International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Migrating Walrus Algorithm: A New Metaheuristic that Hybridizes Migration Algorithm and Walrus Optimization Algorithm

Purba Daru Kusuma[1]* **Anggunmeka Luhur Prasasti[1]**

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

Abstract: Migration algorithm (MA) and walrus optimization algorithm (WaOA) are two new swarm-based metaheuristics which are first introduced in 2023. As new metaheuristics, the modification of these two metaheuristics is still rare. Based on this circumstance, this work constructs a new metaheuristic called as migrating walrus algorithm (MWA) based on the hybridization of both MA and WaOA to create a better metaheuristic than them. MWA consists of five migrations: four directed migrations and one local migration. The references used in these directed migrations are the best walrus, a randomly picked better walrus, a randomly picked walrus, and two randomly picked walruses. In this work, two assessments are carried out: the comparative assessment and the individual migration assessment. The 23 functions are selected as theoretical use cases. In the comparative assessment, MWA is confronted with five new metaheuristics: MA, WaOA, attack leave optimization (ALO), coati optimization algorithm (COA), and osprey optimization algorithm (OOA). The result shows that MWA is better than ALO, COA, MA, OOA, and WaOA in 19, 19, 19, 17, and 17 functions. On the other hand, the individual migration assessment result indicates that the multiple migration approach is important to maintain the superiority of MWA with the directed migration toward the best walrus becoming the most important contributor. This result also strengthens the necessity of the multiple searches strategy rather than a single strategy.

Keywords: Metaheuristic, Optimization, Swarm intelligence, Migration algorithm, Walrus optimization algorithm.

1. Introduction

Metaheuristics is a popular optimization technique that has been used in various optimization works. It is implemented in a wide variety of engineering problems to optimize engineering works. The example is as follows. Mahadevachar and Hosur modified the new battle royale optimization (BRO) into trust-based multi-objective battle royale optimization (T-MBRO) to optimize energy efficiency in the mobile ad-hoc network (MANET) [1]. There are five parameters constructed as its objective: distance, energy consumption, trust, and the packet forwarding rate [1]. Mahdi and Yuhaniz modified the sparrow search algorithm and utilized it to improve the categorization of COVID-19 patients [2]. The ant colony optimization (ACO) and whale optimization algorithm (WOA) has been hybridized to improve the path selection process in the vehicular ad-hoc networks (VANET) system [3]. Bektas, Karaca, Taha, and Zaynal utilized the red deer algorithm (RDA) to eliminate the desired harmonic order in the multi-level inverters (MLI) by finding the optimum switching angles in this MLI [4]. The bat algorithm (BA) has been utilized for the frequency stability improvement in the power system, specifically in the proportional integral differential (PID) controller and battery energy storage system (BESS) [5].

Many swarm-based metaheuristics are introduced in recent years. Nature becomes the main inspiration for these new metaheuristics, especially the food-finding behavior of animals. Some metaheuristics were introduced in 2021, such as the chameleon swarm algorithm (CSA) [6], coronavirus herd immunity optimization (CHIO) [7], three influential members-based optimization (TIMBO) [8], red fox optimization (RFO) [9], remora optimization algorithm (ROA) [10], random selected leader-based optimization (RSLBO) [11], ring toss game-based optimization (RTGBO) [12], and northern goshawk optimization (NGO) [13]. Some metaheuristics were introduced in 2022, such as hybrid leader-based optimization (HLBO) [14], coronavirus optimization algorithm (COVIDOA) [15], chef-based optimization algorithm (CBOA) [16], election-based optimization algorithm (EBOA) [17], zebra optimization algorithm (ZOA) [18], white shark optimization (WSO) [19], driving training based optimization (DTBO) [20], Tasmanian devil optimization (TDO) [21], reptile search algorithm (RSA) [22], Komodo mlipir algorithm (KMA) [23], Siberian tiger optimization (STO) [24], and golden jackal optimization (GJO) [25]. Some metaheuristics were introduced in 2023, such as coati optimization algorithm (COA) [26], migration algorithm (MA) [27], osprey optimization algorithm (OOA) [28], walrus optimization algorithm (WaOA) [29], quad tournament optimization (QTO) [30], total interaction algorithm (TIA) [31], attack-leave optimization (ALO) [32], green anaconda optimization (GAO) [33], and swarm magnetic optimization (SMO) [34].

One problem in the development of metaheuristics is the lack of modification or hybridization of the existing ones. Some metaheuristics, such as grey wolf optimization (GWO) and marine predator algorithm (MPA) are very popular so many studies taken to modify these metaheuristics. Unfortunately, many other metaheuristics receive less attention although they are powerful enough. This circumstance is often faced by newer metaheuristics.

Meanwhile, a study on modifying the existing metaheuristics or hybridizing them with other techniques is as important as introducing brand-new ones. Despite creating potential improvement, modification or hybridization is important to maintain the continuity of the related metaheuristics.

