Received: December 15, 2023. Revised: January 23, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Swarm Bipolar Algorithm: A Metaheuristic Based on Polarization of Two Equal Size Sub Swarms

Purba Daru Kusuma[1]* **Ashri Dinimaharawati[1]**

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

Abstract: This paper presents a new metaphor-free metaheuristic search called the swarm bipolar algorithm (SBA). SBA is developed mainly based on the non-free-lunch (NFL) doctrine, which mentions the non-existence of any general optimizer appropriate to answer all varieties of problems. The construction of SBA is based on splitting the swarm into two equal-sized swarms to diversify the searching process while performing intensification within the sub-swarms. There are two types of finest swarm members: the finest swarm member for the whole swarm and the finest swarm member in every sub-swarm. There are four directed searches performed in every iteration: (1) search toward the finest swarm member, (2) search toward the finest sub-swarm member, (3) search toward the middle between two finest sub-swarm members, and (4) search relative to the randomly picked swarm member from another sub-swarm. The performance of SBA is assessed through two assessments with a set of 23 functions representing the optimization problem. In the benchmark assessment, SBA is contended with five metaheuristics: northern goshawk optimization (NGO), language education optimization (LEO), coati optimization algorithm (COA), fully informed search algorithm (FISA), and total interaction algorithm (TIA). The result presents the superiority of SBA among its contenders by being better than NGO, LEO, COA, FISA, and TIA in 21, 16, 16, 21, and 18 functions. The single search assessment is performed to evaluate each strategy involved in SBA. The result shows that the search toward the middle between the two finest sub-swarm members is the best among the four searches in SBA.

Keywords: Optimization, Stochastic, Metaheuristic, No-free-lunch, Swarm intelligence.

1. Introduction

Optimization is a work to find the most appropriate or acceptable solution among the set of solution candidates for a defined problem. Optimization is essential and critical in the real world, especially in the engineering and industrial fields [1]. In many optimization studies, the set of solution candidates is limited to the defined constraints. In other terms, these constraints are also called hard constraints. Meanwhile, the quality of the chosen solution is measured by using the objective function [2], also called soft constraints. Meanwhile, the decision variables construct the solution set and are used in the objective function. These three aspects (constraints, accuracy, and decision variables) are fundamental to any optimizations [3].

In recent decades, the stochastic optimization method called metaheuristic has been employed to tackle various optimization problems. The pelican optimization algorithm has been used to detect and classify tuberculosis based on an X-ray image of the chest [4]. The red deer algorithm has been combined with deep learning to detect and classify diseases in agricultural plants [5]. Artificial rabbit optimization has been utilized to improve the routing process in wireless mesh networks [6]. Coot optimization has been used to optimize the task classification and scheduling in the cloud computing system to reduce the make-span [7].

The popularity of metaheuristics comes from its stochastic approach, which scans the solution randomly inside the space so that it does not trace all possible solutions through the iterative process [3]. This approach gives an advantage in avoiding excessive computational processes, especially in solving complex problems with many decision variables and a wide range of solution space. This advantage comes with the price that metaheuristic provides the best effort to achieve the quasi-optimal solution [2], and not the global optimal one.

The massive studies that have introduced metaheuristics in recent decades are highly related to the no-free-lunch (NFL) doctrine. This theory or doctrine states that there is no general technique or algorithm, i.e., metaheuristic, whose performance is superior to answer all problems. A metaheuristic may produce high-quality solutions in some issues while moderate or mediocre-quality solutions in others. This circumstance motivates many scientists to develop new metaheuristics and try to be better than the previous ones. In recent years, there have been two notable scientists who are active in the development of many metaheuristics: Mirjalili and Dehghani. Mirjalili was involved in a lot of metaheuristics, such as grey wolf algorithm (GWA) [8], marine predator algorithm (MPA) [9], cheetah optimizer (CO) [10], coronavirus optimization algorithm (COVIDOA) [11], geometric mean optimizer (GMO) [12], geyser inspired algorithm (GEA) [13], inclined planes system optimization (IPO) [14], and so on. Dehghani was involved in many swarm-based metaheuristics, such as northern goshawk optimization (NGO) [15], pelican optimization algorithm (POA) [16], language education optimization (LEO) [17], fully informed search algorithm (FISA) [18], coati optimization algorithm (COA) [19], zebra optimization algorithm (ZOA) [20], walrus optimization algorithm (WaOA) [21], average subtraction based optimization (ASBO) [22], three influential member-based optimization (TIMBO) [23], multileader optimization (MLO) [24], mixed leader based optimization (MLBO) [25], hybrid leader based optimization (HLBO) [26], and so on. Besides, there are also several scientists active in the development of metaheuristics. Kusuma has introduced several metaheuristics such as attack leave optimization (ALO) [27], total interaction algorithm (TIA) [28], modified social force algorithm (MSFA) [29], four-directed search algorithm (FDSA) [30], and so on. Braik was involved in the development of the chameleon swarm algorithm (CSA) [31], white shark optimizer (WSO) [32], and so on. There are also several new metaheuristics, such as golden search optimization (GSO) [33], Komodo mlipir algorithm (KMA) [34], remora optimization algorithm (ROA) [35], reptile search algorithm (RSA) [36] and so on.

In almost all swarm-based metaheuristics, the swarm is not split. This circumstance makes uniformity of action or reference for the whole swarm. This uniformity may lead to the risk of the entrapment of local optimal. Only a few swarm-based metaheuristics where the swarm is divided into sub-swarms. For example, the swarm is split into two equal-sized sub swarms in COA [19]. Meanwhile, in KMA, the swarm is divided into three sub swarms [34].

