Received: March 16, 2024. Revised: April 6, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Focus and Shake Algorithm: A New Stochastic Optimization Employing Strict and Randomized Dimension Mappings

Purba Daru Kusuma[1]*

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

Abstract: The no-free-lunch (NFL) theory has become the main reason of developing new metaheuristics in decades. Besides, the strict dimension mapping has been implemented in many population-based metaheuristics, especially the swarm-based metaheuristics. Regarding this problem, this paper proposes a new swarm-based metaheuristics called focus and shake algorithm (FSA). FSA has novel approach in the dimension mapping between the agent and its reference during the directed motion. It combines the strict dimension mapping called as focus approach and the randomized dimension mapping called as shake approach to enhance its exploration ability. FSA deploys two directed motions based on two references. The first reference is constructed based on the balance mixture between two finer agents while the second reference is constructed based on the balance mixture of the finest agent and a randomly picked agent. In the competing assessment, FSA competes with five brand new swarm-based metaheuristics: migration algorithm (MA), total interaction algorithm (TIA), lyrebird optimization algorithm (LOA), osprey optimization algorithm (OOA), and kookaburra optimization algorithm (KOA). The result exhibits that FSA is finer than MA, TIA, LOA, OOA, and KOA in 19, 21, 21, 19, and 20 functions out of 23 functions respectively. The result also shows that the superiority of FSA takes place in both unimodal and multimodal problems. In the future, the cross-dimension mapping can be more explored to develop finer swarm-based metaheuristics.

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

1. Introduction

Metaheuristics has been utilized in many optimization studies. It has been implemented in a wide range of engineering and computing systems. For example, sunflower optimization (SFO) has been enhanced and implemented to develop encryption in the secure cloud system [1]. Remora optimization algorithm (ROA) was combined with the long-short term memory classifier to predict lung cancer using histopathology images [2]. Golden jackal optimization (GJO) has been combined with a support vector machine (SVM) to detect and classify digital image forgery [3].

There are many recent swarm-based metaheuristics use the animal behavior as metaphor. Some common animal behaviors are searching for food, hunting prey, mating, and avoiding predators. Some of them are green anaconda optimization (GAO) [4], Komodo mlipir algorithm (KMA) [5], lyrebird optimization algorithm (LOA) [6], GJO [7], kookaburra optimization algorithm (KOA) [8], hippopotamus optimization (HO) (9), electric eel foraging optimization (EEFO) [10], white shark optimization (WSO) [11], chameleon swarm algorithm (CSA) [12], osprey optimization algorithm (OOA) [13], graylag goose optimization (GGO) [14], pelican optimization algorithm [15], and so on.

Several metaheuristics adopts the social behavior of human. These social activities range from education or training, social movement, and so on. Some of these metaheuristics are chef-based optimization algorithm (CBOA) [16], driving training-based optimization (DTBO) [17], language education optimization (LEO) [18], migration algorithm (MA) [19], paint optimizer (PO) [20], deep sleep optimization (DSO) [21], mother optimization algorithm (MOA) [22], sewing training-based optimization (STBO) [23], election-based optimization algorithm (EBOA) [24], and so on.

Several metaheuristics are inspired by the mechanism of game. These games may be the traditional games or the modern ones. Some of them are battle royale optimization (BRO) [25], dart game optimization (DGO) [26], shell game optimization (SGO) [27], and so on.

Meanwhile, some metaheuristics are free from metaphors. They use their main or fundamental strategy for their name. Some of them are total interaction algorithm (TIA) [28], one to one based optimization (OOBO) [29], golden search optimization (GSO) [30], attack leave optimization (ALO) [31], four directed search algorithms (FDSA) [32], and so on.

Despites the massive development of the swarm-based metaheuristic, almost all of them employ strict dimension mapping in their directed searches. It means that the dimension of an agent is mapped with the same dimension of its reference. This circumstance will lead to right direction if the reference is finer than the agent in general, but in some dimensions, this rule pushes the agent to a wrong direction. Regarding this problem, there is a chance to construct a swarm-based metaheuristic that does not employ strict dimension manner in a full manner.

Regarding this problem, this work is aimed at proposing a new metaheuristic called focus and shake algorithm (FSA). As the name suggests, FSA combines both strict dimension mapping and randomized dimension mapping between the agent and its reference in the directed motions. Term focus represents the strict dimension mapping while term shake represents the randomized dimension mapping. The randomized dimension mapping represents the cross-dimension mapping highlighted in the previous explanation. Then, both mapping approaches are implemented in each of two directed motions. In the first motion, the reference is the balance mixture between two finer agents which are randomly chosen. In the second motion, the reference is the balance mixture between the finest agent and a randomly picked agent.

The novelty and scientific contributions of this work are as follows.

- FSA employs a new approach namely as randomized dimension mapping which is different from the strict dimension mapping which is common in the swarm-based metaheuristics.
- FSA employs both strict dimension mapping and randomized dimension mapping.
- The performance of FSA is competed with five brand new swarm-based metaheuristics: MA, TIA, LOA, OOA, and KOA.
- The individual search assessment is taken to investigate the contribution of each motion in FSA.

