Received: June 21, 2024. Revised: August 17, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Fossa Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Engineering Applications

Tareq Hamadneh[1,2] Belal Batiha[3] Frank Werner[4] Zeinab Montazeri[5]
Mohammad Dehghani[5]* Gulnara Bektemyssova[6] Kei Eguchi[7]

[1]Department of Mathematics, Al Zaytoonah University of Jordan, Amman 11733, Jordan
[2]Jadara Research Center, Jadara University, Irbid 21110, Jordan
[3]Department of Mathematics, Faculty of Science and Information Technology, Jadara University, Irbid 21110, Jordan
[4]Faculty of Mathematics, Otto-von-Guericke University, P.O. Box 4120, 39016 Magdeburg, Germany
[5]Department of Electrical and Electronics Engineering, Shiraz University of Technology, Shiraz 7155713876, Iran
[6]Department of Computer Engineering, International Information Technology University, Almaty 050000, Kazakhstan
[7]Department of Information Electronics, Fukuoka Institute of Technology, Japan
* Corresponding author's Email: adanbax@gmail.com

Abstract: This paper presents a novel bio-inspired metaheuristic algorithm termed the Fossa Optimization Algorithm (FOA), which emulates the natural hunting behavior of the fossa in its habitat. FOA draws its core inspiration from the two-stage hunting technique of the fossa, involving an initial attack on a spotted lemur followed by a pursuit through the trees. The theoretical framework of FOA is elaborated, and its implementation is mathematically modeled in two distinct phases: (i) exploration, which simulates the fossa's positional adjustments during the initial attack on the lemur, and (ii) exploitation, which models the positional changes of the fossa during the chase. The efficacy of FOA is tested against twenty-two constrained optimization problems from the CEC 2011 test suite, as well as four engineering design challenges. The optimization results demonstrate FOA's strong capabilities in both exploration and exploitation, maintaining a balance that facilitates convergence to optimal solutions. FOA's performance is benchmarked against twelve established algorithms, showing that it consistently outperforms its competitors by delivering superior results and ranking as the top optimizer in most of the evaluated functions. These findings indicate that FOA is highly effective in addressing optimization tasks in real-world scenarios.

Keywords: Metaheuristic, Optimization, Fossa, Exploitation, Exploration.

1. Introduction

Optimization issues are crucial in today's world, making efficient methods to handle them a significant research challenge. The primary aim is to identify the most optimal solution from a spectrum of possibilities [1]. While the ideal is to achieve the global optimum, quasi-optimal solutions are sometimes acceptable [2]. Optimization dilemmas typically involve modeling with decision variables, constraints, and an objective function, aimed at optimizing the latter by determining suitable values for the former while adhering to constraints [3]. Techniques employed to address these optimization predicaments are categorized into deterministic and stochastic approaches [4].

Deterministic algorithms always reach the same solution under the same initial conditions and are effective for simple problems and small solution spaces [5, 6]. However, they struggle with large-scale, nonlinear, and multimodal problems, often getting stuck in local optima [7, 8]. Stochastic approaches, particularly metaheuristic algorithms, offer an alternative by using random search [9]. These algorithms are popular due to their ease of implementation, efficiency in complex and unknown solution spaces, and ability to handle non-convex and non-differentiable problems [10]. Despite their advantages, metaheuristic algorithms do not guarantee the global optimal solution but can provide quasi-optimal solutions close to the global optimum [11]. This has led to the development of numerous metaheuristic algorithms [12].

The No Free Lunch (NFL) theorem asserts that no single metaheuristic algorithm universally excels across all optimization problems, underscoring the imperative to innovate and develop novel algorithms [13]. The inherent stochasticity of metaheuristic algorithms and the uncertainty surrounding optimal solutions emphasize the necessity of creating fresh algorithms aimed at achieving superior solutions. These considerations serve as catalysts prompting researchers to explore and invent new metaheuristic algorithms tailored for scientific optimization tasks.

This study introduces a groundbreaking bio-inspired metaheuristic algorithm known as the Fossa Optimization Algorithm (FOA), specifically developed to address complex optimization problems. The primary innovations and contributions of this research are outlined as follows:

- **Development of FOA**: The FOA is designed by drawing inspiration from the natural behaviors of the fossa, a unique predator in the wild.
- **Hunting Strategy Inspiration**: The FOA's conceptual framework is based on the fossa's two-phase hunting strategy, which includes an initial phase of attacking a lemur and a subsequent phase of chasing the lemur.
- **Theoretical and Mathematical Modeling**: The FOA's methodology is explained and mathematically formulated in two distinct phases:
- **Exploration Phase**: This phase models the positional changes of the fossa during its initial attack on the lemur, aiming to cover a broad search space.
- **Exploitation Phase**: This phase simulates the positional adjustments of the fossa during the chase, focusing on refining and intensifying the search around promising areas.
- **Performance Evaluation**: The effectiveness of FOA is rigorously tested on a set of twenty-two constrained optimization problems from the CEC 2011 test suite, as well as four complex engineering design problems.
- **Comparative Analysis**: The optimization results achieved by FOA are benchmarked against twelve well-established metaheuristic algorithms, providing a comprehensive comparison of performance metrics.

The outcomes of this study highlight the FOA's remarkable ability to balance exploration and exploitation throughout the search process, leading to high-quality solutions for various optimization tasks. The comparative analysis demonstrates that FOA outperforms many existing algorithms, consistently achieving superior results and securing top rankings across most benchmark functions. These findings affirm the FOA's potential as a robust tool for solving real-world optimization problems.

The paper is organized as follows: Section 2 presents the literature review. Section 3 introduces and models the proposed Fossa Optimization Algorithm (FOA). Section 4 assesses the performance of the FOA in tackling a variety of real-world optimization problems, presenting experimental results and comparative analyses to demonstrate its efficacy. Finally, Section 5 offers concluding remarks and proposes directions for future research, suggesting potential improvements and new applications for the FOA.

2. Literature review

Metaheuristic algorithms are inspired by a diverse range of natural phenomena, behaviors of living organisms, scientific principles from genetics, biology, and physics, as well as human activities and evolutionary processes. These algorithms can be broadly categorized into four groups based on their sources of inspiration: swarm-based, evolutionary-based, physics-based, and human-based approaches.

Swarm-based metaheuristic algorithms draw inspiration from the collective behaviors observed in animals, insects, birds, and other creatures. For instance, Particle Swarm Optimization (PSO) [14] mimics the movement of flocks of birds or schools of fish searching for food, where each particle (representing a potential solution) updates its position based on its own experience and that of the swarm. Ant Colony Optimization (ACO) [15] replicates the foraging behavior of ants, using pheromone trails to find the shortest path between their colony and a food source. Similarly, the Grey Wolf Optimizer (GWO) [16] models the hierarchical hunting behavior of grey wolves. Various natural behaviors inspire many swarm-based metaheuristic algorithms, including foraging, hunting, migration, and escape. Examples include Walrus Optimization Algorithm (WaOA)

Received: June 21, 2024. Revised: August 17, 2024.