Regarding to this problem, this work is aimed to introduce a novel metaheuristic called swarm bipolar algorithm (SBA). As its name suggests, SBA is constructed based on swarm intelligence to consist of a certain number of autonomous agents. The bipolar term comes from the concept that the swarm is split into two equal-sized sub-swarms and a local leader in each sub-swarm is elected based on quality. Four references are utilized in SBA, the finest swarm member, the finest sub-swarm member, the middle between two finest sub-swarm members, and a randomly selected sub-swarm member from the opposite sub-swarm.

The objective of splitting the swarm and introducing multiple references as proposed in this work is to diversify the motion of the swarm. This diversification is designed to improve the exploration capability so that it can help the swarm to escape from the local optimal. Some swarm members may move toward the local optimal, but the others may move to other alternatives.

The scientific contributions of this work are mentioned as follows.

- A new metaphor-free swarm-based metaheuristic called SBA is presented including its concept and formalization.
- A local leader called the sub-swarm leader is introduced as a novel reference in the development of metaheuristics.
- A comprehensive review regarding the recent swarm-based metaheuristics is performed including their use of metaphor, strategy, and the existence of non-directed search.
- A benchmark assessment is conducted to investigate the performance of SBA by contending it with five recent swarm-based metaheuristics in solving the 23 functions.
- The effectiveness of each search in the proposed SBA is investigated through the individual search assessment.

The organization of the rest of this paper is as follows. A comprehensive review of the recent metaheuristics, especially the swarm-based metaheuristics, is presented in section two. Then, section three provides a detailed description of the proposed model, including the core concept, formalization, and mathematical formulation. It is followed by the assessment scenario and results conducted to investigate the performance of SBA. The comprehensive and in-depth analysis regarding the

Table 1. The mechanics of recent metaheuristics and the existence of swarm split.

No	Metaheuristic	Metaphor	Searching Strategy	Split Swarm
1	NGO [15]	northern goshawk	searching toward a randomly picked swarm member and searching around the current solution with declining local search space	no
2	LEO [17]	education	searching toward a randomly picked swarm member from the pool consisting of all better swarm members and the finest swarm member; searching relative to a randomly picked swarm member; searching around the current solution with declining local search space	no
3	COA [19]	coati	the first half of swarm members search toward the finest swarm member while the second half of swarm members search relative to a randomized solution within space; searching around the current solution with declining local search space	two equal size sub swarms
4	FISA [18]	-	searching toward the resultant of better swarm members and the finest swarm member and avoid the resultant of worse swarm members and the worst member	no
5	TIA [28]	-	searching relative to all other swarm members	no
7	ALO [27]	-	searching toward the finest swarm member or the finest swarm member avoids the swarm member; searching toward the target or the target avoids the member where the target is the middle between the finest swarm member and a randomly picked swarm member or the middle between two randomly picked swarm members; full random search when stagnation occurs	no
8	POA [16]	pelican	searching relative to a randomized solution within space; searching near current solution with declining local search space	no
9	ZOA [20]	zebra	searching toward the finest swarm member; searching around the current solution with declining local search space or searching toward a randomly selected swarm member.	no
10	TIMBO [23]	-	searching toward the finest swarm member; searching to avoid the worst swarm member; searching relative to the mean member	no
11	KMA [34]	Komodo	searching toward the resultant of better swarm members and avoid the resultant of worse swarm members in the high-quality group; crossover with the finest swarm member or full random search; searching toward the resultant of all swarm members in the high-quality group.	three sub swarms based on quality
13	GSO [33]	-	searching toward the mixture of the global finest unit and local finest unit	no
14	this work	-	searching toward the finest swarm member; searching toward the finest sub-swarm member; searching toward the middle between two finest sub-swarm members; searching relative to the randomly picked swarm member from the opposite sub swarm	two equal size sub swarms

outcome, limitations, and complexity is discussed in section five. In the end, the summary of the conclusion and track for further development is presented in section six.

2. Related Works

Swarm-based metaheuristic is a branch of metaheuristic search that is constructed by a certain number of autonomous agents. These agents are also called swarm members. A swarm member can also be called a unit or entity.

As autonomous agents, there is not any central command or entity that dictates the movement of the swarm members. But rather than moving sporadically, there is a collective intelligence or entity that becomes the reference during the searching so that there is possibility for convergence.

Received: December 15, 2023. Revised: January 23, 2024.

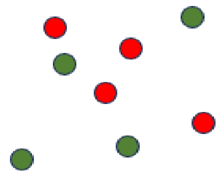

Figure 1. Illustration of two equal size sub swarms

The development of swarm-based metaheuristics cannot be separated from the construction of the reference. There are several common references, such as global finest unit, local finest unit, the finest swarm member, the worst swarm member, a randomly picked swarm member, the resultant of certain number of best members, a randomized solution within space, and so on. A reference can also be constructed based on the mixture of several common references, for example the mixture between the global finest unit and the local finest unit like in MPA. The reference can also be the mixture between the finest swarm member and a randomly picked member like in ALO.

There are also variations in the number of searches performed in the iteration phase. Many metaheuristics perform only a single search, like in GSO, TIA, etc. Meanwhile, many others perform multiple searches. These various searches can be achieved through sequential phases, stochastic options, segregation of roles, or a combination of them. The segregation of functions can also be described as a swarm split. The swarm is split into sub-swarms where each sub-swarm performs a specific search or searches. The review of some recent swarm-based metaheuristics can be seen in Table 1.

Based on the exhibition in Table 1, it is shown that there is rarity in splitting the swarm into sub swarms in the swarm-based metaheuristics. There is uniformity among swarm members in almost all swarm-based metaheuristics. In other word, there is not any segregation of roles or flow pattern among the swarm members. This circumstance becomes the opportunity to develop a new swarm-based metaheuristic that adopts swarm split approach and is competitive compared to the recent metaheuristics.