Another problem in the massive development of metaheuristics is the lack of individual search assessment. Many recent metaheuristics were developed based on the multiple search approach. Meanwhile, the assessment of their performance is still the same as the older ones which is the comparative assessment. This circumstance is related to the trend of beating the older ones with the new ones. Ironically, the investigation of each technique that constructs the metaheuristic is not taken. Many studies assessed the proposed metaheuristic as a whole package rather than the modular assessment. Meanwhile, this modular assessment is important to investigate the contribution, dispensability, and replaceability of each search.

This problem is then transformed into the introduction of a new metaheuristic hybridizing both MA and WaOA which becomes the objective of this work. This introduced metaheuristic is called migrating walrus algorithm (MWA) which gives credit to both metaheuristics as its origin. MWA is designed to improve the performance of MA and MWA. As a hybrid metaheuristic, MWA has some characteristics that are inherited from its origins: (1) swarm-based metaheuristic, (2) multiple search approach, (3) consisting of directed search and local search, and (4) strict replacement rule.

Moreover, this work has several scientific contributions as follows.

1) This work presents a new swarm-based metaheuristic that hybridizes two new metaheuristics (MA and WaOA) where the presentation consists of the main concept and formalization.
2) The comparative assessment of the MWA performance is carried out by confronting it with five new metaheuristics.
3) The individual migration assessment is carried out to investigate each search contribution and strength in constructing the MWA.

The rest of this paper is formulated as follows. The review of several recent metaheuristics, especially the MA and WaOA is carried out in section 2. This review is aimed to evaluate the fundamental concept and strategy used in these reviewed metaheuristics. The detailed presentation of MWA as the proposed metaheuristic can be seen in section 3 which consists of the fundamental concept, algorithm, and formalization. The assessment of the MWA performance is presented in section 4 which consists of the comparative assessment and individual search assessment. The more profound evaluation related to the result, findings, complexity, and limitations is discussed in section 5. Finally, the conclusion and proposal related to the possible tracks of future studies are presented in section 6.

2. Related works

Many recent metaheuristics were developed based on the swarm intelligence approach. By using this approach, the new metaheuristic is constructed based on the population or swarm where each member of a swarm is active and autonomous [34]. Due to this autonomy, this member searches for improvement based on its current condition and its perspective of its environment. Its current condition includes its current location and the quality of its current location or solution. Meanwhile, the environment includes the other members within the swarm and the search space.

This metaheuristics development approach has different methods from the older one, such as the single neighborhood search and the population-based evolutionary approach. In the single solution-neighborhood search, there is only one active solution or agent, and it works by searching for a better solution nearby it. On the other hand, both evolutionary-based metaheuristic and swarm-based metaheuristic are constructed by a population. Meanwhile, each member of the population in the evolutionary-based metaheuristic is passive. This member changes its value or is replaced by another member based on the centralized mechanism.

The directed search becomes the primary search in the swarm-based metaheuristic [34]. There are two fundamental elements in this kind of search: the reference and the step size. This reference can be the best member, other members, some places within the space, and so on. Meanwhile, the step size is usually calculated stochastically with various stochastic patterns, such as uniform [31], normal [23], Brownian [35], Levy [35], and so on. The variety of references and the step size becomes one of several reasons for the popularity of swarm intelligence as the baseline for recent metaheuristics. In most of them, the movement is taken by the corresponding member while in some of them, the movement is carried out by the reference. The summary of the strategy performed by several brand-new metaheuristics is presented in Table 1. The last row depicts the fundamental concept of the proposed metaheuristics.

Table 1. Summary of several metaheuristics first introduced in 2023

No	Metaheuristic	Instance	Fundamental Concept
1	ALO [32]	member	The member moves toward the best member, or the best member avoids the member based on a threshold as the first move. A reference is constructed in the middle between the best member and a randomly picked member, or two randomly picked members based on a threshold. Then, the reference avoids the member, or the member avoids the reference based on quality comparison as the second move. A full random movement is taken as an optional third move if stagnation happens.
2	COA [26]	coati	In the first move, half of the coatis moves toward the iguana on the three (best member) while the rest of the coatis moves toward the iguana on the ground (a randomized member within the space). In the second move, all coatis try to escape from the predator by performing a local search.
3	MA [27]	member	The member migrates to a better place by picking up a better destination (a randomly picked better member) as the first move. The member adapts to the new environment by performing a local search.
4	OOA [28]	osprey	The osprey hunts a random fish (a member from a set of better members plus the best member) as the first move. The osprey carries the fish to a suitable location by performing a local search as the second move.
5	WaOA [29]	walrus	The walrus moves toward the strongest walrus (best member) as the first move. The walrus moves relative to another walrus (a randomly picked member) as the second move. The walrus escapes from predators by performing a local search as a third move.
6	GAO [33]	green anaconda	A male anaconda moves toward a female anaconda (a randomly picked better member based on quality normalization) as the first move. All anaconda hunts prey near them by performing a local search as the second move.
7	TIA [31]	member	The member moves relative to all other members as the sole search.
8	SMO [34]	magnet	The magnet interacts with the best magnet as the first move. The magnet interacts with a randomly picked magnet as the second move. The magnet interacts with a randomized magnet within space. The interaction represents the movement of both the magnet and its reference. The movement can be getting closer or moving away which is controlled by the iteration.
9	this work	walrus	The walrus migrates toward the strongest walrus as the first move. The walrus migrates toward a better walrus (a randomly picked better member) as the second move. The walrus migrates relative to a randomly picked walrus as the third move. The walrus migrates toward the middle between two randomly picked walruses as the fourth move. The walrus adapts to its new environment by performing a local search as the fifth move.

