Enriched Coati Osprey Algorithm: A Swarm-based Metaheuristic and Its Sensitivity Evaluation of Its Strategy

Purba Daru Kusuma, *Member IAENG*, Faisal Candrasyah Hasibuan

***Abstract*—A new swarm-based metaheuristic, namely the enriched coati osprey algorithm (ECOA), is proposed in this paper. As its name suggests, ECOA hybridizes two new metaheuristics, the coati optimization algorithm (COA) and the osprey optimization algorithm (OOA). ECOA is constructed by five searches performed sequentially by the swarm members. The first three are directed searches, while the last two are neighborhood searches. All three directed searches are adopted from COA and OOA. Meanwhile, the four-bordered neighborhood search is developed based on a new approach. During the assessment, ECOA was challenged to overcome the set of 23 functions and contended with five new metaheuristics: total interaction algorithm (TIA), golden search optimization (GSO), average and subtraction-based optimization (ASBO), COA, and OOA. The result shows that ECOA outperforms TIA, GSO, ASBO, COA, and OOA in 16, 23, 18, 21, and 21 functions. Meanwhile, the individual search test result shows that the directed searches perform better than the neighborhood searches. Moreover, the directed search toward the best member becomes the most dominant search.**

***Index Terms*—metaheuristic, swarm intelligence, multi agent, neighborhood search, optimization, coati optimization algorithm, osprey optimization algorithm.**

I. Introduction

METAHEURISTICS has been employed extensively in many optimizations works, especially in various engineering problems. Sand cat swarm optimization (SCSO) has been employed to find the allocation of the distributed generators and shunt capacitors in the distribution system of the power transmission grid [1]. Particle swarm optimization (PSO) has been implemented in manufacturing systems for the production scheduling of remote sensing products [2], in data management system for selecting the features of the high dimension data [3], in robotic path planning system [4], and so on. Artificial bee colony has been employed in the power system to determine the location and capacity of the distributed generations and capacitor banks [5], motion tracking for mobile robots [6], and so on. Meanwhile, a genetic algorithm has been employed to solve the shortest path problem in the computer network [7], the feature selection problem for Arabic-named entity recognition [8], and so on. The sine-cosine algorithm (SCA) has been enriched with Levy mutation to optimize the routing protocol in wireless sensor networks [9].

There are a huge number of new metaheuristics introduced in these recent years. Most of them were constructed based on swarm intelligence and inspired by the behavior of animals, such as Komodo mlipir algorithm (KMA) [10], golden jackal optimization (GJO) [11], Siberian tiger optimization (STO) [12], northern goshawk optimization (NGO) [13], marine predator algorithm (MPA) [14], clouded leopard optimization (CLO) [15], cat and mouse-based optimization (CMBO) [16], chameleon swarm algorithm (CSA) [17], coati optimization algorithm (COA) [18], osprey optimization algorithm (OOA) [19], cheetah optimization (CO) [20], zebra optimization algorithm (ZOA) [21], white shark optimizer (WSO) [22], snake optimizer (SO) [23], pelican optimization algorithm (POA) [24], sparrow search algorithm (SSA) [25], and so on. Some swarm-based metaheuristics imitate human social activities, such as modified social forces algorithm (MSFA) [26], election-based optimization algorithm (EBOA) [27], sewing training-based optimization (STBO) [28], driving training-based optimization (DTBO) [29], and so on. Some metaheuristics imitate the mechanics of traditional games, such as darts game optimization (DGO) [30], ring toss game-based optimization (RTGO) [31], football game-based optimization (FBGO) [32], and so on.

There are several problems or notes regarding the massive development of metaheuristics. The first problem is the massive use of metaphors to cover the novelty of these metaheuristics. Commonly, a new metaheuristic is developed based on modifying previous metaheuristics of hybridization of some metaheuristics. But covering this proposed metaheuristic without crediting the metaheuristic used as the foundation is not fair work. The wise approach is adopting the metaheuristics used for baseline so that the public can trace the root of this proposed metaheuristic easier, such as multi-objective stochastic paint optimization (MOPSO), which is the future development of stochastic paint optimization (SPO) [33], [34], or grey wolf optimizer cuckoo (GWOC) which hybridizes the grey wolf optimization (GWO) and cuckoo search algorithm (CSA) [34]. Some better approach is naming this metaheuristic without using the metaphor but promoting its main strategy, such as in the

Manuscript received August 10, 2023; revised December 10, 2023. This work was financially supported by Telkom University, Indonesia.

Purba Daru Kusuma is an assistant professor in computer engineering, Telkom University, Indonesia (e-mail: purboaru@telkomuniversity.ac.id).

Faisal Candrasyah Hasibuan is a lecturer in computer engineering, Telkom University, Indonesia (e-mail: faicanhasfcb@telkomuniversity.ac.id)

golden search optimization (GSO) [35], total interaction algorithm (TIA) [36], average and subtraction-based optimization (ASBO) [37], attack-leave optimization (ALO) [38], and so on.

The second issue pertains to the assessment conducted during the initial implementation of any metaheuristic algorithm. The primary assessment evaluates the algorithm's performance in solving theoretical problems, such as a comprehensive set of 23 functions. This set of functions is widely employed because it can encompass diverse considerations, including unimodal and multimodal functions, various dimensions, and search spaces. Additionally, some studies have extended the assessment to practical problems spanning engineering and finance domains. Throughout this evaluation, the proposed metaheuristic algorithm is compared against existing ones.

The secondary assessment involves the evaluation of hyper-parameters. Since metaheuristic algorithms comprise several adjustable parameters, such as maximum iteration and population size, it is crucial to assess the sensitivity of these parameters to the algorithm's performance. Unfortunately, due to the construction of recent metaheuristics involving multiple searches, the evaluation of sensitivity for each search within the metaheuristic is rarely found. Nevertheless, this sensitivity assessment holds great significance for the future development of metaheuristics or specific search methodologies. Furthermore, it aids in assessing the strengths and weaknesses of each search approach.

