Received: February 22, 2023. Revised: March 13, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Attack-Leave Optimizer: A New Metaheuristic that Focuses on The Guided Search and Performs Random Search as Alternative

Purba Daru Kusuma[1]* **Faisal Candrasyah Hasibuan[1]**

[1]*Computer Engineering, Telkom University, Indonesia*
* Corresponding author's Email: purbodaru@telkomuniversity.ac.id

Abstract: This paper introduces a new metaphor-free metaheuristic called attack-leave optimizer (ALO). As the name suggests, ALO deploys two strategies to find the optimal solution. The central concept of ALO is to intensify guided searches as a required method. Then, a random search is performed only if the guided search fails to improve the current solution. ALO consists of four guided searches and one random search, performed in three phases: two mandatory and one optional. In the first phase, the guided search is conducted with the best global solution as the reference. In the second phase, the guided search is conducted with a randomly selected solution as the reference. The random search is performed in the third phase. Evaluating ALO, it was tested on 23 classic functions and benchmarked against five existing metaheuristics with known shortcomings: Mixed leader-based optimizer (MLBO), slime mould algorithm (SMA), golden search optimizer (GSO), zebra optimization algorithm (ZOA), and coati optimization algorithm (COA). The results indicate that ALO is highly competitive, outperforming MLBO, SMA, GSO, COA, and ZOA in solving 16, 16, 14, 10, and 9 functions respectively, and demonstrating ALO as a promising new metaheuristic.

Keywords: Metaheuristic, optimization, Swam intelligence, Guided search, Random search.

1. Introduction

Metaheuristics are popular methods extensively applied in various optimization fields, including energy, economy, manufacturing, logistics, engineering, and more. For instance, Northern goshawk optimization (NGO) was utilized to determine the optimal allocation (i.e., size and location) of photovoltaic systems (PV), wind turbine systems, distribution generators, and reactive power sources in the radial distribution system [1]. In the energy sector, artificial rabbit optimization (ARO) was employed to optimize the location and rating of passive power filters and solar photovoltaic systems in the radial distribution system [2]. An enhanced whale optimization algorithm (EWOA) was used to improve the speed and accuracy of classification in the groundwater quality monitoring system [3]. A combination of the genetic algorithm (GA) and firefly algorithm (FA) was utilized to optimize the make-span in job-shop scheduling problems [4]. Non-dominated sorting genetic algorithm (NSGA II) was applied to reduce the total travel time and operational cost of the school bus mixed-load route optimization for students [5]. Grey wolf optimization (GWO) was used to optimize the feature selection and improve the accuracy of tumor detection [6].

Nowadays, there is a vast number of metaheuristics available. The majority of these metaheuristics incorporate metaphors as their novel contribution. Various animal behaviors are commonly used as metaphors for shortcomings, including the tunicate swarm algorithm (TSA) [7], GWO [8], marine predator algorithm (MPA) [9], Komodo mlipir algorithm (KMA) [10], chameleon swarm algorithm (CSA) [11], cheetah optimizer (CO) [12], chimp optimization algorithm (ChOA) [13], clouded leopard optimization (CLO) [14], coati optimization algorithm (COA) [15], zebra optimization algorithm (ZOA) [16], NGO [17], pelican optimization algorithm (POA) [18], fennec fox optimization (FFO) [19], raccoon optimization algorithm (ROA) [20], red fox optimization (RFO) [21], remora optimization algorithm (ROA) [22], Siberian tiger optimization (STO) [23], and many others.

Besides animal behavior, several shortcomings of metaheuristics use social activities, traditional games, and leadership as metaphors. Chef-based optimization algorithm (CBOA) [24], election-based optimization algorithm (EBOA) [25], sewing training-based optimization (STBO) [26], and driving training-based optimization (DTBO) [27] are the example of metaheuristics that use the social activity as a metaphor. Football game-based optimization (FBGO) [28], dart game optimization (DGO) [29], and shell game optimization (SGO) [30] are an example of metaheuristics that use the traditional game as a metaphor. Mixed leader-based optimization (MLBO) [31], multi-leader optimization (MLO) [32], and hybrid leader-based optimization (HLBO) [33] are an example of metaheuristics that use the term leader as a metaphor. Fortunately, some metaheuristics, such as average subtraction-based optimization (ASBO) [34], golden search optimization (GSO) [35], and total interaction algorithm (TIA) [36], do not use metaphor and focus on their distinct mechanics for their novel contribution.

Despite the massive development of metaheuristics and the significant number that exists, developing new ones is still exciting and challenging. There are two reasons for this circumstance. First, there is not any metaheuristic can solve various optimization problems with a good result, as stated in the no-free-lunch theory [37]. Even a superior metaheuristic is not superior in solving all problems. In some problems, the quality of the solution is mediocre [25]. In various studies, a metaheuristic with superior performance in solving theoretical problems may face difficulty solving the practical ones. On the other hand, some old metaheuristics outperformed by many shortcomings become competitive in solving practical problems. Second, there are various optimization problems nowadays, and there will be more problems in the future. The circumstances of these problems are also various, such as ample solution space, a massive number of decision variables or dimension, non-convex problems, multiple objectives, and ambiguous [25]. Third, many new metaheuristics are developed by modifying the existing ones or by combining several of them, such as the hybrid pelican Komodo algorithm (HPKA) [38], stochastic Komodo algorithm (SKA) [39], and other similar approaches.

In light of the current circumstances and potential for further development, this study aims to introduce a new metaphor-free metaheuristic, namely attack-leave optimizer (ALO) that prioritizes guided search over random search. Specifically, the proposed algorithm performs two sequential steps of guided searches and only resorts to random search if the agent fails to improve the quality of its current solution after the guided searches. This approach addresses the limitations of existing metaheuristics and their reliance on animal behavior or other metaphors. Instead, it focuses on novel mechanics prioritizing the guided search for more efficient and effective optimization.

Below are the main scientific contributions of this work:

1) ALO is a novel swarm-based metaheuristic that focuses on guided search and incorporates random search when the guided search fails to improve the current solution.
2) ALO performs multiple searches in sequential steps, each with multiple references, making it a unique approach.
3) The performance of ALO is evaluated on a set of 23 classic benchmark functions, demonstrating its effectiveness in solving a diverse range of optimization problems.
4) ALO is compared against five state-of-the-art metaheuristics: MLBO, SMA, GSO, COA, and ZOA, and found to outperform them in terms of solution quality.
5) A hyperparameter analysis is performed to identify the optimal configuration for ALO, providing insights into the dominant strategies for solving different optimization problems.

The remainder of this paper is as follows. A literature review regarding the shortcomings of metaheuristics is provided in section 2. Section 3 presents the mechanics of ALO, consisting of the central concept and formalization. The evaluation of ALO consisting of the benchmark and hyperparameter tests is presented in section 4. The in-depth evaluation regarding the result, complexity, and limitation is discussed in section 5. The conclusion and potential future studies are summarized in section 6.

2. Related works

Metaheuristic is an optimization technique that relies primarily on stochastic and trial-and-error approaches. Through a stochastic approach, the metaheuristic does not trace all possible search space solutions [25]. This circumstance has advantages and disadvantages.

