Received: June 13, 2023. Revised: July 4, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Walk-Spread Algorithm: A Fast and Superior Stochastic Optimization

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

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

Abstract: This work offers a new stochastic optimization i.e., a metaheuristic algorithm combining both direction-based search and neighbourhood search called as walk-spread algorithm (WSA). These two types of searches become the inspiration for its name where the term walk represents the direction-based search while the term spread represents the neighbourhood search. There are two direction-based searches performed in every iteration where each search produces a single child. Meanwhile, there are two neighbourhood searches performed in every iteration where each search produces several children. The global best unit becomes the first reference while two shuffled units become the second reference in performing the direction-based search. Meanwhile, the local search space of the first neighbourhood search is wide while the second one is narrow. The 23 classic functions are chosen as the assessment of WSA where WSA is confronted with the five latest metaheuristics: mixed leader-based optimization (MLBO), golden search optimization (GSO), pelican optimization algorithm (POA), zebra optimization algorithm (ZOA), and attack-leave optimization (ALO). The assessment result shows that the offered WSA achieves the acceptable result so fast. Moreover, WSA is also superior to these five confronters by outperforming MLBO, GSO, POA, ZOA, and ALO in 23, 23, 22, 21, and 21 functions respectively.

Keywords: Metaheuristic, Optimization, Coati optimization algorithm, Stochastic.

1. Introduction

Optimization can be defined as an effort to find the best solution among a certain number of available solutions to related problems. Optimization has become a popular study for many decades due to its highly related nature to human activities whether these problems are personal, institutional, or state-level problems. In general, optimization is constructed by three aspects: objective, decision variables, and constraints [1].

In the context of optimization, metaheuristic algorithms have become the popular method. Metaheuristics has been used extensively in many optimization studies in many fields. In the energy system, metaheuristics have been implemented such as for designing the PID-controller in the brushless direct current motor control system [2], designing a power system stabilizer for the multi-machine power system [3], optimizing the economic power dispatch in the microgrid power system [4]. In the telecommunication system, metaheuristics have been used to improve the efficiency of MIMO based 5G networks [5], enhancing the spectral efficiency in massive MIMO systems [6], improving intrusion detection in the cloud and IoT networks [7], and so on. In biomedical studies, metaheuristics have been implemented, for example, to optimize the number of layers in the convolutional neural network for EEG signal to detect epilepsy [8], detect epileptic seizure [9], improve the accuracy of breast cancer detection [10], and so on.

In recent years, there are a lot of new metaheuristics have been introduced. The swarm intelligence becomes the favorite baseline for developing these metaheuristics. Many of these metaheuristics were claimed as nature inspired metaheuristics where the animal behavior during foraging or mating becomes the most popular inspiration. The examples of these animal-inspired metaheuristics are cat and mouse-based optimizer (CMBO) [11], clouded leopard optimization (CLO) [12], chameleon swam algorithm (CSA) [13], cheetah

Received: June 13, 2023. Revised: July 4, 2023.

optimizer (CO) [14], coati optimization algorithm (COA) [15], zebra optimization algorithm (ZOA) [16], fennec fox optimization (FFA) [17], golden jackal optimization (GJO) [18], Komodo mlipir algorithm (KMA) [19], snow leopard optimization algorithm (SLOA) [20], northern goshawk optimization (NGO) [21], osprey optimization algorithm (OOA) [22], pelican optimization algorithm (POA) [23], red fox optimization algorithm (RFO) [24], reptile search algorithm (RSA) [25], squirrel search optimizer (SSO) [26], Tasmanian devil optimization (TDO) [27], and so on.

Several metaheuristics do not use metaphors or are called metaphor-free metaheuristics. Several metaheuristics use the term leader representing their main reference, especially in their direction-based search. Examples of them are mixed leader-based optimizers (MLBO) [28], multi leader optimizer (MLO) [29], three influential member-based optimizers (TIMBO) [1], and hybrid leader-based optimizer (HLBO) [30]. On the other hand, some metaheuristics pointed their core strategy for their name, such as total interaction algorithm (TIA) [31], attack-leave optimizer (ALO) [32], quad tournament optimizer (QTO) [33], average and subtraction-based optimizer (ASBO) [34], golden search optimizer (GSO) [35], and so on.