This paper is structured in six sections. The first section explains the background, problem statement, objective, and the scientific contribution of this paper. The comprehensive review regarding the recent development of swarm-based metaheuristics is exhibited in section 2. Section 3 presents the description of FSA including the concept and formalization. Section 4 presents the performance assessment of FSA including the result. Section 5 presents a comprehensive discussion regarding the findings, complexity of the algorithm, and the limitations of this work. Section 6 encapsulates the conclusion and tracks for studies in the future.

2. Related works

Metaheuristic is an optimization or searching method developed based on the stochastic approach. Rather than tracing all possible solutions inside the space, it performs trial and error-based searching or random searching. Due to this nature, a metaheuristic may fail to find the global optimal solution. This circumstance makes the metaheuristic cannot guarantee the finding of the global optimal solution but only the quasi optimal one. There are two capabilities should be had by any metaheuristic: exploitation and exploration. Exploitation is an effort to find enhancement near the current solution. On the other hand, exploration is an effort to find alternatives anywhere inside space.

Metaheuristic relies on the iterative process. It tries to enhance its current solution in every iteration. But the searching effort may lead to enhancement or stagnation which means that the searching result is not finer than the current solution. In this case, there are possible options: strict acceptance, non-strict acceptance, and conditional acceptance. In the strict acceptance, a new solution is accepted only if it provides enhancement. This option can be found in many metaheuristics, such as MA [19], TIA [28], and so on. In the non-strict acceptance, a new solution is accepted without considering the relative quality between the new solution and the current solution. This option is seen in many metaheuristics,

Table 1. The summary of recent swarm-based metaheuristics

No	Meta-heuristic	Number of Search	References	Neighborhood Search	Cross-dimension Mapping	Acceptance
1	MA [19]	2	a randomly picked finer agent	yes	no	strict
2	TIA [28]	1	all other agents	no	no	strict
3	LOA [18]	3	a randomly picked agent from a pool consisting of all finer agents and the finest agent and a randomly picked agent	yes	no	strict
4	OOA [13]	2	a randomly picked agent from a pool consisting of all finer agents and the finest agent	yes	no	strict
5	KOA [8]	2	a randomly picked finer agent	yes	no	strict
6	GSO [30]	1	global finest agent and local finest agent	no	no	non-strict
7	POA [15]	2	a randomly generated agent	yes	no	strict
8	ALO [31]	2	the finest agent, the balance mix between the finest agent and a randomly picked agent, and the balance mixture between two randomly picked agent	no	no	strict
9	GJO [7]	1	the finest agent and the second-finest agent	no	no	non-strict
10	OOBO [29]	2	a randomly picked another agent	no	no	strict
11	this work	4	the balance mixture between two finer agents and the balance mixture between the finest agent and a randomly picked agent	no	yes	strict

such as coronavirus optimization algorithm (COVIDOA) [33], coronavirus herd immunityoptimizer (CHIO) [34], CSA [12], GJO [7], and so on. In the conditional acceptance, new solution with worse quality still can be accepted with certain condition. The example is simulated annealing where the worse solution may be accepted based on the stochastic calculation controlled by the iteration [35].

Swarm intelligence is the most popular approach to develop a new metaheuristic in recent years. Different from the evolutionary based system like genetic algorithm (GA) which relies on the crossover and mutation [36], swarm-based metaheuristic relies on the interaction among the solutions inside the swarm and the shared collective intelligence [37]. This interaction turns to the movement of the agent toward or away from its reference. This movement is called the directed motion which becomes the backbone of any swarm-based metaheuristics. Various swarm-based metaheuristics can be differentiated based on the references implemented in the algorithm and the movement model relative to these references (direction and step size). Table 1.

In Table 1, there are varieties in constructing the reference, the movement strategy, number of searches, and the use of neighborhood search as secondary search. Unfortunately, all these metaheuristics employ strict dimension mapping. In the strict dimension mapping, a vector in every dimension will be mapped with the vector with the same dimension in the reference. On the other hand, the use of cross dimension mapping is not popular. This circumstance gives chance to develop a swarm-based metaheuristic does not employ only the strict direction mapping but also the cross-dimension mapping.

Despite the dominance or popularity of strict dimension mapping, there is possibility of wrong direction circumstance. This circumstance can be found in the multiple or high dimension problems. In this problem, the finest agent or finer agents have finer value in most dimensions compared to other agents. It means that following this finer agent provides higher probability in improving the quality of the solution in majority of dimensions. Meanwhile, the value may get worse in some dimensions where the quality in some dimension owned by the finer agent is worse.

In general, random searches can be chosen as an alternative. This search can be taken to provide exploration. Unfortunately, there are some problems regarding this random search. First, a full random search can reset the current achievement. Second, the neighborhood search which is a random search with narrow search has limitation in reaching farther space.