This work aims to develop a new swarm-based metaheuristic by hybridizing the latest metaheuristics: COA and OOA. This constructed metaheuristic is called an enriched coati osprey algorithm (ECOA). This name is chosen because ECOA is developed based on COA and OOA, so it gives credit to these two metaheuristics used for the foundation. COA and OOA were chosen because they are new, as introduced in 2023. Meanwhile, the term enriched comes from a new search embedded into ECOA. This work also provides scientific contributions as follows.

1) A new swarm-based metaheuristic is constructed by hybridizing COA and OOA, called ECOA.
2) A new local search called a bordered neighborhood search is introduced and embedded in the constructed metaheuristic.
3) The performance assessment of ECOA is taken by challenging it to solve the set of 23 functions.
4) The superiority of ECOA is assessed by confronting it with five new metaheuristics: TIA, GSO, ASBO, COA, and OOA.

This paper is arranged as follows. Section one presents this work's background, problem statement, research objective, and scientific contribution. Section two reviews the strategy of some latest metaheuristics, especially COA and OOA, which become the foundation of the designed metaheuristic in this work. Section three describes the proposed model. Section four consists of the assessment scenario and result. Section five conducts the discussion taken regarding the result, findings, algorithm complexity, and the limitations. Section six contains the conclusion and the proposal for future development.

II. Related Works

Many new metaheuristics are swarm-based metaheuristics. Swarm-based metaheuristics are the subset of population-based metaheuristics where the system consists of a swarm. This swarm can be seen as a collection of autonomous solutions or members. Due to this autonomy, each member searches for a better solution without centralized command [39]. There is interaction among members to boost search performance [39].

In swarm-based metaheuristics, the directed search becomes the primary search. The directed search seeks a better solution by walking toward or away from a reference. This reference can be the best member (local best or global best), the resultant of better members, a randomly selected better member, a randomly generated better member, another member within the swarm, and so on. Meanwhile, neighborhood search becomes a secondary or complementary search. Some swarm-based metaheuristics perform the neighborhood search, while others do not. Some swarm-based metaheuristics perform multiple searches to overcome the weakness of a single search, while others still perform a single search.

COA and OOA are two novel swarm-based metaheuristics that were introduced in 2023. COA mimics the behavior of coatis [18], while OOA imitates the behavior of ospreys [19]. During each iteration, both metaheuristics undergo two steps. Initially, a directed-based search is performed, followed by a neighborhood search. In COA, the swarm is divided into two equal-sized sub-swarms in the first step [18]. The first one conducts a directed search toward the best member, while the second one carries out a directed search toward a randomly member within the search space [18]. On the other hand, in OOA, all members of the swarm perform a directed search toward a randomly selected superior member [19]. Furthermore, both metaheuristics incorporate an iteration-controlled neighborhood search, wherein the local search space is reduced as the iteration count increases. Additionally, Table 1 presents a comprehensive overview of the strategies implemented in several recent metaheuristics. This presentation is crucial for clearly understanding the strategies and assessments employed in different metaheuristics and the position of the proposed metaheuristic within this context.

III. Proposed Model

ECOA is a metaheuristic that is constructed from COA and OOA. ECOA consists of several autonomous members that search independently without any central command. Meanwhile, interaction among members is performed to boost its performance. There are five searches performed by each member sequentially in every iteration. The first three are directed searches, while the last two are neighborhood searches. The 1st search is the walk toward the best member, while the second is the walk relative to a randomly generated member. These two searches are adopted from COA. The third search is the walk toward a randomly selected member. This search is adopted from OOA. The fourth search is a bordered neighborhood search, a novel approach. The fifth search is a neighborhood search. It is adopted from COA. The illustration of these five searches is depicted in Fig. 1.

TABLE I
REVIEW OF SOME NEW MATEHEURISTICS INCLUDING THEIR STRATEGY AND ASSESSMENT

No	Metaheuristics	Directed search	Neighborhood search	Assessment
1	TIA [36]	walking relative to all other members	-	23 functions, hyperparameter assessment, dimension assessment
2	GSO [35]	walking toward the mixture between the global best member and local best member	-	23 functions, convergence assessment
3	ASBO [37]	walking relative to the middle between the best and worst members, walking relative to the difference between the best and worst members, walking away from the best member.	-	23 functions, hyperparameter assessment
4	COA [18]	walking toward the best member, walking relative to a randomly generated member	iteration-controlled neighborhood search	CEC 2011, CEC 2017, mechanical engineering design, dimension assessment, convergence assessment
5	OOA [19]	walking toward a better member	iteration-controlled neighborhood search	CEC 2011, CEC 2017, dimension assessment
6	GJO [11]	walking of two best members toward or away from the corresponding member	-	23 functions, mechanical engineering design, hyperparameter assessment
7	MPA [14]	walking toward the local best member, walking of local best member away from the corresponding member, walking toward the difference between two randomly selected members	iteration-controlled neighborhood search	29 functions, convergence assessment, CEC 2017, mechanical engineering design
8	NGO [13]	walking relative to a randomly selected member	iteration-controlled neighborhood search	23 functions, CEC 2015, Mechanical engineering design, hyperparameter assessment
9	MSFA [26]	walking toward a randomly generated member	fixed-size neighborhood search	Ten functions, hyperparameter assessment
10	STO [12]	walking toward a better member, walking relative to a randomly selected member	iteration-controlled neighborhood search	CEC 2011, CEC 2017, mechanical engineering design, dimension assessment
11	this work	walking toward the best member, walking toward a better member, walking relative to a randomly selected member	bordered neighborhood search, iteration-controlled neighborhood search	23 functions, individual search assessment