[17], Gooseneck Barnacle Optimization (GBO) [18], Termite Alate Optimization Algorithm (TAOA) [19], Orca Predation Algorithm (OPA) [20], Electric Eel Foraging Optimization (EEFO) [21], and Greylag Goose Optimization (GGO) [22].

Evolutionary-based metaheuristic algorithms are grounded in principles of genetics, biology, and evolutionary theory, such as natural selection and survival of the fittest. Notable examples include Genetic Algorithm (GA) [23] and Differential Evolution (DE) [24], which emulate biological reproduction and genetic concepts such as mutation and crossover. In GA, each individual, or chromosome, represents a member of the population. Parents are selected using a selection operator, and reproduction is simulated through crossover and mutation operators. Following the survival of the fittest principle, successive generations evolve, guiding the algorithm toward the optimal solution for the problem.

Physics-based metaheuristic algorithms are inspired by principles and phenomena in physics. Simulated Annealing (SA) [25] mimics the metal annealing process, where metals are melted with heat and slowly cooled to form an ideal crystal, analogous to finding an optimal solution. Gravitational Search Algorithm (GSA) [26] models gravitational attraction between masses, where each mass represents a potential solution, moving towards better solutions based on gravitational force and Newton's laws. Spring Search Algorithm (SSA) [27] is based on Hooke's law, where weights (candidate solutions) move towards heavier weights, leading to convergence on better solutions. Momentum Search Algorithm (MSA) [28] uses collision momentum of balls, each representing a candidate solution, to move towards the best known position. Cosmological concepts inspire algorithms like Multi-Verse Optimizer (MVO) [29] and Black Hole Algorithm (BHA) [30].

Human-based metaheuristic algorithms are inspired by various aspects of human behavior, decision-making, and social interactions. Teaching-Learning Based Optimization (TLBO) [31] models the classroom learning process, where teachers and students represent candidate solutions. The best solution, the teacher, guides students toward optimal solutions during the teacher phase, and student knowledge sharing improves solution quality in the student phase. Mother Optimization Algorithm (MOA) [32] is based on a Eshrat's care and education of her children, with the mother and children as candidate solutions. The algorithm updates solutions through education, advice, and upbringing phases. Driving Training-Based Optimization (DTBO) [7] simulates driving school training, with driving applicants and instructors as candidate solutions. The algorithm updates positions through training, skill patterning, and practice phases.

Despite the extensive range of inspirations for existing metaheuristic algorithms, none have yet been designed based on the natural behaviors of the fossa, a predator native to Madagascar. The fossa's strategy of locating, attacking, and chasing lemurs through the trees exhibits intelligent and efficient hunting tactics that can be harnessed for optimization purposes. To address this gap, this paper introduces a novel metaheuristic algorithm called the Fossa Optimization Algorithm (FOA), inspired by the hunting processes of the fossa. The subsequent sections of this paper will delve into the theoretical foundation, mathematical modeling, and performance evaluation of FOA, demonstrating its potential as an effective tool for solving complex optimization problems.

3. Fossa optimization algorithm

In this section, we delve into the foundational inspiration behind the development of the proposed Fossa Optimization Algorithm (FOA). We begin by exploring the biological and behavioral characteristics of the fossa that have been emulated in the design of FOA. Following this, we present a detailed mathematical modeling of the algorithm's implementation steps, demonstrating how these natural behaviors are translated into computational procedures for optimization.

3.1 Inspiration of FOA

The fossa (Cryptoprocta ferox) is a cat-like mammal native to Madagascar, belonging to the Eupleridae family.

Among the fossa's natural behaviors in wildlife, the strategy of this animal in hunting lemurs is much more prominent. This intelligent strategy has two stages: (i) the attack of the fossa towards the location of the observed lemur and (ii) the chase process between the fossa and the lemur on the trees. Mathematical modeling of these intelligent fossa behaviors during hunting has been employed to design the proposed Fossa Optimization Algorithm (FOA) approach, which is discussed below.

3.2 Algorithm initialization

The proposed Fossa Optimization Algorithm (FOA) operates as a population-based optimization technique where each individual fossa represents a member of the population. FOA efficiently searches for optimal solutions by mimicking the fossas' natural search behaviors within the defined problem space. In this analogy, the fossa's habitat corresponds to the problem-solving space, and the position of each fossa within this habitat represents a potential solution to the optimization problem.

Each fossa's position is characterized by a vector, where each element of the vector signifies the value of a decision variable. Thus, a fossa's position encapsulates a candidate solution. The entire population of fossas, each represented by a position vector, is mathematically described by a matrix, as shown in Eq. (1). The initial positions of the fossas are assigned randomly within the problem space, according to Eq. (2).

This structured approach allows FOA to explore and exploit the search space effectively, using the fossas' dynamic positional adjustments to iteratively hone in on optimal solutions. By leveraging the inherent search capabilities of the fossas, FOA ensures a thorough examination of the problem space, leading to high-quality solutions for complex optimization problems.

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N\times m} = \begin{bmatrix} x_{1,1} \cdots x_{1,d} \cdots x_{1,m} \\ \vdots \ \ \ddots \ \ \vdots \ \ \iddots \ \ \vdots \\ x_{i,1} \cdots x_{i,d} \cdots x_{i,m} \\ \vdots \ \ \iddots \ \ \vdots \ \ \ddots \ \ \vdots \\ x_{N,1} \cdots x_{N,d} \cdots x_{N,m} \end{bmatrix}_{N\times m} \tag{1}$$

$$x_{i,d} = lb_d + r \cdot (ub_d - lb_d) \tag{2}$$

In this context, X represents the population matrix of FOA, where X_i denotes the ith fossa, which is a candidate solution. $x_{i,d}$ represents the dth dimension of the ith fossa in the search space, where N stands for the number of fossas, m indicates the number of decision variables, r is a random number in interval $[0,1]$, lb_d, and ub_d denote the lower and upper bounds of the dth decision variable, respectively.

As previously stated, each fossa's position denotes a potential solution for the problem and can undergo evaluation within the objective function. The resultant values from evaluating the objective function can be encapsulated in a vector, as outlined by Eq. (3).

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N\times 1} \tag{3}$$

Here, F represents the vector of evaluated objective function values, with F_i indicating the objective function value corresponding to the ith fossa.

3.3 Mathematical modelling of FOA

The FOA algorithm is designed by emulating the intelligent movement strategies of fossas in nature. The positions of the FOA members within the problem-solving space are updated through two distinct phases:

Exploration Phase: This phase is inspired by the initial attack behavior of fossas when targeting a lemur. During this phase, the algorithm focuses on exploring the search space broadly to identify promising regions. The positional updates during exploration are modeled based on the changes in the fossa's position as it prepares and initiates its attack.

