Received: January 26, 2024. Revised: February 20, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Swarm Space Hopping Algorithm: A Swarm-based Stochastic Optimizer Enriched with Half Space Hopping Search

Purba Daru Kusuma[1]* **Meta Kallista[1]**

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

Abstract: Many recent swarm-based metaheuristics are trapped in the exploitation of the highest quality as the main or the only reference and the neighbourhood search with the reduction of local search space during the iteration. Regarding to this issue, this paper introduces a novel metaheuristic called swarm space hopping algorithm (SSHA). SSHA consists of three searches. First, a directed search toward the highest quality is performed. Second, the directed search toward the resultant of better agents or away from the other agent is performed. Third, the arithmetic crossover between the agent and a randomized solution in the first half or second half of space is performed. In this work, three evaluations are performed to assess the performance of SSHA. The first evaluation is the benchmark evaluation to compare the performance of SSHA with other recent metaheuristics: northern goshawk optimization (NGO), zebra optimization algorithm (ZOA), clouded leopard optimization (CLO), osprey optimization algorithm (OOA), and total interaction algorithm (TIA). The result exhibits that SSHA is better than NGO, ZOA, CLO, OOA, and TIA in 21, 20, 17, 17, and 21 functions. In the second evaluation, the individual search evaluation to compare the contribution between the first and second searches is performed, with the result that the second search outperforms the first search. The third evaluation is performed to assess the contribution of the third search in the optimization process, and the result shows that the contribution of the third search is significant only in three functions.

Keywords: Optimization, Stochastic, Metaheuristic, Highest quality, Neighbourhood search.

1. Introduction

Optimization is an effort to find the highest quality solution among a certain number of available solutions. This highest quality result is measured by an objective function, whether it is maximization or minimization. In the context of maximization, the highest quality solution is one that produces the highest score, such as profit, power, accuracy, and so on. On the other hand, in the minimization context, the highest quality solution is a solution that produces the lowest score, such as cost, loss, and so on.

In optimization studies, metaheuristic is a popular method that has been utilized in many sectors, especially engineering. Pelican optimization algorithm (POA) has been utilized in the machine learning-based face emotion recognition with the accuracy is 99% for various emotions [1]. Artificial bee colony (ABC) has been hybridized with Jaya algorithm to optimize the synthesis of linear antenna array with the optimized parameters are position, phase, and amplitude [2]. Giant trevally optimizer (GTO) has been introduced and utilized to find the highest quality position and size of the capacitor banks in the low voltage electrical distribution system with the objectives are minimizing operational cost and energy loss [3]. The elephant herd optimization (EHO) has been combined with graph similarity to classify the email [4]. The dandelion optimization algorithm (DOA) has been combined with deep learning to recognize and grade diabetic disease through fundus images [5].

In recent years, there have been plenty of new metaheuristics developed using swarm intelligence approach. Most of them utilized the behavior of animals as fundamental concept and metaphors due to the similarity of swarm intelligence and the animal behavior during searching for food or mating. Some of these animal inspired swarm-based metaheuristics are northern goshawk optimization (NGO) [6], zebra optimization algorithm (ZOA) [7], clouded leopard optimization (CLO) [8], coati optimization algorithm (COA) [9], Tasmanian devil optimization (TDO) [10], pelican optimization algorithm (POA) [11], fennec fox optimization (FFO) [12], cheetah optimization (CO) [13], walrus optimization algorithm (WaOA) [14], white shark optimization (WSO) [15], green anaconda algorithm (GAO) [16], red fox optimization (RFO) [17], golden jackal optimization (GJO) [18], and so on. Besides animal behavior, several metaheuristics imitate the social behavior, such as migration algorithm (MA) [19], mother optimization algorithm (MOA) [20], chef-based optimization algorithm (CBOA) [21], driving training-based optimization (DTBO) [22], sewing training-based optimization (STBO) [23], election-based optimization algorithm (EBOA) [24], and so on. Some swarm-based metaheuristics are free from metaphors, such as total interaction algorithm (TIA) [25], average and subtraction-based optimization (ASBO) [26], attack leave optimization (ALO) [27], four-directed search algorithm (FDSA) [28], golden search optimization (GSO) [29], and so on. Some metaphors promoted the references used in their directed search for their name, such as mixed leader-based optimization (MLBO) [30], hybrid leader-based optimization (HLBO) [31], multi leader optimization (MLO) [32], three influential members-based optimizations (TIMBO) [33], and so on.

Despite the massive development of swarm-based metaheuristics, there is dependence of these metaheuristics to the highest quality solution or highest quality agent. Many of them utilize the highest quality agent as one of their references in their directed search, such as in ASBO [26], COA [9], ZOA [7], GAO [16], GSO [29], and so on. Although the exploitation of the highest quality agent is acceptable, there should be an effort to introduce an alternative reference and a performance comparison between the highest quality and the alternatives. Unfortunately, many studies proposing new metaheuristic with some alternatives despite the highest quality agent, have not measured the performance of the alternative. In these studies, the performance measurement is conducted for the whole metaheuristic. It makes the performance of the alternatives has not been investigated yet.

The neighborhood search with the reduced local space during the iteration is a popular additional technique in many recent swarm-based metaheuristics. This technique was first introduced in the marine predator algorithm (MPA). Then, many metaheuristics proposed by Dehghani utilize this technique too, such as in POA [11], MA [19], WaOA [14], MOA [20], DTBO [22], and so on. This circumstance becomes an opportunity to propose another additional search besides the neighborhood search.