3. Model

The core concept of the swarm bipolar algorithm (SBA) is the splitting of the swarm into two equal-sized sub-swarms. It means the number of the members within each sub-swarm is equal. The splitting mechanism is simple where the first half of the population goes to the first sub-swarm and the second half of the population goes to the second sub swarm. It means that the location of the swarm members within the search space is not considered in the splitting process. This circumstance creates the possibility of cross place among the swarm members.

The distribution of the swam members based on their sub-swarm is illustrated in Fig. 1. In Fig. 1, the first sub-swarm is colored red, while the second sub-swarm is colored green. During the initialization phase, all swarm members are distributed uniformly within the search space. It means that the sub-swarm members are also distributed uniformly within the search space.

Four references are used in SBA, especially during the iteration's improvement effort. The first reference is the finest swarm member. The finest swarm member is the swarm member whose quality is the finest among the swarm members within the population. The second reference is the finest sub-swarm member. As there are two sub-swarms in the system, then there are two finest sub-swarm members. Each sub-swarm has its own finest sub-swarm member. The third reference is the middle between the two finest sub-swarm members. The fourth reference is a randomly picked swarm member from the opposite sub-swarm.

These references are used as guidance for directed searches. In SBA, four directed searches are performed sequentially by each swarm member in every iteration. The first search is the search toward the finest swarm member. The second search is the search for the finest sub-swarm member. Each sub-swarm member follows its own finest sub-swarm member. The third search is the search toward the middle between the two finest sub-swarm members. The fourth search is the search relative to a randomly picked sub-swarm member from the opposite sub-swarm.

The flow, trajectory, or convergence of each motion is described below. The motion toward the finest swarm member makes the trajectory of all swarm members to a specific area where the finest swarm member exists within the space, except the finest swarm member is updated. The motion toward the finest sub-swarm member makes the swarm distribution polarized into two clusters as the iteration increases. The motion toward the middle between the two finest sub-swarm members makes these two sub-swarms get close to each other. The motion relative to a randomly picked sub-swarm member from the opposite sub-swarm disrupts or breaks the polarization. The fourth search is the opposite of the second search. The illustration of these four searches is presented in Fig. 2.

The formalization of SBA based on this core concept is presented in pseudocode in algorithm 1.

Received: December 15, 2023. Revised: January 23, 2024.

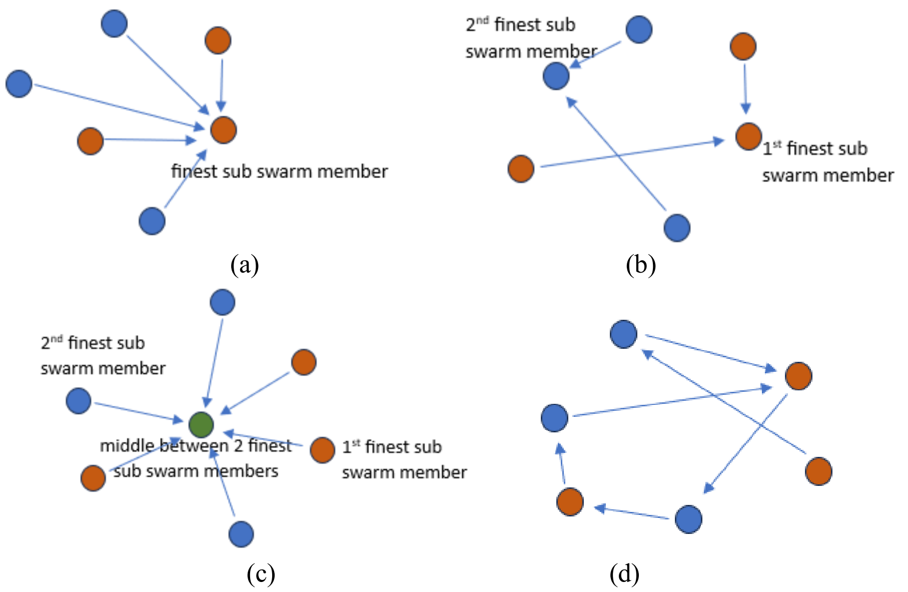

Figure 2. Illustration of four searches: (a) first search, (b) second search, (c) third search, and (d) fourth search

d	dimension
f	objective function
i	index for swarm member
j	index for dimension
s	swarm member
S	swarm/population
s_l	lower boundary
s_u	upper boundary
s_b	the finest swarm member
s_{sb}	the finest sub swarm member
s_t	randomly picked swarm member
r_1	floating point uniform random [0,1]
r_2	integer uniform random [1,2]
t	iteration
t_m	maximum iteration
U	uniform random

Then, Eq. (1) to Eq. (10) are used as mathematical formulation supporting the algorithm. Before that, the annotations used in this paper are listed below.

Eq. (1) to Eq. (4) is utilized during the initialization phase. Eq. (1) represents the uniform distribution used to generate initial solution of swarm members. Eq. (2) represents the strict acceptance role implemented to update the finest swarm member. Eq. (3) is used to update the first best sub swam member while Eq. (4) is used to update the second-finest sub swarm member.

$$s_{i,j} = s_{b,j} + r_1\left(s_{u,j} - s_{l,j}\right) \tag{1}$$

$$s'_b = \begin{cases} s_i, f(s_i) < f(s_b) \\ s_b, else \end{cases} \tag{2}$$

Algorithm 1: Swarm bipolar algorithm

begin
 for each $s \in S$ **do**
  generate initial solution using Eq. (1)
  update s_b and s_{sb} using Eq. (2) to Eq. (4)
 end for
 for t=1 to t_m **do**
  for each $s \in S$ **do**
   first search using Eq. (5) and Eq. (6)
   update s_b and s_{sb} using Eq. (2) to Eq. (4)
   second search using Eq. (7) and Eq. (6)
   update s_b and s_{sb} using Eq. (2) to Eq. (4)
   third search using Eq. (8) and Eq. (6)
   update s_b and s_{sb} using Eq. (2) to Eq. (4)
   fourth search using Eq. (9), Eq. (10), Eq. (6)
   update s_b and s_{sb} using Eq. (2) to Eq. (4)
  end for
 end for
 return s_b
end

$$s'_{sb1} = \begin{cases} s_i, f(s_i) < f(s_{sb1}) \wedge 1 \le i \le \frac{n(S)}{2} \\ s_{sb1}, else \end{cases} \tag{3}$$

$$s'_{sb2} = \begin{cases} s_i, f(s_i) < f(s_{sb2}) \wedge \frac{n(S)}{2} < i \le n(S) \\ s_{sb2}, else \end{cases} \tag{4}$$