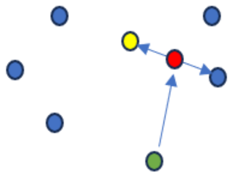
This problem becomes the authors' main motivation in proposing a new metaheuristic. In this proposed metaheuristic, the strict dimension mapping is combines with the cross-dimension mapping. The employment of strict dimension mapping is designed to maintain enhancement in most dimensions. On the other hand, cross-dimension mapping is employed to give an alternative for exploration. As previously mentioned, cross-dimension mapping is preferred rather than full random search to prevent the solution jumping too far and losing its achievement. On the other hand, cross-dimension mapping is preferred rather than neighborhood search to enlarge the search space. Based on this explanation, the cross-dimension mapping is chosen as the trade-off between full random search and neighborhood search.

3. Model

As a swarm-based metaheuristic, FSA performs only the directed move in the iteration phase. It means that each agent searches inside the search space based on the direction of a reference. FSA is not embedded with the neighborhood search as it is commonly found in several new swarm-based metaheuristics like MA [19], LOA [6], LEO [18].

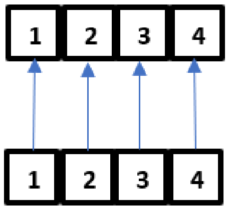
There are two directed moves performed by every agent in every iteration. The first directed move is the movement of agent toward the balance mixture of two randomly selected agents from the pool. The pool contains all the finer agents relative to the agent plus the finest agent. This move is designed to push the targeted enhancement. The second directed move is the movement of agent relative to the balance mixture between the finest agent and a randomly selected agent from the swarm. This search is designed to provide diversification. There are two possible directions in the second directed move. The agent will move to the target if the target is finer than the agent. Otherwise, the agent will move away from the target.

(a)

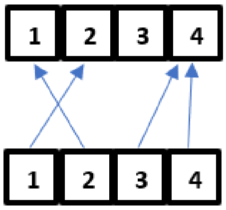


(b)

Figure. 1 Illustration of two directed moves: (a) first directed motion and (b) second directed motion



(a)



(b)

Figure. 2 Illustration of focus and shake: (a) strict dimension mapping and (b) randomized dimension mapping

These two directed moves are illustrated in Fig. 1 where Fig. 1a represents the first move while Fig. 1b represents the second move. In Fig. 1, green circle represents the agent, red circle represents the target, yellow circle represents the finest agent, orange circles represent all finer agents, and blue circles represent the other agents.

As the name suggests, there are two approaches performed in FSA: focus and shake. In the focus approach, the dimension of the agent is mapped to the same dimension of the target. For example, the value of the second dimension of the agent is mapped to the value of the second dimension of the target. In the shake approach, a solution of the agent is mapped with solution in any randomly selected dimension of the target. The focus approach represents exploitation while the shake approach represents exploration. These two approaches are illustrated in Fig. 2 where Fig. 2a represents the focus approach while Fig. 2b represents the shake approach.

The formalization of FSA is displayed in algorithm 1 while its mathematical formulation is displayed in Eq. (1) to Eq. (16). As shown in algorithm 1, the finest agent becomes the final

Received: March 16, 2024. Revised: April 6, 2024.

solution. The annotations used in this work are as follows.

a	agent
A	swarm
a_{best}	the finest agent
a_{bet}	finer agent
A_{bet}	a set of finer agents
a_{betsel}	the randomly selected finer agent
a_{ransel}	a randomly selected agent
a_{t11}, a_{t12}	the first and second targets in 1st move
a_{t2}	target in 2nd move
c_{11}, c_{12}	the 1st and 2nd candidates in 1st move
c_{1sel}	the selected candidate in 1st move
c_{21}, c_{12}	the 1st and 2nd candidates in 2nd move
c_{2sel}	the selected candidate in 2nd move
d	dimension
f	objective function
i	agent index
j	dimension index
lb, ub	lower and upper boundaries
r_1	floating point uniform random [0,1]
r_2	integer uniform random [1,2]
rd	integer uniform random in d
t	iteration
t_m	maximum iteration
U	uniform random

algorithm 1: focus and shake algorithm

1 **begin**
2 **for all** $a \in A$
3 initiate a_i using Eq. (1)
4 update a_{best} using Eq. (2)
5 **end for**
6 **for** t=1: t_m
7 **for all** $a \in A$
8 create pool for a_i using Eq. (3)
9 1st directed move using Eq. (4) to Eq. (10)
10 update a_{best} using Eq. (2)
11 2nd directed move using Eq. (11) to Eq. (15)
12 update a_{best} using Eq. (2)
13 **end for**
14 **end for**
15 **return** a_{best}
16 **end**

The optimization process begins with the initialization phase. It is displayed from lines 2 to 5. There are two processes in the initialization phase: generating an initial solution for each agent using Eq. (1) and updating the finest agent using Eq. (2). Eq. (1) shows that a full random search inside entire space is performed to generate initial solution.

$$a_{i,j} = U\left(lb_j, ub_j\right) \tag{1}$$

$$a'_{best} = \begin{cases} a_i, f(a_i) < f(a_{best}) \\ a_{best}, else \end{cases} \tag{2}$$