Based on this problem, this work is aimed at introducing a new swarm-based metaheuristics designed to provide alternatives for the use of the highest quality solution and the neighborhood search with declining local search space. This metaheuristic is called space hopping algorithm (SSHA). As the name suggests, SSHA is designed as a swarm-based metaheuristic consisting of a certain number of autonomous agents that hop from one place to another in the search space. It consists of two directed searches and one crossover-based search. The first directed search is the motion toward the highest quality agent. The second directed search is the motion toward the resultant of better agents or away from a randomly selected agent. Crossover-based search is the crossover between the agent and a randomized solution in the first half or second half of the space.

The scientific contribution of this work is as follows:

- A novel metaphor-free swarm-based metaheuristic is introduced which consists of three searches (two directed searches and one crossover-based search).
- A novel directed search which is the motion toward the resultant of better agents or away from a randomly selected agent is introduced as a complement and benchmark for the motion toward the highest quality solution.
- A new search called hopping search is introduced which is the crossover between the agent and a randomized solution within the first half or second half of the space.
- A benchmark test is conducted to assess the performance of SSHA in handling 23 classic functions and compare its performance with five recent metaheuristics.
- Individual search tests are performed to compare the performance of the first search and second search.
- Missed search test is performed to assess the contribution of the third search.

The rest of this paper is arranged as follows. The recent studies proposing new metaheuristics are reviewed in section two. The presentation of the proposed metaheuristics, including the fundamental concept and reasoning, formalization through pseudocode and the mathematical formulation are described in section three. The evaluation to

Received: January 26, 2024. Revised: February 20, 2024.

investigate the performance of SSHA including the result is shown in section four. The more comprehensive investigation including the result, relation with the theory, strength and weakness, computational complexity, and limitations of this work are discussed in section five. The conclusion and the suggestion for further studies are presented in section six.

2. Related works

Many swarm-based metaheuristics rely on the highest quality solution or highest quality agent as their reference for their directed search. Some metaheuristics utilize the highest quality solution in a dedicative manner. On the other hand, some metaheuristics mix the highest quality solution with other solutions.

In particle swarm optimization (PSO) [34] and GSO [29], the highest quality solution is mixed with the local highest quality solution. In MLBO, the highest quality agent is mixed with a randomized solution within space [30]. In HLBO, the highest quality agent is mixed with the randomly selected agent within the swarm [31]. In MLO, the highest quality agent is included in a pool consisting of some of the highest quality swarm members [32]. In TIMBO, the highest quality agent becomes one of the three references [33]. In some recent metaphor inspired metaheuristics, the highest quality agent is presented in specific metaphors, such as iguana on the three in COA [9], pioneer zebra in ZOA [7], one

Table 1. Several recent swarm-based metaheuristics

No	Metaheuristic	Directed Search	Additional Search
1	NGO [6]	motion relative to a randomly selected agent	neighborhood search with a declining local search space during iteration and a small local search space in the beginning
2	ZOA [7]	motion toward the highest quality agent; motion toward a randomly selected agent	neighborhood search with a declining local search space during iteration and a small local search space in the beginning
3	CLO [8]	motion relative to a randomly selected agent	neighborhood search with a declining local search space during iteration
4	OOA [35]	motion toward a randomly selected agent from a pool consisting of all better agents and highest quality agent	neighborhood search with a declining local search space during iteration
5	TIA [25]	motion relative to all other agents	-
6	TDO [10]	motion relative to a randomly selected agent	neighborhood search with a declining local search space during iteration; small local search space in the beginning; and the probability of conducting this search in every iteration is 0.5
7	ASBO [26]	motion relative to the middle between the highest quality and worst agents; motion toward the gap between the highest quality and worst agents; motion away from the highest quality agent	-
8	MOA [20]	motion toward the highest quality agent; motion to avoid a randomly selected agent from a pool consisting of the worse agents;	neighborhood search with a declining local search space during iteration
9	ALO [27]	motion of the agent toward the highest quality agent or motion of the highest quality agent away from the agent; motion of the second reference avoiding the agent or the agent avoiding the second reference where the second reference is the middle between the highest quality agent and a randomly selected agent or the middle between two randomly selected agents.	full random search if stagnation occurs
10	this work	motion toward the highest quality agent; motion toward the resultant of better agents or avoid a randomly selected agent	arithmetic crossover between the agent and a randomized solution in the first half or second half of the search space

Received: January 26, 2024. Revised: February 20, 2024.

of the female anaconda in GAO [16], the strongest WaOA [14], mother in MOA [20], one of the preys in TDO [10], one of the driving instructor in DTBO [22], one of the underwater fishes in OOA [35], and so on.

Besides the highest quality agent, there are other references used in the directed search in some recent swarm-based metaheuristics. Once again, in the metaphor-inspired ones, they are represented in some metaphors. In NGO, prey means a randomly selected agent within swarm [6]. In COA, iguana on the ground represents a randomized solution within the space [9]. In ZOA, the other zebra who attacks the lion represents a randomly selected agent within the swarm [7]. The prey in POA represents a randomized solution within space [11].