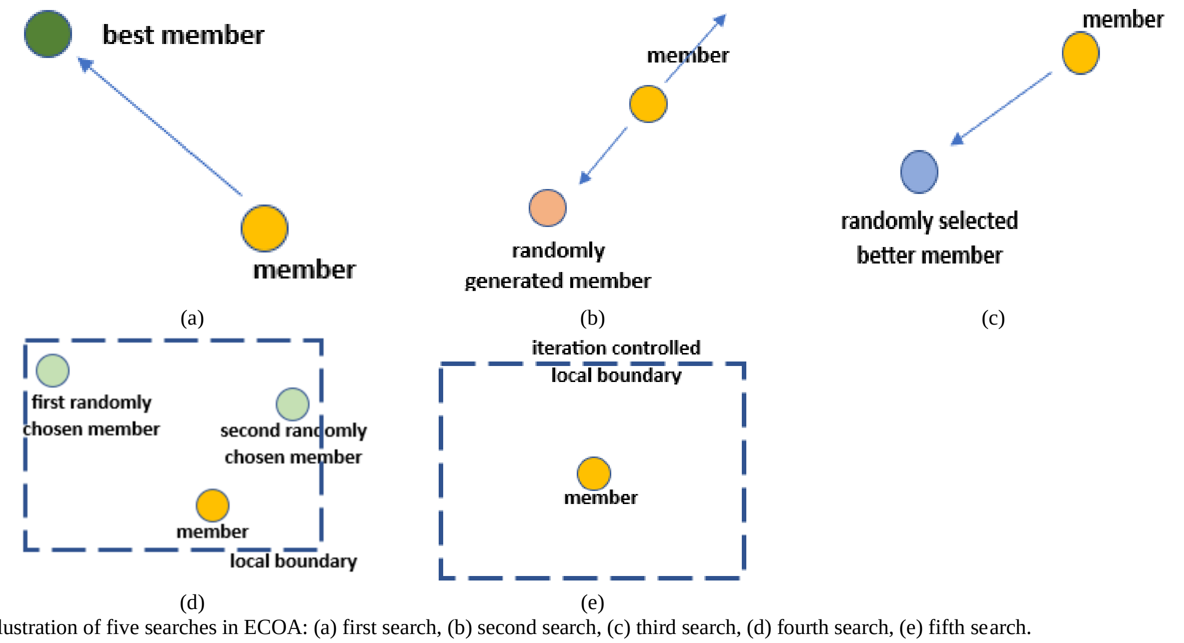

Fig. 1. Illustration of five searches in ECOA: (a) first search, (b) second search, (c) third search, (d) fourth search, (e) fifth search.

The directed search is identical to the motion guided with one or more references. The first search is designed mostly for improvement by approaching closer to the best member. As it is assumed that the best member is in a better location than the corresponding member, walking closer to the best member may produce improvement for the corresponding member. Meanwhile, this search does not guarantee improvement, especially in multimodal problems which are identical with multiple optimal solutions. Leading toward the best member still provides two possibilities: improvement or stagnation in the local optimal solution. This circumstance leads to the two other directed searches. The second search is designed for exploration. In the second search, the reference is generated randomly along the space. It keeps the exploration capability during the iteration. In this second search, the direction depends on the comparative quality between the corresponding member and the corresponding member. If the reference is better, the corresponding member walks toward the reference. Otherwise, this member walks away. The third search is designed for improvement with multiple possible directions. The reference of the third search is a randomly selected member. It means all members better than the corresponding member are collected into a pool. The best member is also included in this pool. Then, a member is chosen randomly from this pool as a reference. Then, the corresponding member walks toward this chosen member.

The ECOA incorporates two distinct neighborhood searches. The first neighborhood search is aimed at achieving convergence. In this initial search, the boundaries of the local search space are determined based on the positions of specific members: the relevant members and two randomly selected members from the swarm. For each dimension, the lowest value among these members is the lower boundary, while the highest is the upper boundary. A candidate is randomly generated within these determined boundaries to generate a new solution.

The second neighborhood search follows a different approach and is commonly employed in various metaheuristics, including MPA. This second search is known as an iteration-controlled neighborhood search. In this case, the corresponding member acts as the central point, and the candidate solutions can be generated in any direction within the search space. The iteration count controls the search space size, where the radius gradually decreases as the iteration progresses. This adaptive behavior represents the shift from exploration to exploitation as the iteration number increases.

This concept is then transformed into the formal model. This formalization consists of two parts: the algorithm and the mathematical model. The algorithm of ECOA is formalized using pseudocode, as depicted in Fig. 1. Meanwhile, the mathematical model presents each process's clear and detailed form. The annotations used in this paper are as follows.

f	objective function
m	corresponding member
M	swarm
m_l	lower bound
m_u	upper bound
m_c	member's candidate
m_{best}	best member
m_t	member's target
m_p	selected better member
M_p	the pool of better members
m_s	randomly selected member
m_b	member's border
r_1	uniform random with the interval [0,1]
r_2	uniform random, whether 1 or 2
t	iteration
t_{max}	maximum iteration
U	uniform random

As common in metaheuristics, ECOA begins with initialization. The first stage is initialization, while the second stage is iteration. During the initialization, all members are distributed uniformly inside the search space. This process is formalized using (1). Meanwhile, each time a member is initialized, the best member is updated using (2). This updating process is important to keep the superiority of the best member up to date.

algorithm 1: enriched coati osprey algorithm

```
1   begin
2     for m=1 to n(M)
3       generate initial m using (1)
4       update m_best using (2)
5     end for
6     for t = 1 to t_max
7       for m=1 to n(M)
8         first search using (3)
9         update m using (4)
10        A second search using (5) and (6)
11        update m using (4)
12        third search using (7) to (9)
13        update m using (4)
14        fourth search using (10) to (13)
15        update m using (4)
16        fifth search using (14)
17        update m using (4)
18        update m_best using (2)
19      end for
20    end for
21  end
```

$$m_{i,j} = ml_j + r_1(mu_j - ml_j) \quad (1)$$

$$m'_{best} = \begin{cases} m_i, f(m_i) < f(m_{best}) \\ m_{best}, otherwise \end{cases} \quad (2)$$

The iteration stage starts after the initialization ends. Each member performs five searches sequentially. In algorithm 1, the iteration is presented from lines 6 to 20. Each time a member completes these five searches. The best member is updated again using (2). Each search generates a solution candidate, which is then compared with the current value of the corresponding member. If the candidate is better than this candidate, replace the corresponding member's current value as presented in (4).

$$mc_{1,j} = m_{i,j} + r_1(m_{best,j} - r_2 . m_{i,j}) \quad (3)$$

$$m'_i = \begin{cases} mc, f(mc) < f(m_i) \\ m_i, otherwise \end{cases} \quad (4)$$

