0.052721	0.38205	9.9449	0.012684
PSO	0.049644	0.307165	13.86793	0.013151
GA	0.049763	0.313345	15.09875	0.012887

The Seybold Report

ISSN 1533-9211

DOI: 10.5281/zenodo.10807204

Table 3: Statistical Results for the Tension/Compression Spring Design Problem

Algorithm	Best	Average	Worst	Std
MHHO	0.012665	0.013504	0.017773	0.001543
HHO	0.012678	0.013839	0.017057	0.001076
ALO	0.01267	0.013577	0.017568	0.001198
AO	0.013261	0.015992	0.020854	0.001685
DOA	0.012666	0.012806	0.014965	0.000372
GWO	0.012672	0.012714	0.012823	2.56E-05
MVO	0.013066	0.017289	0.018126	0.001234
WHO	0.012666	0.012736	0.013193	0.000112
WOA	0.012667	0.013742	0.017774	0.001103
WSO	0.012668	0.015212	0.041668	0.00503
DCSO	0.012684	0.012718	0.01273	1.01E-05
PSO	0.013151	0.014165	0.016378	0.002192
GA	0.012887	0.013178	0.015355	0.002378

The results of optimizing the design variables of the tension/compression spring using the standard HHO, MHHO and competing algorithms are presented in Table 2. The simulation results show that the MHHO algorithm obtained the best solution of this problem, with variable values of (0.0516425, 0.3556, 11.3548) and corresponding objective function values of 0.012665. Table 3 displays the statistical findings obtained from evaluating the performance of the MHHO algorithm and competing algorithms. The table clearly shows the superiority of the MHHO algorithm in providing the best values of the numerical signals, and also reveals its effectiveness in solving the tension/compression spring optimization problem. Figure 4 displays the MHHO convergence curve to demonstrate how the MHHO system converges to the solution of the tension/compression spring.

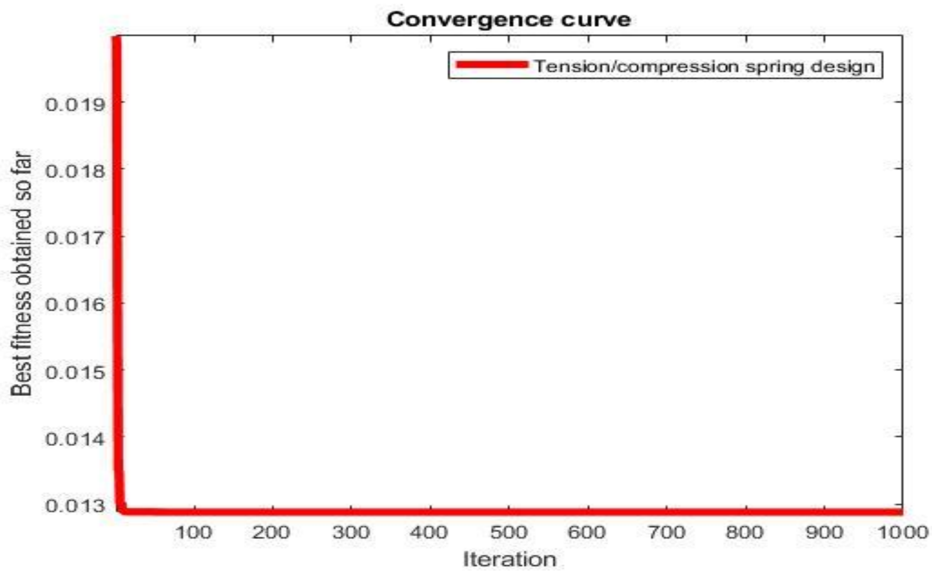

Figure 4: Convergence Curve of the MHHO for the Tension/Compression Spring Design Problem

ISSN 1533-9211
DOI: 10.5281/zenodo.10807204

4.2 Pressure Vessel Design

Pressure vessel design is a real-world challenge that seeks to optimize design costs. A schematic of this system is shown in Figure 5 (Kannan & Kramer, 1994). The pressure vessel design problem is formulated as follows:

Consider $X = [x_1, x_2, x_{3,} x_4] = [T_s, T_h, R, L]$.