Metaheuristics can reduce computational resources, allowing it to be implemented in systems with limited computational resources. In addition,

Received: February 22, 2023. Revised: March 13, 2023.

metaheuristics can address complex problems with a vast search space or an abundance of decision variables. The disadvantage of this approach is that metaheuristics cannot guarantee the global optimal solution, only a quasi-optimal one [25]. Moreover, metaheuristics may be trapped on the optimal local solution.

Metaheuristic employs a trial-and-error method that improves the solution's quality through an iterative process [25]. It is advantageous to the metaheuristic because it abstracts the problem. Metaheuristics disregard the complexity of the problem it faces so that it can be implemented in various optimization problems. It concerns only on the objectives and constraints. The metaheuristic should identify any problems within the constraints. Then, each time it finds a new solution, the objective function is used to evaluate its quality. Some problems have a single objective, whereas others have multiple objectives. This search process is also constrained by the maximum number of iterations implemented during optimization. The primary issue is that the metaheuristic fails to locate the quasi-optimal solution even when the maximum number of iterations has been reached. This circumstance presents a new challenge for developing new metaheuristics, as it must find a quasi-optimal solution in the case of a low maximum number of iterations.

This trial-and-error approach yields various solutions when a metaheuristic explores the search space. Some metaheuristics employ a strict acceptance strategy in which a new solution replaces an existing one only if it is superior. Others use a non-strict acceptance approach, where the new solution replaces the current one without considering its quality. Some metaheuristics, such as simulated annealing, use a gentle acceptance approach, where worse solutions may be accepted based on a time-based stochastic calculation.

Ironically, some metaheuristics employ a static strategy during the search process. It indicates that the searching strategy is implemented in all circumstances without regard to the quality of the results. The inability of metaheuristics to improve the quality of solutions is a significant issue. Only a handful of metaheuristics, such as the artificial bee colony (ABC), implement an alternative strategy if the original strategy fails to improve. Utilizing a roulette wheel strategy, ABC conducts neighborhood searches [40]. Suppose multiple attempts are unsuccessful, the bee searches at random [40]. KMA is also the example of few metaheuristics that is enriched with adaptive strategy. Different from ABC, KMA modifying the population size rather than implementing different searching strategy when facing different circumstance [10]. Unfortunately, many ineffective metaheuristics do not employ this adaptive strategy.

Numerous metaheuristics were tasked with solving a set of functions representing theoretical optimization problems. Popular and widely employed in the initial presentation of many metaheuristics, the set of 23 classic functions is widely used in the first introduction to these algorithms. In addition, several sets of functions, such as CEC 2015 [16] and CEC 2017 [17], are also utilized as theoretical tests. Table 1 outlines the mechanics and evaluation procedure utilized by several shortcoming metaheuristics. There are 16 shortcoming metaheuristics reviewed in Table 1. Meanwhile, the positioning of the proposed metaheuristic is presented in the last row of Table 1. In Table 1, the main scientific contribution of this work is promoting metaphor-free metaheuristic and adaptive strategy where these contributions become more difficult to find in many shortcoming metaheuristics.

Generally, the metaheuristics reviewed in Table 1 are swarm-based, and many rely on metaphors. By abstracting the metaphors,it apparents that guided search is the core strategy underlying these metaheuristics, using the global best solution, a randomly selected solution, or a randomized solution. Nonetheless, one of the most significant opportunities for developing a new metaheuristic is to design an alternative mechanism to promote adaptability, particularly when the metaheuristic is facing stagnation.

3. Model

ALO is constructed based on several reasons. ALO should focus on intensifying the guided search. ALO should implement a multiple phase-multiple strategy approach as it becomes common in many shortcoming metaheuristics. ALO should not depend on only the global best solution but also the randomized solution as a reference during the guided search. Random search becomes the alternative when the guided search faces stagnation.

ALO consists of three sequential phases. The first and second phases are intended for the guided search, whereas the third is for the random search. The first two phases are required, while the final phase is optional. After executing the guided search in the first and second phases, the third phase is executed if the agent cannot improve its current solution.

The agent will randomly select between two potential guided searches in the first phase. The first possible search drives the current solution closer to or

Table 1. The mechanics of shortcoming metaheuristics and the theoretical test used in their first introduction

No	Metaheuristic	Metaphor	Main Strategy	Alternative Strategy	Theoretical Test
1	COA [15]	coati	guided search toward the global best solution, guided search toward a randomized solution within the search space, and random search within the search space	no	CEC 2011, CEC 2017
2	ZOA [16]	zebra	guided search toward the global best solution, neighborhood search, and guided search toward a randomly selected solution.	no	23 classic functions, CEC 2015, CEC 2017
3	POA [18]	pelican	guided search toward a randomized solution within the search space, and neighborhood search	no	23 classic functions,
4	NGO [17]	Northern goshawk	guided search relative to a randomly selected solution and neighborhood search	no	23 classic functions, CEC 2015, CEC 2017
5	MPA [9]	marine predator	guided search toward the local best solution, guided search toward two randomly selected solutions, and neighborhood search	no	29 functions
7	KMA [10]	komodo	guided search toward better solutions and avoid worse solutions, guided search toward some best solutions, crossover with the best solutions among the population, and neighborhood search.	increase population size	23 classic functions
8	MLBO [31]	leader	guided search relative to the mixture between the global best solution and a randomly selected solution.	no	23 classic functions
9	MLO [32]	leader	guided search toward the resultant of some best solutions and neighborhood search	no	23 classic functions
10	EBOA [25]	election	guided search toward a randomly selected solution among some best solutions or global best solution, and neighborhood search	no	23 classic functions
11	CBOA [24]	cooking training	guided search toward the global best solution, neighborhood search, guided search toward a randomly selected solution among some best solutions, and crossover	no	23 classic functions, CEC 2017
13	TIA [36]	-	guided search relative to all other solutions	no	23 classic functions
15	ASBO [34]	-	guided search relative to the average of best and worst solutions, guided search toward the subtraction of best solution with worst solution, and guided search to avoid the best solution	no	23 classic functions
16	SMA [41]	slime mold	full random search, moving toward the gap between two randomly selected solutions, or neighborhood search	no	CEC 2014
17	this work	-	guided search toward the global best solution and guided search relative to a randomized solution	random search within the search space	23 classic functions

beyond the best global solution. The second possible search causes the global best solution to advance and leave behind the corresponding solution. The first search is essential to find a superior solution between the corresponding solution and the global best solution. The second search, meanwhile, focuses on enhancing the quality of the global best solution. The first phase is about exploitation.

In the second phase, there are also two possible guided searches from which the agent will randomly select one. In the first search, the reference combines the best global solution and a randomly chosen solution from the population. The reference for the second search is the mixture of two randomly selected solutions from the population. The movement represents the avoidance of the reference relative to

Received: February 22, 2023. Revised: March 13, 2023.

the corresponding solution if this reference is superior to the corresponding solution. In contrast, the movement represents avoiding the corresponding solution concerning the reference. The first search can be viewed as a combination of exploration and exploitation. The second search, on the other hand, can be viewed as a directed exploration.