Received: September 5, 2023. Revised: September 20, 2023.

Based on this explanation, there are a lot of opportunities to modify, combine, and hybridize the existing metaheuristics, especially the new ones. Table 1 shows that various options can be selected as references. Meanwhile, Table 1 also shows that some metaheuristics perform a local search as a secondary search while others do not. As additional information, the local search performed in metaheuristics in Table 1 is a local search where the search distance degrades during the iteration. This degradation pace can be linear or logarithmic.

3. Model

The fundamental concept of MWA is the hybridization of MA and WaOA. As both are swarm-based metaheuristics, then MWA is a swarm-based metaheuristic too. It means that MWA is constructed by several autonomous agents called walrus that search or migrate independently without any central command and control. Meanwhile, several references are used during the directed migration. These references include the best walrus, a randomly chosen better walrus, a randomly chosen walrus, and two randomly chosen walrus. Meanwhile, as performed in both MA and WaOA, the local migration is also performed in MWA, where the walrus migrates randomly in its local space for improvement.

This fundamental concept of hybridization is then transformed into five migrations, where four of them are directed migrations while another migration is local migration. The first migration is the migration toward the best walrus. The second migration is the migration toward a randomly selected better walrus. The third migration is a migration relative to a randomly selected walrus. The fourth migration is a migration toward the middle between two randomly selected walruses. The fifth walrus is a local migration around the corresponding walrus. Each migration produces a single child. Then, this child is compared with its parent for replacement if the quality of this child is better than its parent. This procedure represents the strict replacement procedure as commonly found in many recent metaheuristics.

This MWA consists of two phases as common in any metaheuristics. The first phase is the initialization where all walruses perform full random migration. Then, the five migrations are carried out by all walruses in every iteration during the iteration phase as the second phase. The formalization of MWA can be seen in algorithm 1 which is presented in pseudocode. Meanwhile, a more detailed presentation of each procedure is presented in Eqs. (1) to (11). Several annotations used in this work are explained below.

d	dimension
f	objective function
r_1	a floating point-based uniform random number between 0 and 1
r_2	integer uniform random number between 1 and 2
r_3	a floating point based on the uniform random number between -1 and 1
t	iteration
t_m	maximum iteration
U	uniform random
w	walrus
W	a collection of walruses
w_b	the best walrus
w_c	walrus child
w_{lo}	the lower boundary of search space
w_{hi}	the higher boundary of the search space
w_{le}	a collection of better walruses
w_{tle}	the walrus target picked from the set of better walruses
w_t	a randomly picked walrus

Algorithm 1: pseudocode of MWA

```
1   output: w_b
2   begin
3     set n(w), t_m
4     for all w in W
5       initialize w_i using Eq. (1)
6       update w_b using Eq. (2)
7     end for
8     for t=1 to t_m
9       for all w in W
10        1st migrate using Eq. (3) and Eq. (4)
12        2nd migrate using Eq. (5) to Eq. (7), Eq. (4)
13        3rd migrate using Eq. (8), Eq. (9), Eq. (4)
14        4th migrate using Eq. (8), Eq. (10), Eq. (4)
15        5th migrate using Eq. (11) and Eq. (4)
16        update w_b using Eq. (2)
17      end for
18    end for
19  end
```

The initialization phase is presented in line 4 to line 7 in Algorithm 1. There are two processes during the initialization phase. The first process is generating the initial walrus based on uniform distribution as stated in Eq. (1). The second process is updating the best walrus as stated in Eq. (2).

$$w_{i,j} = w_{lo,j} + r_1\left(w_{hi,j} - w_{lo,j}\right) \tag{1}$$

$$w_b' = \begin{cases} w_i, f(w_i) < f(w_b) \\ w_b, otherwise \end{cases} \tag{2}$$

The first migration is the migration toward the best walrus. In algorithm 1, this procedure contains two processes. The first process is generating a child based on the movement toward the best walrus as presented in Eq. (3). The second process is performing the strict replacement process as presented in Eq. (4).

$$w_{c,j} = w_{i,j} + r_1\left(w_{b,j} - r_2 w_{i,j}\right) \quad (3)$$

$$w_i' = \begin{cases} w_c, f(w_c) < f(w_i) \\ w_i, otherwise \end{cases} \quad (4)$$