Minimize

$$f(X) = 0.6224x_1x_3x_4 + 1.7881x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

Subject to:

$$g_1(X) = -x_1 + 0.0193x_3 \leq 0,$$

$$g_2(X) = -x_2 + 0.00954x_3 \leq 0,$$

$$g_3(X) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1{,}296{,}000 \leq 0,$$

$$g_4(X) = x_4 - 240 \leq 0,$$

Variable range:

$$x_1, x_2 \epsilon\{1 \times 0.0625, 2 \times 0.0625, 3 \times 0.0625, \ldots, 1600 \times 0.0625\}, 10 \leq x_3, x_4 \leq 200.$$

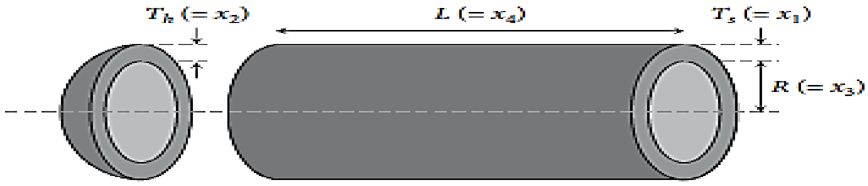

Figure 5: Schematic View Pressure Vessel Design Problem

Table 4: Comparison Results for the Pressure Vessel Design Problem

Algorithm	Optimum variables				Optimum cost
	T_s	T_h	R	L	
MHHO	0.778169	0.383036	40.31962	200	5880.671
HHO	0.811244	0.39881	41.93468	178.6766	5950.075
ALO	0.778393	0.383148	40.33125	199.8384	5881.059
AO	0.798452	0.420413	40.91055	192.3804	6068.143
DOA	0.778169	0.383036	40.31962	200	5882.681
GWO	0.865387	0.427695	44.78171	159.8514	6035.825
MVO	0.780341	0.391693	40.43147	198.4788	5907.15
WHO	0.778169	0.383036	40.31962	200	5881.245
WOA	0.790485	0.393914	40.55615	196.7333	5969.576
WSO	0.77817	0.383403	40.31965	200	5923.719
DCSO	0.778169	0.383576	40.31962	200	5882.232
PSO	0.768881	0.408314	41.3406	201.3501	5914.862
GA	1.103667	0.926399	45.43465	185.5236	6576.185

ISSN 1533-9211
DOI: 10.5281/zenodo.10807204

Table 5: Statistical Results for the Pressure Vessel Design Problem

Algorithm	Best	Average	Worst	Std
MHHO	5880.671	6097.401	6930.734	265.4723
HHO	5950.075	6651.196	7541.441	360.5791
ALO	5881.059	6242.857	7358.018	364.3424
AO	6068.143	6699.043	7598.282	425.6267
DOA	5882.681	6152.946	9728.734	639.3113
GWO	6035.825	6506.604	7383.845	328.3425
MVO	5907.15	6611.37	7341.25	416.5641
WHO	5881.245	6189.408	7299.265	362.3032
WOA	5969.576	7562.618	11610.02	1198.784
WSO	5923.719	7557.022	14753.81	1972.031
DCSO	5882.232	6151.743	6531.004	161.5312
PSO	5914.862	6292.245	7037.343	497.5605
GA	6576.185	6674.924	8041.525	661.4862

Pressure vessel design is optimized through the use of standard HHO, MHHO and competitor algorithms. Table 4 presents the findings for the design variables related to this subject.

The table indicates that MHHO yields the best values for the design variables, which are (0.7781686, 0.3830364, 40.31962, 200). This results in an objective function value of 5880.6708. Table 5 displays the statistical indicator results of the competitor and MHHO algorithm performances. By offering more favorable values for statistical indicators, MHHO has successfully optimized the pressure vessel design challenge, according to statistical results. Figure 6 displays the MHHO convergence curve.

