Triple Shake Algorithm: A New Metaheuristic with Strict and Cross Dimension Mappings

Purba Daru Kusuma, *Member IAENG*

***Abstract*—There are two problems in the development of swarm-based metaheuristic. First, there are not any metaheuristic is able to solve all problems superiorly. Second, the cross-dimension mapping between the entity and its reference during performing the directed search is rare to find. Based on these problems, this work introduces a new metaphor-free swarm-based metaheuristic called the triple-shake algorithm (TSA). As its name suggests, TSA consists of three directed searches. The reference in the first search is the balance mixture between the finest entity and the member of permutation set. The reference in the second search is the balance mixture between the finest entity and a randomly chosen entity. The reference in the third search is the finest entity only. But the cross-dimension mapping is performed in this third search with 50 percent probability. In the benchmark assessment, TSA is compared with zebra optimization algorithm (ZOA), walrus optimization algorithm (WaOA), migration algorithm (MA), total interaction algorithm (TIA), and one-to-one based optimization (OOBO). The result indicates the dominance of TSA among its comparators. TSA is better than ZOA, WaOA, MA, TIA, and OOBO in 21, 19, 17, 19, and 22 functions consecutively out of 23 functions.**

***Index Terms*—optimization, metaheuristic, swarm intelligence, cross dimension mapping.**

I. Introduction

METAHEURISTIC has been implemented in a wide spectrum of optimization problems. Particle swarm optimization (PSO) has been merged with genetic algorithm (GA) and back propagation algorithm to forecast the 3D surface roughness in milling industry [1]. The non-dominated sorting genetic algorithm (NSGA II) has been employed to optimize the agricultural product price recommendation problem where some factors, such as product quality, production level, business competition, risk, and profitability are considered [2]. Run-catch optimization (RCO) has been introduced and employed to solve the outsourcing optimization problem which is a problem in production system [3]. PSO also has been utilized to optimize the online train trajectory planning in which it is proven to catch up the schedule and improve the energy consumption saving [4]. African vulture optimization algorithm (AVOA) has been utilized to improve the power quality of charging station unit which is critical in electric vehicle (EV) system [5]. Bat algorithm (BA) has been employed to optimize the tuning parameters in the combined proportional integral differential (PID) controller and battery energy storage system (BESS) in the load frequency control (LFC) to minimize the frequency oscillation in a power system [6].

There are a lot of swarm-based metaheuristics firstly introduced in 2021 until today. Some metaheuristics introduced in 2021 are chameleon swarm algorithm (CSA) [7], northern goshawk optimization (NGO) [8], mixed leader-based optimization (MLBO) [9], battle royale optimization (BRO) [10], coronavirus herd immunity optimization (CHIO) [11]. Some metaheuristics introduced in 2022 are election-based optimization algorithm (EBOA) [12], Komodo mlipir algorithm (KMA) [13], golden search optimization (GSO) [14], hybrid leader-based optimization (HLBO) [15], clouded leopard optimization (CLO) [16], average subtraction-based optimization (ASBO) [17], golden jackal optimization (GJO) [18], and so on. Some metaheuristics introduced in 2023 are total interaction algorithm (TIA) [19], coati optimization algorithm (COA) [20], walrus optimization algorithm (WaOA) [21], migration algorithm (MA) [22], one-to-one based optimization (OOBO) [23], fully informed search algorithm (FISA) [24], lyrebird optimization algorithm (LOA) [25], language education optimization (LEO) [26], kookaburra optimization algorithm (KOA) [27], attack leave optimization (ALO) [28], mother optimization algorithm (MOA) [29], osprey optimization algorithm (OOA) [30], archery algorithm (AA) [31], four directed search algorithm (FDSA) [32], geyser inspired algorithm (GEA) [33], Nizar optimization algorithm (NOA) [34], and so on.

Many swarm-based metaheuristics utilize the finest swarm member within the population as a reference or a component to construct the reference used for the directed search. This approach can be found in cheetah optimization (CO) [35], KMA [13], COA [20], and so on. The other concern is that almost all swarm-based metaheuristics employ strict dimension mapping approach. In the strict dimension mapping, the dimension of the related entity is mapped to the same dimension of its reference. Meanwhile, the cross-dimensional mapping approach is rare to find. One example of metaheuristic employing the cross-dimensional mapping approach is NOA [34]. Moreover, according to the no-free-lunch (NFL) theory, there is not any metaheuristic that can solve all optimization problems effectively and superiorly compared to other metaheuristics [35]. This theory has become the main or primary factor for many scientists to develop new metaheuristics, whether it is conducted by

Manuscript received November 27, 2023; revised March 10, 2024. This work was financially supported by Telkom University, Indonesia.

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

creating a brand new one or hybridizing some metaheuristics.

Motivated by this problem, this work is aimed at introducing a novel swarm metaheuristic called triple shake algorithm (TSA). TSA is designed as a multiple search algorithm so that it consists of more than one search. The motivation of this approach is related to the NFL theory that each search has its own strengths and weaknesses. By employing multiple searches, the weakness of one search can be covered by the other searches.

The scientific contributions of this work are listed as follows.

1. This work introduces a new swarm-based metaheuristic that employs both strict dimension mapping and cross dimension mapping within its directed searches.
2. The performance of this proposed metaheuristic is investigated by confronting it with five metaheuristics that were introduced in 2022 or 2023.
3. The individual search assessment is also performed to investigate the contribution of each search employed in the proposed metaheuristic.
4. The hyperparameter assessment is performed too to investigate the improvement of TSA due to the change of the adjusted parameters.