Many swarm-based metaheuristics are also enriched with additional search rather than directed search only. The neighborhood search or local search with declining local search space during the iteration becomes a popular method. This method was first introduced in MPA [36]. Then, some metaheuristics use this method with certain metaphors, like chasing and escaping in NGO [6], flying on the water surface in POA [11], carrying the caught fish in OOA [35], adapting to the new environment in MA [19], upbringing in MOA [20], and so on. Meanwhile, some swarm-based metaheuristics still consist of only directed search, such as in TIA [25], ASBO [26], GSO [29], GJO [18], and so on.

Table 1 summarizes the strategy performed by some recent swarm-based metaheuristics. In Table 1, the review is split between the directed searches performed by the related metaheuristics and the additional searches beside the directed searches. The last row consists of the strategy performed by the proposed metaheuristic in this paper.

Based on this explanation and strengthened with the summary in Table 1, the opportunity to propose a new reference in the directed search and new method as an additional search. As presented, the use of neighbourhood search is more popular than the crossover technique.

3. Model

SSHA is constructed based on the fundamental concept that all agents will hop from one location to another location within space. This hopping process is conducted by using directed search and crossover-based search. The neighborhood search is not implemented in SSHA.

SSHA performs three sequential steps during the search to improve the quality of the current agents. In the first step, the agent performs the directed search by moving toward the highest quality solution. In the second step, the agent performs the directed search by moving toward the resultant of the better agents or away from a randomly selected agent within the swarm. The third search is a crossover-based search of the agent with a randomized solution within the first half or second half of the search space. In SSHA, the rigid acceptance rule is implemented to prevent the agent from moving to the worse solution. This objective is obtained by accepting the solution child to replace the current solution only if the improvement takes place.

The first search, which is the motion toward the highest quality solution is used as this search is common in many swarm-based metaheuristics. This search is proven superior, especially for the exploitation purpose. Based on this consideration, the motion toward the highest quality solution is chosen to ensure the competitiveness of SSHA compared to the existing and recent swarm-based metaheuristics. As SSHA implements the rigid acceptance rule, the highest quality agent in the current iteration will always be the highest quality solution.

The second search is perceived as limited exploration. In this search, an agent will trace all other agents in the swarm. All other agents whose quality is better than the corresponding agent will be collected in a pool. Then, the resultant of these better agents is calculated based on the average location of them in every dimension without considering the normalized quality of each better agent. This resultant or average location can be calculated only if there is at least one better agent in the pool. Then, the agent moves toward this resultant of better agents. But there is a case where there are not any better agents that can be found. It means that this corresponding agent is the highest quality agent. When this circumstance happens, then an agent in the swarm is picked randomly to become the reference. As this randomly selected agent is worse than the corresponding agent, then the agent will avoid the randomly selected agent.

The third search is designed for adaptive search. If the improvement occurs after the agent performs the first and second searches, then the agent still focuses on searching in its current area. On the other hand, if the stagnation occurs after the agent performs the first and second searches, then the agent will move toward a certain location in another area. This movement is conducted by an equal arithmetic crossover between the agent and the reference. The reference is generated uniformly within the selected area.

In every dimension, the search space is split into two equal size areas: the first half of the space and the second half of the space. If the agent is in the first half of the space and improvement occurs, then the reference is generated within the first half of the space. Otherwise, the reference is generated within the second half of the space. If the agent is in the second half of the space and improvement occurs, then the reference is generated within the second half of the space. Otherwise, the reference is generated within the first half of the space.

The formal description of SSHA is presented through pseudocode and mathematical formulation. The pseudocode of SSHA is presented in algorithm 1. Meanwhile, the mathematical formulation of SSHA is presented in Eqs. (1)-(16). Below are the annotations used in this paper.

A	agent
A	swarm or set of agents
a_b	highest quality agent
a_l	lower boundary of the space
a_u	upper boundary of the space
a_m	middle of the space
a_{te1}	agent before the first search
a_{te2}	agent after the second search
a_{rb}	resultant of better agents
a_{rs}	randomly selected agent
a_{rc}	randomized agent for the space
a_{sc}	agent's child
D	dimension
F	objective function
I	index for agent
j	index for dimension
n	swarm size
r_1	real uniform random number [0, 1]
r_2	integer uniform random number [1, 2]
t	Iteration
t_m	maximum iteration
U	uniform random

The SSHA is constructed based on swarm intelligence. As a swarm, SSHA consists of a set of swarm or autonomous agents which is presented in Eq. (1).

$$A = \{a_1, a_2, a_3, \dots, a_n\} \tag{1}$$

In the initialization phase, all agents are generated randomly within the search space. This process follows uniform distribution so that all agents can be generated anywhere within the space in equal probability. It is formalized using Eq. (2). Then, each time an agent is generated, the highest quality agent is updated using Eq. (3). Eq. (3) represents the rigid acceptance rule.

algorithm 1: swarm space hopping algorithm

```
output: a_b
begin
 for all a in A
  initialization of a_i using Eq. (2)
  update a_b using Eq. (3)
 end for
 t=1
 while t ≤ t_m
  for all a in A
   run first search using Eqs. (4) and (5)
   run second search using Eqs. (6)-(10)
   run third search using Eqs. (11)-(18)
   update a_b using Eq. (2)
  end for
  t = t + 1
 end while
end
```

$$a_{i,j} = a_{l,j} + r_1\left(a_{u,j} - a_{l,j}\right) \tag{2}$$

$$a_b{}' = \begin{cases} a_i, f(a_i) < f(a_b) \\ a_b, else \end{cases} \tag{3}$$

The first search is formulated using Eqs. (4) and (5). Eq. (4) states that the first child is generated based on the motion toward the highest quality agent with a uniform step size. Then, Eq. (5) represents the rigid acceptance rule for replacing the current value of the agent with the first child.