The third phase is only performed if the agent does not improve after guided searches. This random search can be viewed as a full random search rather than a neighborhood search due to the random generation of a solution within the search space rather than in close proximity to the current solution. This phase entails random exploration.

ALO takes a strict approach to acceptance. It indicates that a candidate solution replaces the current solution only if it is superior to the current solution. The primary objective of this strategy is to prevent the agent from selecting the inferior option. However, this strategy harms the likelihood of finding a better solution close to the worse one. The formalization of ALO is presented in algorithm 1. Meanwhile, the detailed processes are performed using Eq. (1) through Eq. (7).

ALO is composed of two steps as a metaheuristic. The first step is the initialization, specified in algorithm 1, lines 2 through 5. The second step is iteration, formalized in lines 6 through 22 of algorithm 1. Lines 8 through 10 describe the first phase. In lines 11 through 15, the second phase is specified. Lines 16 to 19 constitute the official third phase. In the end, line 24 specifies that the global best solution becomes the final solution.

In the initialization step, the solution is generated uniformly within the search space as formalized using Eq. (1). x denotes the solution. U denotes the uniform random. x_{lb} denotes the lower boundary, and x_{ub} denotes the upper boundary of the search space. Then, the global best solution (x_b) is updated using Eq. (2). Eq. (2) represents the strict acceptance approach. f denotes the objective function.

$$x = U(x_{lb}, x_{ub}) \quad (1)$$

$$x_b' = \begin{cases} x, f(x) < f(x_b) \\ x_b, else \end{cases} \quad (2)$$

Eq. (3) to Eq. (7) formalizes the processes used in the iteration step. Eq. (3) states the search chosen in the first guided search where x_c denotes the solution candidate. r_1 denotes the first phase ratio. Eq. (4) indicates the strict acceptance approach in updating the corresponding solution. Eq. (5) states that the reference in the second phase is randomly generated among the current population. Eq. (6) states the option of choosing the reference in the second phase (x_t), whether the mixture of the global best solution and a randomly selected solution or the mixture of two selected solutions. r_2 denotes the second phase ratio. Eq. (7) states the search chosen in the second guided search.

	algorithm 1: attack-leave optimizer
1	**begin**
2	**for** all x in X
3	generate initial solution x using Eq. (1)
4	update x_b using Eq. (2)
5	**end for**
6	**for** t = 1 to t_{max}
7	**for** all x in X
8	perform a guided search using Eq. (3)
9	update x using Eq. (4)
10	update x_b using Eq. (2)
11	generate x_{s1} and x_{s2} using Eq. (5)
12	generate x_t using Eq. (6)
13	perform a guided search using Eq. (7)
14	update x using Eq. (4)
15	update x_b using Eq. (2)
16	**if** improvement fails, **then**
17	generate x_c using Eq. (1)
18	update x using Eq. (4)
19	update x_b using Eq. (2)
20	**end if**
21	**end for**
22	**end for**
23	**end**
24	**output:** x_b

$$x_c = \begin{cases} x + U(0,1)(x_b - 2x), U(0,1) \leq r_1 \\ x_b + U(0,1)(x_b - 2x), else \end{cases} \quad (3)$$

$$x' = \begin{cases} x_c, f(x_c) < f(x) \\ x, else \end{cases} \quad (4)$$

$$x_s = U(X) \quad (5)$$

$$x_t = \begin{cases} \frac{x_b + x_{s1}}{2}, U(0,1) \leq r_2 \\ \frac{x_{s1} + x_{s2}}{2}, else \end{cases} \quad (6)$$

$$x_c = \begin{cases} x_t + U(0,1)(x_t - 2x), f(x_t) < f(x) \\ x + U(0,1)(x - 2x_t), else \end{cases} \quad (7)$$

4. Simulation and result

The test performed to evaluate the proposed ALO, and its results are presented in this section. Two tests were conducted in this work: the benchmark test and the hyperparameters test. The benchmark test was

Table 2. A detailed description of the set of 23 functions.

No	Function	Model	Dim	Space	Target
1	Sphere	$\sum_{i=1}^{d} x_i^2$	35	[-100, 100]	0
2	Schwefel 2.22	$\sum_{i=1}^{d}\lvert x_i\rvert + \prod_{i=1}^{d}\lvert x_i\rvert$	35	[-100, 100]	0
3	Schwefel 1.2	$\sum_{i=1}^{d}\left(\sum_{j=1}^{i} x_j\right)^2$	35	[-100, 100]	0
4	Schwefel 2.21	$\max\{\lvert x_i\rvert, 1 \le i \le d\}$	35	[-100, 100]	0
5	Rosenbrock	$\sum_{i=1}^{d-1}(100(x_{i+1}+x_i^2)^2+(x_i-1)^2)$	35	[-30, 30]	0
6	Step	$\sum_{i=1}^{d-1}(x_i+0.5)^2$	35	[-100, 100]	0
7	Quartic	$\sum_{i=1}^{d} i\, x_i^4 + random\,[0,1]$	35	[-1.28, 1.28]	0
8	Schwefel	$\sum_{i=1}^{d} -x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	35	[-500, 500]	-12,569
9	Ratsrigin	$10d+\sum_{i=1}^{d}\left(x_i^2-10cos(2\pi x_i)\right)$	35	[-5.12, 5.12]	0
10	Ackley	$-20\cdot exp\left(-0.2\cdot\sqrt{\frac{1}{d}\sum_{i=1}^{d} {x_i}^2}\right)-$ $exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos 2\pi x_i\right)+20+exp(1)$	35	[-32, 32]	0
11	Griewank	$\frac{1}{4000}\sum_{i=1}^{d} x_i^2-\prod_{i=1}^{d} cos\left(\frac{x_i}{\sqrt{i}}\right)$+1	35	[-600, 600]	0
12	Penalized	$\frac{\pi}{d}\left\{10\sin(\pi y_1)+\sum_{i=1}^{d-1}\left((y_i-1)^2(1+\right.\right.$ $\left.\left.10\, sin^2(\pi y_{i+1}))\right)+(y_d-1)^2\right\}+\sum_{i=1}^{d} u(x_i,10,100,4)$	35	[-50, 50]	0
13	Penalized 2	$0.1\left\{sin^2(3\pi x_1)+\sum_{i=1}^{d-1}\left((x_i-1)^2(1+\right.\right.$ $\left.\left.sin^2(3\pi x_{i+1}))\right)+(x_d-1)^2(1+sin^2(2\pi x_d))\right\}+$ $\sum_{i=1}^{d} u(x_i,5,100,4)$	35	[-50, 50]	0
14	Shekel Foxholes	$\left(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right)^{-1}$	2	[-65, 65]	1
15	Kowalik	$\sum_{i=1}^{11}\left(a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}\right)^2$	4	[-5, 5]	0.0003
16	Six Hump Camel	$4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	[-5, 5]	-1.0316
17	Branin	$\left(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos(x_1)+10$	2	[-5, 5]	0.398
18	Goldstein-Price	$\left(1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+\right.$ $\left.6x_1x_2+3x_2^2)\right).\left(30+(2x_1-3x_2)^2(18-32x_1+\right.$ $\left.12x_1^2+48x_2-36x_1x_2+27x_2^2)\right)$	2	[-2, 2]	3
19	Hartman 3	$-\sum_{i=1}^{4}\left(c_i exp\left(-\sum_{j=1}^{d}\left(a_{ij}(x_j-p_{ij})^2\right)\right)\right)$	3	[1, 3]	-3.86
20	Hartman 6	$-\sum_{i=1}^{4}\left(c_i exp\left(-\sum_{j=1}^{d}\left(a_{ij}(x_j-p_{ij})^2\right)\right)\right)$	6	[0, 1]	-3.32
21	Shekel 5	$-\sum_{i=1}^{5}\left(\sum_{j=1}^{d}(x_j-c_{ji})^2+\beta_i\right)^{-1}$	4	[0, 10]	-10.153
22	Shekel 7	$-\sum_{i=1}^{7}\left(\sum_{j=1}^{d}(x_j-c_{ji})^2+\beta_i\right)^{-1}$	4	[0, 10]	-10.402
23	Shekel 10	$-\sum_{i=1}^{10}\left(\sum_{j=1}^{d}(x_j-c_{ji})^2+\beta_i\right)^{-1}$	4	[0, 10]	-10.536