The second migration consists of four processes. The first process is filling the leaders' pool using Eq. (5). The second process is picking a better walrus randomly as stated in Eq. (6). The third process is generating a child by migrating toward this randomly picked better walrus as stated in Eq. (7). The fourth process is strict replacement rule as it is presented in Eq. (4).

$$W_{le,i} = \{w \in W, f(w) < f(w_i)\} \quad (5)$$

$$w_{tle,i} = U(W_{le,i}) \quad (6)$$

$$w_{c,j} = w_{i,j} + r_1\left(w_{tle,i,j} - r_2 w_{i,j}\right) \quad (7)$$

The third migration is the migration relative to a randomly selected walrus. This migration is presented in line 13 in Algorithm 1. It consists of three processes. The first process is selecting a walrus uniformly within the set of walruses as presented in Eq. (8). Then, a child is generated based on the migration of the corresponding walrus relative to this reference where the direction is determined based on the quality comparison between the corresponding walrus and this randomly selected walrus as presented in Eq. (9). Then, the strict replacement is performed as presented in Eq. (4).

$$w_t = U(W) \quad (8)$$

$$w_{c,j} = \begin{cases} w_{i,j} + r_1\left(w_{t,j} - r_2 w_{i,j}\right), f(w_t) < f(w_i) \\ w_{i,j} + r_1\left(w_{i,j} - r_2 w_{t,j}\right), otherwise \end{cases} \quad (9)$$

The fourth migration is the migration toward the middle between two randomly selected walruses. This migration is presented in line 14 in Algorithm 1. It consists of three processes. The first process is selecting two walruses randomly by using Eq. (8). The second process is generating a child by using Eq. (10). The third process is performing the strict replacement by using Eq. (4).

Table 2. List of 23 functions

No	Function	*d*	Space	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]	-16,759
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

$$w_{c,j} = w_{i,j} + r_1\left(\frac{w_{t1,j} + w_{t2,j}}{2} - r_2 w_{i,j}\right) \quad (10)$$

The fifth migration is the local migration. It is presented in line 15 in Algorithm 1. Meanwhile, the formalization is presented in Eq. (11). In Eq. (11), the local space is reduced by the iteration where the initial search space is as wide as the space between the lower boundary and the higher boundary. Then, the child generated in this migration is also evaluated by using Eq. (4) for the replacement procedure.

$$w_{c,j} = w_{i,j} + \frac{r_3\left(w_{hi,j} - w_{lo,j}\right)}{t} \quad (11)$$

4. Simulation and result

There are two assessments carried out to evaluate the performance of MWA. The first assessment is a comparative assessment while the second assessment is an individual migration assessment.

In the first assessment, the performance of

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

F	Parameter	ALO [32]	COA [26]	MA [27]	OOA [28]	WaOA [29]	MWA
1	mean	0.0012	3.4953×10^{1}	1.1373×10^{1}	4.1902	0.0241	0.0000
	std-dev	0.0025	3.3440×10^{1}	6.2852	2.4034	0.0406	0.0000
	mean rank	2	6	5	4	3	1
2	mean	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	std-dev	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	mean rank	1	1	1	1	1	1
3	mean	6.7399	8.9714×10^{3}	1.5283×10^{4}	5.3916×10^{3}	2.6454×10^{2}	2.2527
	std-dev	2.1496×10^{1}	6.2035×10^{3}	1.2735×10^{4}	3.8780×10^{3}	4.1231×10^{2}	3.2758
	mean rank	2	5	6	4	3	1
4	mean	0.0369	7.8869	6.7296	2.9877	0.2565	0.0003
	std-dev	0.0514	2.5815	1.3433×10^{1}	1.1220	0.1763	0.0003
	mean rank	2	6	5	4	3	1
5	mean	3.9051×10^{1}	1.4857×10^{3}	1.9658×10^{2}	9.0236×10^{1}	3.9289×10^{1}	3.8937×10^{1}
	std-dev	0.1645	1.7446×10^{3}	1.5145×10^{2}	2.5397×10^{1}	0.3059	0.0267
	mean rank	2	6	5	4	3	1
6	mean	8.7987	4.3809×10^{1}	1.1522×10^{1}	1.5204×10^{1}	8.2021	8.2772
	std-dev	0.3491	1.7510×10^{1}	0.0000	4.6086	0.4487	0.3984
	mean rank	3	6	4	5	1	2
7	mean	0.0384	0.1054	0.0691	0.0611	0.0245	0.0109
	std-dev	0.0339	0.0573	0.0507	0.0336	0.0163	0.0078
	mean rank	3	6	5	4	2	1