Formalization through mathematical formulation during the improvement is presented in Eq. (5) to Eq. (10). Eq. (5) states that the first search is the search toward the finest swarm member. Eq. (6) represents the strict acceptance role in accepting the solution candidate to replace the current value of the swarm member. Eq. (7) generates the solution candidate of the second search where each sub-swarm member moves toward its own finest sub-swarm member. Eq. (8) is used for the third search, where all swarm members move toward the middle between the two finest sub-swarm members. Eq. (9) defines the randomly picked swarm member from the opposite sub-swarm. Eq. (10) represents the movement in the fourth search, which depends on the quality comparison between the swarm member and its reference.

$$c_{i,j} = s_{i,j} + r_1\left(s_{b,j} - r_2 s_{i,j}\right) \quad (5)$$

$$s_i' = \begin{cases} c_i, f(c_i) < f(s_i) \\ s_i, else \end{cases} \quad (6)$$

$$c_{i,j} = \begin{cases} s_{i,j} + r_1\left(s_{sb1,j} - r_2 s_{i,j}\right), 1 \le i \le \frac{n(S)}{2} \\ s_{i,j} + r_1\left(s_{sb2,j} - r_2 s_{i,j}\right), \frac{n(S)}{2} < i \le n(S) \end{cases} \quad (7)$$

$$c_{i,j} = s_{i,j} + r_1\left(\frac{s_{sb1,j}+s_{sb2,j}}{2} - r_2 s_{i,j}\right) \quad (8)$$

$$s_t = \begin{cases} U\left(s_1, s_{\frac{n(S)}{2}}\right), \frac{n(S)}{2} < i \le n(S) \\ U\left(s_{\frac{n(S)}{2}+1}, s_{n(S)}\right), 1 \le i \le \frac{n(S)}{2} \end{cases} \quad (9)$$

$$c_{i,j} = \begin{cases} s_{i,j} + r_1\left(s_{t,j} - r_2 s_{i,j}\right), f(s_t) < f(s_i) \\ s_{i,j} + r_1\left(s_{i,j} - r_2 s_{t,j}\right), else \end{cases} \quad (10)$$

4. Simulation and result

This section presents the assessment performed in this work to investigate the performance of SBA. The presentation includes the assessment scenario and its result. In this work, there are two assessments. The first assessment is the benchmark assessment. The second assessment is the single search assessment. The 23 classic functions are chosen as the optimization problems to answer. The population size and maximum iteration are set to 10. The dimensions for the high-dimension functions are 50. The detailed description of these 23 functions is described in Table 2.

The reasons for choosing these 23 functions are as follows. First, this set of functions covers various circumstances of problems. Seven functions are unimodal while fifteen functions are multimodal. Thirteen functions are high dimension functions where the dimension can be expanded to hundreds or thousands while ten functions are fixed dimension functions where the dimension ranges from two to four. Some functions have large space while the others have narrow search space. The global optimal of some functions is in the center of space while the global optimal of some functions is far from the center of the space. The terrain of some functions is smooth while the terrain of some others is ripple or flat with narrow and steep holes so that the ambiguity and the risk to be entrapped in the local optimal are high. Second, this set of functions has been used in many previous studies proposing metaheuristic, such as in TIA [28], KMA [34], GSO [33], TIMBO [23], and so on. Based on this explanation, the use of the set of 23 functions is acceptable.

Table 2. Detailed description of the set of 23 functions

No	Function	Dim	Space	Target
1	Sphere	50	[-100, 100]	0
2	Schwefel 2.22	50	[-100, 100]	0
3	Schwefel 1.2	50	[-100, 100]	0
4	Schwefel 2.21	50	[-100, 100]	0
5	Rosenbrock	50	[-30, 30]	0
6	Step	50	[-100, 100]	0
7	Quartic	50	[-1.28, 1.28]	0
8	Schwefel	50	[-500, 500]	-12,569
9	Ratsrigin	50	[-5.12, 5.12]	0
10	Ackley	50	[-32, 32]	0
11	Griewank	50	[-600, 600]	0
12	Penalized	50	[-50, 50]	0
13	Penalized 2	50	[-50, 50]	0
14	Shekel Foxholes	2	[-65, 65]	1
15	Kowalik	4	[-5, 5]	0.0003
16	Six Hump Camel	2	[-5, 5]	-1.0316
17	Branin	2	[-5, 5]	0.398
18	Goldstein-Price	2	[-2, 2]	3
19	Hartman 3	3	[1, 3]	-3.86
20	Hartman 6	6	[0, 1]	-3.32
21	Shekel 5	4	[0, 10]	-10.153
22	Shekel 7	4	[0, 10]	-10.402
23	Shekel 10	4	[0, 10]	-10.536

The objective of the benchmark assessment is to investigate the improvement of SBA compared to the existing metaheuristics. The improvement is measured based on the fitness score produced by SBA and its contenders. There are five recent metaheuristics chosen as contenders: NGO, LEO, COA, FISA, and TIA. All these contenders are new. NGO is firstly introduced in 2021. Meanwhile, LEO, COA, FISA, and TIA are firstly introduced in 2023. Among them, TIA is the only contender that performs single search only.