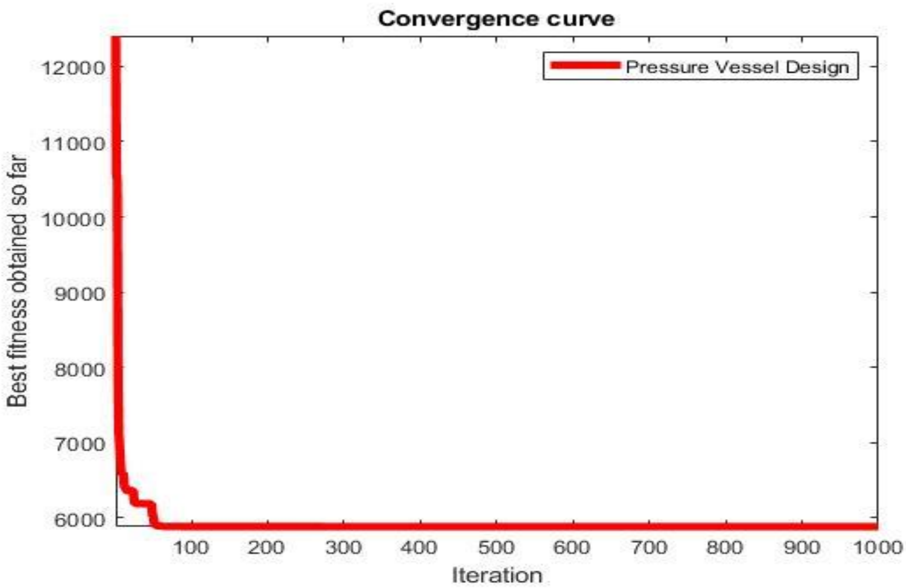

Figure 6: Convergence Curve of the MHHO for the Pressure Vessel Design Optimization Problem

The Seybold REPORT
ISSN 1533-9211
DOI: 10.5281/zenodo.10807204

4.3 Speed Reducer Design

The developed speed reducer is a real-world engineering challenge of speed reduction aimed at reducing the weight of the reducer. A schematic of this system is shown in Figure 7 (Sattar & Salim, 2021).The design problem is formulated as follows:

Consider $X = [x_1, x_2, x_3, x_4, x_5, x_6, x_7] = [b, m, z, l_1, l_2, d_1, d_2]$.

Minimize

$$f(X) = 0.7854x_1x_2^2(3.333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2)$$

Subject to:

$$g_1(X) = \frac{27}{x_1x_2^2x_3} - 1 \leq 0,$$

$$g_2(X) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0,$$

$$g_3(X) = \frac{1.93x_4^3}{x_2x_6^4x_3} - 1 \leq 0,$$

$$g_4(X) = \frac{1.93x_5^3}{x_2x_7^4x_3} - 1 \leq 0,$$

$$g_5(X) = \frac{\sqrt{(\frac{745x_4}{x_2x_3})^2 + 157.5 \times 10^6}}{110x_6^3} - 1 \leq 0,$$

$$g_6(X) = \frac{\sqrt{(\frac{745x_5}{x_2x_3})^2 + 16.9 \times 10^6}}{85x_7^3} - 1 \leq 0,$$

$$g_7(X) = \frac{x_2x_3}{40} - 1 \leq 0,$$

$$g_8(X) = \frac{5x_2}{x_1} - 1 \leq 0,$$

$$g_9(X) = \frac{x_1}{12x_2} - 1 \leq 0,$$

$$g_{10}(X) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0,$$

$$g_{11}(X) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0,$$

ISSN 1533-9211
DOI: 10.5281/zenodo.10807204

Variable range:

$2.6 \leq x_1 \leq 3.6, 0.7 \leq x_2 \leq 0.8, x_3 \in \{17, 18, 19, \ldots, 28\}, 7.3 \leq x_4, x_5 \leq 8.3,$
$2.9 \leq x_6 \leq 3.9, 5 \leq x_7 \leq 5.5.$