The arrangement of the rest of this paper is as follows. The second section reviews the recent development of swarm-based metaheuristics, especially the new ones which were introduced in 2023. Section three presents the inspiration, fundamental concept, and formulation of the proposed TSA. Section four presents the performance evaluation of TSA, including the evaluation scenario and the result. Section five discusses the findings, limitations, and computing complexity of TSA. Section six summarizes the concluding remarks and the proposal for future studies.

II. Related Works

The swarm-based metaheuristic is a branch of metaheuristic that employs both population-based system and multi-agent system. As a population-based system, swarm-based metaheuristic consists of multiple solutions rather than a single solution which is commonly found in some old metaheuristics, such as simulated annealing (SA), tabu search (TS), variable neighborhood search (VNS), and so on. As a multi agent system, the swarm-based metaheuristic consists of certain number of autonomous agents that acts independently without any central command. This approach is different from other population-based metaheuristics such as genetic algorithm, where the improvement is controlled by central entity, such as in selecting the parents to breed new children.

PSO is the early version of swarm-based metaheuristic. Its mechanism is simple, and it adopts the flocking pattern of birds during flight. Its simplicity can be seen from its single search which is the directed search. Each entity moves toward the mixture of the global finest entity and local finest entity with certain speed [36]. In its classic form, PSO does not employ stringent acceptance rules so that the entity will move to its new location although this new location does not provide improvement.

Then, the massive development of swarm intelligence or swarm-based metaheuristic evolve its mechanism. There are a lot of innovations performed by many scientists in developing new swarm-based metaheuristics. By abstracting the metaphors used as their inspiration, many swarm-based metaheuristics can be observed based on some aspects, such as the interaction with the finest entity, the interaction with other entities, the use of neighborhood search, the acceptance role, and the dimension mapping between the entity and its reference during the directed search. The summary of some swarm-based metaheuristics which were firstly introduced in 2023 is presented in Table 1.

Some swarm metaheuristics employ multiple search strategy while some others still employ single search strategy. Many metaheuristics associated with Dehghani employ multiple search strategy, such as COA [20], ZOA [37], WaOA [21], MA [22], and so on. Some other metaheuristics also employ multiple search strategy, such as ALO [28], FDSA [32], KMA [13], and so on. In some metaheuristics, the multiple search strategy is followed by the split or roles for the members of the swarm. In this case, some members perform certain searches while others perform other searches. For example, in the first phase of COA, the first half of the population performs the motion toward the finest entity while the second half of the population performs the motion relative to a randomized solution within space [20]. In KMA, the swarm is split into three groups based on their quality. The first group consists of high-quality entities [13]. The second group consists of moderate-quality entities [13]. Meanwhile, the third group consists of poor-quality entities [13]. In some metaheuristics, such as ZOA [37], WaOA [21], MA [22], and OOA [30], the multiple search strategy is implemented into multiple phases of searches. Meanwhile, some metaheuristics still employ single search strategy, such as OOBO [23], GSO [14], FISA [24], CO [35], TIA [19], and so on.

TABLE I
List of Some Swarm Metaheuristics Which is Firstly Introduced in 2023

No	Metaheuristic	Interaction with The Finest Entity	Interaction with Other Entities	Neighborhood Search	Stringent Acceptance	Multiple Strategy	Cross Dimension Mapping
1	TIA [19]	yes	yes	no	yes	no	no
2	ALO [28]	yes	yes	no	yes	yes	no
3	FDSA [32]	yes	yes	no	yes	yes	no
4	WaOA [21]	yes	yes	yes	yes	yes	no
5	MA [22]	no	yes	yes	yes	yes	no
6	OOBO [23]	no	yes	no	yes	no	no
7	COA [20]	yes	no	yes	yes	yes	no
8	FISA [24]	yes	yes	no	yes	no	no
9	GEA [33]	no	yes	yes	no	yes	no
10	KOA [27]	no	yes	yes	yes	yes	no
11	this work	yes	yes	no	yes	yes	yes

The neighborhood search with declining local search space is employed in some metaheuristics, such as COA [20], WaOA [21], ZOA [37], and so on. This method was first introduced in the marine predator algorithm (MPA) [38]. This search becomes the complementary of the directed search which becomes the backbone of the swarm-based metaheuristic. This search is designed to focus on the exploration in the early iteration then it shifts to exploitation as iteration goes.

As shown in Table 1, there are a lot of variations in the recent swarm-based metaheuristics. But there is not any metaheuristics presented in Table 1 employs the cross-dimension mapping between the related entity and its reference. Based on this fact, there is an opportunity in developing a new swarm-based metaheuristic that employs both strict dimension mapping and cross dimension mapping. Both mapping approaches can be accommodated through multiple search strategies where some searches perform strict dimension mapping while some others perform cross dimension mapping.

III. Proposed Model

TSA is a swarm-based metaheuristic so that it is constructed based on a set or collection of entities. These entities represent the solutions. Meanwhile, due to the nature of swarm intelligence, these entities are active and independent tracers. It means that all entities trace for improvement actively along the iteration. Meanwhile, each entity moves independently without any forcing command from other entities. But interaction among entities exists to improve the quality of the search. The term shake becomes the fundamental strategy in TSA. It means that there are three shaking processes implemented in it.

The finest entity plays a significant role in TSA. The finest entity is the entity whose quality is the best among the swarm. It is involved in every search in TSA. This value is always updated every time an entity finds a better solution. Moreover, the finest entity becomes the final solution after the termination criteria, i.e., maximum iteration is reached. There are three sequential searches performed by each entity in the swarm in every iteration.