Received: February 22, 2023. Revised: March 13, 2023.

Table 3. Fitness score comparison in solving high-dimension unimodal functions.

F	Parameter	MLBO [31]	SMA [41]	GSO [35]	COA [15]	ZOA [16]	ALO
1	mean	1.1041×10^4	2.7774×10^4	1.0582×10^4	0.0000	0.0000	0.0000
	st dev	2.0277×10^3	6.3390×10^3	2.3287×10^3	0.0000	0.0000	0.0000
	min	6.5627×10^3	1.4393×10^4	6.2995×10^3	0.0000	0.0000	0.0000
	max	1.4513×10^4	3.7251×10^4	1.4835×10^4	0.0001	0.0000	0.0000
	mean rank	5	6	4	1	1	1
2	mean	3.1565×10^{20}	0.0000	0.0000	0.0000	0.0000	0.0000
	st dev	1.4805×10^{21}	0.0000	0.0000	0.0000	0.0000	0.0000
	min	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	max	6.9443×10^{21}	0.0000	0.0000	0.0000	0.0000	0.0000
	mean rank	6	1	1	1	1	1
3	mean	2.3327×10^4	8.3962×10^4	1.9500×10^4	3.6800×10^1	0.0145	0.0000
	st dev	9.8174×10^3	2.1132×10^4	6.6162×10^3	4.4598×10^1	0.0274	0.0000
	min	1.1874×10^4	4.8148×10^4	1.0697×10^4	0.0241	0.0000	0.0000
	max	5.0327×10^4	1.2873×10^5	3.8757×10^4	1.7670×10^2	0.1017	0.0000
	mean rank	5	6	4	3	2	1
4	mean	3.7564×10^1	6.7273×10^1	3.7849×10^1	0.0451	0.0000	0.0000
	st dev	4.9217	6.8186	3.8682	0.0276	0.0000	0.0000
	min	2.7736×10^1	4.0000×10^1	3.1105×10^1	0.0107	0.0000	0.0000
	max	4.8563×10^1	7.5000×10^1	4.4781×10^1	0.1009	0.0000	0.0000
	mean rank	4	6	5	3	1	1
5	mean	7.0458×10^6	6.8417×10^7	7.5194×10^6	3.3933×10^1	3.3904×10^1	3.3929×10^1
	st dev	3.4508×10^6	2.8784×10^7	4.4168×10^6	0.0575	0.0458	0.0250
	min	1.9697×10^6	7.3345×10^6	2.7617×10^6	3.3694×10^1	3.3794×10^1	3.3881×10^1
	max	1.4708×10^7	1.1583×10^8	2.0912×10^7	3.3975×10^1	3.3948×10^1	3.3960×10^1
	mean rank	4	6	5	3	1	2
6	mean	1.0211×10^4	2.3517×10^4	1.0791×10^4	6.9839	6.3068	6.8256
	st dev	2.1187×10^3	6.6797×10^3	2.6193×10^3	0.4775	0.5981	0.3318
	min	6.2812×10^3	1.3597×10^4	7.2930×10^3	5.7624	5.2201	6.1439
	max	1.4570×10^4	3.7446×10^4	1.7229×10^4	7.9519	7.3329	7.6304
	mean rank	4	6	5	3	1	2
7	mean	4.2438	2.1004×10^2	3.9706	0.0208	0.0051	0.0075
	st dev	3.0119	3.2388×10^1	1.6190	0.0096	0.0026	0.0047
	min	0.8877	1.5617×10^2	1.4139	0.0057	0.0011	0.0000
	max	1.4507×10^1	2.8647×10^2	7.1755	0.0481	0.0106	0.0166
	mean rank	5	6	4	3	1	2

performed to compare the performance of ALO with other metaheuristics. In contrast, the hyperparameters test evaluated the sensitivity of ALO's adjusted parameters to improve the final solution quality. In this test, decimal points less than 10^{-4} were rounded to 10^{-4}.

The optimization problem used in the tests consisted of a set of 23 functions, which can be divided into three groups: high-dimensional unimodal functions (functions 1-7), high-dimensional multimodal functions (functions 8-13), and fixed-dimensional multimodal functions (functions 14-23). A detailed description of this set of functions is presented in Table 2.

The first test benchmarked ALO against five other metaheuristics: MLBO, Slime Mold Algorithm (SMA), GSO, COA, and ZOA. MLBO, COA, and ZOA are metaheuristics that implement the strict acceptance approach, while SMA and GSO do not. The population size was generally set to 10, and the maximum iteration was set to 35. No other parameters needed to be set for MLBO, COA, and ZOA. Meanwhile, in SMA, variable z was set to 0.3, and both ratios in ALO were set to 0.5. The comparison results for the first, second, and third groups of functions are presented in Tables 3, 4, and 5, respectively. Each table includes five indicators related to the fitness score: mean, standard deviation, minimum, maximum, and mean rank.