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

F	Parameter	ALO [32]	COA [26]	MA [27]	OOA [28]	WaOA [29]	MWA
8	mean	-3.3021×10^{3}	-3.7163×10^{3}	-3.1263×10^{3}	-3.3478×10^{3}	-3.2851×10^{3}	-3.2308×10^{3}
	std-dev	5.3603×10^{2}	7.4504×10^{2}	5.3851×10^{2}	4.5320×10^{2}	7.2969×10^{2}	4.8123×10^{2}
	mean rank	3	1	6	2	4	5
9	mean	0.0135	3.4124×10^{1}	6.6801×10^{1}	2.2081×10^{1}	0.3835	0.0000
	std-dev	0.0519	3.7880×10^{1}	6.8032×10^{1}	1.8564×10^{1}	1.1704	0.0000
	mean rank	2	5	6	4	3	1
10	mean	0.0054	2.0320	3.9855	0.8259	0.0299	0.0000
	std-dev	0.0093	0.6117	6.3164	0.2788	0.0154	0.0000
	mean rank	2	5	6	4	3	1
11	mean	0.0339	1.4200	1.0155	0.7459	0.0364	0.0034
	std-dev	0.1200	0.3104	0.2392	0.2643	0.0721	0.0154
	mean rank	2	6	5	4	3	1
12	mean	1.1874	1.9034	1.3283	1.2437	1.0187	0.9996
	std-dev	0.1209	0.5746	0.2890	0.2398	0.1729	0.1405
	mean rank	3	6	5	4	2	1
13	mean	3.1545	7.3788	4.3816	4.0316	3.2060	3.0788
	std-dev	0.0360	3.0115	0.3317	0.4281	0.0805	0.0503
	mean rank	2	6	5	4	3	1

MWA is compared with other metaheuristics. On the other hand, in the second assessment, each migration of MWA is assessed individually. In both assessments, the swarm size is 5 while the maximum iteration is 15. The value less than 10^{-4} is rounded to 0.

In both assessments, the set of 23 functions is used as the use case. The list of these functions can be seen in Table 2. There are three types of functions in these 23 functions: high-dimension unimodal functions (function 1 to function 7), high-dimension multimodal functions (function 8 to function 13), and fixed-dimension multimodal functions (function 14 to function 23).

There are five existing metaheuristics used as confronters in the first assessment: ALO [32], COA [26], MA [27], OOA [28], and WaOA [29]. All these metaheuristics are new as they were first introduced in 2023. All of them are swarm-based metaheuristics. All of them implement a strict

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

F	Parameter	ALO [32]	COA [26]	MA [27]	OOA [28]	WaOA [29]	MWA
14	mean	7.4669	8.4985	9.9130	8.2698	9.5510	8.3798
	std-dev	5.4011	5.3438	3.9681	4.7209	3.7439	2.9740
	mean rank	1	4	6	2	5	3
15	mean	0.0103	0.0101	0.0086	0.0058	0.0019	0.0060
	std-dev	0.0111	0.0096	0.0081	0.0097	0.0018	0.0169
	mean rank	6	5	4	2	1	3
16	mean	-0.9975	-1.0280	-1.0122	-1.0235	-1.0116	-1.0271
	std-dev	0.0443	0.0043	0.0220	0.0123	0.0341	0.0086
	mean rank	6	1	4	3	5	2
17	mean	0.5910	0.4708	0.4110	0.4230	0.4080	0.4104
	std-dev	0.2269	0.2429	0.0285	0.0780	0.0129	0.0325
	mean rank	6	5	3	4	1	2
18	mean	8.7663	8.2369	3.7114	4.3716	$2.0953x10^1$	7.3574
	std-dev	8.1830	$1.0078x10^1$	2.2464	5.8071	$3.0096x10^1$	$1.7594x10^1$
	mean rank	5	4	1	2	6	3
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	-2.4373	-2.9616	-2.9686	-2.9655	-2.9696	-3.0835
	std-dev	0.3052	0.1892	0.1397	0.1956	0.1979	0.1496
	mean rank	6	5	3	4	2	1
21	mean	-1.6549	-3.4033	-3.4168	-1.7364	-2.3906	-4.5161
	std-dev	1.6766	1.4104	1.3967	1.0690	1.4253	1.7653
	mean rank	6	3	2	5	4	1
22	mean	-1.7479	-3.4544	-3.3698	-3.0821	-2.6182	-3.8515
	std-dev	0.9280	1.2979	1.3315	2.3265	1.4615	1.1018
	mean rank	6	2	3	4	5	1
23	mean	-1.8457	-3.2817	-4.1415	-2.3249	-2.4130	-3.4877
	std-dev	0.8839	0.8030	1.7308	0.7445	0.6844	0.9268
	mean rank	6	3	1	5	4	2

replacement procedure. The result is presented in Tables 3 to 5. In these tables, there are three data: average fitness score (mean), standard deviation, and mean rank. Then, this data is summarized in Table 6 to evaluate the superiority of MWA to the related metaheuristic.

Table 3 indicates the superiority of MWA in solving the high-dimension unimodal functions. MWA sits on the first rank in solving six functions (Sphere, Schwefel 2.22, Schwefel 1.2, Schwefel 2.21, Rosenbrock, and Quartic) and on the second rank in solving one function (Step). Moreover, MWA can find the global optimal solution in two functions (Sphere and Schwefel 2.22). As a note, all metaheuristics in this assessment also can find the global optimal solution in solving Schwefel 2.22.