A permutation set is generated as the first shake. This permutation set consists of the index of the members of the swarm where the order is scrambled. This process is performed in the beginning of every iteration. This permutation set is used in the first search. The objective is to ensure that all entities in the swarm are involved as a reference. This permutation creates a one-to-one mapping in the first search. It means that each entity will be used only once in the first search. This mapping is illustrated in Fig. 1. The reference used in the first search is the balance mixture between the finest entity and the entity whose index is at the same order as the index in the permutation set. The corresponding entity performs two direction movement in the first search. The first direction is toward the reference while the second direction is away from the reference.

In the second shake, the reference is the balance mixture of the finest entity and a randomly chosen entity. This process presents the second shake. This second shake creates a many-to-one mapping between the corresponding entity and the entity chosen to construct its reference. It means that there is a condition where an entity will be used multiple times in a single iteration while another entity has never been used as reference in one iteration. This many-to-one mapping is illustrated in Fig. 2. Same as in the first search, the corresponding entity performs two direction movement in the second search. The first direction is toward the reference while the second direction is away from the reference.

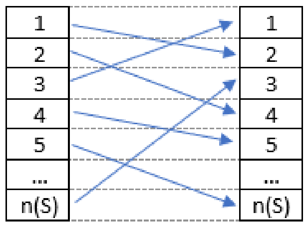

Fig. 1. One-to-one mapping using permutation set in the first shake.

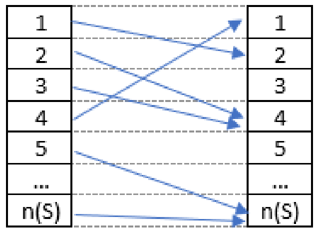

Fig. 2. Many-to-one mapping using randomly chosen approach in the second shake.

In the third search, the reference is the finest entity only. The corresponding entity moves only in a single direction which is toward the finest entity. Meanwhile, the interaction between the corresponding entity and the finest entity may not be in the same dimension. There is a 50 percent probability that the corresponding entity interacts with the finest entity in the same dimension. Meanwhile, there is also a 50 percent probability that the corresponding entity interacts with the finest entity in the different dimension. This cross-dimension mapping represents the third shake. Like in the second shake, this third shake is also many-to-one mapping. This many-to-one dimension mapping is illustrated in Fig. 3.

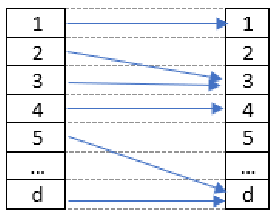

Fig. 3. Many-to-one dimension mapping in the third shake.

The TSA is then formalized using pseudocode and mathematical formulation. The pseudocode of the TSA is presented in algorithm 1. Meanwhile, the mathematical formulation used for a more detailed description of TSA is presented in (1) to (16).

d	dimension
h	objective function
i	solution index
j	dimension index
p	member of permutation set
P	permutation set

α, γ	floating point uniform random number [0,1]
β	integer uniform random number [1,2]
s	solution
S	set of solution
s_b	the finest solution
s_{lo}	lower boundary of the space
s_{up}	upper boundary of the space
s_{c1}, s_{c2}, s_{c3}	the candidate of 1st, 2nd, and 3rd searches
s_{c11}, s_{c12}	the 1st and 2nd candidates of the 1st search
s_{c21}, s_{c22}	the 1st and 2nd candidates of the 2nd search
t	iteration
t_m	maximum iteration
U	uniform random

The construction of TSA can be split into two phases as seen in algorithm 1. The first phase, which is the initialization, is presented in lines 2 to 5. Meanwhile, the second phase, which is the iteration, is presented in lines 6 to 16. In the end, line 17 shows that the finest entity becomes the final solution.

algorithm 1: TSA

```
1   begin
2    for all s ∈ S
3     initialize s_i using (1)
4     update s_b using (2)
5    end for
6    for t=1 to t_m
7     create P using (3)
8     for all s ∈ S
9      perform first search using (4) to (8)
10     update s_b using (3)
11     perform second search using (9) to (14)
12     update s_b using (3)
13     perform third search using (15) and (16)
14     update s_b using (3)
15    end for
16   end for
17   return s_b
18  end
```

The initialization phase contains two processes. The first process is a full random search within the space to generate the initial value of all entities. This process is formalized using (1) where the uniform random is chosen so that the probability of the initial entity is generated in any location within the search space is equal. Then, this process is followed by the updating of the finest entity as formalized using (2). The updating of the finest entity also occurs at the end of every search as presented in lines 10, 12, and 14 in algorithm 1.

$$s_{i,j} = s_{lo,j} + \alpha\left(s_{up,j} - s_{lo,j}\right) \quad (1)$$

$$s_b' = \begin{cases} s_i, h(s_i) < h(s_b) \\ s_b, else \end{cases} \quad (2)$$

The setup of permutation set is performed every time the optimization enters the new iteration. It is conducted before the loop for whole swarm begins as shown in line 7 in algorithm 1. This process is formalized using (3). Equation (3) that the permutation set consists of index from 1 to the dimension of the problem.

$$P = permutation(1,2,3,\ldots,d) \quad (3)$$

The first search is formalized using (4) to (8). Equation (4) states that the first reference is balance mixture between the finest entity and the entity where the index is acquired from the permutation set. Equation (5) formalizes the first candidate of the first search based on the motion toward the first reference while (6) formalizes the second candidate of the first search based on the motion away from the first reference. Equation (7) formalizes the selection of the final first search candidate based on the quality of the first and second candidates previously generated using (6) and (7). Equation (8) formalizes the stringent acceptance approach for the first search.