The second phase is the iteration. It is performed to enhance the quality of the solution generated in the initialization phase. There are three processes performed in the iteration phase: creating the pool consisting of all finer agents, performing the first directed move, and performing the second directed move. The finest agent is updated every time a directed move is performed. Eq. (3) states that the pool contains all finer agents relative to the agent plus the finest agent.

$$A_{bet,i} = \{a \in A, f(a) < f(a_i) \cup a_{best}\} \tag{3}$$

The first directed move is formalized using Eq. (4) to Eq. (10). Eq. (4) states that an agent is randomly picked from the pool. Then, two targets are obtained based on the balance mixture of two randomly selected finer agents. Eq. (5) represents the focus approach while Eq. (6) represents the shake approach. Eq. (7) is used to generate the first candidate based on the focus approach while Eq. (8) is used to generate the second candidate based on the shake approach. Eq. (9) states that the finer candidate between these two candidates will be chosen as the final candidate for the first directed move. Eq. (10) states that this candidate replaces the current value of the agent only if it provides enhancement.

$$a_{betsel,i} = U\left(A_{bet,i}\right) \tag{4}$$

$$a_{t11,i,j} = \frac{a_{betsel1,i,j} + a_{betsel1,i,j}}{2} \tag{5}$$

$$a_{t12,i,j} = \frac{a_{betsel1,i,rd} + a_{betsel1,i,rd}}{2} \tag{6}$$

$$c_{11,i,j} = a_{i,j} + r_1\left(a_{t11,i,j} - r_2 a_{i,j}\right) \tag{7}$$

$$c_{12,i,j} = a_{i,j} + r_1\left(a_{t12,i,j} - r_2 a_{i,j}\right) \tag{8}$$

$$c_{1sel,i} = \begin{cases} c_{11,i}, f\left(c_{11,i}\right) < f\left(c_{12,i}\right) \\ c_{12,i}, else \end{cases} \tag{9}$$

$$a'_i = \begin{cases} c_{1sel,i}, f\left(c_{1sel,i}\right) < f(a_i) \\ a_i, else \end{cases} \tag{10}$$

The second directed move is formalized using Eq. (11) to Eq. (16). Eq. (11) states that an agent is randomly picked among the swarm. Eq. (12) states that the target for the second directed move is obtained based on the balance mixture between the finest agent and a randomly selected agent. Eq. (13) is used to generate the first candidate of the second directed move using the focus approach. Meanwhile, Eq. (14) is used to generate the second candidate of the second directed move using the shake approach. Eq. (15) states that the finer candidate between these two candidates will be chosen as the final candidate for the second directed move. Eq. (16) states that this candidate replaces the current value of the agent only if it provides enhancement.

$$a_{ransel,i} = U(A) \tag{11}$$

$$a_{t2,i,j} = \frac{a_{best,j} + a_{ransel,i,j}}{2} \tag{12}$$

$$c_{21,i,j} = \begin{cases} a_{i,j} + r_1\left(a_{t2,i,j} - r_2 a_{i,j}\right), f\left(a_{t2,i}\right) < f(a_i) \\ a_{i,j} + r_1\left(a_{i,j} - r_2 a_{t2,i,j}\right), else \end{cases} \tag{13}$$

$$c_{22,i,j} = \begin{cases} a_{i,j} + r_1\left(a_{t2,i,rd} - r_2 a_{i,j}\right), f\left(a_{t2,i}\right) < f(a_i) \\ a_{i,j} + r_1\left(a_{i,j} - r_2 a_{t2,i,rd}\right), else \end{cases} \tag{14}$$

$$c_{2sel,i} = \begin{cases} c_{21,i}, f\left(c_{21,i}\right) < f\left(c_{22,i}\right) \\ c_{22,i}, else \end{cases} \tag{15}$$

$$a_i' = \begin{cases} c_{2sel,i}, f\left(c_{2sel,i}\right) < f(a_i) \\ a_i, else \end{cases} \tag{16}$$

4. Simulation and result

The performance assessment of FSA is carried out in two ways. The first one is the competing assessment. In this assessment, FSA competes with five brand new swarm-based metaheuristics: MA, TIA, LOA, OOA, and KOA. These five competitors are chosen as all of them are first introduced in 2023 so that the scientific enhancement of FSA can be fairly investigated. The old metaheuristics such as GA, particle swarm optimization (PSO), or grey wolf optimization (GWO) are not chosen as these metaheuristics have been beaten by many new metaheuristics. All these competitors deploy strict dimension motion as displayed in Table 1.

The second one is the individual search assessment. As FSA deploys two directed motions, the contribution of each motion should be investigated. This assessment is also called a single missing assessment because there are only two searches performed by every agent in every iteration. The existence of a search means the non-existence of another one. In this assessment, when the first motion is assessed, the second motion is deactivated. On the other hand, when the second motion is assessed, the first motion is deactivated. In this second assessment, FSA does not compete with any other metaheuristics as its objective is not to investigate the competitiveness of FSA compared to other existing metaheuristics, but to investigate the competitive comparison between two motions in FSA.