Table 3 indicates the superiority of ALO among other metaheuristics. ALO is four times (Sphere, Schwefel 2.22, Schwefel 1.2, and Schwefel 2.21) on the first and three times on the second (Rosenbrock, Step, and Quartic). Meanwhile, other metaheuristics are also on the first rank in solving three functions

Table 4. Fitness score comparison in solving high-dimension multimodal functions

F	Parameter	MLBO [31]	SMA [41]	GSO [35]	COA [15]	ZOA [16]	ALO
8	mean	-3.7040×10^3	-5.4247×10^3	-3.8952×10^3	-5.0538×10^3	-2.6241×10^3	-3.8520×10^3
	st dev	7.0996×10^2	3.9399×10^2	7.0568×10^2	5.6921×10^2	4.7529×10^2	4.7839×10^2
	min	-5.7019×10^3	-6.3320×10^3	-5.4766×10^3	-6.4397×10^3	-3.5720×10^3	-5.2734×10^3
	max	-2.8687×10^3	-4.7354×10^3	-2.7333×10^3	-3.9613×10^3	-1.8446×10^3	-3.1506×10^3
	mean rank	5	1	3	2	6	4
9	mean	2.5086×10^2	8.5678×10^1	2.5306×10^2	0.0005	0.0000	0.0000
	st dev	3.0289×10^1	1.9417×10^1	2.7290×10^1	0.0011	0.0000	0.0000
	min	1.8483×10^2	4.6002×10^1	2.0644×10^2	0.0000	0.0000	0.0000
	max	3.0346×10^2	1.2701×10^2	2.9256×10^2	0.0050	0.0000	0.0000
	mean rank	5	4	6	3	1	1
10	mean	1.5013×10^1	1.5605×10^1	1.5180×10^1	0.0014	0.0000	0.0000
	stdev	0.7470	0.5009	0.8745	0.0008	0.0000	0.0000
	min	1.3615×10^1	1.4626×10^1	1.3596×10^1	0.0005	0.0000	0.0000
	max	1.6308×10^1	1.6596×10^1	1.6695×10^1	0.0036	0.0000	0.0000
	mean rank	4	6	5	3	1	1
11	mean	9.7373×10^1	2.2982×10^2	9.7472×10^1	0.0278	0.0015	0.0000
	st dev	2.7975×10^1	5.9770×10^1	2.1867×10^1	0.1013	0.0070	0.0000
	min	5.7267×10^1	1.2684×10^2	5.5135×10^1	0.0000	0.0000	0.0000
	max	1.6675×10^2	3.4254×10^2	1.4170×10^2	0.4713	0.0330	0.0000
	mean rank	4	6	5	3	2	1
12	mean	1.0532×10^6	1.0750×10^8	2.5069×10^6	0.4892	0.8156	0.8798
	st dev	1.4062×10^6	5.0940×10^7	2.3589×10^6	0.1332	0.1030	0.1719
	min	1.7242×10^4	2.2288×10^6	5.5821×10^4	0.2634	0.6480	0.5105
	max	6.3908×10^6	2.3493×10^8	7.7657×10^6	0.6993	0.9746	1.1976
	mean rank	4	6	5	1	2	3
13	mean	1.0431×10^7	2.4634×10^8	1.2590×10^7	3.0740	2.9665	3.1046
	st dev	7.4544×10^6	1.1841×10^8	1.0165×10^7	0.0845	0.1183	0.0418
	min	4.8351×10^5	7.0763×10^7	2.0530×10^6	2.8471	2.6066	3.0017
	max	3.1107×10^7	5.4621×10^8	4.6446×10^7	3.2427	3.1281	3.1411
	mean rank	4	6	5	2	1	3

Table 5. Fitness score comparison in solving fixed dimension multimodal functions

F	Parameter	MLBO [31]	SMA [41]	GSO [35]	COA [15]	ZOA [16]	ALO
14	mean	6.8537	3.3819	5.3771	3.4697	8.4571	1.8140
	st dev	5.4470	2.4416	2.8177	2.6787	3.4077	1.1451
	min	0.9980	0.9980	1.0132	0.9980	0.9981	0.9980
	max	2.1988×10^1	7.8740	1.1719×10^1	1.0763×10^1	1.2670×10^1	4.9667
	mean rank	5	2	4	3	6	1
15	mean	0.0093	0.1065	0.0114	0.0032	0.0039	0.0028
	st dev	0.0081	0.0401	0.0077	0.0060	0.0141	0.0050
	min	0.0014	0.0220	0.0017	0.0003	0.0003	0.0006
	max	0.0224	0.1484	0.0245	0.0222	0.0667	0.0227
	mean rank	4	6	5	2	3	1
16	mean	-1.0270	-0.0530	-1.0178	-1.0316	-0.9923	-1.0267
	st dev	0.0077	0.2052	0.0257	0.0000	0.0853	0.0053
	min	-1.0316	-0.9618	-1.0316	-1.0316	-1.0316	-1.0316
	max	-1.0054	0.0000	-0.9459	-1.0313	-0.6957	-1.0128
	mean rank	2	6	4	1	5	3
17	mean	0.3996	0.6347	0.4023	0.3981	0.8492	0.4463
	st dev	0.0041	0.0426	0.0068	0.0000	0.6775	0.0465
	min	0.3981	0.4439	0.3981	0.3981	0.3981	0.4064
	max	0.4170	0.6438	0.4217	0.3984	2.6056	0.5938
	mean rank	2	5	3	1	6	4
18	mean	5.5318	3.0000	3.0957	3.2568	1.7196×10^1	3.8003
	st dev	7.8480	0.0000	0.1929	1.1970	1.9873×10^1	1.3374
	min	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000

	max	3.3764x10^{1}	3.0000	3.5877	8.6160	8.4000x10^{1}	8.7115
	mean rank	5	1	2	3	6	4
19	mean	-0.0495	-0.0495	-0.0495	-0.0495	-0.0495	-0.0495
	st dev	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	min	-0.0495	-0.0495	-0.0495	-0.0495	-0.0495	-0.0495
	max	-0.0495	-0.0495	-0.0495	-0.0495	-0.0495	-0.0495
	mean rank	1	1	1	1	1	1
20	mean	-3.0108	-1.1433	-3.0897	-3.2282	-2.7282	-2.7893
	st dev	0.2269	0.6027	0.1566	0.0639	0.3590	0.2359
	min	-3.1926	-2.6914	-3.3056	-3.3127	-3.2005	-3.1180
	max	-2.1833	-0.2985	-2.6743	-3.1074	-1.8374	-2.2192
	mean rank	3	6	2	1	5	4
21	mean	-3.0156	-2.8450	-3.9857	-8.6455	-4.8695	-2.3631
	st dev	1.9124	3.0227	2.3509	1.4844	2.3500	1.0256
	min	-9.3354	-1.0153x10^{1}	-9.0130	-1.0126x10^{1}	-8.4505	-4.9718
	max	-1.0106	-0.5099	-1.3702	-4.8191	-1.7392	-0.7532
	mean rank	4	5	3	1	2	6
22	mean	-4.2014	-5.2307	-4.5314	-8.4132	-4.8429	-2.9188
	st dev	2.9412	3.9320	2.6870	1.5438	2.3781	1.3791
	min	-1.0221x10^{1}	-1.0403x10^{1}	-9.2841	-1.0332x10^{1}	-9.1936	-7.5892
	max	-1.5689	-0.6591	-2.0634	-5.0096	-1.7021	-1.2333
	mean rank	5	2	4	1	3	6
23	mean	-3.3278	-3.8977	-3.6730	-8.6204	-4.0117	-2.4825
	st dev	1.9788	2.6000	1.8425	1.6747	2.1175	0.6174
	min	-1.0039x10^{1}	-1.0536x10^{1}	-8.4964	-1.0484x10^{1}	-1.0176x10^{1}	-4.6240
	max	-1.4031	-1.4314	-1.8474	-4.0317	-1.8541	-1.8307
	mean rank	5	3	4	1	2	6

(Sphere, Schwefel 2.22, and Schwefel 2.21). This result indicates fierce competition in solving the high-dimension unimodal functions.