The confrontation in this first group of functions is dynamic enough. ALO is competitive enough due to its performance to sit on the first rank four times and on the third rank twice. WaOA is also competitive as it sits on the first rank once, on the second rank once, and on the third rank four times. Meanwhile, COA and MA can be said to be the two worst performers as both sit in the fifth and sixth ranks several times. Meanwhile, the performance of OOA is moderate.

The performance gap among the functions in this first group of functions varies. The very close gap can be seen in solving Schwefel 2.22. The performance gap in solving Quartic is also close. Moderate performance gaps among metaheuristics can be found in solving Sphere, Schwefel 2.21, and Step. In the end, a large performance gap can be found in solving Schwefel 1.2 and Rosenbrock.

The superiority of MWA among its confronters is continuous in solving the high-dimension multimodal functions. MWA becomes the best performer in solving five functions (rastrigin, ackley, griewank, penalized, and penalized 2). Unfortunately, MWA becomes the second worst performer (on the fifth rank) in solving one function (Schwefel). COA becomes the best performer in solving this function. Moreover, MWA can find the global optimal solution in two functions (rastrigin

Table 6. Group-based superiority of MWA

Group	Number of Functions Where MWA is Better				
	ALO [32]	**COA [26]**	**MA [27]**	**OOA [28]**	**WaOA [29]**
1	6	6	6	6	5
2	5	5	6	5	5
3	8	8	7	6	7
Total	19	19	19	17	17

and ackley).

Table 5 indicates that MWA is less superior following the fierce confrontation among metaheuristics in solving the fixed dimension multimodal functions. MWA sits on the first rank in four functions (hartman 3, hartman 6, shekel 5, and shekel 7), second rank in three functions (six hump camel, branin, and shekel 10), and third rank in three functions (shekel foxholes, kowalik, and goldstein price). As presented, MWA sits never worse than on the third rank. On the other hand, the confrontation among the confronters is very dynamic. There exists a condition where each confronter sits on the fifth or sixth rank. Moreover, the performance gap among the metaheuristics is narrow except in solving goldstein-price.

Table 6 strengthens the superior performance of MWA among its confronters. MWA is better than ALO, COA, MA, OOA, and WaOA in solving 19, 19, 19, 17, and 17 functions consecutively. Meanwhile, there are two functions where all six metaheuristics achieve equal performance, which are Schwefel 2.22 and Hartman 3. This result also shows that the superiority of MWA takes place in all three groups of functions.

The second assessment is carried out to compare each migration constructing the MWA performance. As an individual migration assessment, each migration is challenged to solve these 23 functions without the other migration contributions. It means that when migration is assessed, the other migrations are set to passive. The result is presented in Table 7 where the best result in each function is written bold.

Table 7 shows that the first search plays a dominant role in MWA. By neglecting two functions (schwefel 2.22 and hartman 3) where multiple migrations perform the best result, the first migration becomes the best performer in eleven functions. Most of these functions are high-dimension functions. The second and third migrations become the best performer in the two functions. The fourth and fifth migrations become the best performers in three functions.

5. Discussion

In general, MWA achieves successful performance as a superior metaheuristic. First, MWA has found the quasi-optimal solution in most of the functions. Second, MWA has found the global optimal solution in several functions. Third, MWA is superior to its confronters although all these confronters are new metaheuristics. Its superiority occurs in all groups of functions.

The result in Table 7 highlights the necessity of the directed migration toward the best walrus. In general terms, it is the directed search toward the best member. This migration should be performed dedicatedly because this migration plays a dominant role in tackling the multi-dimension functions, whether they are unimodal functions or multimodal ones. In some high dimension functions, the performance of this single migration is better than COA, MA, and OOA that does not implement this search dedicatedly. In MA, the reference is the only better member [27]. In OOA, the best osprey is mixed with the better ospreys [28]. In COA, only half of the coatis attack the iguana on the tree (the best member) [26]. As the migration toward the best walrus is the most centralized migration among all migrations in MWA, it can be said that centralized migration is crucial to solve the high-dimension functions.

On the other hand, centralized migration should be balanced with more diverse migration as implemented in the four other migrations. The necessity of diverse migration becomes more important in handling fixed-dimension multimodal functions. Although the dimension of these functions is low, the more irregular pattern in these functions creates ambiguity that makes centralized migration or approach falls to the local optimal. On the other hand, a more diverse approach is needed to create many alternatives. This diverse direction can be held by does not depend only on one member. Interaction with other members is also important. On the other hand, tracing solutions within the space as performed in the fifth migration is important too.

The result in Table 7 which is then combined with the result in Tables 3 to 5 highlights the necessity of a multiple search approach. Although every single search has its strength, these searches should be combined into a single package so that this metaheuristic can compete with other metaheuristics that conduct multiple search approaches.

The computational complexity of MWA can be investigated based on the loop implemented in it. Due to this context, its complexity can be split into two parts: during the initialization phase and the iteration phase. In both phases, the numbers of loops in them are different.