$$a_{sc1,j} = a_{i,j} + r_1\left(a_{b,j} - r_2 a_{i,j}\right) \tag{4}$$

$$a_i' = \begin{cases} a_{sc1,j}, f(a_{sc1}) < f(a_i) \\ a_i, else \end{cases} \tag{5}$$

The second search is formulated using Eqs. (6)-(10). Eq. (6) represents the process of collecting all better agents into a pool. Eq. (7) formulates the resultant of better agents. Eq. (8) states that an agent is selected uniformly from the swarm to become the reference. Eq. (9) shows that the motion toward the resultant of better agents is performed if the pool consisting of the better agents is not empty. Otherwise, the motion avoiding the randomly selected agent is performed. Eq. (10) represents the rigid acceptance rule for the replacement of the current value of the agent with the second child.

$$A_{be,i} = \{a \mid a \in A, f(a) < f(a_i)\} \tag{6}$$

$$a_{rb,i,j} = \frac{\sum_{A_{be,i}} a_{be,i,j}}{n(A_{be,i})} \tag{7}$$

$$a_{rs,i} = U(A) \tag{8}$$

$$a_{sc2,j} = \begin{cases} a_{i,j} + a_{r1}\left(a_{rb,i,j} - r_2 a_{i,j}\right), A_{be,i} \neq \emptyset \\ a_{i,j} + a_{r1}\left(a_{i,j} - r_2 a_{rs,i,j}\right), else \end{cases} \tag{9}$$

$$a_i' = \begin{cases} a_{sc2,j}, f(a_{sc2}) < f(a_i) \\ a_i, else \end{cases} \tag{10}$$

The third search is formulated using Eqs. (11)-(17). Eq. (11) is used to calculate the middle point in every dimension. Eq. (12) states that the first hop reference is generated uniformly within the first half of the space. Eq. (13) states that the second hop reference is generated uniformly within the second half of the space. Eq. (14) shows that the improvement-based reference is chosen in another area of the agent. Eq. (15) shows that the stagnation-based reference is chosen in the same area as the agent. Eq. (16) states that the reference of the third search is chosen based on the condition of stagnation or improvement. Eq. (17) represents the balance arithmetic crossover between the agent and the third reference. Eq. (18) represents the rigid acceptance rule in the third search.

$$a_{m,j} = \frac{a_{l,j} + a_{u,j}}{2} \tag{11}$$

$$a_{fh,j} = a_{l,j} + r_1\left(a_{m,j} - a_{l,j}\right) \tag{12}$$

$$a_{sh,j} = a_{m,j} + r_1\left(a_{u,j} - a_{u,j}\right) \tag{13}$$

$$a_{im,i,j} = \begin{cases} a_{fh,j}, a_{l,j} \leq a_{i,j} \leq a_{m,j} \\ a_{sh,j}, a_{m,j} < a_{i,j} \leq a_{u,j} \end{cases} \tag{14}$$

$$a_{sg,i,j} = \begin{cases} a_{sh,j}, a_{l,j} \leq a_{i,j} \leq a_{m,j} \\ a_{fh,j}, a_{m,j} < a_{i,j} \leq a_{u,j} \end{cases} \tag{15}$$

$$a_{th,i,j} = \begin{cases} a_{im,i,j}, f\left(a_{te2,j}\right) < f(a_{te1,j}) \\ a_{sg,i,j}, else \end{cases} \tag{16}$$

$$a_{sc3,j} = \frac{a_{i,j} + a_{th,i,j}}{2} \tag{17}$$

$$a_i' = \begin{cases} a_{sc3,j}, f(a_{sc3}) < f(a_i) \\ a_i, else \end{cases} \tag{18}$$

4. Simulation and result

There are three assessments in this work to evaluate the performance of SSHA. The first assessment is the benchmark test to compare the performance of SSHA as a whole package with five recent metaheuristics. The second assessment is the individual search test to compare the performance of the first search with the second search. The third assessment is the missed search test to measure the contribution of the third search.

The reasoning behind these assessments is as follows. The first assessment is conducted to measure the improvement of the current work, i.e., the proposed SSHA in the development of metaheuristics. Due to this consideration, SSHA should be benchmarked with several recent metaheuristics rather than the older ones such as PSO. The second assessment is conducted to measure the performance of the first and second searches as a single search. By drawing back to the objective of this work, this second assessment is important to measure the comparative performance of the second search which use the resultant of the better agents or the randomly selected agent with the first search whose reference is the highest quality agent. The third assessment is conducted to measure the contribution of the third search in SSHA.

The mechanism of each assessment is as follows. In the first assessment, SSHA is benchmarked with five recent swarm-based metaheuristics: NGO, ZOA, CLO, OOA, and TIA. NGO, ZOA, and were first introduced in 2022. Meanwhile, OOA and TIA were first introduced in 2023. In the second assessment, both the first and second searches are tested individually. In the third assessment, SSHA is tested with the third search is inactive. This third assessment is performed with this mechanism because the option of whether a randomized solution is generated in the first half, or second half of the search space depends on whether the improvement takes place after the first and second half searches are conducted. It means that the third search cannot be conducted independently like in the second assessment. In all three assessments, the set of 23 functions is chosen as the use case. The detailed description of these functions is exhibited in Table 2. These functions can be grouped as seven high dimension unimodal functions (HDUF), six high dimension multimodal functions (HDMF), and ten fixed dimension multimodal functions (FDMF). In this assessment, the swarm size is set to 5 while the maximum iteration is set to 10.