The competition for solving high-dimensional unimodal functions can be divided into two groups according to the proximity of the results. Three metaheuristics comprise the first group: ALO, COA, and ZOA. Also comprising three metaheuristics, the second group includes MLBO, SMA, and GSO. The metaheuristics in the first group produce superior final solutions than those in the second group. In general, this gap is substantial, and it exists in nearly all functions. In all functions belonging to the first group, MLBO, SMA, and GSO belong to the best of the three.

Table 3 also indicates that ALO can find the optimal global solution or is near the optimal global solution. ALO can find the global optimal in solving four functions (Sphere, Schwefel 2.22, Schwefel 1.2, Schwefel 2.21, and Quartic). Meanwhile, the average fitness score produced by ALO is near the global optimal in solving Quartic. Moreover, the minimum fitness score of ALO is the global optimal in solving Quartic. The average fitness score produced by ALO is not so near to the global optimal in solving Rosenbrock and Step. However, this score is still much better than the average fitness score produced by MLBO, SMA, and GSO.

Table 4 indicates that ALO can solve multimodal functions with high dimensions. ALO ranks first in the solution of three functions (Rastrigin, Ackley, and Griewank), third in the solution of two functions (Penalized and Penalized 2), and fourth in the solution of one function (Schwefel). ZOA also holds the top spot for solving Rastrigin and Ackley in this group. Similar to the first group, these six metaheuristics can be divided into two groups within this second group. The members of the first group are ALO, ZOA, and COA.

In contrast, the second group includes MLBO, SMA, and GSO. In solving Schwefel, the performance gap between the first and second groups is narrow. In solving other functions, however, the performance gap between the first and second groups is substantial.

Table 5 shows intense competition among the metaheuristics in solving the fixed dimension multimodal functions. ALO achieved the first rank in solving three functions (Shekel Foxholes, Kowalik, and Hartman 3), the third rank in solving one function (Six Hump Camel), the fourth rank in solving three functions (Branin, Goldstein-Price, and Hartman 6), and the sixth rank in solving three functions (Shekel 5, Shekel 7, and Shekel 10). The narrow gap in the metaheuristics' average fitness scores shows the

Table 6. Group-based superiority of ALO

Group	Number of Functions Where ESCO is Better				
	MLBO [31]	SMA [41]	GSO [35]	COA [15]	ZOA [16]
1	7	6	6	5	1
2	6	5	5	3	2
3	3	5	3	2	6
Total	16	16	14	10	9

Table 7. Relation between r_1 and the average fitness score

F	Average Fitness Score		Which r_1 is Significantly Better?
	$r_1 = 0.2$	$r_1 = 0.8$	
1	0.0000	0.0000	none
2	0.0000	0.0000	none
3	0.0000	0.0000	none
4	0.0000	0.0000	none
5	3.3952×10^1	3.3924×10^1	none
6	6.9768	6.8266	none
7	0.0131	0.0062	low
8	-3.4881×10^3	-3.7167×10^3	none
9	0.0000	0.0000	none
10	0.0000	0.0000	none
11	0.0000	0.0000	none
12	0.8967	0.9174	none
13	3.1156	3.0711	none
14	1.9044	3.6088	high
15	0.0051	0.0035	none
16	-1.0279	-1.0236	none
17	0.4746	0.4692	none
18	3.5931	4.0419	none
19	-0.0495	-0.0495	none
20	-2.8029	-2.8508	none
21	-2.6516	-3.2277	none
22	-2.7144	-3.2269	none
23	-2.5167	-2.6192	none

intense competition among them. In addition, all metaheuristics achieved the same average fitness score in solving Hartman 3. The wider gap was observed in Shekel Foxholes, Shekel 5, Shekel 7, and Shekel 10, where a metaheuristic showed a significantly better performance than the others.

Tables 3 to 5 show an intense competition among metaheuristics, particularly between ALO and the other five. ALO performs better than MLBO, especially in solving high-dimensional functions. ALO is superior to MLBO in solving high-dimensional unimodal and multimodal functions. However, ALO is inferior to MLBO in solving fixed-dimension multimodal functions. ALO outperforms SMA in almost all high-dimensional functions, unimodal or multimodal. At the same time, ALO is slightly superior to SMA in solving fixed-dimension multimodal functions. ALO is also superior to GSO,

Table 8. Relation between r_2 and the average fitness score

F	Average Fitness Score		Which r_2 is Significantly Better?
	$r_2 = 0.2$	$r_2 = 0.8$	
1	0.0000	0.0000	none
2	0.0000	0.0000	none
3	0.0000	0.0000	none
4	0.0000	0.0000	none
5	3.3909×10^1	3.3907×10^1	none
6	6.6657	6.7970	none
7	0.0076	0.0067	none
8	-3.7820×10^3	-3.6294×10^3	none
9	0.0000	0.0000	none
10	0.0000	0.0000	none
11	0.0068	0.0000	none
12	0.9029	0.8750	none
13	3.0887	3.1032	none
14	2.2824	2.8987	none
15	0.0034	0.0045	none
16	-1.0247	-1.0281	none
17	0.4468	0.4700	none
18	3.7711	3.3609	none
19	-0.0495	-0.0495	none
20	-2.7828	-2.7737	none
21	-2.4600	-2.7652	none
22	-2.7639	-2.8616	none
23	-2.6183	-2.7476	none

especially in solving high dimensional functions, where it outperforms GSO in almost all these functions. ALO is comparable to COA, as ALO performs better than COA in 10 functions, is equal in 3 functions, and is worse in three functions. The superiority of ALO over COA is primarily observed in solving high dimensional unimodal functions. Finally, ALO is slightly superior to ZOA, as it performs better than ZOA in 9 functions, is equal in 6 functions, and is worse than ZOA in 8.

The second test is the hyperparameters evaluation. Two sub-tests are performed in this work. The first subtest aims to determine the relationship between the first ratio and the mean fitness score. The objective of the second subtest is to determine the relationship between the second ratio and the average fitness score. Each subtest contains two values representing a low-ratio and a high-ratio scenario. Table 7 displays the outcome of the first subtest. In the meantime, the results of the second subtest are shown in Table 8.

Table 7 indicates that, in general, there is no significant influence of the first ratio on the performance of the proposed ALO. In solving almost all functions, there is no significant performance difference between the low first ratio scenario and the high first ratio scenario. The low first ratio scenario is significantly better in solving Quartic. On the other

Received: February 22, 2023. Revised: March 13, 2023.

hand, the high first ratio scenario is significantly better in solving Shekel Foxholes.

Table 8 indicates that there is not any significant difference between the low r_2 scenario and the high r_2 scenario. This circumstance occurs in all functions, whether high-dimension unimodal, high-dimension multimodal, or fixed-dimension multimodal functions.