Exploitation Phase: This phase mimics the pursuit of the lemur through the trees, where the fossa fine-tunes its approach to home in on the target. In the exploitation phase, the algorithm intensifies the search around the identified promising areas, refining the solutions. The positional updates during exploitation are based on the dynamic adjustments made by the fossa during the chase.

The mathematical modeling and detailed explanation of each update phase in FOA are presented below.

3.3.1 Attacking and moving towards the lemur (exploration phase)

In the initial phase of the FOA, the positions of the population members within the problem-solving space are updated by simulating the fossa's attack on an observed lemur. Fossas can accurately locate lemurs using their keen sense of smell, hearing, and vision. Once the lemur's position is identified, the fossa advances toward it. This modeled displacement during the attack phase results in significant changes in the positions of the population members, thereby enhancing FOA's global exploration capabilities.

For each fossa, the positions of other population members with better objective function values are regarded as the lemurs' locations within its habitat. Consequently, the set of candidate lemurs for each fossa is determined by comparing objective function values, as described by Eq. (4):

$$CL_i = \{X_k : F_k < F_i \ and \ k \neq i\}, \\ \text{where } i = 1,2,\dots,N \text{ and } k \in \{1,2,\dots,N\} \tag{4}$$

In this equation, CL_i represents the set of candidate lemur locations for the i th fossa, X_k denotes the population member with a superior objective function value than the ith fossa, and F_k is its corresponding objective function value.

The FOA assumes that the fossa randomly selects one of these candidate lemurs in its habitat and launches an attack. Based on the fossa's positional change during the attack on the identified lemur, a new random position for each member of the FOA population is calculated using Eq. (5). If this new position yields a better objective function value, it replaces the previous position of the respective population member, as outlined in Eq. (6).

$$x_{i,j}^{P1} = x_{i,j} + r_{i,j} \cdot (SL_{i,j} - I_{i,j} \cdot x_{i,j}), \quad (5)$$

$$X_i = \begin{cases} X_i^{P1}, & F_i^{P1} \leq F_i, \\ X_i, & else, \end{cases} \quad (6)$$

In this context, SL_i denotes the lemur chosen by the ith fossa and $SL_{i,j}$ refers to the jth dimension of this selected lemur's position. X_i^{P1} represents the newly computed position for the ith fossa during the attack phase of the FOA, with $x_{i,j}^{P1}$ being its jth dimension. The objective function value at this new position is F_i^{P1}. The terms $r_{i,j}$ are random nvalues within the range $[0, 1]$, and $I_{i,j}$ are random integers, either 1 or 2.

3.3.2 Phase 2: Chasing to catch lemur (exploitation phase)

In the second phase of FOA, the positions of the population members are updated by simulating the fossa's pursuit of the lemur. The fossa utilizes its exceptional climbing abilities to follow the lemur through the trees and across branches. This pursuit takes place in a confined area within the hunting ground. By modeling the fossa's movements during the chase, the algorithm introduces minor adjustments to the positions of the population members, thereby enhancing the FOA's exploitation capabilities in local search optimization.

In the FOA design, the chase process between the fossa and the lemur in nature is represented by small positional changes in the population members. The positional updates during the lemur chase are mathematically modeled, with a new position for each FOA member calculated using Eq. (7). If this new position results in an improved objective function value, it replaces the member's previous position, as outlined in Eq. (8).

$$x_{i,j}^{P2} = x_{i,j} + \left(1 - 2\, r_{i,j}\right) \cdot \frac{ub_j - lb_j}{t} \quad (7)$$

$$X_i = \begin{cases} X_i^{P2}, & F_i^{P2} \leq F_i \\ X_i, & else \end{cases} \quad (8)$$

In this context, X_i^{P2} represents the updated position computed for the ith f fossa during the chasing phase in the proposed FOA. Each $x_{i,j}^{P2}$ signifies the jth dimension of this new position, while F_i^{P2} denotes its corresponding objective function value. The variables $r_{i,j}$ are random numbers ranging from $[0, 1]$, and t stands for the current iteration count.

4. Simulation studies

In this segment, the efficacy of FOA in addressing optimization challenges in practical applications is assessed. To achieve this, a total of twenty-six constrained optimization problems have been chosen. These comprise twenty-two real-world problems sourced from the CEC 2011 test suite, alongside an additional four problems from the realm of engineering design.

4.1 Evaluation of CEC 2011 test suite

In this section, we evaluate the effectiveness of FOA and its competitors in tackling the challenges posed by the CEC 2011 test suite. This test suite encompasses twenty-two distinct constrained optimization problems derived from real-world applications. Comprehensive details, including the mathematical models and descriptions of the CEC 2011 test suite, can be accessed in [33].

Table 1 presents the optimization outcomes obtained by FOA and competing algorithms on the CEC 2011 test suite. The results highlight FOA's capacity to deliver effective solutions across problems C17-F1 to C17-F22, demonstrating its adeptness in exploration, exploitation, and maintaining a balance throughout the search process. According to the findings from simulations, FOA consistently outperformed all other algorithms across all twenty-two optimization problems, positioning it as the top-performing optimizer. Statistical analyses, specifically the Wilcoxon rank sum test conducted on experimental data, further underscored FOA's significant statistical superiority over the twelve alternative algorithms in optimizing the CEC 2011 test suite.

Received: June 21, 2024. Revised: August 17, 2024.