The result of the first assessment is presented in Table 3 to Table 6. Table 3 to Table 5 exhibit the results in solving HDUF, HDMF, and FMDF consecutively. In these tables, there are three parameters: the average fitness score (mean), standard deviation, and the mean rank. Then, the result in these tables is summarized in Table 6, that presents the superiority of SSHA compared to the benchmark metaheuristics based on the number of

Table 2. List of 23 Functions

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

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

F	Parameter	NGO [6]	ZOA [7]	CLO [8]	OOA [35]	TIA [25]	SSHA
1	mean	6.6935x10^{3}	1.5097x10	1.4629x10^{3}	2.4215x10^{2}	3.8223x10	0.4651
	std deviation	3.8910x10^{3}	1.0371x10	5.8192x10^{2}	1.3656x10^{2}	1.1235x10	0.4238
	mean rank	6	2	5	4	3	1
2	mean	2.8971x10^{41}	0.0000	0.0000	0.0000	0.0000	0.0000
	std deviation	1.3276x10^{42}	0.0000	0.0000	0.0000	0.0000	0.0000
	mean rank	6	1	1	1	1	1
3	mean	1.2614x10^{5}	6.5189x10^{3}	7.0659x10^{4}	2.4752x10^{4}	4.3191x10^{3}	1.0452x10^{3}
	std deviation	6.0468x10^{4}	4.7197x10^{3}	3.4872x10^{4}	1.3948x10^{4}	3.7101x10^{3}	1.3509x10^{3}
	mean rank	6	3	5	4	2	1
4	mean	5.0224x10	2.9712	4.5555x10	1.3173x10	4.5422	0.5342
	std deviation	1.6462x10	0.8237	1.3980x10	3.6644	0.9554	0.2291
	mean rank	6	2	5	4	3	1
5	mean	2.4020x10^{6}	2.5396x10^{2}	4.4654x10^{5}	1.4721x10^{4}	6.9186x10^{2}	5.4297x10
	std deviation	1.9270x10^{6}	1.5894x10^{2}	5.5740x10^{5}	1.3323x10^{4}	3.2103x10^{2}	4.2573
	mean rank	6	2	5	4	3	1
6	mean	6.2219x10^{3}	2.4536x10	1.5854x10^{3}	2.3537x10^{2}	4.4562x10	1.1527x10
	std deviation	4.4632x10^{3}	8.8552	9.9245x10^{2}	1.2769x10^{2}	1.2788x10	0.7695
	mean rank	6	2	5	4	3	1
7	mean	2.5952	0.0597	0.6881	0.1789	0.0960	0.0376
	std deviation	2.0984	0.0418	0.4793	0.0828	0.0830	0.0256
	mean rank	6	2	5	4	3	1

functions where SSHA is better than the benchmark metaheuristics in every group of functions.

The result of the benchmark test on solving the high dimension unimodal indicates the superiority of SSHA among its benchmarks. SSHA is placed on the first rank in all seven functions. SSHA is on the distinct first rank in six functions (Sphere, Schwefel 1.2, Schwefel 2.21, Rosenbrock, Step, and Quartic). Meanwhile, four benchmarks (ZOA, CLO, OOA< and TIA) are also on the first rank in solving Schwefel 2.22. NGO is the only benchmark that fails to find global optimal solution in solving Schwefel

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

F	Parameter	NGO [6]	ZOA [7]	CLO [8]	OOA [35]	TIA [25]	SSHA
8	mean	-2.9010×10^3	-2.6135×10^3	-3.7549×10^3	-3.2200×10^3	-2.2191×10^3	-2.7835×10^3
	std deviation	4.8799×10^2	4.3289×10^2	7.4709×10^2	5.3006×10^2	4.9380×10^2	3.5067×10^2
	mean rank	3	5	1	2	6	4
9	mean	4.5224×10^2	3.4043x10	3.2789×10^2	1.4311×10^2	9.9151x10	0.6584
	std deviation	5.1058x10	2.6274x10	5.8803x10	5.2123x10	5.9712x10	0.8328
	mean rank	6	2	5	4	3	1
10	mean	1.2389x10	1.2686	8.3157	4.1323	2.1621	0.1436
	std deviation	2.6812	0.4219	2.0913	0.7806	0.2054	0.0950
	mean rank	6	2	5	4	3	1
11	mean	4.8793x10	0.9348	1.5395x10	2.6572	1.2710	0.1774
	std deviation	2.5282x10	0.2986	7.0892	0.6676	0.1375	0.2236
	mean rank	6	2	5	4	3	1
12	mean	1.5424×10^6	1.2421	1.2127×10^4	3.5396	1.1535	1.0838
	std deviation	4.4874×10^6	0.2281	3.5356×10^4	1.0136	0.2782	0.1215
	mean rank	6	3	5	4	2	1
13	mean	7.0407×10^6	4.1103	7.6172×10^5	3.2147×10^2	4.8336	3.4418
	std deviation	9.2873×10^6	0.5167	1.4166×10^6	1.4743×10^3	0.7374	0.1574
	mean rank	6	2	5	4	3	1