$$s_{t1,i,j} = \frac{s_{b,j} + s_{p_i,j}}{2} \quad (4)$$

$$s_{c11,i,j} = s_{i,j} + \alpha\left(s_{t1,i,j} - \beta s_{i,j}\right) \quad (5)$$

$$s_{c12,i,j} = s_{i,j} + \alpha\left(s_{i,j} - \beta s_{t1,i,j}\right) \quad (6)$$

$$s_{c1,i} = \begin{cases} s_{c11,i}, h\left(s_{c11,i}\right) < h\left(s_{c12,i}\right) \\ s_{c12,i}, else \end{cases} \quad (7)$$

$$s_i' = \begin{cases} s_{c1,i}, h\left(s_{c1,i}\right) < h(s_i) \\ s_i, else \end{cases} \quad (8)$$

The second search is formalized using (9) and (14). Equation (9) shows the uniformly chosen entity among the swarm. Equation (10) shows the balance mixture between the finest entity and the randomly chosen entity to form the second reference. Equation (11) formalizes the first candidate of the first search based on the motion toward the second reference while (12) formalizes the second candidate of the second search based on the motion away from the second reference. Equation (13) formalizes the selection of the final second search candidate based on the quality of the first and second candidates previously generated using (11) and (12). Equation (14) formalizes the stringent acceptance approach for the second search.

$$s_{sel} = U(S) \quad (9)$$

$$s_{t2,i,j} = \frac{s_{b,j} + s_{sel,j}}{2} \quad (10)$$

$$s_{c21,i,j} = s_{i,j} + \alpha\left(s_{t2,i,j} - \beta s_{i,j}\right) \quad (11)$$

$$s_{c22,i,j} = s_{i,j} + \alpha\left(s_{i,j} - \beta s_{t2,i,j}\right) \quad (12)$$

$$s_{c2,i} = \begin{cases} s_{c21,i}, h\left(s_{c21,i}\right) < h\left(s_{c22,i}\right) \\ s_{c22,i}, else \end{cases} \quad (13)$$

$$s_i' = \begin{cases} s_{c2,i}, h\left(s_{c2,i}\right) < h(s_i) \\ s_i, else \end{cases} \quad (14)$$

The third search is formalized using (15) and (16). Equation (15) presents the two equal options where the first option is the same dimension mapping while the second option is the cross-dimension mapping. In the cross-dimension mapping, the dimension used as reference is uniformly chosen among the dimensions. Equation (16) represents the stringent acceptance approach in the third search.

$$s_{c3,i,j} = \begin{cases} s_{i,j} + \alpha\left(s_{b,j} - \beta s_{i,j}\right), \gamma < 0.5 \\ s_{i,j} + \alpha\left(s_{b,U(1,d)} - \beta s_{i,j}\right), else \end{cases} \tag{15}$$

$$s_i' = \begin{cases} s_{c3,i}, h\left(s_{c3,i}\right) < h(s_i) \\ s_i, else \end{cases} \tag{16}$$

IV. Simulation

This section presents the assessment performed to investigate the performance of TSA. TSA is challenged to find the optimal solution of the set of 23 functions. This set of functions consists of seven high dimension unimodal (HDU) functions (f_1 to f_7), six high dimension multimodal (HDM) functions (f_8 to f_{13}), and ten fixed dimension high dimension (FDM) functions (f_{14} to f_{23}). The detailed description of these functions can be seen in [28] or [8]. The HDU functions are designed to investigate the exploitation capability as they consist of only one optimal solution. The HDM functions are designed to investigate the exploration capability as they consist of multiple optimal solutions. The FDM capability is designed to investigate the balance between exploitation and exploration capabilities. In this assessment, the dimension is set to 40. The maximum iteration is set to 15 while the swarm size is set to 5. There are three assessments in this work: the benchmark assessment, individual search assessment, and the hyperparameter test.

In the benchmark assessment, TSA is compared with five swarm-based metaheuristics: ZOA, WaOA, MA, TIA, and OOBO. All these five are new metaheuristics. ZOA is firstly introduced in 2022 while WaOA, MA, TIA, and OOBO is firstly introduced in 2023. The old metaheuristics like GA, HS, PSO, and ABC are not chosen as the benchmarks as they have been already beaten many times by many recent swarm-based metaheuristics like in [34], [8], or [18]. The result of the first assessment is presented in Table 2 to Table 4 representing the assessment result to solve HDU, HDM, and FDM functions respectively. The decimal point smaller than 10^{-4} is rounded to zero. The summary of this result is presented in Table 5.

The assessment result in solving the HDU functions is released in Table 2. This result shows that TSA is superior in solving these functions. TSA becomes the first best algorithm in six functions (f_1, f_2, f_3, f_4, f_5, and f_7) and the second-best performer in one function (f_6). Moreover, TSA can provide the global optimal solution in three functions (f_1, f_2, and f_4). All algorithms except OOBO provide the same result in f_2. ZOA becomes the second-best performer while OOBO becomes the worst performer. The performance difference between the best performer and the worst performer is wide.

TSA maintains its dominance among these algorithms in the HDM functions as released in Table 3. TSA becomes the best performer in five functions (f_9, f_{10}, f_{11}, f_{12}, and f_{13}). TSA is on the second rank in solving f_8. Meanwhile, the performance difference between the best performer and the worst performer is wide except in f_8.