Table 7. Individual search assessment result

Function	Average Fitness Score				
	1st Search	2nd Search	3rd Search	4th Search	5th Search
1	**2.1085**	1.5873×10^{1}	1.5771×10^{2}	9.9251	8.9277×10^{4}
2	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
3	**9.7908×10^{2}**	8.2158×10^{3}	2.0567×10^{4}	3.3178×10^{3}	1.6667×10^{5}
4	**1.8097**	2.1759×10^{1}	1.6542×10^{1}	2.1295	9.1056×10^{1}
5	**6.7191×10^{1}**	2.3863×10^{2}	1.2865×10^{4}	1.4274×10^{2}	3.2315×10^{8}
6	**9.9654**	2.0198×10^{1}	1.5914×10^{2}	1.6633×10^{1}	8.6577×10^{4}
7	0.0530	**0.0512**	0.1190	0.0528	2.0899×10^{2}
8	-2.2040×10^{3}	-2.3228×10^{3}	-2.4786×10^{3}	-2.1410×10^{3}	**-3.4471×10^{3}**
9	**8.9516**	6.1510×10^{1}	1.5204×10^{2}	4.3131×10^{1}	6.1072×10^{2}
10	**0.4037**	1.3307	4.0904	1.1365	2.0720×10^{1}
11	**0.4867**	0.9058	2.6368	0.7924	8.2681×10^{2}
12	**1.0820**	1.3118	3.7540	1.2765	8.3343×10^{8}
13	**3.6600**	4.4784	1.6104×10^{1}	4.2471	1.4687×10^{9}
14	**1.1523×10^{1}**	1.7785×10^{1}	1.2030×10^{1}	1.1752×10^{1}	3.2664×10^{1}
15	0.0256	**0.0182**	0.0196	0.0197	0.0334
16	-0.8879	-0.9429	**-0.9527**	-0.8412	-0.7565
17	3.5306	4.5791	**1.9067**	6.0537	0.7456
18	4.6218×10^{1}	6.8383×10^{1}	1.6242×10^{1}	5.9449×10^{1}	**1.4048×10^{1}**
19	**-0.0495**	**-0.0495**	**-0.0495**	**-0.0495**	-0.0267
20	-1.9139	-1.9973	-2.1647	-2.1562	**-2.4807**
21	-1.4072	-1.5372	-1.2944	**-1.9761**	-1.5865
22	-1.5736	-1.5645	-1.5953	**-1.9178**	-1.5532
23	-1.6155	-1.7925	-1.8363	**-1.9152**	-1.8060

The initialization phase contains a nested loop consisting of two loops. The outer loop is the loop for whole walruses. The inner loop is the loop for the whole dimensions. Based on this explanation, the complexity during the initialization phase can be presented as $O(n(W).n(D))$.

On the other hand, the iteration phase contains a nested loop consisting of three loops. The outer loop is the loop from the first iteration to the maximum iteration. The intermediate loop is the loop for whole walruses. Then, the inner loop is the loop for the whole dimension. Meanwhile, there is a loop needed to trace the better walruses for every walrus. Then, there are five migrations performed by every walrus. Based on this explanation, the complexity during the iteration phase can be presented as $O(t_m.n(W).(n(W)+5n(D)))$.

6. Conclusion

In this paper, a new metaheuristic called migrating walrus algorithm (MWA) is introduced. MWA is a product of hybridization between two new metaheuristics: migration algorithm (MA) and walrus optimization algorithm (WaOA). As a hybrid metaheuristic, it consists of five migration processes where four migrations are directed migrations and one local migration. Through assessment, MWA performs superiorly compared to two five new metaheuristics as its confronters. MWA is better than ALO, COA, MA, OOA, and WaOA in 19, 19, 19, 17, and 17 functions consecutively. Meanwhile, through individual migration assessment, the migration toward the best walrus becomes the most contributing migration while other migrations are also important. Meanwhile, although local migration becomes the least significant migration, its contribution is critical in three multimodal functions. This result also strengthens the necessity of the multiple searches strategy rather than a single strategy.

In the future, the upcoming studies can be carried out in two tracks. The first track is constructing MWA, WaOA, or MA with other existing metaheuristics. The second track is implementing this MWA to solve various practical problems.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, Kusuma; methodology, Kusuma; software, Kusuma; formal analysis, Kusuma and Prasasti; investigation, Kusuma and

Prasasti; data curation, Kusuma; writing-original paper draft, Kusuma; writing-review and editing: Prasasti; supervision: Prasasti; funding acquisition, Kusuma.

Acknowledgments

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

References

[1] V. K. Mahadevachar and N. T. Hosur, “Metaheuristic Based Energy Efficient Routing Protocol in MANET Using Battle Royale Optimization”, *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 4, pp. 1-11, 2023, doi: 10.22266/ijies2023.0831.01.