Table 1. Performance of metaheuristic algorithms on CEC 2011 test suite

		GA	PSO	GSA	TLBO	GWO	MVO	WOA	TSA	MPA	RSA	AVOA	WSO	FOA
C11-F1	Best	20.99485	10.47427	17.97987	16.71953	1.130611	11.58557	7.604293	16.90901	0.453935	18.45749	8.642607	14.08941	2.00E-10
	Mean	22.14845	17.20023	20.60315	17.65027	10.72532	13.59232	12.90498	17.62112	7.718053	20.85771	12.63879	16.95377	5.920103
	Median	21.54986	17.5065	21.08212	17.37924	12.49167	13.61517	13.57618	17.20423	8.845166	20.79824	12.72897	16.97953	5.687176
	Worst	24.49924	23.31363	22.2685	19.12309	16.78732	15.55336	16.86329	19.16701	12.72794	23.37688	16.45459	19.76661	12.30606
	std	1.756925	6.23952	2.006092	1.188596	7.3733	2.470244	4.664845	1.157165	6.115793	2.501392	4.807401	2.952731	7.476538
	rank	13	8	11	10	3	6	5	9	2	12	4	7	1
C11-F2	Best	-16.1717	-24.0945	-20.9656	-13.286	-24.7265	-12.1764	-22.2907	-15.9609	-25.6183	-13.168	-21.8573	-16.597	-27.0676
	Mean	-13.9987	-22.8058	-16.3567	-12.1633	-22.7616	-10.2743	-19.1379	-12.5182	-24.9761	-12.7716	-21.3238	-15.3149	-26.3179
	Median	-13.6499	-23.2107	-15.984	-12.04	-23.4923	-10.1089	-19.3403	-11.7486	-25.3448	-12.7422	-21.3444	-15.1796	-26.3856
	Worst	-12.5232	-20.7072	-12.4933	-11.2874	-19.3352	-8.70316	-15.5801	-10.6146	-23.5963	-12.4343	-20.749	-14.3035	-25.4328
	std	1.942177	1.600367	4.174713	0.942061	2.60868	1.618708	3.696526	2.751276	1.0288	0.389213	0.577737	1.227751	0.767703
	rank	9	3	7	12	4	13	6	11	2	10	5	8	1
C11-F4	Best	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06
	Mean	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06
	Median	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06
	Worst	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06	3.26E-06
	std	1.39E-16	1.40E-16	1.40E-16	7.27E-14	3.59E-15	9.22E-13	1.40E-16	2.20E-14	1.28E-15	4.62E-11	2.36E-09	2.05E-11	2.08E-19
	rank	5	2	3	9	7	10	4	8	6	12	13	11	1
C11-F4	Best	72.55129	71.10386	37.99406	58.62053	28.13873	31.40391	38.65784	30.85337	23.23436	44.19588	26.19046	40.48998	17.74098
	Mean	74.28086	85.67926	43.4327	85.41443	29.73022	35.40336	45.08551	35.25719	26.66493	57.83315	31.42147	44.24001	22.20814
	Median	74.25833	83.45319	44.05978	80.81618	29.87288	34.8409	43.82622	34.54418	26.20191	55.04009	30.50303	43.94568	21.62299
	Worst	75.9203	91.35037	51.37966	93.81534	31.89236	37.15296	46.47603	36.80899	28.24343	60.29817	32.97873	46.81271	24.33469
	std	1.951844	10.28216	6.117383	16.85201	1.74757	2.705872	3.899698	3.123915	2.324352	8.210035	3.490426	3.505964	3.071087
	rank	11	13	9	12	3	6	7	5	2	10	4	8	1
C11-F5	Best	-12.9633	-14.0945	-31.4583	-15.0617	-34.1245	-31.6367	-28.3603	-31.7808	-33.7811	-23.2641	-29.6017	-26.7142	-34.7494
	Mean	-11.8745	-11.103	-27.9156	-13.0451	-31.6952	-27.596	-28.1683	-27.7213	-33.2163	-21.2904	-28.5962	-25.6376	-34.1274
	Median	-12.0577	-10.3539	-27.5065	-12.7468	-32.3842	-26.6221	-28.2	-28.0587	-33.6269	-21.4602	-28.4014	-25.6157	-34.1871
	Worst	-10.4193	-9.60958	-25.1912	-11.6251	-27.8878	-25.503	-27.913	-22.9871	-31.8304	-18.977	-27.9804	-24.605	-33.3862
	std	1.250673	2.30699	3.004306	1.594488	2.917966	3.142033	0.227202	3.948511	1.012393	2.411024	0.768531	0.990576	0.612958
	rank	12	13	6	11	3	8	5	7	2	10	4	9	1
C11-F6	Best	-10.4176	-7.99292	-25.9196	-4.88561	-22.6153	-17.8135	-22.7897	-16.9438	-25.6097	-14.4772	-20.8568	-15.2174	-27.4298
	Mean	-5.89114	-5.07774	-21.849	-4.30091	-19.8288	-10.7684	-20.1179	-9.00058	-22.4991	-13.9198	-19.2912	-14.8115	-24.1119
	Median	-4.52608	-4.13065	-21.4943	-4.13065	-19.2485	-10.6016	-21.9742	-6.56902	-21.5949	-14.1527	-19.3295	-14.7852	-23.0059
	Worst	-4.09474	-4.05674	-18.4879	-4.05674	-18.2027	-4.05674	-13.7337	-5.92048	-21.197	-12.8964	-17.6491	-14.4582	-23.0059
	std	3.318344	2.122875	3.541293	0.428562	2.28315	7.977012	4.701624	5.812291	2.296303	0.769418	1.554489	0.367927	2.415463
	rank	11	12	3	13	5	9	4	10	2	8	6	7	1
C11-F7	Best	1.293914	0.854726	0.879567	1.474091	0.827545	0.841207	1.56562	1.097291	0.766328	1.608278	1.125389	1.468141	0.582266
	Mean	1.656328	1.106615	1.067479	1.636933	1.056742	0.890619	1.65915	1.265574	0.933894	1.816199	1.249677	1.536587	0.860699
	Median	1.742515	1.119582	1.066934	1.656858	1.065555	0.87712	1.630522	1.183451	0.977372	1.839325	1.244905	1.523598	0.91775