TSA is still competitive in solving the FDM functions as released in Table 4. Meanwhile, the dominance of TSA is not so strong as it is shown in Table 2 and Table 3. TSA is on the first rank in seven functions (f_{14}, f_{15}, f_{16}, f_{19}, f_{21}, f_{22}, and f_{23}), second rank in one function (f_{18}), and third rank in two functions (f_{17} and f_{20}).

Table 5 summarizes the assessment result in Table 2 to Table 4. This summary released the number of functions where TSA is better than related metaheuristics in every group of functions. The result is obtained based on the mean value of each metaheuristic tested in this work. The result strengthens the dominance of TSA among its competitors. TSA is better than ZOA, WaOA, MA, TOA, and OOBO in 21, 19, 17, 19, and 22 functions respectively. Since all metaheuristics achieve the result in f_2 and f_{19}, it means that ZOA and OOBO never outperform TSA.

TABLE II
BENCHMARK SIMULATION RESULT ON SOLVING HIGH DIMENSION UNIMODAL FUNCTIONS

F	Parameter	ZOA [37]	WaOA [21]	MA [22]	TIA [19]	OOBO [23]	TSA
1	mean	0.0604	0.4468	8.5369	0.2588	8.8238×10^{2}	0.0000
	std dev	0.0583	0.4297	5.0352	0.0895	5.1949×10^{2}	0.0000
	mean rank	2	4	5	3	6	1
2	mean	0.0000	0.0000	0.0000	0.0000	2.2505×10^{21}	0.0000
	std dev	0.0000	0.0000	0.0000	0.0000	1.0313×10^{22}	0.0000
	mean rank	1	1	1	1	6	1
3	mean	4.2078×10^{2}	9.8939×10^{2}	1.4712×10^{4}	8.1733×10^{2}	3.5343×10^{4}	4.5907
	std dev	5.5171×10^{2}	1.8586×10^{3}	1.1975×10^{4}	1.3297×10^{3}	1.8532×10^{4}	7.4257
	mean rank	2	4	5	3	6	1
4	mean	0.3434	0.7117	6.4527	0.6076	3.0230×10^{1}	0.0000
	std dev	0.1630	0.4255	1.2220×10^{1}	0.1683	1.0547×10^{1}	0.0000
	mean rank	2	4	5	3	6	1
5	mean	3.9753×10^{1}	4.0892×10^{1}	2.4362×10^{2}	4.3064×10^{1}	3.1920×10^{5}	3.8906×10^{1}
	std dev	0.4829	1.4301	2.8834×10^{2}	1.8579	5.7340×10^{5}	0.0364
	mean rank	2	3	5	4	6	1
6	mean	8.5380	8.4889	2.1197×10^{1}	7.3744	1.0367×10^{3}	7.4626
	std dev	0.4531	0.6127	1.1128×10^{1}	0.5601	6.8936×10^{2}	0.8901
	mean rank	4	3	5	1	6	2
7	mean	0.0254	0.0352	0.0618	0.0412	0.4116	0.0067
	std dev	0.0172	0.0227	0.0406	0.0318	0.3736	0.0041
	mean rank	2	3	5	4	6	1

TABLE III

BENCHMARK SIMULATION RESULT ON SOLVING HIGH DIMENSION MULTIMODAL FUNCTIONS

F	Parameter	ZOA [37]	WaOA [21]	MA [22]	TIA [19]	OOBO [23]	TSA
8	mean	-2.3087x10^{3}	-3.1402x10^{3}	-3.3070x10^{3}	-1.9081x10^{3}	-2.7687x10^{3}	-3.2874x10^{3}
	std dev	4.1522x10^{2}	5.8214x10^{2}	4.7968x10^{2}	5.5397x10^{2}	6.8098x10^{2}	3.4693x10^{2}
	mean rank	5	3	1	6	4	2
9	mean	0.1360	2.1920	9.7182x10^{1}	2.6438	2.7782x10^{2}	0.0000
	std dev	0.1702	3.7524	9.1725x10^{1}	3.3305	5.4140x10^{1}	0.0000
	mean rank	2	3	5	4	6	1
10	mean	0.0526	0.1653	2.1095	0.1204	6.8057	0.0000
	std dev	0.0408	0.0807	3.3050	0.0318	0.9862	0.0000
	mean rank	2	4	5	3	6	1
11	mean	0.0511	0.0871	0.9973	0.1412	9.8010	0.0000
	std dev	0.0650	0.0762	0.1898	0.1643	6.3394	0.0000
	mean rank	2	3	5	4	6	1
12	mean	1.0355	0.9890	1.3432	0.8393	2.8373x10^{3}	0.8121
	std dev	0.1247	0.1862	0.3522	0.1467	1.2922x10^{4}	0.2136
	mean rank	4	3	5	2	6	1
13	mean	3.2441	3.3826	4.3418	3.3791	4.7485x10^{4}	3.0914
	std dev	0.1032	0.1862	0.5069	0.1508	8.4708x10^{4}	0.1519
	mean rank	2	4	5	3	6	1