As presented in algorithm 1, there are three directed searches. The procedure of walking toward the best member as the first directed search is formalized using (3). Meanwhile, walking relatively to a randomly generated member is formalized using (5) and (6). Equation (5) formalizes the generation of the target, while (6) formalizes the walking procedure where the direction can be toward or away. Equation (7) to (9) formalizes the third directed search. Equation (7) formalizes that the pool consists of a set of better members and the best member. Equation (8) formalizes that the selected better member is uniformly picked up from the pool. Equation (9) formalizes walking toward this selected better member.

$$mt_j = ml_j + r_1(mu_j - ml_j) \quad (5)$$

$$mc_{2,j} = \begin{cases} m_{i,j} + r_1(mt_j - r_2 . m_{i,j}), f(mt) < f(m_i) \\ m_{i,j} + r_1(m_{i,j} - r_2 . mt_j), otherwise \end{cases} \quad (6)$$

$$Mp_i = \{m \in M | f(m) < f(m_i) \cup m_{best}\} \quad (7)$$

$$mp_i = U(Mp_i) \tag{8}$$

$$mc_{3,j} = m_{i,j} + r_1(mp_{i,j} - r_2 . m_{i,j}) \tag{9}$$

The two neighborhood searches are performed after the member performs the directed searches. The bordered neighborhood search is formalized using (10) to (14). Meanwhile, the iteration-controlled neighborhood search is formalized using (14). Equation (10) formalizes the uniform random selection of a member from the swarm. Equation (11) formalizes the lower border based on the lowest value among the corresponding member, first selected member, and second selected member. On the other hand, (13) formalizes the upper border based on the highest value among the corresponding member, first selected member, and second selected member. Equation (14) shows that the corresponding member becomes the central point in the second neighborhood search, and its local space declines as the iteration increases.

$$ms = U(M) \tag{10}$$

$$mb_{min,i,j} = \min(m_{i,j}, ms_{1,j}, ms_{2,j}) \tag{11}$$

$$mb_{max,i,j} = \max(m_{i,j}, ms_{1,j}, ms_{2,j}) \tag{12}$$

$$mc_{4,j} = mb_{min,i,j} + r_1(mb_{max,i,j} - mb_{min,i,j}) \tag{13}$$

$$mc_{5,j} = m_{i,j} + \frac{(1-2r_1)\left(ml_j + r_1(mu_j - ml_j)\right)}{t} \tag{14}$$

IV. Simulation

Three assessments are performed in this work to evaluate the performance of the constructed ECOA. The first assessment is performed to evaluate the performance of ECOA in solving the theoretical problems. The second assessment is performed to evaluate the contribution or performance of each search individually. The third assessment is the hyperparameter assessment designed to investigate the relation between the adjusted parameters and the performance of the algorithm.

The set of 23 functions is chosen as the theoretical problem. It is selected due to its popularity in assessing many metaheuristics. It consists of three groups of functions: seven high-dimension unimodal functions, six high-dimension multimodal functions, and ten fixed-dimension multimodal functions. A detailed description of these functions can be seen in [35]. These functions are used in both assessments. In this assessment, the swarm size is five, while the maximum iteration is 25. The dimension is set to 40 for the high-dimension functions.

The initial assessment compares ECOA against five newly developed metaheuristics: TIA, GSO, ASBO, COA, and OOA. Among these metaheuristics, TIA [36] and GSO [35] utilize a single search approach, while ASBO [37], COA [18], and OOA [19] employ multiple search methodologies. TIA [36], GSO [35], and ASBO [37] exclusively utilize directed search, whereas COA [18] and OOA [19] incorporate directed search and neighborhood search strategies. The selection of COA and OOA for comparison is particularly significant as ECOA is developed by hybridizing these two metaheuristics. This choice allows a meaningful evaluation of the proposed metaheuristic concerning its foundational components. Notably, GSO is the only metaheuristic that does not employ a strict acceptance criterion [35]. In the subsequent assessment, each search within ECOA is individually evaluated, with the inactive searches temporarily disabled during the assessment of a specific search. This approach ensures a focused analysis of each search's performance within ECOA. In this second assessment, ECOA is not compared against other metaheuristics but is evaluated on its merits and the effectiveness of its constituent searches.

The assessment result is depicted in Table 2 to Table 6. Table 2 to Table 5 depict the first assessment result, while Table 6 depicts the second. There are three pieces of data in Table 2 to Table 4: the mean of the average fitness score, the standard deviation of the average fitness score, and the mean rank.

TABLE II
PERFORMANCE ASSESSMENT ON SOLVING HIGH-DIMENSION UNIMODAL FUNCTIONS

F	Parameter	TIA [36]	GSO [35]	ASBO [37]	COA [18]	OOA [19]	ECOA
1	mean	0.0000	4.8593×10^{4}	0.0575	0.0716	0.0051	0.0000
	std deviation	0.0000	1.2112×10^{4}	0.0401	0.0567	0.0059	0.0000
	mean rank	1	6	4	5	3	1
2	mean	0.0000	1.1764×10^{53}	0.0000	0.0000	0.0000	0.0000
	std deviation	0.0000	3.2516×10^{53}	0.0000	0.0000	0.0000	0.0000
	mean rank	1	6	1	1	1	1
3	mean	6.4554	1.3546×10^{5}	1.3556×10^{3}	2.0058×10^{3}	1.5562×10^{3}	1.4953×10^{1}
	std deviation	1.0891×10^{1}	8.5764×10^{4}	1.1465×10^{3}	2.7571×10^{3}	2.6782×10^{3}	4.7989×10^{1}
	mean rank	1	6	3	5	4	2
4	mean	0.0126	6.1203×10^{1}	0.3525	1.1140	0.1558	0.0000
	std deviation	0.0052	7.0465	0.1308	0.4719	0.0972	0.0000
	mean rank	2	6	4	5	3	1
5	mean	3.8883×10^{1}	9.5107×10^{7}	3.9545×10^{1}	4.0694×10^{1}	3.9046×10^{1}	3.8934×10^{1}
	std deviation	0.0453	4.8982×10^{7}	0.4933	1.4729	0.1355	0.0389
	mean rank	1	6	4	5	3	2
6	mean	7.1509	4.9907×10^{4}	7.5779	8.9644	8.3829	8.2469
	std deviation	0.5242	9.9001×10^{3}	0.5798	0.6032	0.5284	0.5573
	mean rank	1	6	2	5	4	3
7	mean	0.0251	6.2590×10^{1}	0.0605	0.0578	0.0219	0.0091
	std deviation	0.0162	3.2591×10^{1}	0.0336	0.0262	0.0136	0.0053
	mean rank	3	6	5	4	2	1