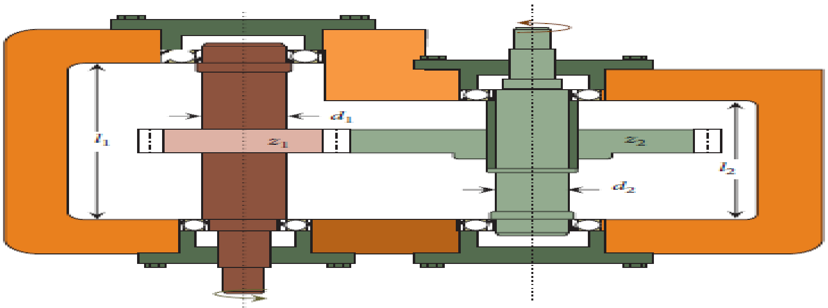

Figure 7: A Schematic Representation of Speed Reducer

Table 6: Comparison Results for the Speed Reducer Design Problem

Algorithm	Optimum variables							Optimum cost
	b	m	p	l1	l2	d1	d2	
MHHO	3.5	0.7	17	7.3	7.71532	3.35054	5.28665	2994.425
HHO	3.51957	0.7	17	7.3	7.80311	3.36278	5.28668	3007.186
ALO	3.5	0.7	17	7.30067	7.73969	3.35054	5.28666	2994.971
AO	3.53806	0.7	17	8.11482	8.02291	3.36103	5.29534	3031.574
DOA	3.5	0.7	17	7.3	7.71532	3.35054	5.28665	2994.485
GWO	3.50277	0.7	17	7.46229	7.74759	3.35365	5.28683	2998.559
MVO	3.50113	0.7	17	7.30129	8.03155	3.35599	5.28693	3003.387
WHO	3.5	0.7	17	7.3	7.71532	3.35054	5.28665	2994.465
WOA	3.5	0.7	17	7.3	7.86772	3.35106	5.30093	3007.013
WSO	3.50256	0.7	17	7.3	7.7179	3.35054	5.28666	2995.489
DCSO	3.5	0.7	17	7.3	7.75145	3.35054	5.28704	2995.463
PSO	3.506677	0.698881	16.97277	8.337218	7.787524	3.358732	5.282308	3066.024
GA	3.516538	0.698879	16.97289	8.357188	7.787516	3.363496	5.283263	3027.485

Table 7: Statistical Results for the Speed Reducer Design Problem

Algorithm	Best	Average	Worst	Std
MHHO	2994.4245	2994.4247	2994.4307	0.00097163
HHO	3007.1864	3140.0792	4051.1245	233.9707
ALO	2994.9713	3002.5002	3014.3303	5.9549
AO	3031.5739	3451.4741	4637.0591	491.7015
DOA	2994.4845	12195.0482	456961.726	64184.5075
GWO	2998.5588	3005.0827	3014.8602	3.9903
MVO	3003.387	3033.6065	3088.1227	15.9462
WHO	2994.4645	2996.1823	3033.7016	7.8481
WOA	3007.0128	3143.9143	4669.3732	229.0892
WSO	2995.4889	3054.8957	4861.2553	321.5473
DCSO	2995.4632	3006.9478	3023.9603	6.4479
PSO	3066.0235	3185.8769	3312.5289	17.11507
GA	3027.4851	3294.6642	3618.7832	57.01175

DOI: 10.5281/zenodo.10807204

Table 6 reports the outcomes of the speed reducer design optimization using MHHO, rival algorithms, and the standard HHO. The findings indicate that MHHO outperformed other metaheuristics in solving this problem, with the objective function value of 2994.4245 and the variable values of (3.5, 0.7, 17, 7.3, 7.71532, 3.35054, 5.28665). The superiority of the suggested MHHO is demonstrated by the statistical results obtained from MHHO and the algorithms compared in Table 7. Figure 8 displays the MHHO convergence curve obtained from solving the speed reducer design problem.

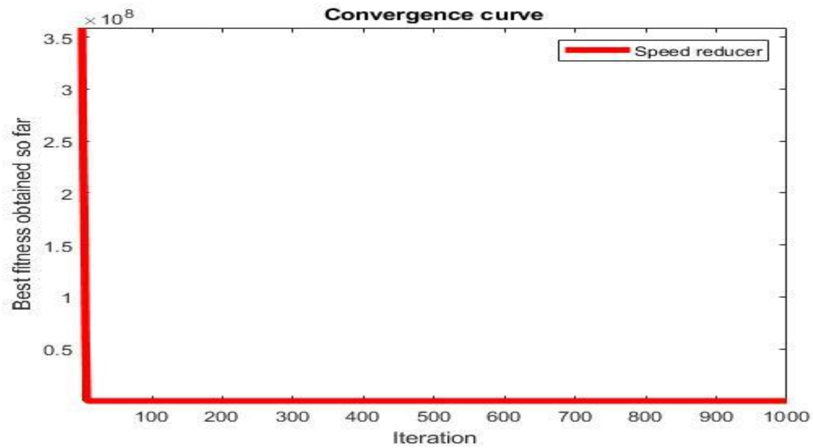

Figure 8: Convergence Curve of the MHHO for the Speed Reducer Design Optimization Problem

4.4 Welded Beam Design

Welded beam design is considered a global challenge in engineering sciences, with the primary aim of reducing the cost of fabricating welded beams. The schematic of this system is depicted in Figure 9 (Coello, 2000). The formulation of this system is as follows:

Consider $X = [x_1, x_2, x_{3,} x_4] = [h, l, t, b]$.