TABLE IV

BENCHMARK SIMULATION RESULT ON SOLVING FIXED DIMENSION MULTIMODAL FUNCTIONS

F	Parameter	ZOA [37]	WaOA [21]	MA [22]	TIA [19]	OOBO [23]	TSA
14	mean	1.3028x10^{1}	8.1253	9.1855	1.0348x10^{1}	1.5400x10^{1}	6.9100
	std dev	7.5195	3.2222	3.3896	4.7719	1.2381x10^{1}	3.6660
	mean rank	5	2	3	4	6	1
15	mean	0.0127	0.0044	0.0095	0.0094	0.0232	0.0020
	std dev	0.0168	0.0079	0.0082	0.0167	0.0208	0.0024
	mean rank	5	2	4	3	6	1
16	mean	-0.9319	-1.0236	-1.0010	-1.0007	-0.9957	-1.0291
	std dev	0.2358	0.0186	0.0499	0.0997	0.0466	0.0051
	mean rank	6	2	3	4	5	1
17	mean	2.9927	0.3992	0.4234	3.6686	0.6500	0.6057
	std dev	4.4460	0.0023	0.0284	5.4982	0.1926	0.3493
	mean rank	5	1	2	6	4	3
18	mean	7.5856x10^{1}	1.9224x10^{1}	7.5248	1.6129x10^{1}	2.3160x10^{1}	1.2138x10^{1}
	std dev	1.3050x10^{2}	2.3586x10^{1}	1.9912x10^{1}	1.2487x10^{1}	2.5774x10^{1}	1.2278x10^{1}
	mean rank	6	4	1	3	5	2
19	mean	-0.0495	-0.0495	-0.0495	-0.0495	-0.0495	-0.0495
	std dev	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	mean rank	1	1	1	1	1	1
20	mean	-2.2112	-2.9411	-2.9378	-2.3943	-2.1531	-2.6073
	std dev	0.6065	0.2399	0.1875	0.3982	0.4183	0.4450
	mean rank	5	2	1	4	6	3
21	mean	-1.5907	-2.1274	-2.7671	-2.2224	-1.0289	-4.6115
	std dev	0.7416	1.2697	0.1875	1.1666	0.5895	2.1632
	mean rank	5	4	2	3	6	1
22	mean	-2.3045	-2.5576	-2.9304	-2.1419	-1.5281	-3.5667
	std dev	1.1566	0.9617	1.0951	1.0497	1.4937	1.7279
	mean rank	4	3	2	5	6	1
23	mean	-2.0156	-2.9496	-3.4118	-1.9849	-1.3279	-3.6884
	std dev	0.7870	1.5513	1.7255	1.2462	0.6928	0.9981
	mean rank	4	3	2	5	6	1

TABLE V

GROUP BASED SUPERIORITY COMPARISON

Cluster	ZOA [37]	WaOA [21]	MA [22]	TIA [19]	OOBO [23]
1	6	6	6	5	7
2	6	6	5	6	6
3	9	7	6	8	9
Total	21	19	17	19	22

The second assessment is the individual search assessment. This assessment is designed to assess the contribution of each search in TSA since TSA is the multiple-search metaheuristic. Since TSA is constructed from three searches, there are three individual searches assessed in this work. The result is released in Table 6. The best result in each function is written in bold font.

Table 6 releases the fact that the performance difference among these three searches is narrow. In two functions (f_2 and f_{19}), each search provides the same result. The first search becomes the distinct best performer in ten functions. The second search becomes the distinct best performer in eight functions. Finally, the third search becomes the distinct best performer in three functions. It means that the first search becomes the main contributor in TSA.

The third assessment is the hyperparameter assessment. This assessment is designed to evaluate the relation between the adjusted parameters and the performance of TSA. There are two adjusted parameters evaluated in this assessment: maximum iteration and swarm size. There are two values of maximum iteration: 20 and 40. On the other hand, there are two values of swarm size: 10 and 20. The 23 functions are still used for the use case. The result of the assessment regarding the change of maximum iteration is presented in Table 7 while the result of the assessment regarding the change of swarm size is presented in Table 8. The data is only the average fitness score. As in general, the increase of maximum iteration and swarm size tends to improve the quality of the solution, the consideration is whether the improvement is significant or not.

TABLE VI
INDIVIDUAL SEARCH ASSESSMENT RESULT

Function	Average Fitness Score		
	1st search	2nd search	3rd search
1	**1.5746**	2.0798	2.5265
2	**0.0000**	**0.0000**	**0.0000**
3	**1.6634x10^3**	2.0422x10^3	7.3334x10^3
4	**1.1711**	1.3129	1.6279
5	**5.7067x10^1**	7.1256x10^1	6.4967x10^1
6	1.0448x10^1	1.0314x10^1	**8.0940**
7	0.0400	**0.0391**	0.0617
8	**-3.0180x10^3**	-2.8964x10^3	-2.2522x10^3
9	2.6564x10^1	**1.2706x10^1**	3.1598x10^1
10	0.5205	**0.5181**	0.5263
11	0.4520	**0.4198**	0.5774
12	1.1295	1.1240	**0.7502**
13	3.7080	3.7753	**3.5188**
14	9.3065	**8.2844**	9.7176
15	0.0087	**0.0059**	0.0100
16	**-1.0110**	-1.0092	-0.8353
17	**1.7855**	2.4077	2.0932
18	**1.8571x10^1**	2.6038x10^1	4.1890x10^1
19	**-0.0495**	**-0.0495**	**-0.0495**
20	**-2.2567**	-2.2145	-2.1965
21	-2.4165	**-2.6226**	-1.8794
22	**-2.7147**	-2.1461	-2.3959
23	-2.1316	**-2.7918**	-2.0986