TABLE III
PERFORMANCE ASSESSMENT ON SOLVING HIGH-DIMENSION MULTIMODAL FUNCTIONS

F	Parameter	TIA [36]	GSO [35]	ASBO [37]	COA [18]	OOA [19]	ECOA
	mean	-1.9781×10^3	-3.3635×10^3	-3.7795×10^3	-4.4685×10^3	-3.8588×10^3	-4.7311×10^3
8	std deviation	3.9901×10^2	9.0826×10^2	5.4571×10^2	5.5682×10^2	5.1586×10^2	8.2378×10^2
	mean rank	6	5	4	2	3	1
	mean	0.0002	3.9887×10^2	1.4731×10^1	1.7623	0.0916	0.0000
9	std deviation	0.0002	5.3494×10^1	3.1314	4.4273	0.2521	0.0000
	mean rank	2	6	5	4	3	1
	mean	0.0011	1.9196×10^1	2.5895	0.0574	0.0129	0.0000
10	std deviation	0.0004	0.7100	0.3384	0.0326	0.0051	0.0000
	mean rank	2	6	5	4	3	1
	mean	0.0010	4.4872×10^2	0.3703	0.1347	0.0096	0.0025
11	std deviation	0.0032	9.1480×10^1	0.1787	0.2273	0.0319	0.0098
	mean rank	1	6	5	4	3	2
	mean	0.8091	1.6920×10^8	0.1302	0.8591	1.0994	0.6968
12	std deviation	0.1380	1.1919×10^8	0.1413	0.1956	0.2061	0.1948
	mean rank	3	6	1	4	5	2
	mean	3.0974	3.6655×10^8	9.0205	3.3365	3.1676	3.0254
13	std deviation	0.0935	2.2205×10^8	0.9045	0.1303	0.0618	0.0897
	mean rank	2	6	5	4	3	1

TABLE IV
PERFORMANCE ASSESSMENT ON SOLVING FIXED-DIMENSION MULTIMODAL FUNCTIONS

F	Parameter	TIA [36]	GSO [35]	ASBO [37]	COA [18]	OOA [19]	ECOA
	mean	9.2838	1.1623×10^1	5.1110	5.5668	6.4955	3.8626
14	std deviation	3.1887	5.7309	3.7357	4.6510	3.2340	2.6207
	mean rank	5	6	2	3	4	1
	mean	0.0038	0.0920	0.1232	0.0053	0.0036	0.0032
15	std deviation	0.0114	0.2933	0.0361	0.0103	0.0066	0.0058
	mean rank	3	5	6	4	2	1
	mean	-0.9973	-0.9345	-0.0373	-1.0307	-1.0304	-1.0315
16	std deviation	0.0570	0.2343	0.1752	0.0014	0.0018	0.0003
	mean rank	4	5	6	2	3	1
	mean	3.0893	2.1793	1.2252	0.3988	0.3988	0.3982
17	std deviation	4.2417	4.0203	1.6390	0.0010	0.0008	0.0003
	mean rank	6	5	4	2	2	1
	mean	1.4979×10^1	1.8156×10^1	3.0000	6.4216	4.3038	3.0011
18	std deviation	1.9261×10^1	2.6659×10^1	0.0000	9.1134	5.8891	0.0021
	mean rank	5	6	1	4	3	2
	mean	-0.0495	-0.0148	-0.0495	-0.0495	-0.0495	-0.0495
19	std deviation	0.0000	0.0175	0.0000	0.0000	0.0000	0.0000
	mean rank	1	6	1	1	1	1
	mean	-2.3154	-2.2219	-0.9751	-3.0632	-3.0770	-3.2804
20	std deviation	0.7154	0.6296	0.7305	0.1847	0.1196	0.0791
	mean rank	4	5	6	3	2	1
	mean	-2.7876	-2.2786	-3.0005	-5.8289	-3.2824	-7.5231
21	std deviation	1.4439	2.2894	3.2542	2.2278	1.7509	2.3328
	mean rank	5	6	4	2	3	1
	mean	-2.5620	-1.9601	-3.7365	-5.9375	-3.3221	-7.4799
22	std deviation	1.2146	1.2709	3.5917	2.5457	1.9400	2.6759
	mean rank	5	6	3	2	4	1
	mean	-2.5619	-2.8784	-4.0350	-4.7437	-2.8749	-5.9493
23	std deviation	1.4250	2.3311	2.7824	2.1196	0.9101	3.3693
	mean rank	6	4	3	2	5	1

Table 4 indicates that ECOA is also powerful in solving fixed-dimension multimodal functions. ECOA is in the second rank in solving one function (Goldstein Price), while ECOA is always in the first rank in solving the rest functions in this group. In Goldstein-Price, ASBO is in the first rank. Like in the first and the second groups, ECOA is always better than COA and OOA in solving all functions in the third group except in Hartman 3.

Table 5 depicts the summarized superiority of ECOA with the five other metaheuristics. The data presented in Table 5 is the number of functions where ECOA outperforms the related functions in every group of functions. The performance comparison is based on the average fitness score between ECOA and its confronter. Table 5 indicates that overall, ECOA outperforms its confronters. ECOA is better than TIA, GSO, ASBO, COA, and OOA in solving 16, 23, 18, 21, and 21 functions consecutively. This result shows that TIA becomes the most difficult-to-beat metaheuristic while GSO becomes the one that is the easiest to outperform.