The result of benchmark assessment is presented in Table 3 to Table 6. Table 3 exhibits the result in solving seven high dimension unimodal functions.

Table 3. Performance comparison in solving high-dimension unimodal functions

F	Parameter	NGO [15]	LEO [17]	COA [19]	FISA [18]	TIA [28]	SBA
1	mean	3.1769×10^3	1.3765×10^1	6.2789×10^2	4.6075×10^4	8.5371	0.0000
	std deviation	2.5467×10^3	6.5385	1.6041×10^2	5.8965×10^3	1.9850	0.0000
	mean rank	5	3	4	6	2	1
2	mean	0.0000	0.0000	0.0000	0.0000	3.3572×10^{49}	0.0000
	std deviation	0.0000	0.0000	0.0000	0.0000	1.5014×10^{50}	0.0000
	mean rank	1	1	1	1	6	1
3	mean	7.7811×10^4	8.5848×10^3	2.3711×10^4	1.9106×10^5	1.5739×10^3	2.5562
	std deviation	2.4361×10^4	6.3199×10^3	1.1636×10^4	7.6949×10^4	1.1638×10^3	4.0294
	mean rank	5	3	4	6	2	1
4	mean	4.0413×10^1	2.7670	1.9054×10^1	8.3108×10^1	2.1499	0.0020
	std deviation	1.2050×10^1	0.6905	5.4061	1.2122×10^1	0.4355	0.0006
	mean rank	5	3	4	6	2	1
5	mean	8.3463×10^5	1.9511×10^2	6.2683×10^4	1.3468×10^8	1.7922×10^2	4.8908×10^1
	std deviation	7.1789×10^5	9.8121×10^1	4.8483×10^4	1.1863×10^8	3.6422×10^1	0.0540
	mean rank	5	3	4	6	2	1
6	mean	2.8397×10^3	4.1105×10^2	6.8246×10^2	4.9715×10^4	1.4421×10^1	9.9386
	std deviation	1.1134×10^3	1.0552×10^2	2.9752×10^2	7.6410×10^3	3.0113	0.4509
	mean rank	5	3	4	6	2	1
7	mean	0.9113	0.0347	0.2595	1.1789×10^2	0.0382	0.0074
	std deviation	0.6996	0.0256	0.0927	9.4495×10^1	0.0206	0.0046
	mean rank	5	2	4	6	3	1

Table 4. Performance comparison in solving high-dimension multimodal functions

F	Parameter	NGO [15]	LEO [17]	COA [19]	FISA [18]	TIA [28]	SBA
8	mean	-3.1630×10^3	-3.8791×10^3	-4.1617×10^4	-3.0392×10^3	-2.2075×10^3	-3.6188×10^3
	std deviation	6.3406×10^2	6.5381×10^2	4.7473×10^2	4.9193×10^2	2.6773×10^2	4.5629×10^2
	mean rank	4	2	1	5	6	3
9	mean	4.2954×10^2	2.3564×10^2	1.5968×10^2	5.8325×10^2	2.9265×10^1	0.0000
	std deviation	3.3106×10^1	5.7377×10^1	4.2894×10^1	3.9627×10^1	2.6773×10^1	0.0000
	mean rank	5	4	3	6	2	1
10	mean	9.0689	1.2556	5.4447	1.8997×10^1	1.1498	0.0003
	std deviation	1.6713	0.4761	0.6268	0.4463	0.2564	0.0000
	mean rank	5	3	4	6	2	1
11	mean	2.6605×10^1	0.9168	6.4970	4.9144×10^2	0.7448	0.0017
	std deviation	1.0326×10^1	0.2761	1.0889	1.9029×10^2	0.1724	0.0077
	mean rank	5	3	4	6	2	1
12	mean	2.1022×10^4	1.0606	7.1373	2.6702×10^8	0.6422	0.9527
	std deviation	4.0023×10^4	0.1766	2.2877	2.6727×10^8	0.1577	0.1560
	mean rank	5	3	4	6	1	2
13	mean	8.8259×10^5	4.1531	1.5266×10^3	7.0558×10^8	3.3652	3.1133
	std deviation	1.5116×10^6	0.3432	1.8255×10^3	6.0505×10^8	0.2347	0.0267
	mean rank	5	3	4	6	2	1

Table 4 exhibits the result in solving six high dimension multimodal functions. Table 5 exhibits the result in solving ten fixed dimension multimodal functions. There are three parameters presented in Table 3 to Table 5: average fitness score (mean), standard deviation, and mean rank. The result more precise than 10^{-4} is rounded to nearest 10^{-4}.

Table 3 exhibits the fine quality of SBA in solving the high dimension unimodal functions. SBA becomes the best performer in solving all seven functions. Meanwhile, SBA becomes the distinct best performer in seven functions (f_1, f_3, f_4, f_5, f_6, and f_7). In f_2, SBA and four contenders (NGO, LEO, FISA, and TIA) achieve the same result. Moreover, SBA can find the global optimal solution for two functions (f_1 and f_2). The performance difference between SBA and its contenders are also wide enough.

Table 4 also exhibits the fine quality and competitiveness of SBA in solving six high dimension multimodal functions. SBA becomes the best performer in solving four functions (f_9, f_{10}, f_{11}, and f_{13}), second best performer in solving f_{12}, and third best performer in solving f_8. SBA achieves the

Received: December 15, 2023. Revised: January 23, 2024.