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

F	Parameter	NGO [6]	ZOA [7]	CLO [8]	OOA [35]	TIA [25]	SSHA
14	mean	2.8906x10	1.2015x10	6.2461	1.1465x10	1.2062x10	8.6731
	std deviation	5.2816x10	3.6126	3.6463	5.9028	4.2835	3.5572
	mean rank	6	4	1	3	5	2
15	mean	0.0359	0.0087	0.0195	0.0206	0.0135	0.0099
	std deviation	0.0260	0.0194	0.0223	0.0280	0.0229	0.0224
	mean rank	6	1	4	5	3	2
16	mean	-0.8334	-0.8073	-1.0175	-1.0120	-0.9713	-0.9824
	std deviation	0.2356	0.2698	0.0180	0.0213	0.1205	0.0756
	mean rank	5	6	1	2	4	3
17	mean	1.4290	1.7241	0.5310	0.4299	1.8852	0.4524
	std deviation	1.7531	2.5879	0.2960	0.0539	2.3960	0.1380
	mean rank	4	5	3	1	6	2
18	mean	3.0933x10	5.1131x10	1.3631x10	9.8168	2.7783x10	1.6619x10
	std deviation	2.0710x10	9.0647x10	1.2212x10	1.8432x10	3.4359x10	1.4874x10
	mean rank	5	6	2	1	4	3
19	mean	-0.0495	-0.0495	-0.0495	-0.0495	-0.0495	-0.0495
	std deviation	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	mean rank	1	1	1	1	1	1
20	mean	-2.1506	-2.1701	-2.6642	-2.7675	-2.2435	-2.8172
	std deviation	0.4684	0.5728	0.3864	0.3433	0.4664	0.2644
	mean rank	6	5	3	2	4	1
21	mean	-0.9393	-1.7589	-2.1284	-1.6190	-1.3853	-2.8429
	std deviation	0.6434	0.8859	0.9312	0.8122	0.6772	1.4576
	mean rank	6	3	2	4	5	1
22	mean	-1.0776	-1.6840	-2.3917	-1.5679	-1.8269	-2.9753
	std deviation	0.7077	0.9235	1.1228	0.6123	0.9398	1.4986
	mean rank	6	4	2	5	3	1
23	mean	-1.1107	-1.8512	-1.9761	-2.0995	-2.0372	-3.1037
	std deviation	0.3108	1.1212	0.6019	0.4490	1.0836	1.7178
	mean rank	6	5	4	2	3	1

Table 6. Group based superiority of SSHA

Group	Number of Functions Where SSHA is Better				
	NGO [6]	ZOA [7]	CLO [8]	OOA [35]	TIA [25]
1	7	6	6	6	6
2	5	6	5	5	6
3	9	8	6	6	9
Total	21	20	17	17	21

Table 7. Individual search of the first and second searches

Function	First Search	Second Search
1	$1.2953x10^2$	**7.6227x10**
2	0.0000	0.0000
3	$1.4930x10^4$	**$1.0691x10^4$**
4	**1.0467x10**	1.1976x10
5	**$3.3259x10^3$**	$4.1609x10^3$
6	$1.3882x10^2$	**8.7650x10**
7	0.1322	**0.1019**
8	**$-2.5414x10^3$**	$-2.3756x10^3$
9	**$1.1454x10^2$**	$1.3746x10^2$
10	3.2360	**2.8068**
11	1.9605	**1.7657**
12	2.2572	**1.8758**
13	9.8091	**6.6766**
14	**1.1927x10**	1.9388x10
15	0.0248	**0.0213**
16	-0.9447	**-0.9699**
17	4.9858	**2.3364**
18	3.0237x10	**1.6916x10**
19	-0.0495	-0.0495
20	-1.9298	**-2.0196**
21	**-1.2960**	-1.2029
22	-1.2082	**-1.3758**
23	**-1.5572**	-1.5025

2.22. The gap between SSHA as the highest quality metaheuristic and the worst metaheuristic in solving HDUF is wide.

The result of the benchmark test on solving the high dimension multimodal functions still indicates the superiority of the SSHA among its benchmarks. SSHA is on the first rank on five functions (Rastrigin, Ackley, Griewank, Penalized, and Penalized 2) out of six functions. Meanwhile, SSHA is on the fourth rank after CLO, OOA, and NGO in solving Schwefel. The performance gap between the highest quality performer and the worst performer in the high dimension multimodal functions is also wide except in solving Schwefel.

Table 5 shows that SSHA is still competitive in solving the fixed dimension multimodal functions. Among these ten functions, SSHA becomes the highest quality performer in five functions, the second-highest quality performer in three functions, and the third highest quality performer in two functions. It means that SSHA is never in the lower half of the metaheuristics involved in this test. SSHA is the highest quality performer in solving Hartman 3, Hartman 6, Shekel 5, Shekel 7, and Shekel 10. But the note is that all metaheuristics achieve the same result in solving Hartman 3. It means that SSHA becomes the distinct highest quality of only four functions. SSHA is the second-highest quality performer in solving Shekel Foxholes, Kowalik, and Branin. Meanwhile, SSHA becomes the third highest quality performer in solving Six hump camel and Goldstein price. Different from the first and second groups, the performance gap between the highest quality and worst performers in this third group is narrow.