	Worst	1.846369	1.332569	1.25648	1.759925	1.268311	0.967026	1.809935	1.598104	1.014501	1.977866	1.383506	1.63101	1.025027
	std	0.272084	0.265478	0.186178	0.139852	0.197138	0.066569	0.114757	0.246191	0.125447	0.167329	0.152119	0.076051	0.219737
	rank	11	6	5	10	4	2	12	8	3	13	7	9	1
C11-F8	Best	220	245.9004	220	220	220	220	242.6022	220	220	278.3859	224.0273	255.2005	220
	Mean	222.7139	443.6161	244.049	224.1316	227.048	224.1316	261.6804	253.8425	222.6734	314.7225	238.7947	278.5874	220
	Median	220.5788	497.1932	234.7096	220.3934	226.9554	220.9722	250.2866	226.9554	222.5807	313.1747	238.3086	274.4072	220
	Worst	229.6981	534.1777	286.7768	235.7397	234.2814	234.582	303.5461	341.4594	225.5322	354.1547	254.5343	310.3349	220
	std	5.119385	148.666	34.59718	8.460498	8.888807	7.626272	30.73647	64.18699	3.375126	33.97846	14.33194	26.45289	0
	rank	3	12	7	4	5	4	9	8	2	11	6	10	1
C11-F9	Best	1669561	779171.2	633837.2	305005.7	17992.06	70020.37	187100	46780.75	11358.35	623943.8	303703.8	336624.4	5457.674
	Mean	1741609	971669.5	739743.3	368599.6	41337.18	122209.7	338205.6	62107.96	20911.18	953353.7	341487	501589	8789.286
	Median	1726937	958802.2	765168.3	348439.9	37918.27	117413.7	297579	61996.84	21185.22	1035051	348103.4	546223.3	7828.591
	Worst	1843000	1189903	794799.3	472513	71520.11	183991.2	570564.2	77657.41	29915.93	1119369	366037.4	577285	14042.29
	std	92744.64	239797	78603.68	80205.77	24690.8	51444.35	190694.6	14680.84	8914.331	244708.1	29948.31	123361	4040.59
	rank	13	12	10	8	3	5	6	4	2	11	7	9	1
C11-F10	Best	-11.8777	-12.1392	-14.1377	-12.09	-14.9635	-20.8499	-13.9637	-18.7065	-19.2772	-13.2432	-17.2126	-15.4171	-21.8299
	Mean	-11.7997	-12.0639	-13.6434	-11.9747	-14.4933	-15.0277	-13.3928	-14.7489	-18.8938	-12.8612	-17.0117	-14.3738	-21.4889
	Median	-11.8093	-12.0602	-13.7626	-11.974	-14.7956	-13.5846	-13.3078	-13.8074	-18.8929	-12.7759	-17.0947	-14.1093	-21.669
	Worst	-11.7024	-11.996	-12.9109	-11.8606	-13.4183	-12.0915	-12.9919	-12.6743	-18.5122	-12.6498	-16.6447	-13.8597	-20.7878
	std	0.082719	0.0703	0.657711	0.116756	0.796964	4.308425	0.446373	2.968027	0.431168	0.292805	0.287749	0.774897	0.518028
	rank	13	11	8	12	6	4	9	5	2	10	3	7	1
C11-F11	Best	5677943	4876156	1355754	4874518	3479200	797431.5	1212878	4648869	1605925	7898344	933669.6	5170959	260837.9
	Mean	5715298	4909100	1503283	4899197	3668455	1411326	1328293	5556217	1724848	8161789	1127735	5428932	571712.3
	Median	5710274	4911323	1494330	4896975	3602463	1089997	1309812	5450789	1720208	8204838	1140258	5389551	598725.2
	Worst	5762701	4937598	1668719	4928322	3989695	2667880	1480672	6674421	1853049	8339135	1296753	5765667	828560.9
	std	39470.81	31430.76	140055.5	30459.3	243250.4	927121.6	123049	912761.6	127928.3	204150.5	175849.4	298944.6	271080
	rank	12	9	5	8	7	4	3	11	6	13	2	10	1
C11-F12	Best	12985236	2075391	5069911	12214423	1267206	1204649	4976933	4413862	1212610	11129535	3073561	7330982	1155937
	Mean	13101652	2218684	5325393	12957822	1423395	1337048	5347886	4645979	1289778	11980897	3167584	7650391	1199805
	Median	13099661	2198262	5365007	13042668	1435195	1340014	5444858	4702619	1288578	12032002	3179857	7668092	1196965
	Worst	13222048	2402821	5501647	13531529	1555984	1463515	5524894	4764815	1369345	12730048	3237059	7934399	1249353
	std	106367	148292.6	203566.6	603530.9	130905.5	116064.8	277918.8	179484.8	74652.8	716923.4	77855.94	272918.8	48993.46
	rank	13	5	8	12	4	3	9	7	2	11	6	10	1
C11-F13	Best	15461.56	15473.13	84613.22	15610.79	15491.54	15486.67	15490.35	15479.1	15461.74	15849.47	15449.72	15650.7	15444.19
	Mean	28468.14	15488.92	116417.9	15884.56	15498.11	15504.19	15528.85	15488.35	15464.18	16221.86	15450.61	15816.98	15444.2
	Median	15619.13	15480.19	110751.1	15769.62	15495.75	15495.83	15521.86	15487.34	15463.33	15945.42	15450.75	15698.87	15444.2
	Worst	67172.77	15522.16	159556.3	16388.19	15509.41	15538.42	15581.33	15499.63	15468.33	17147.14	15451.22	16219.49	15444.21
	std	28177.61	24.47431	36845.87	384.045	8.636126	26.33876	46.4354	10.80619	3.179862	679.1709	0.707511	295.8929	0.009445
	rank	12	5	13	10	6	7	8	4	3	11	2	9	1
C11-F14	Best	18818.75	18938.08	18796.93	29049.17	19043.13	19250.66	19030.28	19218.19	18539.58	152908.3	18451.81	78523.85	18241.58