TABLE VII
RELATION BETWEEN MAXIMUM ITERATION AND FITNESS

Function	Average Fitness Score		Significant Improvement
	t_m = 20	t_m = 40	
1	0.0000	0.0000	no
2	0.0000	0.0000	no
3	0.1410	0.0000	yes
4	0.0000	0.0000	no
5	3.8882x10^1	3.8892x10^1	no
6	7.3422	7.2310	no
7	0.0080	0.0022	yes
8	-3.4307x10^3	-3.6685x10^3	no
9	0.0000	0.0000	no
10	0.0000	0.0000	no
11	0.0000	0.0000	no
12	0.7669	0.7423	no
13	3.0607	3.0018	no
14	7.5709	6.6958	no
15	0.0010	0.0005	yes
16	-1.0166	-1.0127	no
17	1.2696	1.5103	no
18	8.1110	9.5660	no
19	-0.0495	-0.0495	no
20	-2.6424	-2.9142	no
21	-4.3501	-5.2551	no
22	-4.6737	-5.5611	no
23	-4.4351	-5.2419	no

Result in Table 7 shows that the significant improvement occurs in only few functions due to the increase of maximum iteration. There are only three functions where the improvement is significant: f_3, f_7, and f_{15}. In some cases, the stagnation occurs because in these functions, the global optimal solution has been found or the final solution is near the global optimal solution. This circumstance can be seen in some functions, such as f_1, f_2, f_4, f_9, and so on. Meanwhile, in some functions, such as in f_{21}, f_{22}, or f_{23}, the stagnation occurs although the final solution has not been near the global optimal yet.

Result in Table 8 also shows that the significant improvement regarding the increase of swarm size occurs in only few functions. In this assessment, there are only two functions where significant improvement is found: f_3 and f_{15}. Meanwhile, as also found in Table 7, in some functions, stagnation occurs because the global optimal solution has been found or the final solution is near the global optimal.

TABLE VIII
RELATION BETWEEN SWARM SIZE AND FITNESS

Function	Average Fitness Score		Significant Improvement
	$n(S)$ = 10	$n(S)$ = 20	
1	0.0000	0.0000	no
2	0.0000	0.0000	no
3	0.3923	0.0372	yes
4	0.0000	0.0000	no
5	3.8883x10^1	3.8847x10^1	no
6	6.4268	6.0550	no
7	0.0037	0.0024	no
8	-3.6636x10^3	-3.8451x10^3	no
9	0.0000	0.0000	no
10	0.0000	0.0000	no
11	0.0000	0.0000	no
12	0.6007	0.5898	no
13	2.9875	2.9331	no
14	4.5303	2.5654	no
15	0.0016	0.0005	yes
16	-1.0274	-1.0316	no
17	0.4192	0.3983	no
18	4.5938	3.0042	no
19	-0.0495	-0.0495	no
20	-2.9325	-3.0467	no
21	-5.4213	-5.7816	no
22	-5.1487	-6.3390	no
23	-5.1409	-6.5297	no

V. DISCUSSION

Through the result on benchmark assessment, it is shown that TSA is a promising metaheuristic. TSA becomes the dominant performer among the five metaheuristics chosen as the competitors. Table 5 strengthens this statement as TSA is superior in all three groups of functions. This means that STA has acceptable capability in exploration, exploitation, and balancing the exploration and exploitation. The significant performance difference, especially in the HDU functions, shows the significant improvement in these problems.

The stochastic cross dimension mapping employed in the third search provides moderate contribution for the search process. As seen in Table 6, the performance of the third search is still close to the performance of the first and second searches. Moreover, the third search can become the best performer in three functions where all these functions are high dimension functions. As the first and second searches are more competitive, this cross-dimension mapping approach has potential for future development.

Based on the result of hyper-parameter assessment, TSA successfully found the quasi-optimal solution in the scenario of low swarm size and low maximum iteration. This circumstance takes place in many functions. Meanwhile, in some functions, stagnation still happens although the swarm size or maximum iteration increases significantly. It becomes a note that there is modification in the searching strategy so that this stagnation can be overcome.

There are some limitations in this work, especially in TSA despite its significant improvement compared to its five competitors. First, TSA employs a stringent acceptance approach so that the solution candidate substitutes the value of current entity only if it provides improvement. In one side, this approach provides a guarantee that the optimization process will not move to a worse location. But, on the other side, this approach makes the entity stand on its current value due to the stagnation of the searching process. Meanwhile, the stringent acceptance approach may put the risk of local optimal entrapment. It is better to construct a more adaptive acceptance approach. Second, TSA adopts only one cross-dimension mapping which is performed stochastically. Although the cross-dimension mapping proves the improvement in some functions, developing a more deterministic cross-dimension mapping is important and it can be conducted by analyzing the pattern during the iteration. Third, this work has not accommodated the assessment with real world problems. In general, this assessment is important to investigate the improvement of TSA in a more comprehensive and realistic manner. In general, the real-world optimization problems are simpler than theoretic ones.

The computing complexity of TSA can be traced back by observing the looping process in its algorithm. TSA employs a nested loop consisting of two loops during the initialization. The size of the outer loop is swarm size while the size of the inner loop is the dimension. Based on this explanation, the computing complexity during the initialization is presented as $O(n(S).d)$. Meanwhile, the looping process during the iteration is more complex. The size of the outer loop is the maximum iteration. Then, there is a loop with the size of the swarm size to generate the permutation set. This loop is then followed by the loop with the size of the swarm size too that represents the searching process of each agent. Inside this loop, there are three searches where there is a loop with the size of dimension which is employed in every search. Based on this explanation, the computing complexity during the iteration can be presented as $O(t_m.n(S).(1+3d))$.