Table 5. Performance comparison in solving fixed dimension multimodal functions

F	Parameter	NGO [15]	LEO [17]	COA [19]	FISA [18]	TIA [28]	SBA
14	mean	8.8582	7.4578	5.5187	1.3382×10^1	8.1209	7.7745
	std deviation	3.9616	3.6826	3.5836	9.5784	3.0713	4.3806
	mean rank	5	2	1	6	4	3
15	mean	0.0196	0.0030	0.0056	0.0190	0.0011	0.0024
	std deviation	0.0154	0.0029	0.0071	0.0165	0.0016	0.0072
	mean rank	6	3	4	5	1	2
16	mean	-0.9937	-1.0284	-1.0211	-0.9452	-1.0300	-1.0296
	std deviation	0.0543	0.0068	0.0188	0.15670	0.0045	0.0053
	mean rank	5	3	4	6	1	2
17	mean	0.6691	0.4016	0.4094	0.4986	0.4517	0.4410
	std deviation	0.4112	0.0043	0.0227	0.2857	0.1941	0.0900
	mean rank	6	1	2	5	4	3
18	mean	9.2298	3.0645	4.7666	1.8245×10^1	6.5924	8.6078
	std deviation	1.3899×10^1	0.1273	5.8123	2.5476×10^1	8.0834	1.0457×10^1
	mean rank	5	1	2	6	3	4
19	mean	-0.0495	-0.0495	-0.0495	-0.0409	-0.0495	-0.0495
	std deviation	0.0000	0.0000	0.0000	0.0145	0.0000	0.0000
	mean rank	1	1	1	1	1	1
20	mean	-2.3514	-3.1139	-2.9608	-2.6931	-2.9109	-2.6972
	std deviation	0.4349	0.0910	0.1621	0.3569	0.2603	0.3590
	mean rank	6	1	2	5	3	4
21	mean	-1.1890	-2.7277	-2.3776	-1.8543	-3.7564	-4.1828
	std deviation	0.5785	1.2160	1.0716	1.5026	1.5494	1.4770
	mean rank	6	3	4	5	2	1
22	mean	-1.4904	-3.3370	-2.6343	-1.8842	-2.9086	-3.4780
	std deviation	0.6338	1.5012	1.2486	0.9484	1.0830	0.9810
	mean rank	6	2	4	5	3	1
23	mean	-1.9497	-3.4386	-2.9586	-2.2705	-2.6710	-3.6457
	std deviation	1.1982	1.4197	0.7477	1.8491	0.7562	1.6614
	mean rank	6	2	3	5	4	1

global optimal solution in solving f_9. The performance difference between the best performer and worst performer is wide in five functions (f_9 to f_{13}). Meanwhile, the performance difference between the best and worst performers is narrow in solving f_8.

Table 5 shows that the competition in solving the fixed dimension multimodal functions is fair and fierce. The performance difference between the best performer and the worst performer is narrow. SBA becomes the distinct best performer in solving only three functions (f_{21}, f_{22}, and f_{23}). SBA becomes the second-best performer in solving two functions (f_{15} and f_{16}), third best performer in two functions (and f_{17}), and fourth best performer in two functions (f_{18} and f_{20}).

Table 6 shows that SBA is superior to its contenders by being better than NGO, LEO, COA, FISA, and TIA in 12, 16, 16, 21, and 18 functions respectively. SBA is superior to its contenders, especially in solving the high dimension functions, whether they are unimodal functions or multimodal ones. Meanwhile, SBA is still superior to NGO and FISA in solving fixed dimension multimodal functions. On the other hand, the advantage of SBA compared to LEO, COA, FISA, and TOA is moderate in this group of functions as it is just better than its contenders only a half of functions.

Table 6. Group-based superiority of SBA.

Group	Number of Functions Where SBA is Better				
	NGO [15]	LEO [17]	COA [19]	FISA [18]	TIA [28]
1	6	6	6	6	7
2	6	5	5	6	6
3	9	5	5	9	5
Total	21	16	16	21	18

The second assessment is performed to investigate the contribution of each strategy or search in solving the 23 functions. This assessment is taken by activating only one search in one session. This assessment is called single search assessment. There are four searches investigated individually as there are four searches constructing the SBA. The result is presented in Table 7. The parameter in the average

Table 7. Result of single search assessment.

F	Average Fitness Score			
	First Search	Second Search	Third Search	Fourth Search
1	4.7456×10^1	8.9700×10^1	**4.0432×10^1**	1.4106×10^3
2	**0.0000**	**0.0000**	**0.0000**	**0.0000**
3	6.9866×10^3	1.1660×10^4	**3.8096×10^3**	6.7888×10^4
4	5.1749	8.3354	**3.6263**	4.1386×10^1
5	8.3665×10^2	2.6969×10^3	**5.5967×10^2**	2.7002×10^5
6	5.8532×10^1	1.0003×10^2	**4.2632×10^1**	1.7101×10^3
7	0.0512	0.0767	**0.0507**	0.5691
8	-2.8361×10^3	-2.7252×10^3	-2.4696×10^3	**-3.2389×10^3**
9	7.3568×10^1	7.2062×10^1	**6.3026×10^1**	3.8596×10^2
10	2.4511	3.0938	**2.0068**	8.0702
11	1.4487	1.8927	**1.2687**	1.5584×10^1
12	**1.3074**	2.0209	1.3277	1.8699×10^3
13	5.5797	7.3361	**4.7624**	1.2002×10^5
14	**9.1193**	1.0638×10^1	1.1543×10^1	1.2422×10^1
15	0.0109	**0.0098**	0.0136	0.0159
16	-1.0031	-1.0015	-0.9546	**-1.0043**
17	1.0163	1.7132	1.1056	**0.6455**
18	**1.4423×10^1**	1.5454×10^1	3.7207×10^1	1.4602×10^1
19	**-0.0495**	**-0.0495**	**-0.0495**	**-0.0495**
20	**-2.5300**	-2.2300	-2.2664	-2.3487
21	**-1.7997**	-1.7831	-1.6781	-1.1580
22	-1.7807	-1.6873	**-1.9820**	-1.5496
23	**-2.5561**	-1.8042	-2.1085	-1.5993

fitness score. The best result in every function is written in bold font.