 DOI: 10.22266/ijies2024.1031.78

	Mean	19077.55	19088.96	19059.83	280023.5	19185.03	19352.06	19178.82	19452.19	18626.04	206926.2	18546.95	102602.5	18295.35
	Median	19068.96	19095.56	19102.76	276190.7	19169.03	19362.31	19195.3	19319.71	18631.88	188733.6	18547.18	94586.94	18275.87
	Worst	19353.51	19226.62	19236.88	538663.5	19358.95	19432.94	19294.4	19951.15	18700.84	297329.2	18641.62	142712.3	18388.08
	std	239.1445	129.5361	210.3944	267168.7	152.4139	84.72372	133.1554	367.0078	76.74049	70641.04	98.64241	31350.13	74.38679
	rank	5	6	4	13	8	9	7	10	3	12	2	11	1
C11-F15	Best	3237130	33247.73	240949.8	2895904	33032.56	33005.93	33000.47	33047.85	32875.58	720290.4	42110.22	338511.1	32782.17
	Mean	7107922	33322.55	271873.5	13807658	33134.41	33154.23	199218.4	52358.2	33018.48	1717634	100060.8	816433.8	32883.58
	Median	6507633	33264.59	276807.7	15872946	33101.93	33113.48	240023.2	33165.02	32990.01	836157.3	96682.82	439828.3	32897.86
	Worst	12179293	33513.28	292928.8	20588838	33301.23	33384.02	283826.7	110054.9	33218.31	4477930	164767.4	2047568	32956.46
	std	4477377	139.348	26377.34	8785227	129.6209	177.9281	123599.1	42004.12	159.5038	2012667	72064.12	899537.6	79.94256
	rank	12	5	9	13	3	4	8	6	2	11	7	10	1
C11-F16	Best	55289650	58934654	8526963	77564482	142909.8	133833.5	136624.1	141930.9	135804.4	438659.6	134084.5	273214.9	131374.2
	Mean	68406462	71243770	16774448	79596055	145238.3	141600.4	141918.6	144570.6	137885.4	1759603	135665.3	860740.2	133550
	Median	65421652	70454300	14117049	79466231	143717.1	141657.3	141973	145080.4	137067.7	1124036	136090.1	577936.7	133257.5
	Worst	87492894	85131824	30336734	81887274	150609.3	149253.4	147104.1	146190.7	141601.9	4351682	136396.6	2013873	136310.8
	std	14938222	12330851	10298312	1978328	3947.89	7018.439	4767.445	2214.617	2825.563	1921850	1161.754	854983.8	2485.329
	rank	11	12	10	13	7	4	5	6	3	9	2	8	1
C11-F17	Best	1.83E+10	1.65E+10	8.84E+09	1.92E+10	5297236	6661968	6.19E+09	9.49E+08	5224908	9.98E+09	1.89E+09	6.84E+09	1916953
	Mean	1.96E+10	1.87E+10	1E+10	2E+10	6922237	7005034	8.68E+09	1.15E+09	6269965	1.39E+10	2.08E+09	8.03E+09	1926615
	Median	1.89E+10	1.83E+10	1.03E+10	1.99E+10	6583310	6977244	8.5E+09	1.17E+09	6198317	1.43E+10	2.07E+09	8.18E+09	1923412
	Worst	2.21E+10	2.16E+10	1.06E+10	2.09E+10	9225091	7403679	1.15E+10	1.32E+09	7458320	1.7E+10	2.27E+09	8.9E+09	1942685
	std	1.89E+09	2.51E+09	8.93E+08	7.35E+08	1909349	357222.7	2.46E+09	2.06E+08	1123218	3.28E+09	1.86E+08	9.95E+08	12470.83
	rank	12	11	9	13	3	4	8	5	2	10	6	7	1
C11-F18	Best	98886898	1.01E+08	7538917	22101606	968293.4	977576.3	3793961	1707773	953554.7	73305536	3631280	33958804	938416.2
	Mean	1.03E+08	1.21E+08	10068634	27859754	1029725	991222.8	8682456	1941970	976570.8	1.06E+08	5975105	49330249	942057.5
	Median	1.03E+08	1.24E+08	10026993	29600639	980595.5	994076.1	7882971	1906696	957999.8	1.15E+08	5044596	53633053	942553.5
	Worst	1.06E+08	1.34E+08	12681633	30136132	1189417	999162.8	15169920	2246712	1036729	1.21E+08	10179946	56096087	944706.9
	std	3364918	15984910	2508023	4209691	116531.1	10532.68	5246197	280616.2	43947.61	24445003	3329758	11321428	2882.138
	rank	11	13	8	9	4	3	7	5	2	12	6	10	1
C11-F19	Best	1E+08	1.4E+08	2281996	22432725	1222677	1135717	1969595	2110354	1075570	89781843	5585104	41462669	967927.7
	Mean	1.03E+08	1.55E+08	5739620	31991581	1348953	1450893	9276615	2334247	1147559	1.04E+08	6088032	48581106	1025341
	Median	1.03E+08	1.5E+08	6591645	32831756	1322786	1394055	9220230	2248793	1106128	97677003	5711338	45566352	983146.6
	Worst	1.06E+08	1.79E+08	7493195	39870088	1527561	1879745	16696406	2729050	1302409	1.31E+08	7344349	61729050	1167142
	std	2532248	18224008	2582122	8246265	139976.6	339653.7	7576039	296442.5	113940.5	20772114	919788	9980797	103555.5
	rank	11	13	6	9	3	4	8	5	2	12	7	10	1
C11-F20	Best	98348523	1.3E+08	8591082	30375571	979533.8	965606.1	6245290	1574189	961236.8	98191740	4755991	45431180	936143.2
	Mean	1.03E+08	1.43E+08	12880053	31054116	998175.3	975145.4	6622515	1739427	963984.7	1.12E+08	5379388	51622517	941250.4
	Median	1.04E+08	1.43E+08	11524473	31026530	999801.4	974999.6	6559891	1685079	964071	1.09E+08	5357579	49973759	940995.9
	Worst	1.07E+08	1.55E+08	19880187	31787834	1013565	984976.4	7124987	2013362	966560.1	1.33E+08	6046402	61111371	946866.6
	std	4043353	14919097	5387546	641381.5	15945.97	9334.779	410649.8	227709.2	2403.706	16376455	584973.9	7296408	5208.733
	rank	11	13	8	9	4	3	7	5	2	12	6	10	1
C11-F21	Best	54.60476	83.80605	34.18591	45.26357	20.14849	23.76096	33.88851	25.41455	14.00337	53.05298	19.85679	39.1355	9.974206
	Mean	93.74696	96.54937	38.32035	92.04052	21.90974	26.53102	36.60882	28.57921	16.1408	70.34437	21.25232	46.86645	12.71443
	Median	103.2593	97.69051	39.02586	94.17921	21.62156	26.43952	36.22601	29.53327	16.06672	70.33024	21.01376	46.55892	12.95425
	Worst	113.8644	107.0104	41.04376	134.5401	24.24733	29.48409	40.09472	29.83577	18.42639	87.66401	23.12498	55.21246	14.97499
	std	30.11547	12.62395	3.297139	39.95036	2.008403	3.429023	2.988624	2.320265	2.250136	16.67677	1.538698	7.551228	2.506594
	rank	12	13	8	11	4	5	7	6	2	10	3	9	1
C11-F22	Best	84.01881	82.03603	37.22109	61.58839	23.72063	24.30535	37.80818	27.11712	16.46137	43.54151	21.82335	38.765	11.50133
	Mean	85.1019	97.6976	44.00065	94.08526	24.56654	31.10542	43.68071	30.96559	19.27119	59.04216	26.76287	44.11097	16.12513
	Median	84.84462	100.4589	43.42257	101.7697	24.65206	32.16406	44.42632	31.66513	19.61731	62.42624	26.8069	44.23459	16.72317
	Worst	86.69957	107.8366	51.93637	111.2133	25.24143	35.7882	48.062	33.41497	21.38878	67.77466	31.61431	49.20969	19.55286
	std	1.242318	12.64539	6.634157	24.32807	0.710153	5.511919	5.075941	2.956052	2.556775	11.66582	5.059597	4.902887	4.36122
	rank	11	13	8	12	3	6	7	5	2	10	4	9	1
Sum rank		234	210	165	233	99	123	151	150	56	240	112	198	22
Mean rank		10.63636	9.545455	7.5	10.59091	4.5	5.590909	6.863636	6.818182	2.545455	10.90909	5.090909	9	1
Total rank		12	10	8	11	3	5	7	6	2	13	4	9	1
p-value		1.32E-18	6.28E-19	2.11E-18	1.32E-18	1.75E-18	9.84E-16	4.24E-19	9.03E-19	1.75E-18	4.24E-19	2.41E-18	4.24E-19	-

4.2 Evaluation of engineering design problems

In this section, we assess the efficacy of FOA and rival algorithms in tackling four distinct engineering optimization tasks. These challenges encompass pressure vessel design, speed reducer design, welded beam design, and tension/compression spring design. Pressure vessel design aims primarily at minimizing construction costs within real-world engineering applications. Detailed mathematical models and comprehensive information on pressure vessel design can be found in reference [34]. Similarly, speed reducer design focuses on minimizing the weight of the speed reducer, with detailed models and information available in references [35, 36]. Welded beam design aims to reduce fabrication costs, and its detailed mathematical model is outlined in reference [37]. Lastly, tension/compression spring design aims to minimize the weight of these components, with detailed models also available in reference [37].