The set of well-known functions are used as a problem in both assessments. It contains 23 functions including 7 high dimension unimodal functions (HDU), 6 high dimension multimodal functions (HDM), and 10 fixed dimension multimodal functions (FDM). The HDUs are designed to assess the exploitation ability as each function contains only one optimal solution [16]. The HDMs are designed to assess the exploration ability as each function contains multiple optimal solutions so that the main challenge is avoiding the local optimal entrapment [16]. The FDMs are designed to investigate the balance between exploration and exploitation capabilities [16]. A

Table 2. Description of the set of 23 functions

No	Function	*d*	[*lb*, *ub*]	Target
1	Sphere	40	[-100, 100]	0
2	Schwefel 2.22	40	[-100, 100]	0
3	Schwefel 1.2	40	[-100, 100]	0
4	Schwefel 2.21	40	[-100, 100]	0
5	Rosenbrock	40	[-30, 30]	0
6	Step	40	[-100, 100]	0
7	Quartic	40	[-1.28, 1.28]	0
8	Schwefel	40	[-500, 500]	-12,569
9	Ratsrigin	40	[-5.12, 5.12]	0
10	Ackley	40	[-32, 32]	0
11	Griewank	40	[-600, 600]	0
12	Penalized	40	[-50, 50]	0
13	Penalized 2	40	[-50, 50]	0
14	Shekel Foxholes	2	[-65, 65]	1
15	Kowalik	4	[-5, 5]	0.0003
16	Six Hump Camel	2	[-5, 5]	-1.0316
17	Branin	2	[-5, 5]	0.398
18	Goldstein-Price	2	[-2, 2]	3
19	Hartman 3	3	[1, 3]	-3.86
20	Hartman 6	6	[0, 1]	-3.32
21	Shekel 5	4	[0, 10]	-10.153
22	Shekel 7	4	[0, 10]	-10.402
23	Shekel 10	4	[0, 10]	-10.536

Received: March 16, 2024. Revised: April 6, 2024.

description of the 23 functions is displayed in Table 2. The dimension of the multimodal functions is set to 40.

The adjusted parameters are set as follows. The swarm size is set to 10 while the maximum iteration is also set to 10. The result of the first assessment is displayed in Table 3 to Table 5. Then, this result is summarized in Table 6. The result of the second assessment is displayed in Table 7. The decimal point smaller than 10^{-4} is rounded to zero.

Table 3 exhibits that FSA is superior in tackling the HDUs. FSA is the distinct best method in tackling six functions: f_1, f_3 to f_7. Meanwhile, it is shown that all metaheuristics in this assessment achieve the same result in tackling f_2. Meanwhile, TIA is the second-best method while LOA is the worst method. The result also exhibits that the disparity between the best method and the worst method in this first group of functions is wide.

Table 3. Fitness score comparison in tackling HDUs

F	Parameter	MA [19]	TIA [28]	LOA [6]	OOA [13]	KOA [8]	FSA
1	average	1.3135×10^{2}	4.2969	2.5664×10^{3}	1.5886×10^{2}	1.6049×10^{2}	0.0004
	std deviation	4.6457×10^{1}	1.9387	7.2164×10^{2}	7.3544×10^{1}	7.2631×10^{1}	0.0003
	average rank	3	2	6	4	5	1
2	average	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	std deviation	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	average rank	1	1	1	1	1	1
3	average	1.7112×10^{4}	6.9952×10^{2}	5.4899×10^{4}	1.3050×10^{4}	1.8728×10^{4}	7.1358
	std deviation	1.0773×10^{4}	3.8766×10^{2}	2.1960×10^{4}	7.6222×10^{3}	1.3079×10^{4}	6.9396
	average rank	4	2	6	3	5	1
4	average	7.6587	1.7803	2.8890×10^{1}	9.6908	7.7402	0.0174
	std deviation	1.8387	0.4141	4.8292	2.3143	1.7848	0.0057
	average rank	3	2	6	5	4	1
5	average	3.7571×10^{3}	1.0776×10^{2}	8.4804×10^{5}	4.6819×10^{3}	3.6225×10^{3}	3.8862×10^{1}
	std deviation	2.4448×10^{3}	2.6858×10^{1}	6.1222×10^{5}	3.2662×10^{3}	2.5533×10^{3}	0.0406
	average rank	4	2	6	5	3	1
6	average	1.6006×10^{2}	8.2273	2.4976×10^{3}	1.7451×10^{2}	1.4730×10^{2}	5.6808
	std deviation	4.8849×10^{1}	1.4478	1.2949×10^{3}	7.1085×10^{1}	5.1522×10^{1}	0.7909
	average rank	4	2	6	5	3	1
7	average	0.0839	0.0438	0.9631	0.0916	0.0827	0.0110
	std deviation	0.0605	0.0318	0.7233	0.0450	0.0448	0.0080
	average rank	4	2	6	5	3	1