TABLE V
CLUSTER BASED COMPARISON RESULT OF ECOA

Cluster	Number of Functions Beaten by ECOA				
	TIA [36]	GSO [35]	ASBO [37]	COA [18]	OOA [19]
1	2	7	5	6	6
2	5	6	5	6	6
3	9	10	8	9	9
Total	16	23	18	21	21

TABLE VI
SINGLE SEARCH ASSESSMENT RESULT

F	Average Fitness Score				
	1st search	2nd search	3rd search	4th search	5th search
1	**0.0014**	6.0087×10^4	0.0047	3.9921×10^4	9.4785×10^4
2	**0.0000**	2.1141×10^{46}	**0.0000**	1.4211×10^{46}	2.6164×10^{53}
3	**1.1393×10^2**	1.4399×10^5	1.1832×10^2	8.1885×10^4	2.0040×10^5
4	**0.0636**	7.7867×10^1	0.0937	7.0222×10^1	8.6315×10^1
5	**3.8954×10^1**	1.8354×10^8	3.8985×10^1	9.7643×10^7	3.5158×10^8
6	**8.2699**	5.9663×10^4	8.4839	3.9279×10^4	9.6258×10^4
7	0.0192	1.1040×10^2	**0.0189**	6.3638×10^1	2.2021×10^2
8	-2.5053×10^3	-2.8578×10^3	-2.3983×10^3	-2.6642×10^3	**-4.2447×10^3**
9	**0.0296**	5.1489×10^2	0.0557	3.8214×10^2	5.5374×10^2
10	**0.0054**	1.9742×10^1	0.0122	1.8797×10^1	2.0701×10^1
11	**0.0117**	5.1110×10^2	0.0426	3.7461×10^2	8.4491×10^2
12	**1.0649**	3.4944×10^8	1.1138	1.2199×10^8	8.2702×10^8
13	**3.1349**	7.2386×10^8	3.1533	3.3800×10^8	1.5263×10^9
14	9.3335	1.0663×10^1	9.7483	7.2885×10^1	**6.2246**
15	0.0225	0.0234	**0.0098**	0.1823	0.0979
16	-0.8885	-0.8829	**-0.8892**	2.6722×10^1	-0.1733
17	4.2234	**0.6256**	4.9839	7.9764	3.6633
18	6.9090×10^1	**9.4627**	3.7450×10^1	1.4849×10^2	8.6549×10^1
19	**-0.0495**	**-0.0495**	**-0.0495**	-0.0002	-0.0308
20	-2.0990	**-2.4069**	-2.2345	-1.6599	-2.3557
21	-1.8578	-1.0538	**-2.5080**	-0.9038	-2.1584
22	**-2.0676**	-1.2289	-1.9679	-1.3615	-1.8514
23	-1.8064	-1.5891	-2.2841	-1.3458	**-2.4525**

Table 6 depicts the assessment result regarding the strength of each search in ECOA. The strength of each search can be measured based on the number of functions where the corresponding member is in the first rank. The first rank is written in bold font. Based on this category, the first, second, third, fourth, and fifth search is consecutively ranked first in solving 13, 4, 6, 0, and 3 functions. This result shows the dominance of the first search. On the other hand, the second, third, and fifth searches are less dominant. Ironically, the fourth search is the least contributor among these searches. The first search is crucial in solving the high-dimension functions, while the second and third searches perform superior in solving fixed-dimension functions. Compared between the bordered neighborhood search and the iteration-controlled neighborhood search, the bordered neighborhood search is superior to the iteration-controlled neighborhood search in 12 functions where most of these functions are high dimension functions.

TABLE VII
RELATION BETWEEN SWARM SIZE AND FTNESS SCORE

F	Average Fitness Score	
	$n(M)$=10	$n(M)$=20
1	0.0000	0.0000
2	0.0000	0.0000
3	0.5811	0.0285
4	0.0000	0.0000
5	3.8905×10^1	3.886×10^1
6	7.1472	6.3034
7	0.0056	0.0033
8	-4.9610×10^3	-5.4395×10^3
9	0.0000	0.0000
10	0.0000	0.0000
11	0.0014	0.0000
12	0.5180	0.4716
13	2.8644	2.6486
14	3.4457	1.9963
15	0.0026	0.0006
16	-1.0316	-1.0316
17	0.3981	0.3981
18	3.0004	3.0001
19	-0.0495	-0.0495
20	-3.2806	-3.3013
21	-8.1950	-9.3818
22	-9.5038	-1.0344×10^1
23	-7.2238	-7.8538

There are two adjusted parameters investigated in the third assessment. The first parameter is the swarm size while the second parameter is the maximum iteration. There are two values for the swarm size: 10 and 20. On the other hand, there are two values for the maximum iteration: 20 and 40. When the swarm size is investigated, the maximum iteration is set to 25. On the other hand, when the maximum iteration is investigated, the swarm size is set to 5. The relation between the swarm size and the performance of ECOA is presented in Table 7. The relation between the maximum iteration and the performance of ECOA is presented in Table 8.

TABLE VIII
RELATION BETWEEN MAXIMUM ITERATION AND FTNESS SCORE

F	Average Fitness Score	
	t_m=20	t_m=40
1	0.0000	0.0000
2	0.0000	0.0000
3	1.6891×10^2	0.4410
4	0.0008	0.0000
5	3.8936×10^1	3.8936×10^1
6	7.9929	7.9935
7	0.0131	0.0046
8	-4.2198×10^3	-5.0773×10^3
9	0.0000	0.0000
10	0.0000	0.0000
11	0.0005	0.0000
12	0.8888	0.3583
13	3.0384	2.9795
14	5.5386	4.2508
15	0.0010	0.0062
16	-1.0314	-1.0316
17	0.3983	0.3981
18	1.0056×10^1	3.0002
19	-0.0495	-0.0495
20	-3.2287	-3.2689
21	-6.9950	-8.2174
22	-6.8719	-8.1311
23	-6.6109	-7.1488

Table 7 exposes that the increase of swarm size from 10 to 20 improves the average fitness score significantly in only four functions. Two functions are high dimension unimodal functions (Schwefel 1.2 and Quartic), one function is high dimension multimodal functions (Griewank), and one function is fixed dimension multimodal functions (Kowalik).

Many stagnations occur because the final solution is the global optimal solution or near the global optimal solution in 13 functions.

Similar circumstances also can be found in Table 8. Table 8 exposes that the increase of maximum iteration improves the quality of solution significantly in only six functions. Three functions are high dimension unimodal functions (Schwefel 1.2, Schwefel 2.21, and Quartic), two functions are high dimension multi dimension multimodal functions (Griewank and Penalized), and one function is fixed dimension multimodal function (Goldstein-Price) Many stagnations occurs because the final solution is the global optimal solution or near the global optimal solution in 11 functions.