Table 2 presents the implementation results and performance of competing algorithms in optimizing these engineering challenges. FOA has demonstrated significant achievements in each design scenario:

Table 2. Performance of metaheuristic algorithms on engineering design problems

DP		FOA	WSO	AVOA	RSA	MPA	TSA	WOA	MVO	GWO	TLBO	GSA	PSO	GA
TCS	x1	0.051689	0.051737	0.051257	0.050326	0.05169	0.051073	0.051233	0.050326	0.051924	0.052324	0.054699	0.052053	0.051888
	x2	0.356718	0.357867	0.346439	0.319486	0.356738	0.342104	0.345876	0.324558	0.362385	0.372754	0.430976	0.366437	0.362179
	x3	11.28897	11.22211	11.92786	14.29026	11.2879	12.22009	11.96323	13.56401	10.96935	10.81158	8.238095	11.25212	11.33667
	OF	0.012665	0.012665	0.01267	0.013099	0.012665	0.01268	0.01267	0.01274	0.01267	0.012783	0.013024	0.012792	0.012752
WB	x1	0.778027	0.77803	0.778152	1.194252	0.778029	0.779498	0.911327	0.834505	0.778847	0.908547	1.129191	0.933719	0.886017
	x2	0.384579	0.384583	0.384653	0.639897	0.384581	0.385818	0.450989	0.416446	0.386002	0.628703	1.155857	0.641424	0.557939
	x3	40.31228	40.31245	40.3187	60.51195	40.31237	40.38643	46.22319	43.22087	40.34048	42.75299	44.1012	44.05702	43.02611
	x4	200	199.999	199.9263	48.28486	200	200	133.9229	163.8891	199.7992	182.9717	190.8091	168.7006	176.7552
	OF	5882.901	5882.941	5883.219	7756.095	5882.934	5909.346	6270.268	6004.221	5891.301	7762.927	11970.25	7816.691	7181.082
SR	x1	3.5	3.500037	3.500004	3.581054	3.500014	3.511351	3.576922	3.502079	3.500663	3.50624	3.520229	3.506347	3.50428
	x2	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.700726	0.70242	0.700726	0.700484
	x3	17	17	17	17	17	17	17	17	17	17.09734	17.32445	17.09734	17.06489
	x4	7.3	7.3	7.30074	8.110958	7.3	7.3	7.3	7.3	7.304521	7.491246	7.757514	7.545238	7.47249
	x5	7.8	7.800502	7.800076	8.205554	7.800498	8.205554	7.984485	8.036471	7.8	7.823629	7.878762	7.823629	7.815752
	x6	3.350215	3.350249	3.350272	3.355059	3.350215	3.350501	3.360247	3.367261	3.362298	3.366437	3.401692	3.367212	3.361677
	x7	5.286683	5.286693	5.286703	5.4595	5.286687	5.289792	5.286754	5.286861	5.288556	5.313197	5.373942	5.313517	5.304633
	OF	2996.348	2996.378	2996.386	3160.309	2996.367	3011.773	3033.212	3006.84	3000.933	3043.037	3148.787	3043.958	3028.259
PV	x1	0.886017	0.20573	0.205728	0.205044	0.19779	0.205728	0.20438	0.212751	0.205952	0.205601	0.228928	0.283135	0.228904
	x2	0.557939	3.470489	3.470516	3.485375	3.526928	3.470516	3.492384	3.346984	3.465724	3.473457	3.273343	2.812744	3.273462
	x3	43.02611	9.036624	9.036654	9.036704	9.817037	9.036654	9.060873	8.981482	9.043723	9.036264	8.612117	7.617417	8.61337
	x4	176.7552	0.20573	0.20573	0.205729	0.216335	0.20573	0.206105	0.219144	0.206017	0.205792	0.232667	0.295529	0.232664
	OF	7181.082	1.724852	1.724862	1.725806	1.945043	1.724862	1.732763	1.809626	1.727966	1.725466	1.819821	2.040803	1.819994

- For pressure vessel design, FOA yielded optimal design variable values of (0.7780271, 0.3845792, 40.312284, 200) and an objective function value of 5882.8955.
- In speed reducer design, FOA achieved superior results with design variable values of (3.5, 0.7, 17, 7.3, 7.8, 3.3502147, 5.2866832) and an objective function value of 2996.3482.
- For welded beam design, FOA obtained optimal design variable values of (0.2057296, 3.4704887, 9.0366239, 0.2057296) and an objective function value of 1.7246798.
- In tension/compression spring design, FOA produced optimal design variable values of (0.0516891, 0.3567177, 11.288966) and an objective function value of 0.0126019.

Analysis of simulation results indicates that FOA consistently outperforms competing algorithms in solving these engineering design challenges. Its capability to effectively explore, exploit, and maintain balance throughout the search process underscores its robust performance in real-world optimization applications.

5. Concluding remarks and future works

This study introduced the Fossa Optimization Algorithm (FOA), a newly developed metaheuristic inspired by the hunting strategies of fossas in the wild. FOA was designed to emulate the fossa's approach to hunting lemurs, which involved initial attacks followed by strategic pursuits through trees. The algorithm was meticulously explained and mathematically modeled to optimize exploration and exploitation during these distinct phases. To evaluate its efficacy, FOA was tested on twenty-two challenging optimization problems sourced from the CEC 2011 test suite, alongside four practical engineering designs. Results highlighted FOA's capability in achieving optimal solutions by effectively balancing exploration, exploitation, and search management. Comparative assessments against twelve prominent metaheuristic algorithms consistently demonstrated FOA's superiority across diverse benchmarks.

Future research directions include developing binary and multi-objective versions of FOA and applying it across diverse scientific and practical domains.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, T.H, B.B, G.B, and F.W; methodology, TH, M.D, and K.E; software, K.E, B.B, and F.W; validation, K.E, M.D, G.B., and Z.M; formal analysis, Z.M, M.D, and K.E; investigation, B.B, Z.M, and F.W; resources, T.H, Z.M, B.B, and G.B; data curation, K.E, G.B, and F.W; writing—original draft preparation, M.D, T.H, and F.W; writing—review and editing, F.W, Z.M, B.B, G.B, and K.E; visualization, K.E; supervision, M.D;

project administration, K.E, T.H, and Z.M; funding acquisition, K.E.

References

[1] A. Taheri *et al.*, "Partial reinforcement optimizer: An evolutionary optimization algorithm", *Expert Systems with Applications*, Vol. 238, p. 122070, 2024, doi: 10.1016/j.eswa.2023.122070.

[2] S. Barua and A. Merabet, "Lévy Arithmetic Algorithm: An enhanced metaheuristic algorithm and its application to engineering optimization", *Expert Systems with Applications*, Vol. 241, p. 122335, 2024, doi: 10.1016/j.eswa.2023.122335.

[3] N. Singh, X. Cao, S. Diggavi, and T. Başar, "Decentralized multi-task stochastic optimization with compressed communications", *Automatica*, Vol. 159, p. 111363, 2024, doi: 10.1016/j.automatica.2023.111363.

[4] D. A. Liñán, G. Contreras-Zarazúa, E. Sánchez-Ramírez, J. G. Segovia-Hernández, and L. A. Ricardez-Sandoval, "A hybrid deterministic-stochastic algorithm for the optimal design of process flowsheets with ordered discrete decisions", *Computers & Chemical Engineering*, Vol. 180, p. 108501, 2024, doi: 10.1016/j.compchemeng.2023.108501.

[5] H. Qawaqneh, "New contraction embedded with simulation function and cyclic (α, β)-admissible in metric-like spaces", *International Journal of Mathematics and Computer Science*, Vol. 15, No. 4, pp. 1029-1044, 2020.