Table 4. Fitness score comparison in tackling HDMs

F	Parameter	MA [19]	TIA [28]	LOA [6]	OOA [13]	KOA [8]	FSA
8	average	-3.1339×10^{3}	-2.0439×10^{3}	-2.7704×10^{3}	-3.2162×10^{3}	-3.0788×10^{3}	-3.0855×10^{3}
	std deviation	4.4389×10^{2}	4.1264×10^{2}	3.5682×10^{2}	5.0957×10^{2}	3.7277×10^{2}	4.8014×10^{2}
	average rank	2	6	5	1	4	3
9	average	1.7959×10^{2}	2.8264×10^{1}	3.4396×10^{2}	1.2619×10^{2}	1.9446×10^{2}	0.0194
	std deviation	6.6704×10^{1}	1.8535×10^{1}	2.4889×10^{1}	5.8339×10^{1}	6.1710×10^{1}	0.0264
	average rank	4	2	6	3	5	1
10	average	4.1798	0.8443	1.0339×10^{1}	3.7820	4.4102	0.0039
	std deviation	0.7325	0.1667	2.2269	0.5139	1.7505	0.0010
	average rank	4	2	6	3	5	1
11	average	2.2123	0.7135	2.1784×10^{1}	2.3798	2.2739	0.0007
	std deviation	0.5297	0.1769	1.2004×10^{1}	0.4967	0.4253	0.0006
	average rank	3	2	6	5	4	1
12	average	2.2624	0.5590	7.8894×10^{4}	2.7604	2.2304	0.5419
	std deviation	0.7188	0.1090	1.6395×10^{5}	1.0863	0.6326	0.1592
	average rank	4	2	6	5	3	1
13	average	1.0960×10^{1}	3.0340	4.8863×10^{5}	1.0655×10^{1}	8.9627	3.0042
	std deviation	2.6831	0.2453	8.9325×10^{5}	2.9390	1.9130	0.3013
	average rank	5	2	6	4	3	1

 DOI: 10.22266/ijies2024.0630.43

Table 5. Fitness score comparison in tackling FDMs

F	Parameter	MA [19]	TIA [28]	LOA [6]	OOA [13]	KOA [8]	FSA
14	average	9.2575	7.9889	1.1243×10^{1}	8.0682	6.7007	6.5511
	std deviation	3.9227	3.5338	4.0403	3.5392	3.4573	3.5063
	average rank	5	3	6	4	2	1
15	average	0.0077	0.0010	0.0251	0.0056	0.0080	0.0008
	std deviation	0.0074	0.0008	0.0192	0.0080	0.0085	0.0006
	average rank	4	2	6	3	5	1
16	average	-1.0167	-1.0300	-0.9511	-1.0240	-1.0180	-1.0309
	std deviation	0.0240	0.0023	0.0934	0.0139	0.0190	0.0011
	average rank	5	2	6	3	4	1
17	average	0.4317	0.4438	0.5060	0.4098	0.4091	0.4007
	std deviation	0.0479	0.1492	0.1058	0.0137	0.0131	0.0043
	average rank	4	5	6	3	2	1
18	average	3.7869	5.2992	1.1529×10^{1}	3.2180	3.6477	3.1261
	std deviation	1.7172	4.1888	1.8497×10^{1}	0.3812	1.2325	0.2909
	average rank	4	5	6	2	3	1
19	average	-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
	average rank	1	1	1	1	1	1
20	average	-2.9714	-2.7415	-2.6081	-3.0083	-2.9275	-2.7968
	std deviation	0.1299	0.2538	0.4023	0.1281	0.1386	0.2898
	average rank	2	5	6	1	3	4
21	average	-3.0348	-3.7725	-1.7789	-1.8971	-2.8453	-5.1443
	std deviation	1.4849	2.0518	1.2732	0.7654	1.6532	1.9714
	average rank	3	2	6	5	4	1
22	average	-2.9941	-3.3549	-1.5196	-2.2759	-2.7873	-5.2565
	std deviation	1.3680	2.1072	0.3708	0.9831	0.9159	2.0886
	average rank	3	2	6	5	4	1
23	average	-2.4187	-2.3689	-1.6572	-2.5846	-3.2240	-5.3974
	std deviation	0.4941	0.7006	0.4208	0.8189	1.2713	2.5758
	average rank	4	5	6	3	2	1

Table 4 exhibits that FSA is still superior in tackling HDMs. FSA is the best method in tackling five functions (f_9 to f_{13}). Meanwhile, FSA is the third best method in tackling f_8 where OOA is the first best method and MA is the second-best method. The disparity performance between the best method and the worst method is wide except in f_8. In this group of functions, TIA is the second-best method.

Table 5 exhibits the superiority of FSA in tackling FDMs. FSA is the distinct best method in eight functions. Meanwhile, all metaheuristics achieve same result in tackling f_{19}. FSA is on the fourth rank in tackling f_{20} where MA, OOA, and KOA are finer than FSA. Different from the high dimension functions, the competition in this group is tough so that the performance disparity between the best and worst methods is narrow.

Table 6 strengthens the superiority of FSA among its competitors. Overall, FSA is finer than MA, TIA, LOA, OOA, and KOA in 19, 21, 21, 19, and 20 functions respectively. This result also summarizes the superiority of FSA in all three groups of functions. As all metaheuristics achieve same result in f_2 and f_{19}, TIA and LOA never beat FSA.