V. Discussion

The first assessment indicates that the ECOA performs well and competently solves the set of 23 functions. This acceptable performance can be traced from two aspects: the absolute result and the comparative result. In the first aspect, ECOA can find quasi-optimal solutions in all functions. Moreover, ECOA can find the global optimal solution of five high-dimension functions. Fortunately, ECOA is also ranked first in solving 17 functions split into eight high-dimension functions and nine fixed-dimension functions.

The comparative results demonstrate the superiority of ECOA over its competitors. ECOA consistently outperforms its foundational metaheuristics, COA and OOA, as it achieves better results in 21 functions and performs on par in 2. Additionally, ECOA surpasses GSO in all 23 functions, establishing its complete superiority over GSO. On the other hand, TIA proves to be a strong contender, with ECOA outperforming it in only two functions within the first group.

Through the analysis of strategies, it is evident that implementing multiple strategies in a metaheuristic enhances its effectiveness, although the magnitude of improvement may vary. Notably, the strict acceptance strategy emerges as a new standard, with GSO being the only metaheuristic not incorporating this approach.

The individual search indicates that the directed searches are more powerful than the neighborhood searches, although neighborhood searches are still competitive in a few functions. This circumstance strengthens the strategy that directed searches should be the primary searches while neighborhood searches are the secondary ones. Compared between the neighborhood searches, the proposed bordered neighborhood search is better in the high-dimension functions, while the iteration-controlled neighborhood search is better in the fixed-dimension functions.

The result of hyper-parameter assessment shows that ECOA performs well in the circumstance where the swarm size and maximum iteration are low. This finding comes from the fact that stagnation is found in many functions when the swarm size or maximum iteration increases. In many functions, the global optimal solution has been found or the final solution is close to the global optimal solution.

The computational complexity of ECOA can be drawn back by evaluating its loops. As ECOA contains two stages (initialization and iteration), the computational complexity between these stages differs, so the evaluation is taken separately. There are two loops during the initialization. The first and outer loop is a loop for all members. Meanwhile, the second and inner loop is a loop for whole dimensions. Based on this explanation, the complexity in the initial stages is presented as $O(n(X).d)$. On the other hand, there are four loops. The outer loop relates to the iteration. Then, in every iteration, there is a loop for all members. There are two loops performed by each member sequentially. The first loop is used to trace all the best members. Meanwhile, a loop is needed in every search. Based on this explanation, the complexity of ECOA can be presented as $O(t_{max}.n(X).(n(X)+5d))$.

The superiority of ECOA comes with several notes. First, although overall, ECOA is superior to all its confronters, ECOA is inferior to TIA in solving the high-dimension unimodal functions. Meanwhile, TIA performs only a single strategy which is performed more extensively. Second, the bordered neighborhood search's performance still needs much improvement to compete with other neighborhood searches, especially the iteration-controlled neighborhood search. Fourth, ECOA has not been tested to solve the practical optimization problem yet due to the wide variety of this problem. It means that evaluation of ECOA by challenging it to solve various practical optimization problems is needed to evaluate the performance of ECOA, including its strengths and weaknesses, more comprehensively.

VI. Conclusion

This work presents ECOA, a novel swarm-based metaheuristic combining COA and OOA features. The performance of ECOA has been evaluated, and the results demonstrate its superiority over the compared metaheuristics. ECOA outperforms TIA, GSO, ASBO, COA, and OOA in 16, 23, 18, 21, and 21 functions across the entire set of 23 functions. TIA is the most challenging to surpass among the evaluated metaheuristics, particularly in high-dimensional unimodal functions. Through the assessment of individual searches, it is observed that the directed search holds more dominance than the neighborhood search. Specifically, the directed search toward members emerges as the most influential search ally in high-dimensional multimodal functions. Meanwhile, the directed search towards a better member and the directed search relative to other members complement each other due to their strengths in fixed-dimensional multimodal functions.

Future studies can be carried out in several tracks. The first one is performing exploration to improve the bordered neighborhood search performance. The second is exploring other directed search methods to contend the directed search toward the best member. The third one is implementing ECOA to solve various practical optimization problems.

References

[1] H. Long, Y. He, Y. Xu, C. You, D. Zeng, and H. Lu, “Optimal allocation research of distribution network with DGs and SCs by improved sand cat swarm optimization algorithm,” *IAENG International Journal of Computer Science*, vol. 50, no. 2, pp. 646-661, 2023.

[2] Y. Hou, W. He, X. Zuo, L. Dang, and H. Han, "A task scheduling approach based on particle swarm optimization for the production of remote sensing products," *IAENG International Journal of Computer Science*, vol. 50, no. 1, pp. 23-31, 2023.

[3] T. Saw, and W. M. Oo, "Ranking-based feature selection with wrapper PSO search in high-dimensional data classification," *IAENG International Journal of Computer Science*, vol. 50, no. 1, pp. 32-41, 2023.

[4] A. A. Alabdalbari and I. A. Abed, “New robot path planning optimization using hybrid GWO-PSO algorithm,” *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 3, pp. 1289–1296, 2022.

[5] M. Ahmadi Kamarposhti, S. M. Mousavi Khormandichali, and A. A. Ahmed Solyman, “Locating and sizing of capacitor banks and multiple DGs in distribution system to improve reliability indexes and reduce loss using ABC algorithm,” *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 2, pp. 559–568, 2021.

[6] A. Y. Abdalla and T. Y. Abdalla, “A modified artificial bee colony based fuzzy motion tracking scheme for mobile robot,” *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 4, pp. 2160–2168, 2022.

[7] Fahad. A. Alghamdi, A. Younes Hamed, A. M. Alghamdi, A. Ben Salah, T. H. Farag, and W. Hassan, “A genetic algorithm for shortest path with real constraints in computer networks,” *International Journal of Electrical and Computer Engineering*, vol. 13, no. 1, pp. 435–442, 2023.