[6] T. Hamadneh, N. Athanasopoulos, and M. Ali, "Minimization and positivity of the tensorial rational Bernstein form", In: *Proc. of 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology*, pp. 474-479, 2019.

[7] M. Dehghani, E. Trojovská, and P. Trojovský, "A new human-based metaheuristic algorithm for solving optimization problems on the base of simulation of driving training process", *Scientific Reports*, Vol. 12, No. 1, p. 9924, 2022, doi: 10.1038/s41598-022-14225-7.

[8] T. Hamadneh and R. Wisniewski, "The Barycentric Bernstein Form for Control Design", In: *Proc. of 2018 Annual American Control Conference (ACC)*, pp. 3738-3743, 2018, doi: 10.23919/ACC.2018.8431599.

[9] F.-A. Zeidabadi, M. Dehghani, P. Trojovský, Š. Hubálovský, V. Leiva, and G. Dhiman, "Archery Algorithm: A Novel Stochastic Optimization Algorithm for Solving Optimization Problems", *Computers, Materials & Continua*, Vol. 72, No. 1, pp. 399-416, 2022.

[10] A.-Q. Tian, F.-F. Liu, and H.-X. Lv, "Snow Geese Algorithm: A novel migration-inspired meta-heuristic algorithm for constrained engineering optimization problems", *Applied Mathematical Modelling*, Vol. 126, pp. 327-347, 2024, doi: 10.1016/j.apm.2023.10.045.

[11] A. Seyyedabbasi and F. Kiani, "Sand Cat swarm optimization: a nature-inspired algorithm to solve global optimization problems", *Engineering with Computers*, Vol. 39, No. 4, pp. 2627-2651, 2023, doi: 10.1007/s00366-022-01604-x.

[12] S. Zhao, T. Zhang, L. Cai, and R. Yang, "Triangulation topology aggregation optimizer: A novel mathematics-based meta-heuristic algorithm for continuous optimization and engineering applications", *Expert Systems with Applications*, Vol. 238, p. 121744, 2024, doi: 10.1016/j.eswa.2023.121744.

[13] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization", *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, pp. 67-82, 1997.

[14] J. Kennedy and R. Eberhart, "Particle swarm optimization", In: *Proc. of ICNN'95 - International Conference on Neural Networks*, Perth, WA, Australia, Vol. 4, pp. 1942-1948, 1995, doi: 10.1109/ICNN.1995.488968.

[15] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 26, No. 1, pp. 29-41, 1996.

[16] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer", *Advances in Engineering Software*, Vol. 69, pp. 46-61, 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[17] P. Trojovský and M. Dehghani, "A new bio-inspired metaheuristic algorithm for solving optimization problems based on walruses behavior", *Scientific Reports*, Vol. 13, No. 1, p. 8775, 2023.

[18] M. Ahmed, M. H. Sulaiman, A. J. Mohamad, and M. Rahman, "Gooseneck barnacle optimization algorithm: A novel nature inspired optimization theory and application", *Mathematics and Computers in Simulation*, Vol. 218, pp. 248-265, 2024, doi: 10.1016/j.matcom.2023.10.006.

[19] A. Majumder, "Termite alate optimization algorithm: a swarm-based nature inspired algorithm for optimization problems", *Evolutionary Intelligence*, Vol. 16, No. 3, pp. 997-1017, 2023, doi: 10.1007/s12065-022-00714-1.

[20] Y. Jiang, Q. Wu, S. Zhu, and L. Zhang, "Orca predation algorithm: A novel bio-inspired algorithm for global optimization problems", *Expert Systems with Applications*, Vol. 188, p. 116026, 2022.

[21] W. Zhao *et al.*, "Electric eel foraging optimization: A new bio-inspired optimizer for engineering applications", *Expert Systems with Applications*, Vol. 238, p. 122200, 2024, doi: https://doi.org/10.1016/j.eswa.2023.122200.

[22] E.-S. M. El-kenawy, N. Khodadadi, S. Mirjalili, A. A. Abdelhamid, M. M. Eid, and A. Ibrahim, "Greylag Goose Optimization: Nature-inspired optimization algorithm", *Expert Systems with Applications*, Vol.

238, p. 122147, 2024, doi: 10.1016/j.eswa.2023.122147.

[23] D. E. Goldberg and J. H. Holland, “Genetic Algorithms and Machine Learning”, *Machine Learning*, Vol. 3, No. 2, pp. 95-99, 1988, doi: 10.1023/A:1022602019183.

[24] R. Storn and K. Price, “Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces”, *Journal of global optimization*, Vol. 11, No. 4, pp. 341-359, 1997.

[25] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing”, *Science*, Vol. 220, No. 4598, pp. 671-680, 1983.

[26] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, “GSA: a gravitational search algorithm”, *Information sciences*, Vol. 179, No. 13, pp. 2232-2248, 2009.

[27] M. Dehghani *et al.*, “A spring search algorithm applied to engineering optimization problems”, *Applied Sciences*, Vol. 10, No. 18, p. 6173, 2020.

[28] M. Dehghani and H. Samet, “Momentum search algorithm: A new meta-heuristic optimization algorithm inspired by momentum conservation law”, *SN Applied Sciences*, Vol. 2, No. 10, pp. 1-15, 2020.

[29] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, “Multi-verse optimizer: a nature-inspired algorithm for global optimization”, *Neural Computing and Applications*, Vol. 27, No. 2, pp. 495-513, 2016.

[30] A. Hatamlou, “Black hole: A new heuristic optimization approach for data clustering”, *Information Sciences*, Vol. 222, pp. 175-184, 2013.

[31] R. V. Rao, V. J. Savsani, and D. Vakharia, “Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems”, *Computer-Aided Design*, vol. 43, No. 3, pp. 303-315, 2011.

[32] I. Matoušová, P. Trojovský, M. Dehghani, E. Trojovská, and J. Kostra, “Mother optimization algorithm: a new human-based metaheuristic approach for solving engineering optimization”, *Scientific Reports*, Vol. 13, No. 1, p. 10312, 2023, doi: 10.1038/s41598-023-37537-8.

[33] S. Das and P. N. Suganthan, “Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems”, *Jadavpur University, Nanyang Technological University, Kolkata*, pp. 341-359, 2010.

[34] B. Kannan and S. N. Kramer, “An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design”, *Journal of Mechanical Design*, Vol. 116, No. 2, pp. 405-411, 1994.

[35] A. H. Gandomi and X.-S. Yang, “Benchmark problems in structural optimization”, *Computational Optimization, Methods and Algorithms*, pp. 259-281, 2011.

[36] E. Mezura-Montes and C. A. C. Coello, “Useful infeasible solutions in engineering optimization with evolutionary algorithms”, In: *Proc. of Mexican International Conference on Artificial Intelligence*, pp. 652-662, 2005.

[37] S. Mirjalili and A. Lewis, “The whale optimization algorithm”, *Advances in Engineering Software*, Vol. 95, pp. 51-67, 2016.