Table 6. Group-based superiority of FSA.

Group	Number of Functions Where SBA is Finer				
	MA [19]	TIA [28]	LOA [6]	OOA [13]	KOA [8]
1	6	6	6	6	6
2	5	6	6	5	6
3	8	9	9	8	8
Total	19	21	21	19	20

Table 7 exhibits that both searches are complementary to each other. But the first search provides more contribution that the second search. The first search is finer than the second search in 15 functions. On the other hand, the second search is finer than the first search in 6 functions.

Received: March 16, 2024. Revised: April 6, 2024.

5. Discussion

The competing assessment result shows that FSA is a good and superior metaheuristic. It has found the quasi-optimal solution of all 23 functions. Moreover, it is superior to its five competitors in almost all functions. FSA is the first best method in 21 functions with the note that it is the sole first best method in 19 functions due to the same result has been achieved by all metaheuristics in two functions (f_2 and f_{19}).

The superiority of FSA in all groups of functions proves that FSA has good exploration and exploitation abilities. Its exploitation ability is outstanding as it is superior in tackling HDUs [9]. Its exploration ability is outstanding as it is superior in tackling HDMs [9]. Moreover, both of its capabilities are balance as it is superior in tackling FDMs. Meanwhile, the result still proves the existence of NFL theory as FSA is not the best method in two functions (f_8 and f_{20}). But FSA is still competitive in these two functions as the performance disparity among the metaheuristics is narrow.

The competing assessment result also indicates that the existence of the neighbourhood search with declining search space along the iteration is not critical. FSA is still superior although it does not implement this search. It is still finer than MA, LOA [6], OOA [13], and KOA [8] that implement this search as complementary to the directed search. This finding is also strengthened by the fact that overall, TIA becomes the second-best method, and it also does not implement the neighbourhood search [28].

Meanwhile, the existence of the directed search toward the finer agent which is conducted in a dedicated manner is important. This finding comes from the fact that LOA becomes the worst method. In LOA, the existence of the directed search toward a finer agent is only 50 percent as it should be shared with the neighbourhood search with declining search space along the iteration [6].

The result also indicates two strengths of FSA. The first factor is the combination of strict dimension mapping (focus approach) and randomized dimension mapping (shake approach). The second factor is its two directed motions. In FSA, the existence of the finer agent can be found in both directed motions. In the first search, the reference is constructed by two finer agents. Meanwhile, the finest agent contributes to constructing the reference in the second directed motion besides the randomly chosen agent as diversification.

Table 7. Single search assessment result.

F	Average Fitness Score	
	1st Search	**2nd Search**
1	2.6086	3.6961
2	0.0000	0.0000
3	5.0224×10^{2}	1.2194×10^{3}
4	0.9191	1.2034
5	6.7040×10^{1}	8.2205×10^{1}
6	7.9244	8.7964
7	0.0232	0.0473
8	-2.3733×10^{3}	-3.3122×10^{3}
9	6.1006	1.5828×10^{2}
10	0.6599	0.7146
11	0.5539	0.6402
12	0.5853	0.6814
13	2.8488	3.0594
14	8.0790	8.2462
15	0.0013	0.0015
16	-1.0049	-1.0253
17	0.8067	0.4217
18	1.5389×10^{1}	3.6353
19	-0.0495	-0.0495
20	-2.5703	-2.7249
21	-3.0580	-3.3452
22	-3.4972	-3.1658
23	-3.4488	-3.1520

The individual search assessment result shows that both searches are complementary to each other so that both searches are important. Although the first reference performs finer than the second reference, the second reference provides more contribution in the multimodal functions as its superiority takes place in six multimodal functions. Besides, the performance disparity between these two motions is narrow in almost all functions.

The computational complexity of FSA can be displayed in big-O, and it depends on the number of loops. In the initialization, the nested loop contains only two loops with the factor is the swarm size (outer loop) and the dimension (inner loop). Meanwhile, each agent performs only one search in this phase which is the full random search. Based on this explanation, the computational complexity during the initialization phase can be displayed as $O(n(A).d)$. On the other hand, the nested loop in the iteration phase contains three factors: maximum iteration (outer loop), swarm size (intermediate loop), and dimension (inner loop). Each agent also creates pool first before conducting the first search which means tracing all agents inside swarm. Then, there are four searches in every iteration which is grouped into two motions. Each motion performs two searches (focus and shake). This explanation concludes that the computational complexity of FSA during the iteration can be presented as $O(t_m.n(A).(1+4d))$.

Despites its superiority in tackling theoretical optimization problems with various circumstance, there are several limitations in this work which can be used as baselines for future works. First, FSA has not been tested to solve any practical problems. Second, FSA has not been employed to tackle any combinatorial optimization problems as the 23 functions used in this work is numerical optimization problems. Third, FSA cannot accommodate all techniques available in the stochastic optimization methods as it is impossible for any algorithm to cover all tools.