Table 7 exhibits that the third search offers the highest contribution among the others. There are two functions where all searches achieve the same result (f_2 and f_{19}). By neglecting these two functions, the third search achieves the distinct best result in 11 functions. Then, the first search follows as the second best by achieving distinct best result in six functions. The fourth search achieves distinct best result in three functions and the second search achieves distinct best result in one function.

5. Discussion

In general, the benchmark assessment result shows that SBA is highly accepted metaheuristic due to its superiority among its contenders. The significant superiority of SBA, especially in solving both high dimension unimodal functions and high dimension multimodal functions means that SBA has superior both exploitation and exploration capabilities. Meanwhile, the fierce competition in the fixed dimension multimodal functions shows that all metaheuristics in this assessment have equal capability on balancing the exploitation and exploration. The superior performance in the first group of functions proofs the superior exploitation capability while the superior performance in the second group of functions proofs the exploration capability. In the end, SBA has a balance between exploitation and exploration as it is competitive in the third group of functions.

The single assessment result indicates that the mixture between two finest sub-swarm members is better than the finest swarm member as reference. As mentioned previously, the finest swarm member as a dedicated reference has been used in many swarm-based metaheuristics, such as COA [19], ZOA [20], KMA [34], WaOA [21], TIMBO [23], and so on. This result poses that the mixture of the finest swarm member with other entities can be explored deeper to construct a better reference. Due to the nature of the strict acceptance method, the finest swarm member always comes from the finest sub-swarm members. It means that the reference in the third search is the finest swarm member with another finest sub-swarm member. Meanwhile, there is not any guarantee that this another finest sub-swarm member is the second-finest swarm member. In some circumstance, a sub-swarm may consist of better swarm members than another sub-swarm. But there is a guarantee that the finest sub-swarm member will always the best among at least half of population of the swarm. The note is the performance difference between the first search and the third search is not wide. This fact can be used as baseline to construct a new entity as reference that far better than the finest swarm member.

Received: December 15, 2023. Revised: January 23, 2024.

The comparison between the first search and second search exposes some other findings. The superiority of the third search is in the high dimension functions, whether they are unimodal or multimodal functions. It means that the third search has superior both exploration and exploitation capabilities, relative to the first search. On the other hand, the first search has comparative advantage in balancing its exploitation and exploitation capabilities as its superiority occurs generally in fixed dimension multimodal functions.

The comparison between the second search and the fourth search exposes some findings. The contribution of both searches is less dominant rather than the first and third searches. The second search becomes the distinct best performer only in one function while the fourth search becomes the distinct best performer only in three functions. This achievement occurs in the multimodal functions. It means that the contribution of the second and fourth search is mostly in the exploration rather than in the exploitation.

Despites the superior performance in most of functions, SBA still has weaknesses. The narrow performance difference between SBA and its contenders in solving the fixed dimension multimodal functions exposes the limited improvement of SBA. In some functions, the contenders are better than SBA. This fact also strengthens the NFL theory. Moreover, SBA is still far from the global finest unit after the iteration stops like in Shekel 5, Shekel 7, and Shekel 10. This limitation can be seen as potential prospect for development of more powerful metaheuristic in the future studies.

There are also limitations in the performance assessment. First, SBA is tested to answer theoretical optimization problems which are represented by the 23 functions. In these functions, the precision of the solution does matter as these problems are numerical problems with floating point representation. In the real world, the solution may be presented in the integer number or floating-point number with limited decimal points. These problems can be found in many hardware related system where the design cannot be highly precise so that it is impossible to manufacture. This circumstance makes the wide performance difference becomes more difficult to achieve.

The second limitation is that there are several other sets of functions that can be used to assess the performance of any metaheuristics. These sets are the series of IEEE CEC functions. For example, COA was tested by using CEC 2011 in its first appearance [19]. GEA was tested by using CEC 2005, CEC 2014, and CEC 2017 [13]. NGO was tested by using CEC 2015 and CEC 2017 besides the classic 23 functions in its first appearance [15]. It is shown that IEEE CEC functions can be used as alternatives to assess the performance of any optimization method. But it is impossible to conduct all IEEE CEC series in a single paper. Based on this explanation, the future studies can be performed by assessing SBA or the modification of SBA by using any IEEE CEC series.

The computational complexity of SBA can be traced based on the use of loop inside the algorithm. The nested loop for whole swarm as outer loop and whole dimension as inner loop is performed in both initialization and iteration phases. Meanwhile, the loop representing the iteration to the maximum iteration covering the previously explained nested loop is only performed in the iteration phase. There is only one search in the initialization phase which is the full random search within the space. On the other hand, there are four sequential directed searches in the iteration phase performed by each swarm member. This explanation becomes the reasoning that the complexity during the initialization phase is $O(n(S).d)$ while the complexity during the iteration phase is $O(4.t_m.n(S).d)$.

6. Conclusion

This paper presents a new metaphor-free metaheuristic search called swarm bipolar algorithm (SBA). The fundamental concept of SBA comes from the splitting of the swarm into two equal size sub swarms. Besides focusing on the intensification within every sub swarm, the diversification is also performed by interaction with entities from the opposite sub swarm. The benchmark assessment has been performed to investigate the performance of SBA as a whole package by contending it with five recent swarm-based metaheuristics. The result shows that SBA is better than NGO, LEO, COA, FISA, and TIA in 21, 16, 16, 21, and 18 functions respectively. The superiority of SBA comes especially in solving the multimodal problems where the performance difference between SBA and the worst performer is wide. Meanwhile, the individual search assessment result shows that the search toward the middle between two finest sub-swarm members performs the best among the searches which also means better than the search toward the finest swarm member.