Minimize

$$f(X) = 1.1047x_1^2 x_2 + 0.04811x_3 x_4(14.0 + x_2)$$

Subject to:

$$g_1(X) = \tau(X) - \tau_{max} \le 0,$$

$$g_2(X) = \sigma(X) - \sigma_{max} \le 0,$$

$$g_3(X) = \delta(X) - \delta_{max} \le 0,$$

$$g_4(X) = x_1 - x_4 \le 0,$$

$$g_5(X) = P - P_c(X) \le 0,$$

$$g_6(X) = 0.125 - x_1 \le 0,$$

ISSN 1533-9211

DOI: 10.5281/zenodo.10807204

$$g_7(X) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 0.5 \leq 0,$$

$$\tau(X) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2},$$

$$\tau' = \frac{P}{\sqrt{2x_1x_2}},$$

$$\tau'' = \frac{MR}{J},$$

$$M = P(L + \frac{x_2}{2}),$$

$$R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1 + x_3}{2})^2},$$

$$J = 2\left\{\sqrt{2x_1x_2}\left[\frac{x_2^2}{4}(\frac{x_1 + x_3}{2})^2\right]\right\},$$

$$\sigma(X) = \frac{6PL}{x_4x_3^2},$$

$$\delta(X) = \frac{6PL^2}{Ex_3^2x_4},$$

$$P_c(X) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right),$$

$P = 6000Lb,$

$L = 14$ in,

$\delta_{max} = 0.25$ in,

$E = 30 \times 10^6$ psi,

$G = 12 \times 10^6$ psi,

$\tau_{max} = 13{,}600$ psi,

$\sigma_{max} = 30000$ psi,

Variable range:

$0.1 \leq x_1, x_4 \leq 2, x_3 \leq 10.$

ISSN 1533-9211
DOI: 10.5281/zenodo.10807204

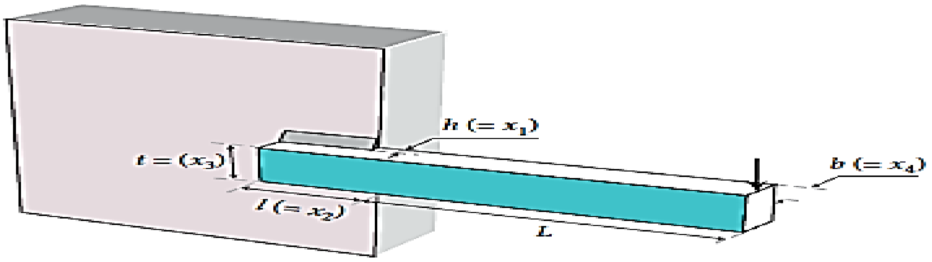

Figure 9: Schematic of the Welded Beam Structure with Indication of Design Variables

Table 8: Comparison Results for the Welded Beam Design Problem

Algorithm	Optimum variables				Optimum cost
	h	*l*	*t*	*b*	
MHHO	0.18297	2.4073	9.5829	0.18298	**1.4731**
HHO	0.175	2.621	9.585	0.18296	1.491
ALO	0.18298	2.4073	9.5818	0.18298	1.4738
AO	0.18728	2.4662	9.403	0.19055	1.515
DOA	0.18298	2.4073	9.5818	0.18298	1.4733
GWO	0.1829	2.4098	9.5827	0.18302	1.4736
MVO	0.18262	2.4152	9.593	0.18296	1.4751
WHO	0.18298	2.4073	9.5818	0.18298	1.4736
WOA	0.1791	2.4572	9.9563	0.18066	1.5112
WSO	0.17721	2.4993	9.5765	0.18319	1.4792
DCSO	0.1823	2.4174	9.5854	0.18296	1.4739
PSO	0.164214	4.033338	9.87542	0.223684	1.876522
GA	0.206527	3.636602	9.95423	0.204201	1.838802

Table 9: Statistical Results for the Welded Beam Design Problem

Algorithm	Best	Mean	Worst	Std
MHHO	1.4731	1.5056	1.5717	0.025956
HHO	1.491	1.6128	2.1813	0.15605
ALO	1.4738	1.4901	1.6343	0.030342
AO	1.5150	1.6926	2.2152	0.16808
DOA	1.4733	1.7108	3.8309	0.46821
GWO	1.4736	1.4749	1.4795	0.0011788
MVO	1.4751	1.4881	1.5129	0.010688
WHO	1.4736	1.4768	1.5516	0.013334
WOA	1.5112	1.8765	4.2271	0.50215
WSO	1.4792	2.3264	3.9759	0.53178
DCSO	1.4739	1.4806	1.5053	0.0061549
PSO	1.8765	2.1233	2.3246	0.035104
GA	1.8388	1.3662	2.0392	0.139689

MHHO, HHO and competitive algorithms are applied to the welded beam design problem, and the results are presented in Table 8. Based on these results, MHHO has the best solution for this problem with values of variables equal (0.18297, 2.4073, 9.5829, 0.18298) and the corresponding objective function value 1.4731. The statistical results is reported in Table 9.