6. Conclusion

In this paper, a new swarm-based called focus and shake algorithm (FSA) has been introduced. There are two approaches of the dimension mapping between the agent and its reference in FSA. Strict dimension mapping becomes the first approach and randomized dimension mapping becomes the second approach. These two approaches become the novelty and difference between FSA and other swarm-based metaheuristics. Through competing assessment, it is shown that FSA is proven as a good metaheuristic because it can find the quasi-optimal solution of all 23 functions. Meanwhile, FSA is proven as a superior metaheuristic as it is finer than its competitors in almost all functions.

In the future, the following studies can be conducted in several ways. There are more stochastic methods that can be explored in shaking the dimension mapping so that the first track for further studies can be performed by developing shaking alternative rather than uniform random as presented in this work. The second track is implementing FSA to solve practical optimization problems in many sectors. The third track is combining FSA with other optimization methods to create a more powerful optimization tool. The fourth track is combining FSA with machine learning methods whether they are supervised or unsupervised methods to solve a wider range of problems.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

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

Acknowledgments

Telkom University provides the funding for the publication of this paper.

References

[1] M. Mageshwari and R. Naresh, "Improved Sunflower Optimization Algorithm Based Encryption with Public Auditing Scheme in Secure Cloud Computing", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 6, pp. 13–23, 2023, doi: 10.22266/ijies2023.1231.02.

[2] M. Pradhan, I. L. Coman, S. Mishra, T. Thieu, and A. Bhuiyan, "LSTM based Modified Remora Optimization Algorithm for Lung Cancer Prediction", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 6, pp. 46–59, 2023, doi: 10.22266/ijies2023.1231.05.

[3] M. R. Archana, D. Jamkhandikar, and D. N. Biradar, "Image Copy-move Forgery Detection and Classification Using Golden Jackal Optimization Based Multi-Support Vector Machine", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 6, pp. 123–133, 2023, doi: 10.22266/ijies2023.1231.11.

[4] 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.

[5] Suyanto, A. A. Ariyanto, and A. F. Ariyanto, "Komodo Mlipir Algorithm", *Applied Soft Computing*, Vol. 114, ID: 108043, 2022.

[6] 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.

[7] 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.

[8] 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.

[9] M. H. Amiri, N. M. Hashjin, M. Montazeri, S. Mirjalili, and N. Khodadadi, “Hippopotamus Optimization Algorithm: A Novel Nature-inspired Optimization Algorithm”, *Scientific Reports*, Vol. 14, ID. 5032, pp. 1-50, 2024.

[10] W. Zhao, L. Wang, Z. Zhang, H. Fan, J. Zhang, S. Mirjalili, N. Khodadadi, and Q. Cao, “Electric Eel Foraging Optimization: A New Bio-inspired Optimizer for Engineering Applications”, *Expert Systems with Applications*, Vol. 238, ID. 122200, 2024.

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

[12] 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.

[13] 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.

[14] E. M. El-kenawy, N. Khodadadi, S. Mirjalili, A. A. Abdelhamid, M. M. Eid, A. Ibrahim, “Greylag Goose Optimization: Nature-inspired Optimization Algorithm”, *Expert Systems with Applications*, Vol. 238, ID. 122147, 2024.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[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] A. Kaveh, S. Talatahari, and N. Khodadadi, “Stochastic Paint Optimizer: Theory and Application in Civil Engineering”, *Engineering with Computers*, Vol. 38, pp. 1921-1952, 2022.

[21] S. O. Oladejo, S. O. Ekwe, L. A. Akinyemi, and S. A. Mirjalili, "The Deep Sleep Optimizer: A Human-Based Metaheuristic Approach", In: *IEEE Access*, Vol. 11, pp. 83639-83665, 2023.

[22] 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.

[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] T. R. Farshi, “Battle Royale Optimization Algorithm”, *Neural Computing and Applications*, Vol. 33, pp. 1139-1157, 2021.2

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

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

[28] P. D. Kusuma and A. Novianty, “Total Interaction Algorithm: A Metaheuristic in Which Each Agent Interacts with All Other Agents”, *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 1, pp.

224-234, 2023, doi: 10.22266/ijies2023.0228.20.

[29] M. Dehghani, E. Trojovska, P. Trojovsky, and O. P. Malik, “OOBO: A New Metaheuristic Algorithm for Solving Optimization Problems”, *Biomimetics*, Vol. 8, ID. 468, pp. 1-48, 2023.

[30] 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.

[31] 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.

[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, doi: 10.22266/ijies2023.1031.51.

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

[34] 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.

[35] P. Gonzalez-Ayala, A. Alejo-Reyes, E. Cuevas, and A. Mendoza, “A Modified Simulated Annealing (MSA) Algorithm to Solve the Supplier Selection and Order Quantity Allocation Problem with Non-Linear Freight Rates”, *Axioms*, Vol. 12, ID. 459, pp. 1-17, 2023.

[36] S. Katoch, S. S. Chauhan, and V. Kumar, “A Review on Genetic Algorithm: Past, Present, and Future”, *Multimedia Tools and Applications*, Vol. 80, pp. 8091-8126, 2021.

[37] 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.