Further studies are important in developing the transformation of SBA to answer combinatorial optimization problems which are different from the numerical optimization problems like the 23 functions. The combinatorial problems can be found in many works, especially in scheduling, assignment, timetabling, and so on. These combinatorial problems

 DOI: 10.22266/ijies2024.0430.31

have different challenges from the numerical optimization problems where the significant improvement is not so easy as in the numerical optimization problems.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

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

Acknowledgments

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

References

[1] T. R. Farshi, "Battle Royale Optimization Algorithm", *Neural Computing and Applications*, Vol. 33, pp. 1139-1157, 2021.

[2] 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.

[3] E. Trojovska, M. Dehghani, and V. Leiva, "Drawer Algorithm: A New Metaheuristic Approach for Solving Optimization Problems in Engineering", *Biomimetics*, Vol. 8, ID. 239, pp. 1-35, 2023.

[4] K. Manivannan and S. Sathiamoorthy, "Pelican Optimization with Majority Voting Ensemble Model for Tuberculosis Detection and Classification on Chest X-Ray Images", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 5, pp. 1-10, 2023, doi: 10.22266/ijies2022.1231.14.

[5] D. Raja and M. Karthikeyan, "Red Deer Optimization with Deep Learning Enabled Agricultural Plant Disease Detection and Classification Model", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 5, pp. 21-30, 2023, doi: 10.22266/ijies2023.1031.03.

[6] K. N. Prema and M. B. Ushadevi, "Effective Routing Using Multi-Objective Levy flight-Artificial Rabbit Optimization Algorithm for Wireless Mesh Networks", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 5, pp. 666-676, 2023, doi: 10.22266/ijies2023.1031.56.

[7] S. Karimunnisa and Y. Pachipala, "Task Classification and Scheduling Using Enhanced Coot Optimization in Cloud Computing", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 5, pp. 501-511, 2023, doi: 10.22266/ijies2023.1031.43.

[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 System with Applications*, Vol. 152, ID: 113377, 2020.

[10] 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.

[11] A. M. Khalid, K. M. Hosny, and S. Mirjalili, "COVIDOA: A Novel Evolutionary Optimization Algorithm Based on Coronavirus Disease Replication Lifecycle", *Neural Computing and Applications*, Vol. 34, pp. 22465-22492, 2022.

[12] F. Rezaei, H. R. Safavi, M. A. Elaziz, and S. Mirjalili, "GMO: Geometric Mean Optimizer for Solving Engineering Problems", *Soft Computing*, Vol. 27, pp. 10571–10606, 2023.

[13] M. Ghasemi, M. Zare, A. Zahedi, M. A. Akbari, S. Mirjalili, and L. Abualigah, "Geyser Inspired Algorithm: A New Geological-inspired Meta-heuristic for Real-parameter and Constrained Engineering Optimization", *Journal of Bionic Engineering*, in press, 2023.

[14] A. Mohammadi, F. Sheikholeslam, and S. Mirjalili, "Nature-Inspired Metaheuristic Search Algorithms for Optimizing Benchmark Problems: Inclined Planes System Optimization to State-of-the-Art Methods", *Archives of Computational Methods in Engineering*, Vol. 30, pp. 331–389, 2023.

[15] 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.

[16] P. Trojovsky and M. Dehghani, "Pelican Optimization Algorithm: A Novel Nature-Inspired Algorithm for Engineering

Received: December 15, 2023. Revised: January 23, 2024.

Applications", *Sensors*, Vol. 22, ID: 855, pp. 1-34, 2022.

[17] P. Trojovský, M. Dehghani, E. Trojovská, and E. Milkova, "Language Education Optimization: A New Human-Based Metaheuristic Algorithm for Solving Optimization Problems", *Computer Modeling in Engineering & Sciences*, Vol. 136, No. 2, pp. 1527-1573, 2023.

[18] M, Ghasemi, A. Rahimnejad, E. Akbari, R. V. Rao, P. Trojovský, E. Trojovská, and S. A. Gadsden, "A New Metaphor-Less Simple Algorithm Based on Rao Algorithms: A Fully Informed Search Algorithm (FISA)", *PeerJ Computer Science*, ID. 1431, pp. 1-24, 2023.

[19] 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.

[20] 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.

[21] P. Trojovsky and M. Dehghani, "A New Bio Inspired Metaheuristic Algorithm for Solving Optimization Problems based on Walruses Behavior", *Scientific Reports*, Vol. 13, ID. 8775, pp. 1-32, 2023.

[22] 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.

[23] F. A. Zeidabadi, M. Dehghani, and O. P. Malik, "TIMBO: Three Influential Members Based Optimizer", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 5, pp. 121-128, 2021, doi: 10.22266/ijies2021.1031.12.

[24] 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.

[25] 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.

[26] M. Dehghani and P. Trojovský, "Hybrid Leader Based Optimization: A New Stochastic Optimization Algorithm for Solving Optimization Applications", *Scientific Reports*, Vol. 12, No. 1, pp. 1–16, 2022.

[27] 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, doi: 10.22266/ijies2023.0630.19.

[28] 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.

[29] 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, doi: 10.22266/ijies2022.0630.25.

[30] P. D. Kusuma and A. Dinimaharawati, "Four Directed Search Algorithm: A New Optimization Method and Its Hyper Strategy Investigation", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 5, pp. 598–611, 2023, doi: 10.22266/ijies2023.1031.51.

[31] 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.

[32] M. Braik, A. Hammouri, J. Atwan, M. A. A. Betar, and M. A. Awadallah, "White Shark Optimizer: A Novel Bio-Inspired Meta-Heuristic Algorithm for Global Optimization Problems", *Knowledge-Based Systems*, Vol. 243, ID. 108457, pp. 1-29, 2022.

[33] 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.

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

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

[36] L. Abualigah, M. A. Elaziz, P. Sumari, Z. W. Geem, and A. H. Gandomi, "Reptile Search Algorithm (RSA): A Nature-Inspired Meta-Heuristic Optimizer", *Expert Systems with*

Applications, Vol. 191, ID. 116158, pp. 1-33, 2022.