ISSN 1533-9211
DOI: 10.5281/zenodo.10807204

This table shows that MHHO performs well in terms of statistical signals. Figure 10 shows the convergence for this system.

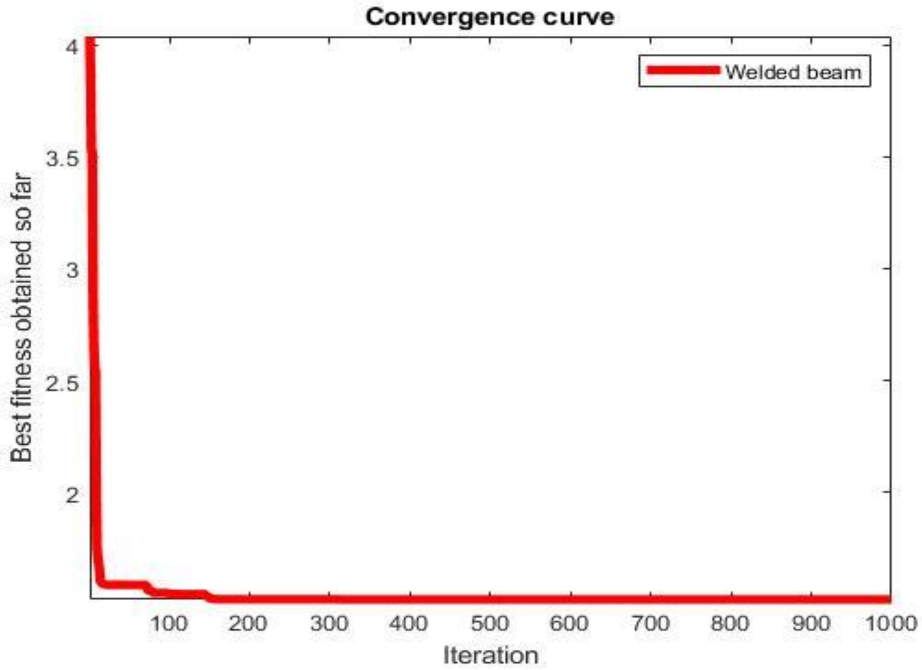

Figure 10: Convergence Curve of the MHHO for the Welded Beam Design Optimization Problem

5. CONCLUSION

In this work, a Modified Harris Hawks optimization (MHHO) algorithm is proposed using a mutation-selection strategy and crossover operator. This adaptability enables the algorithm to find optimal or nearly optimal solutions in a variety of optimization scenarios and search environments, potentially improving performance. Numerous restricted structural engineering design problems have been used to test the suggested approach, and it has also been compared to well-known metaheuristic algorithms and the standard HHO.

Systematic experiments showed that compared to other well-known algorithms, the MHHO algorithm produced more dependable solutions. Moreover, the experimental results demonstrate that, in terms of optimization performance, MHHO performed better than the standard HHO and other metaheuristic algorithms. When compared to constrained engineering design benchmark functions, the MHHO performs significantly better than other state-of-the-art algorithms, demonstrating its potential to handle a wide range of constrained optimization problems.

ISSN 1533-9211

DOI: 10.5281/zenodo.10807204

References

1) Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., Al-qaness, M. A. A., & Gandomi, A. H. (2021). Aquila Optimizer: A novel meta-heuristic optimization algorithm. *Computers & Industrial Engineering*, *157*, 107250. https://doi.org/10.1016/j.cie.2021.107250

2) Ahmed, A. M., Rashid, T. A., & Saeed, S. Ab. M. (2021). Dynamic Cat Swarm Optimization algorithm for backboard wiring problem. *Neural Computing and Applications*, *33*(20), 13981–13997. https://doi.org/10.1007/s00521-021-06041-3

3) Arora, J. S. (2011). *Introduction to optimum design* (3rd ed). Academic Press.