VI. CONCLUSION

A new stochastic optimization, i.e., metaheuristic named triple shake algorithm (TSA) has been presented in this paper. The fundamental concept of TSA which is constructed by three directed searches including the permutation, reference selection, and the stochastic cross dimension mapping has been presented and it is followed by the formalization through pseudocode and the mathematical formulation. The performance assessment for TSA has been performed where TSA is benchmarked with five brand new metaheuristics. The result shows that TSA is superior to its five competitors where it is better than ZOA, WaOA, MA, TIA, and OOBO in 21, 19, 17, 19, and 22 functions respectively. It means that the dominance of TSA occurs in all three groups of functions: HDUs, HDMs, and FDMs. Meanwhile, through the individual search assessment, it is shown that the first search where the reference is the balance mixture between the finest entity and the permutation indexed entity becomes the most dominant search.

In the future, it is important to improve and employ TSA to solve various optimization problems. The improvement can be taken by combining TSA with other metaheuristics or exploring the cross-dimension mapping in a better way. Meanwhile, employing TSA to solve various problems will give a more comprehensive perspective in evaluating the performance of TSA.

REFERENCES

[1] J. Zhao, Y. Guan, and Z. Chen, "Numerical modelling and optimization of 3D surface roughness forecasting in milling", *Engineering Letters*, vol. 31, no.4, pp. 1343-1347, 2023.

[2] F. D. Wihartiko, S. Nurdiati, A. Buono, and E. Santosa, "Multi-objective entropy optimization model for agricultural product price recommendation problem", *Engineering Letters*, vol. 31, no.4, pp. 1908-1918, 2023.

[3] P. D. Kusuma, and F. M. Dirgantara, "Run-catch optimizer: a new metaheuristic and its application to address outsourcing optimization problem", *Engineering Letters*, vol. 31, no.3, pp. 1045-1053, 2023.

[4] Z. He, Y. Li, H. Li, and N. Xu, "Multi-objective optimization for online train trajectory planning with moving window method", *IAENG International Journal of Computer Science*, vol. 50, no.3, pp. 1074-1082, 2023.

[5] S. M. A. Altbawi, S. A. Khalid, A. S. Mokhtar, R. H. Alsisi, Z. A. Arfeen, H. Shareef, and M. K. Azam, "Improve power quality of charging station unit using African vulture optimization algorithm", *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 5, pp. 2605-2614, 2023.

[6] D. F. U. Putra, A. A. Firdaus, H. Arof, N. P. U. Putra, and V. A. Kusuma, "Improved load frequency control performance by tuning parameters of PID controller and BESS using Bat algorithm", *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 5, pp. 2624-2634, 2023.

[7] 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.

[8] 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.

[9] 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.

[10] T. R. Farshi, "Battle royale optimization algorithm", *Neural Computing and Applications*, vol. 33, pp. 1139-1157, 2021.

[11] M. A. Al-Betar, Z. A. A. Alyasseri, M. A. Awadallah, and I. A. Doush, "Coronavirus herd immunity optimizer (CHIO)", *Neural Computing and Applications*, vol. 33, pp. 5011–5042, 2021.

[12] 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.

[13] P. D. Kusuma and M. Kallista, "Stochastic komodo algorithm", *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 4, pp. 156-166, 2022.

[14] 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.

[15] 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.

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

[17] 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.

[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. 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.

[20] 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.

[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] 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.

[23] M. Dehghani, E. Trojovská, P. Trojovský, and O. P. Malik, "OOBO: a new metaheuristic algorithm for solving optimization problems", *Biomimetics*, vol. 8, ID. 468, pp. 1-48, 2023.

[24] 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*, vol. 9, ID. e1431, pp. 1-24, 2023.

[25] M. Dehghani, G. Bektemyssova, Z. Montazeri, G. Shaikemelev, O. M. Malik, and G. Dhiman, “Lyrebird Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems”, *Biomimetics*, vol. 8, ID. 507, pp. 1-62, 2023.

[26] P. Trojovsky, M. Dehghani, E. Trojovska, and E. Milkova, “Language education optimization: a new human-based metaheuristic algorithm for solving optimization problems”, *Computer Modeling in Engineering and Sciences*, vol. 136, no. 2, pp. 1527-1573, 2023.

[27] M. Dehghani, Z. Montazeri, G. Bektemyssova, O. P. Malik, G. Dhiman, and A. E. M. Ahmed, “Kookaburra optimization algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems”, *Biomimetics*, vol. 8, ID. 470, pp. 1-54, 2023.

[28] 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.

[29] I. Matousova, P. Trojovsky, M. Dehghani, E. Trojovska, 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.

[30] 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.

[31] F. A. Zeidabadi, M. Dehghani, P. Trojovský, S. Hubálovský, V. Leiva, and G. Dhiman, "Archery algorithm: a novel stochastic optimization algorithm for solving optimization problems", *Computers, Materials & Continua*, vol. 72, no. 1, pp. 399-416, 2023.

[32] 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.

[33] 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*, pp. 1-35, 2023.

[34] S. E. Khouni and T. Menacer, "Nizar optimization algorithm: a novel metaheuristic algorithm for global optimization and engineering applications", *The Journal of Supercomputing*, pp. 1-53, 2023.

[35] 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.

[36] A. G. Gad, “Particle swarm optimization algorithm and its applications: a systematic review”, *Archives of Computational Methods in Engineering*, vol. 29, pp. 2531-2561, 2022.

[37] 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.

[38] 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.

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