[8] B. A. Benali, S. Mihi, I. El Bazi, and N. Laachfoubi, “New approach for Arabic named entity recognition on social media based on feature selection using genetic algorithm,” *International Journal of Electrical and Computer Engineering*, vol. 11, no. 2, pp. 1485–1497, 2021.

[9] X. -L. Guo, Y. -F. Ye, M. -H. Liu, L. Li, and Z. -L. Chang, “Inter-cluster multi-relay routing protocol based on SCA and Levy mutation”, *Engineering Letters*, vol. 31, no. 3, pp. 948-953, 2023.

[10] S. Suyanto, A. A. Ariyanto, and A. F. Ariyanto, “Komodo mlipir algorithm,” *Applied Soft Computing*, vol. 114, pp. 1–17, 2022.

[11] N. Chopra and M. Mohsin Ansari, “Golden jackal optimization: A novel nature-inspired optimizer for engineering applications,” *Expert Syst Appl*, vol. 198, p. 116924, 2022.

[12] 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.

[13] 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, doi: 10.1109/ACCESS.2021.3133286.

[14] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, “Marine predators algorithm: a nature-inspired metaheuristic,” *Expert Systems with Applications*, vol. 152, p. 113377, 2020.

[15] E. Trojovska and M. Dehghani, “Clouded leopard optimization: a new nature-inspired optimization algorithm,” *IEEE Access*, vol. 10, pp. 102876–102906, 2022.

[16] M. Dehghani, Š. Hubálovský, and P. Trojovský, “Cat and mouse based optimizer: a new nature-inspired optimization algorithm,” *Sensors*, vol. 21, no. 15, p. 5214, 2021.

[17] M. S. Braik, “Chameleon swarm algorithm: a bio-inspired optimizer for solving engineering design problems,” *Expert Syst Appl*, vol. 174, p. 114685, 2021.

[18] M. Dehghani, Z. Montazeri, E. Trojovská, and P. Trojovský, “Coati optimization algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems,” *Knowledge-based Systems*, vol. 259, p. 110011, 2023.

[19] M. Dehghani and P. Trojovský, “Osprey optimization algorithm: a new bio-inspired metaheuristic algorithm for solving engineering optimization problems,” *Frontiers in Mechanical Engineering*, vol. 8, p. 1126450, 2023.

[20] M. A. Akbari, M. Zare, R. Azizipanah-abarghooee, S. Mirjalili, and M. Deriche, “The cheetah optimizer: a nature-inspired metaheuristic algorithm for large-scale optimization problems,” *Scientific Reports*, vol. 12, no. 1, p. 10953, 2022.

[21] 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.

[22] M. Braik, A. Hammouri, J. Atwan, M. A. Al-Betar, and M. A. Awadallah, “White shark optimizer: a novel bio-inspired meta-heuristic algorithm for global optimization problems,” *Knowledge-based Systems*, vol. 243, p. 108457, 2022.

[23] F. A. Hashim and A. G. Hussien, “Snake optimizer: a novel meta-heuristic optimization algorithm,” *Knowledge-based Systems*, vol. 242, p. 108320, Apr. 2022.

[24] P. Trojovský and M. Dehghani, “Pelican optimization algorithm: a novel nature-inspired algorithm for engineering applications,” *Sensors*, vol. 22, no. 3, pp. 1–34, 2022.

[25] J. Xue and B. Shen, “A novel swarm intelligence optimization approach: sparrow search algorithm,” *Systems Science & Control Engineering*, vol. 8, no. 1, pp. 22–34, 2020.

[26] P. D. Kusuma and D. Adiputra, “Modified social forces algorithm: from pedestrian dynamic to metaheuristic optimization,” *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 3, pp. 294–303, 2022.

[27] P. Trojovský and M. Dehghani, “A new optimization algorithm based on mimicking the voting process for leader selection,” *PeerJ Computer Science*, vol. 8, p. e976, 2022.

[28] M. Dehghani, E. Trojovská, and T. Zuščák, “A new human-inspired metaheuristic algorithm for solving optimization problems based on mimicking sewing training,” *Scientific Reports*, vol. 12, no. 1, p. 17387, Oct. 2022.

[29] 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.

[30] M. Dehghani, Z. Montazeri, H. Givi, J. 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.

[31] S. Doumari, H. Givi, M. Dehghani, and O. Malik, “Ring toss game-based optimization algorithm for solving various optimization problems,” *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 3, pp. 545–554, 2021.

[32] M. Dehghani, M. Mardaneh, J. Guerrero, O. 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.

[33] N. Khodadadi, L. Abualigah, and S. Mirjalili, “Multi-objective stochastic paint optimizer (MOSPO),” *Neural Computing Applications*, vol. 34, no. 20, pp. 18035–18058, 2022.

[34] M. Nanjappan, P. Krishnadoss, J. Ali, G. Natesan, and B. Ananthakrishnan, “Task scheduling based on cost and execution time using ameliorate grey wolf optimizer algorithm in cloud computing,” *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 3, pp. 417–427, 2023.

[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.

[37] M. Dehghani, Š. Hubálovský, and P. Trojovský, “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, pp. 1–40, 2022.

[38] P. D. Kusuma and F. C. Hasibuan, “Attack-leave optimizer: a new metaheuristic that focuses on the guided search and performs random search as alternative,” *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 3, pp. 244–257, 2023.

[39] A. G. Gad, “Particle swarm optimization algorithm and its applications: a systematic review,” *Archives of Computational Methods in Engineering*, vol. 29, no. 5, pp. 2531–2561, 2022.

Purba Daru Kusuma is an assistant professor in computer engineering in Telkom University, Indonesia. He received his bachelor’s and master’s degrees in electrical engineering from Bandung Institute of Technology, Indonesia. He received his doctoral degree in computer science from Gadjah Mada University, Indonesia. His research interests are in artificial intelligence, machine learning, and operational research. He is currently becoming a member of IAENG.

Faisal Candrasyah Hasibuan is a computer engineering lecturer at Telkom University, Bandung, West Java, Indonesia. Status as an alma mater, he completed a bachelor's program at the place where he works now. He was granted a master’s degree in electrical engineering with a Computer Engineering specialization at Bandung Institute of Technology (ITB). His research interests include the embedded system, the Internet of Things and intelligent control systems.