5. Discussion

This section discusses significant aspects of this work, such as ALO, the results, and metaheuristics in general. ALO has demonstrated intense exploitation and exploration capabilities. Its exploitation capability has been evaluated based on its performance in solving unimodal functions [10], while its exploration capability has been evaluated based on its performance in solving multimodal functions [10]. In solving unimodal functions, ALO has shown the ability to find the optimal global solution in four functions and quasi-optimal solutions in three functions. ALO has found the optimal global solution in three functions and quasi-optimal solutions for high-dimensional multimodal functions in two out of six evaluated functions. Additionally, for fixed-dimension multimodal functions, ALO has found six quasi-optimal solutions.

The benchmark test result indicates the competitiveness of ALO compared to the benchmark metaheuristics. ALO is superior to MLBO, SMA, and GSO, especially in solving high-dimension functions. ALO is still competitive compared to ZOA and COA as the newest metaheuristics among these five benchmarks. The fierce competition takes place in solving the fixed dimension multimodal functions. In this group, the performance gap among these metaheuristics is narrow.

This result indicates two essential findings. First, this result strengthens the no-free-lunch theory [24]. ALO is not superior in solving all these 23 functions. On the other hand, although MLBO, SMA, and GSO fall behind in solving high-dimension functions, they are very competitive in solving fixed-dimension multimodal functions.

The hyperparameters test indicates essential findings regarding the dominant strategy deployed in ALO. The less significance of the r_1 ratio indicates the less significance of starting point of the guided search relative to the global best solution. Whether this guided search starts from the corresponding solution or the global best solution is not essential. The more important thing is the existence of the guided search toward the global best solution as it is also deployed in various metaheuristics. The less significance of the r_2 ratio also indicates that the mixture of the randomly generated reference in the second guided search is not essential. It is not important whether this second target is the mixture of the global best solution and a randomly selected solution or the mixture of two selected solutions. The more important thing is also the existence of the guided search relative to a randomized reference.

In ALO, several parameters contribute to its complexity. They determine the looping process in the iteration phase. The outer loop runs until the maximum iteration is reached. The intermediate loop iterates over the entire population, while the inner loop iterates over all decision variables (also known as the dimension). The number of solutions in the population and the number of moves or jumps in each iteration also affect the algorithm's complexity. They can be adjusted to balance the algorithm's exploration and exploitation capabilities and convergence speed. The algorithm's complexity is also affected by the optimized function, affecting the number of function evaluations required to find the optimal solution. Meanwhile, there are three steps performed inside the looping. Based on this explanation, the algorithm complexity of ALO can be presented as $O(3t_{max}.n(X).n(d))$.

Regarding this work and the proposed metaheuristics, there are limitations. The set of 23 functions is selected as the theoretical problems. Other sets of functions, such as CEC 2015 [16] or CEC 2017 [15] can be used to evaluate a variety of metaheuristics. In addition, ALO has not been tested to solve real-world optimization problems. ALO only supports five strategies, including four guided and one random search. Many strategies have not yet been accommodated due to the inherent inability of metaheuristics to deal with diverse strategies. They can serve as a baseline for future development.

6. Conclusion

This work presents the concept, formalization, and evaluation of a new metaheuristic known as an attack-leave optimizer (ALO). It is intended to move toward the reference and abandon the current solution if it is no longer productive. ALO has been tested to solve a theoretical optimization problem with a set of 23 classic functions as the problem. This test demonstrates the effectiveness of ALO in locating global optimal or quasi-optimal solutions. It demonstrates that ALO has excellent exploration and exploitation abilities. ALO is competitive among the benchmarked metaheuristics, outperforming MLBO, SMA, GSO, COA, and ZOA in the solution of 16, 16, 14, 10, and 9 functions respectively.

Proposing ALO opens up several avenues for future research. ALO should be further tested to solve many real-world optimization problems. Additionally, the competition between ALO and two other metaheuristics, COA and ZOA, can be leveraged to explore hybridization strategies that combine the strengths of these algorithms to create more powerful ones.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, Kusuma; methodology, Kusuma; software, Kusuma; formal analysis, Kusuma and Hasibuan; investigation, Kusuma and Hasibuan; data curation, Kusuma; writing-original paper draft, Kusuma; writing-review and editing: Hasibuan; supervision: Hasibuan; funding acquisition, Kusuma.

Acknowledgments

This work is funded and supported by Telkom University, Indonesia.

References

[1] M. Manohara, V. C. V. Reddy, and M. V. Kumar, "Northern Goshawk Optimization for Optimal Allocation of Multiple Types of Active and Reactive Power Distribution Generation in Radial Distribution Systems for Techno-Environmental Benefits", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 1, pp. 91-99, 2023, doi: 10.22266/ijies2023.0228.08.

[2] C. R. Rao, R. Balamurugan, and R. K. R. Alla, "Artificial Rabbits Optimization Based Optimal Allocation of Solar Photovoltaic Systems and Passive Power Filters in Radial Distribution Network for Power Quality Improvement", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 1, pp. 100-109, 2023, doi: 10.22266/ijies2023.0228.09.

[3] N. D. S. S. K. Relangi, A. Chaparala, and R. Sajja, "Effective Groundwater Quality Classification Using Enhanced Whale Optimization Algorithm with Ensemble Classifier", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 1, pp. 214-223, 2023, doi: 10.22266/ijies2023.0228.19.

[4] C. E. Nugraheni, D. Swastiani, and L. Abednego, "Hybrid Metaheuristics for Job Shop Scheduling Problems", *Engineering Letters*, Vol. 30, No. 4, pp. 1444-1451, 2022.

[5] M. Zhao, Y. Zhao, C. Ma, and X. Wu, "Robust Optimization of Mixed-Load School Bus Route Based on Multi-Objective Genetic Algorithm", *Engineering Letters*, Vol. 30, No. 4, pp. 1543-1550, 2022.

[6] K. D. S. Devi, P. Karthikeyan, U. Moorthy, K. Deeba, V. Maheshwari, and S. M. Allayear, "Tumor Detection on Microarray Data Using Grey Wolf Optimization with Gain Information", *Mathematical Problems in Engineering*, Vol. 2022, ID. 4092404, pp. 1-10, 2022.

[7] S. Kaur, L. K. Awasthi, A. L. Sangal, and G. Dhiman, "Tunicate Swarm Algorithm: A New Bio-inspired based Metaheuristic Paradigm for Global Optimization", *Engineering Applications of Artificial Intelligence*, Vol. 90, ID. 103541, 2020.

[8] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer" , *Advances in Engineering Software*, Vol. 69, pp. 46–61, 2014.

[9] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine Predators Algorithm: A Nature-inspired Metaheuristic", *Expert Systems with Applications*, Vol. 152, ID: 113377, 2020.

[10] S. Suyanto, A. A. Ariyanto, and A. F. Ariyanto, "Komodo Mlipir Algorithm", *Applied Soft Computing*, Vol. 114, pp. 1–17, 2022.

[11] M. S. Braik, "Chameleon Swarm Algorithm: A Bio-inspired Optimizer for Solving Engineering Design Problems", *Expert Systems with Applications*, Vol. 174, ID. 114685, pp. 1-25, 2021.