Table 8. Result of the third search

F	with 3rd Search	without 3rd Search	Significantly Affected
1	0.4651	0.3264	no
2	0.0000	0.0000	no
3	$1.0452x10^3$	$1.3164x10^3$	no
4	0.5342	0.7527	no
5	5.4297x10	5.3576x10	no
6	1.1527x10	1.1072x10	no
7	0.0376	0.0380	no
8	$-2.7835x10^3$	$-3.0083x10^3$	no
9	0.6584	1.6059	yes
10	0.1436	0.1277	no
11	0.1774	0.1529	no
12	1.0838	1.0717	no
13	3.4418	3.4149	no
14	8.6731	1.1034x10	no
15	0.0099	0.0128	no
16	-0.9824	-0.9128	no
17	0.4524	4.1126	yes
18	1.6619x10	3.9495x10	yes
19	-0.0495	-0.0495	no
20	-2.8172	-2.2825	no
21	-2.8429	-1.8243	no
22	-2.9753	-2.0512	no
23	-3.1037	-1.7698	no

Table 6 enhances the superiority of SSHA among its benchmarks. Overall, SSHA is better than NGO, ZOA, CLO, OOA, and TIA in 21, 20, 17, 17, and 21 functions respectively. It means that SSHA is superior to NGO, ZOA, and TIA in almost all functions. SSHA is never worse than TIA as SSHA and TIA achieve equal results in two functions. SSHA is worse than ZOA only in one function and SSHA is worse than NGO only in two functions. SSHA is superior to CLO and OOA in almost all functions in the first and second groups of functions. Meanwhile, SSHA is slightly superior to CLO and OOA in the third group of functions. Although the total number of functions where SSHA is better than

CLO is equal with the total number of functions where SSHA is better than OOA, the composition of these functions is different in the fixed dimension multimodal functions as it can be seen in Table 5. This circumstance also occurs between NGO and TIA. The number of functions where SSHA is better than NGO is equal to the number of functions where SSHA is better than TIA. But the composition of these functions is different as the summary can be seen in Table 6 and the detail rank can be seen in Table 3 to Table 5.

The result of the second assessment is presented in Table 7. The only parameter presented in Table 7 is the average fitness score obtained from the individual search test. The second row presents the result regarding the first search, while the third row presents the result regarding the second search. The better result is presented in bold font.

Table 7 indicates that the second search is superior to the first search. The first search is better than the second search in seven functions, while the second search is better than the first search in fourteen functions. In two functions, both searches achieve the equal result. The superiority of the second search to the first search occurs in all groups of functions.

Table 8 presents the result of the third test. Same as the second test, the only parameter is the average fitness score. Table 8 consists of four columns: the function, the average fitness score where all searches are active, the average fitness score with the absence of the third search, and the status whether the performance of SSHA drops significantly in the absence of the third search.

Table 8 indicates that the presence of the third search is not so significant as the first and second searches. The performance of SSHA drops significantly only in three functions where all of them are multimodal functions.

5. Discussion

This section presents the comprehensive analysis regarding the results. There are four aspects discussed in this section. The first aspect is the performance of SSHA in solving the 23 functions and the comparison with the benchmark metaheuristics. The second aspect is the result of the individual search test that compares the first directed search with the second directed search. The third aspect is the result of the missed search to evaluate the contribution of the crossover-based search. The fourth aspect is the computational complexity of the SSHA. The fifth aspect is the limitations of this work, especially the SSHA.

The superiority of SSHA in all groups of functions can be perceived as superiority of exploration and exploitation capabilities of SSHA. The superiority of SSHA in solving the high dimension unimodal functions indicates that SSHA has superior exploitation capability. The superiority of SSHA in solving the high dimension multimodal functions indicates that SSHA has superior exploration capability, especially in avoiding the local optimal entrapment. The competitiveness of SSHA in solving the fixed dimension multimodal functions indicates that SSHA has a good balance between exploration and exploitation capabilities.

The result in the second test shows that the combination between the resultant of better agents and a randomly selected agent is more powerful than the highest quality agent. This result also meets the one of the objectives of this work which is creating an alternative reference despite being the highest quality agent.

The computational complexity is important part in the investigation of any algorithm as it affects the computational consumption. Specifically, the computational complexity is important to investigate which parameters increase the computational process and how the computational process grows due to the change of the value of these parameters. When all agents are firstly generated or initialized, there is a loop that runs based on the swarm size where each iteration represents the generation of an agent. Then, within this iteration, there is a loop that runs for the whole dimension to generate the initial value of each dimension regarding this agent. Based on this analysis, the complexity during the initialization is presented as $O(n(S)\times d)$. Then, when the optimization process comes for the iteration process, there is a loop that runs from the first iteration until the maximum iteration. Then, within this iteration, there is a loop that runs for the whole swarm which means that all agents perform searching process. Each agent performs the three searches. Within every search, tracing for whole dimension is performed. Meanwhile, in the second search, each agent will trace the whole swarm to collect all better agents and find its resultant. Based on this explanation, the computational complexity during the iteration process can be presented as $O(t_m\times n(S)\times(2+n(S))\times d)$.

Despite the superiority of SSHA and its successful attempt as a new metaheuristic, there are limitations exposed in this work. These limitations can be split between the algorithm and the evaluation. There are various stochastic approaches that already exist, but it is impossible to accommodate all these approaches. SSHA uses only the uniform distribution although there are other stochastic distributions, such as normal, Poisson, exponential, Levy, Brownian, and so on. Each of them has its own characteristic, strength, and weakness. Applying other stochastic distribution into SSHA will be interesting work. Meanwhile, there are also a lot of entities that can be chosen as reference despite the highest quality agent which is proven inferior to the combination of the resultant of better agents and a randomly selected agent. Moreover, there are various local searches and evolutionary-based search can be further explored. In the use case aspect, there are several other sets of functions already exist and available to be chosen as a theoretical use cases. Moreover, there are also various practical problems can be chosen to investigate the performance of SSHA in the more comprehensive manner, whether these problems are numerical problems or combinatorial problems.