4) Braik, M., Hammouri, A., Atwan, J., Al-Betar, M. A., & Awadallah, M. A. (2022). White Shark Optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems. *Knowledge-Based Systems*, *243*, 108457. https://doi.org/10.1016/j.knosys.2022.108457

5) Coello, C. (2000). *Use of a self-adaptive penalty approach for engineering optimization problems*. *41*, 113–127.

6) Eskandar, H., Sadollah, A., Bahreinineja, A., & Abd Shukor, M. (2012). Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers & Structures*, *110–111*, 151–166. https://doi.org/10.1016/j.compstruc.2012.07.010

7) Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, *97*, 849–872. https://doi.org/10.1016/j.future.2019.02.028

8) Holland, J. H. (1992). Genetic Algorithms. *Scientific American*, *267*(1), 66–73.

9) Houssein, E. H., Helmy, B. E., Rezk, H., & Nassef, A. M. (2021). An enhanced Archimedes optimization algorithm based on Local escaping operator and Orthogonal learning for PEM fuel cell parameter identification. *Engineering Applications of Artificial Intelligence*, *103*, 104309. https://doi.org/10.1016/j.engappai.2021.104309

10) Houssein, E., Helmy, B., Rezk, H., & Nassef, A. (2021). An enhanced Archimedes optimization algorithm based on Local escaping operator and orthogonal learning for PEM fuel cell parameter identification. *Engineering Applications of Artificial Intelligence*, *103*, 104309. https://doi.org/10.1016/j.engappai.2021.104309

11) Kannan, B. K., & Kramer, S. N. (1994). An Augmented Lagrange Multiplier Based Method for Mixed Integer Discrete Continuous Optimization and Its Applications to Mechanical Design. *Journal of Mechanical Design*, *116*(2), 405–411. https://doi.org/10.1115/1.2919393

12) Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95-International Conference on Neural Networks*, *4*, 1942–1948. https://ieeexplore.ieee.org/abstract/document/488968/

13) Mirjalili, S. (2015). The Ant Lion Optimizer. *Advances in Engineering Software*, *83*, 80–98. https://doi.org/10.1016/j.advengsoft.2015.01.010

14) Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software*, *95*, 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008

15) Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-Verse Optimizer: A nature-inspired algorithm for global optimization. *Neural Computing and Applications*, *27*(2), 495–513. https://doi.org/10.1007/s00521-015-1870-7

16) Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, *69*, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

The Seybold REPORT

ISSN 1533-9211

DOI: 10.5281/zenodo.10807204

17) Naruei, I., & Keynia, F. (2022). Wild horse optimizer: A new meta-heuristic algorithm for solving engineering optimization problems. *Engineering with Computers*, *38*(S4), 3025–3056. https://doi.org/10.1007/s00366-021-01438-z

18) Peraza-Vázquez, H., Peña-Delgado, A. F., Echavarría-Castillo, G., Morales-Cepeda, A. B., Velasco-Álvarez, J., & Ruiz-Perez, F. (2021). A Bio-Inspired Method for Engineering Design Optimization Inspired by Dingoes Hunting Strategies. *Mathematical Problems in Engineering*, *2021*, 1–19. https://doi.org/10.1155/2021/9107547

19) Rao, S. S. (2009). *Engineering optimization: Theory and practice* (4th ed). John Wiley & Sons.

20) Sattar, D., & Salim, R. (2021). A smart metaheuristic algorithm for solving engineering problems. *Engineering with Computers*, *37*(3), 2389–2417. https://doi.org/10.1007/s00366-020-00951-x

21) Sergeyev, Y., Kvasov, D., & Mukhametzhanov, M. (2018). On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget. *Scientific Reports*, *8*. https://doi.org/10.1038/s41598-017-18940-4

22) Trojovský, P., & Dehghani, M. (2023). A new bio-inspired metaheuristic algorithm for solving optimization problems based on walruses behavior. *Scientific Reports*, *13*(1), 8775. https://doi.org/10.1038/s41598-023-35863-5

23) Wang, S., Jia, H., Liu, Q., Zheng, R., Wang, S., Jia, H., Liu, Q., & Zheng, R. (2021). An improved hybrid Aquila Optimizer and Harris Hawks Optimization for global optimization. *Mathematical Biosciences and Engineering*, *18*(6), Article mbe-18-06-352. https://doi.org/10.3934/mbe.2021352