[12] M. A. Akbari, M. Zare, R. A. Abarghooee, S. Mirjalili, and M. Deriche, "The Cheetah Optimizer: A Nature-inspired Metaheuristic Algorithm for Large-scale Optimization Problems", *Scientific Reports*, Vol. 12, ID. 10953, pp. 1-20, 2022.

[13] M. Khishe and M. R. Mosavi, "Chimp Optimization Algorithm", *Expert Systems with Applications*, Vol. 149, ID. 113338, pp. 1-26, 2020.

[14] E. Trojovska and M. Dehghani, "Clouded Leopard Optimization: A New Nature-Inspired Optimization Algorithm", *IEEE Access*, Vol. 10, pp. 102876-102906, 2022.

[15] M. Dehghani, Z. Montazeri, E. Trojovska, and P. Trojovsky, "Coati Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems", *Knowledge-Based Systems*, Vol. 259, ID. 110011, pp. 1-43, 2023.

[16] E. Trojovska, M. Dehghani, and P. Trojovsky, "Zebra Optimization Algorithm: A New Bio-Inspired Optimization Algorithm for Solving Optimization Algorithm", *IEEE Access*, Vol. 10, pp. 49445-49473, 2022.

[17] M. Dehghani, S. Hubalovsky, and P. Trojovsky, "Northern Goshawk Optimization: A New Swarm-Based Algorithm for Solving Optimization Problems", *IEEE Access*, Vol. 9, pp. 162059–162080, 2021.

[18] P. Trojovsky and M. Dehghani, "Pelican Optimization Algorithm: A Novel Nature-Inspired Algorithm for Engineering Applications", *Sensors*, Vol. 22, ID. 855, pp. 1-34, 2022.

[19] E. Trojovska, M. Dehghani, and P. Trojovsky, "Fennec Fox Optimization: A New Nature-Inspired Optimization Algorithm", *IEEE Access*, Vol. 10, pp. 84417-84443, 2022.

[20] S. Z. Koohi, N. A. W. A. Hamid, M. Othman, and G. Ibragimov, "Raccoon Optimization Algorithm", *IEEE Access*, Vol. 7, pp. 5383-5399, 2019.

[21] D. Polap and M. Wozniak, "Red Fox Optimization Algorithm", *Expert Systems with Applications*, Vol. 166, ID. 114107, pp. 1-21, 2021.

[22] H. Jia, X. Peng, and C. Lang, "Remora Optimization Algorithm", *Expert Systems with Applications*, Vol. 185, ID. 115665, pp. 1-25, 2021.

[23] P. Trojovsky, M. Dehghani, and P. Hanus, "Siberian Tiger Optimization: A New Bio-Inspired Metaheuristic Algorithm for Solving Engineering Optimization Problems", *IEEE Access*, Vol. 10, pp. 132396-132431, 2022.

[24] E. Trojovska and M. Dehghani, "A New Human-based Metaheuristic Optimization Method based on Mimicking Cooking Training", *Scientific Reports*, Vol. 12, ID. 14861, pp. 1-24, 2022.

[25] P. Trojovský and M. Dehghani, "A New Optimization Algorithm based on Mimicking the Voting Process for Leader Selection", *PeerJ Computer Science*, Vol. 8, ID. e976, pp. 1-40, 2022.

[26] M. Dehghani, E. Trojovska, and T. Zuscak, "A New Human-Inspired Metaheuristic Algorithm for Solving Optimization Problems based on Mimicking Sewing Training", *Scientific Reports*, Vol. 12, ID. 17387, pp. 1-24, 2022.

[27] 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, pp. 1–21, 2022.

[28] M. Dehghani, M. Mardaneh, J. S. Guerrero, O. P. Malik, and V. Kumar, "Football Game Based Optimization: An Application to Solve Energy Commitment Problem", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 5, pp. 514-523, 2020, doi: 10.22266/ijies2020.1031.45.

[29] M. Dehghani, Z. Montazeri, H. Givi, J. M. Guerrero, and G. Dhiman, "Darts Game Optimizer: A New Optimization Technique Based on Darts Game", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 5, pp. 286–294, 2020, doi: 10.22266/ijies2020.1031.26.

[30] M. Dehghani, Z. Montazeri, O. P. Malik, H. Givi, and J. M. Guerrero, "Shell Game Optimization: A Novel Game-Based Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 3, pp. 246-255, 2020, doi: 10.22266/ijies2020.0630.23.

[31] F. Zeidabadi, S. Doumari, M. Dehghani, and O. P. Malik, "MLBO: Mixed Leader Based Optimizer for Solving Optimization Problems", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 4, pp. 472–479, 2021, doi: 10.22266/ijies2021.0831.41.

[32] M. Dehghani, Z. Montazeri, A. Dehghani, R. A. R. Mendoza, H. Samet, J. M. Guerrero, and G. Dhiman, "MLO: Multi Leader Optimizer", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 6, pp. 364–373, 2020, doi: 10.22266/ijies2020.1231.32.

[33] M. Dehghani and P. Trojovsky, "Hybrid Leader Based Optimization: A New Stochastic Optimization Algorithm for Solving Optimization Applications", *Scientific Reports*, Vol. 12, ID: 5549, pp. 1-16, 2022.

[34] M. Dehghani, S. Hubalovsky, and P. Trojovsky, "A New Optimization Algorithm based on Average and Subtraction of the Best and Worst Members of the Population for Solving Various Optimization Problems", *PeerJ Computer Science*, Vol. 8, ID: e910, pp. 1-29, 2022.

[35] M. Noroozi, H. Mohammadi, E. Efatinasab, A. Lashgari, M. Eslami, and B. Khan, "Golden Search Optimization Algorithm", *IEEE Access*, Vol. 10, pp. 37515–37532, 2022.

[36] P. D. Kusuma and A. Novianty, "Total Interaction Algorithm: A Metaheuristic in Which Each Agent Interacts with All Other Agents", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 1, pp. 224-234, 2023, doi: 10.22266/ijies2023.0228.20.

[37] J. Swan, S. Adriaensen, A. E. I. Brownlee, K. Hammond, C. G. Johnson, A. Kheiri, F. Krawiec, J. J. Merelo, L. L. Minku, E. Ozcan, G. L. Pappa, P. G. Sanchez, K. Sorensen, S. Vob, M. Wagner, and D. R. White, "Metaheuristics in the Large", *European Journal of Operational Research*, Vol. 297, pp. 393-406, 2022.

[38] P. D. Kusuma and A. Dinimaharawati, "Hybrid Pelican Komodo Algorithm", *International Journal of Advanced Computer Science and Applications*, Vol. 13, No. 6, pp. 46-55, 2022.

[39] P. D. Kusuma and M. Kallista, "Stochastic Komodo Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 4, pp. 156-166, 2022, doi: 10.22266/ijies2022.0831.15.

[40] A. Banharnsakun, "A New Approach for Solving The Minimum Vertex Cover Problem Using Artificial Bee Colony", *Decision Analytics Journal*, Vol. 6, ID. 100175, pp. 1-6, 2023.

[41] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime Mould Algorithm: A New Method for Stochastic Optimization", *Future Generation Computer Systems*, Vol. 111, pp. 300-323, 2020.