6. Conclusion

This paper has presented the proposed swarm-based metaheuristic called space hopping algorithm (SSHA). The fundamental concept of SSHA has been presented which is accommodating both directed search and arithmetic crossover. There are three references used in the directed search: highest quality agent, the resultant of better agents, and a randomly selected agent. The result of the benchmark evaluation shows that SSHA is superior to its five benchmarks by outperforming NGO, ZOA, CLO, OOA, and TIA in 21, 20, 17, 17, and 21 functions out of 23 functions consecutively. The result of the comparison test between the first search and second search shows that the combination of the resultant of better agents and a randomly selected agent is better than the highest quality agent as the second search achieves better result in 14 functions and same result in only 2 functions. The result of the significance test shows that the contribution of the crossover search run in the third search is less significant as the directed search is proven to be more dominant as the existence of the third result improves the quality of the solution significantly only in 3 functions. Fortunately, this crossover search has potential as a replacement for the neighbourhood search with declining local search space as it is used in many recent swarm-based metaheuristics.

In the future, this proposed SSHA can be refined in several ways. The directed search toward the resultant of better agents or away from the randomized worse agent can be improved by mixing it with other references. Moreover, more exploration of arithmetic crossover should be improved to make it more significant compared to the directed search. Besides, more evaluations, especially based on the practical problems, are needed to investigate its strengths and weaknesses in a more comprehensive manner.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

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

Acknowledgments

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

References

[1] M. Sumithra and N. Rajkumar, “A Systematic Pelican Optimization based Weight Extreme Learning Machine Algorithm for Face Emotion Recognition”, *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 5, pp. 539-551, 2023, doi: 10.22266/ijies2023.1031.46.

[2] A. M. Alturfi, S. Goyal, and A. Kaur, “Hybrid Artificial Bee Colony and JAYA Algorithm for Linear Antenna Array Synthesis”, *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 4, pp. 669-681, 2023, doi: 10.22266/ijies2023.0831.54.

[3] K. Bhavana, V. Rajeswari, V. Rajan, V. R. Velmurugan, and P. Muthukumar, “An Efficient Hybrid Approach for Optimal Integration of Capacitors in Radial Distribution Networks with Realistic Load Models Using Giant Trevally Optimizer and Voltage Stability Index”, *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 4, pp. 181-190, 2023, doi: 10.22266/ijies2023.0831.15.

[4] B. A. Kumara and M. M. Kodabagi, “Email Net: Efficient Email Classification Approach Based on Graph Similarity Measure”, *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 4, pp. 512-522, 2023, doi: 10.22266/ijies2023.0831.41.

[5] R. Ramesh and S. Sathiamoorthy, “Blood Vessel Segmentation and Classification for Diabetic Retinopathy Grading Using Dandelion

Optimization Algorithm with Deep Learning Model", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 5, pp. 11-20, 2023, doi: 10.22266/ijies2023.1031.02.

[6] 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.

[7] 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.

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

[9] 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.

[10] M. Dehghani, S. Hubalovsky, and P. Trojovsky, "Tasmanian Devil Optimization: A New Bio-Inspired Optimization Algorithm for Solving Optimization Algorithm", *IEEE Access*, Vol. 10, pp. 19599-19620, 2022.

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

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

[13] 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, ID. 10953, pp. 1-20, 2022.

[14] 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.

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

[16] M. Dehghani, P. Trojovsky, and O. P. Malik, "Green Anaconda Optimization: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems", *Biomimetics*, Vol. 8, ID. 121, pp. 1-60, 2023.

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

[18] N. Chopra and M. M. Ansari, "Golden Jackal Optimization: A Novel Nature-Inspired Optimizer for Engineering Applications", *Expert Systems with Applications*, Vol. 198, ID. 116924, pp. 1-15, 2022.

[19] P. Trojovsky and M. Dehghani, "Migration Algorithm: A New Human-Based Metaheuristic Approach for Solving Optimization Problems", *Computer Modeling in Engineering & Sciences*, Vol. 137, No. 2, pp. 1695-1730, 2023.

[20] Matoušová, P. Trojovský, M. Dehghani, E. Trojovská, and J. Kostra, "Mother Optimization Algorithm: A New Human-Based Metaheuristic Approach for Solving Engineering Optimization", *Scientific Reports*, Vol. 13, ID. 10312, pp. 1-26, 2023.

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

[22] 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.

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

[24] 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.

[25] 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.

[26] 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.

[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. 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.

[29] 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.

[30] F. A. Zeidabadi, S. A. 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.

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

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

[33] 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.

[34] D. Freitas, L. G. Lopes, and F. Morgado-Dias, "Particle Swarm Optimisation: A Historical Review Up to the Current Developments", *Entropy*, Vol. 22, No. 3, ID. 362, pp. 1-36, 2020.

[35] M. Dehghani and P. Trojovsky, "Osprey Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Solving Engineering Optimization Problems", *Frontiers in Mechanical Engineering*, Vol. 8, ID. 1126450, pp. 1-43, 2023.

[36] 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.