[2] A. Y. Mahdi and S. S. Yuhaniz, “An Intelligent Feature Selection Approach Based on a Novel Improve Binary Sparrow Search Algorithm for COVID-19 Classification”, *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 4, pp. 39-59, 2023, doi: 10.22266/ijies2023.0831.04.

[3] F. A. Dolaimy, R. R. Ali, N. Nabeel, W. S. A. Dayyeni, F. H. Abbas, H. M. Hariz, S. A. Mostafa, and M. A. Jubair, “Hybrid Optimization with Enhanced QoS-based Path Selection in VANETs”, *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 4, pp. 69-80, 2023, doi: 10.22266/ijies2023.0831.06.

[4] Y. Bektas, H. Karaca, T. A. Taha, and H. I. Zaynal, “Red Deer Algorithm-based Selective Harmonic Elimination Technique for Multilevel Inverters”, *Bulletin of Electrical Engineering and Informatics*, Vol. 12, No. 5, pp. 2643-2650, 2023.

[5] 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.

[6] M. S. Braik, "Chameleon Swarm Algorithm: A Bio-inspired Optimizer for Solving Engineering Design Problems", *Expert Systems with Applications*, Vol. 174, No. 114685, pp. 1-25, 2021.

[7] M. A. A. 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.

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

[9] D. Polap and M. Wozniak, “Red Fox Optimization Algorithm”, *Expert Systems with Applications*, Vol. 166, No. 114107, pp. 1-21, 2021.

[10] H. Jia, X. Peng, and C. Lang, "Remora Optimization Algorithm", *Expert Systems with Applications*, Vol. 185, No. 115665, pp. 1-25, 2021.

[11] F. A. Zeidabadi, M. Dehghani, and O. P. Malik, “RSLBO: Random Selected Leader Based Optimizer”, *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 5, pp. 529–538, 2021, doi: 10.22266/ijies2021.1031.46.

[12] S. A. Doumari, H. Givi, M. Dehghani, and O. P. Malik, “Ring Toss Game-Based Optimization Algorithm for Solving Various Optimization Problems”, *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 3, pp. 545–554, 2021, doi: 10.22266/ijies2021.0630.46.

[13] M. Dehghani, S. Hubalovsky, and P. Trojovsky, “Northern Goshawk Optimization: A New Swarm-Based Algorithm for Solving Optimization Problems”, *IEEE Access*, Vol. 9, pp. 162059–162080, 2021.

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

[15] 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.

[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] P. Trojovský and M. Dehghani, “A New Optimization Algorithm based on Mimicking the Voting Process for Leader Selection”, *PeerJ Computer Science*, Vol. 8, No. e976, pp. 1-40, 2022.

[18] 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.

[19] M. Braik, A. Hammouri, J. Atwan, M. A. A. Betar, and M. A. Awadallah, "White Shark Optimizer: A Novel Bio-Inspired Meta-Heuristic Algorithm for Global Optimization Problems", *Knowledge-Based Systems*, Vol. 243, No. 108457, pp. 1-29, 2022.

[20] 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.

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

[22] L. Abualigah, M. A. Elaziz, P. Sumari, Z. W. Geem, and A. H. Gandomi, "Reptile Search Algorithm (RSA): A Nature-Inspired Meta-Heuristic Optimizer", *Expert Systems with Applications*, Vol. 191, No. 116158, pp. 1-33, 2022.

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

[24] P. Trojovsky, M. Dehghani, and P. Hanus, "Siberian Tiger Optimization: A New Bio-Inspired Metaheuristic Algorithm for Solving Engineering Optimization Problems", *IEEE Access*, Vol. 10, pp. 132396-132431, 2022.

[25] N. Chopra and M. M. Ansari, "Golden Jackal Optimization: A Novel Nature-Inspired Optimizer for Engineering Applications", *Expert Systems with Applications*, Vol. 198, No. 116924, pp. 1-15, 2022.

[26] 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, No. 110011, pp. 1-43, 2023.

[27] 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.

[28] 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, No. 1126450, pp. 1-43, 2023.

[29] P. Trojovsky and M. Dehghani, "A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems based on Walruses Behavior", *Scientific Reports*, Vol. 13, No. 8775, pp. 1-32, 2023.

[30] P. D. Kusuma and M. Kallista, "Quad Tournament Optimizer: A Novel Metaheuristic Based on Tournament Among Four Strategies", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 2, pp. 268-278, 2023, doi: 10.22266/ijies2023.0430.22.

[31] 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.

[32] 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.

[33] M. Dehghani, P. Trojovsky, and O. P. Malik, "Green Anaconda Optimization: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems", *Biomimetics*, Vol. 8, No. 121, pp. 1-60, 2023.

[34] P. D. Kusuma and F. C. Hasibuan, "Swarm Magnetic Optimizer: A New Optimizer that Adopts Magnetic Behaviour", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 4, pp. 264-275, 2023, doi: 10.22266/ijies2023.0831.22.

[35] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine Predators Algorithm: A Nature-inspired Metaheuristic", *Expert System with Applications*, Vol. 152, No. 113377, 2020.