A new metaheuristic can also be constructed by modifying the existing metaheuristic or hybridizing some metaheuristics. The distributed bi-behaviors crow search algorithm (DB-CSA) [36] was developed by enriching the original form of CSA with two Gaussian beta functions. The modified honey badger algorithm (MHBA) is developed from the original form of the honey badger algorithm by embedding the opposition-based learning strategy [37]. The extended stochastic coati optimizer (ESCO) was developed from the original form of COA by expanding the searching strategy and the reference used in its direction-based search [38].

There are several existing problems and challenges despite the massive development of metaheuristics. The first problem is that each metaheuristic still has weakness although it has offered significant improvement. This circumstance is related to the no-free-lunch theory that states that although a metaheuristic is powerful in solving some problems, its performance is mediocre or poor in solving other problems [33]. The second problem is many metaheuristics need extensive iteration to achieve its acceptable performance. Many metaheuristics were implemented in the high maximum iteration to solve their use cases in their first introduction. For example, in its first introduction, CSA is tested with a maximum iteration set to 1,000 and the swarm size is set to 30 [13]. Meanwhile, the swarm size is set to 30 while the maximum iteration is set to 200 in the first introduction of GJO [18]. It means that there is a challenge to develop a metaheuristic that is superior in the low maximum iteration circumstance. Meanwhile, various stochastic methods are available but have not been explored yet. Based on this circumstance, the third challenge is the opportunity to explore stochastic methods to develop a more powerful metaheuristic. The fourth problem is that many metaheuristics were developed by using metaphors as novelty [39]. This approach is often criticized as an effort to hide the mere distinct novelty of the proposed metaheuristics [39].

Based on these problems or challenges, the objective of this work is to offer a new metaheuristic that is more powerful than the existing ones. This work is also the continuation of the authors' effort on continuing the research in artificial intelligence, especially in the development of metaheuristics.

The novelty of this work is mainly in offering a new swarm-based metaheuristic that is free from metaphors and provides fast and superior performance by implementing multiple searches. Meanwhile, the contributions of this work are as follows.

- WSA is designed as a fast and superior metaheuristic to tackle many kinds of optimization problems.
- The strategy designed for WSA is then formalized through algorithm and mathematical model.
- The performance of WSA is assessed through simulation using 23 classic functions as theoretical problems.
- The performance of WSA is confronted with the five latest metaheuristics: MLBO, GSO, POA, ZOA, and ALO.
- The hyper-parameter test is performed to evaluate the significance of the adjusted parameters to the performance improvement.

The continuation of this paper is arranged as follows. Section two presents the literature review regarding some metaheuristics proposed in recent years. Then, the description and formalization of the offered WSA are described in section three. The performance assessment of WSA and its related results is presented in section four. The discussion regarding the findings, complexity, and limitations is explained in section five. Finally, the conclusion and the prospects regarding future studies based on this work are provided in section six.

2. Related works

Metaheuristics has two perspectives. Based on the computer science perspective, metaheuristics is a searching method. Meanwhile, based on the operation research perspective, a metaheuristic is an optimization method. As a searching method, metaheuristics is performed by iterating its strategy until the stopping criteria are reached. There are two possible stopping criteria. First, the goal or destination has been achieved. Second, the maximum iteration has been reached. Meanwhile, as an optimization method, metaheuristics are performed to achieve the best solution among a set of possible solutions [30]. This quality of the solution is measured based on the given objective function. The set of possible solutions can be finite like in combinatorial problems or integer-based numerical problems or infinite like in floating-point numbers based on numerical problems.

Metaheuristic is constructed based on the stochastic approach. Based on this approach, metaheuristics does not guarantee to find the global optimal solution but only the quasi-optimal one [1]. The advantage of this approach is metaheuristics is feasible enough to be implemented to solve optimization problems with limited computation resources [32]. It is different from many greedy-based approaches that trace all possible solutions, so it needs excessive computation resources. Moreover, metaheuristics is known for its flexibility because it can be implemented to solve various problems with less customization [18]. This advantage comes from its nature that abstracts the problem formulation and complexity and focuses on three aspects: objective function, constraint (boundary), and decision variables [32].

In general, metaheuristics performs two strategies: exploration and exploitation. Exploration is also known as diversification while exploitation is also known as intensification. Exploration is the effort to improve the solution by searching for new solutions within the search space [13]. Exploration is also designed to overcome the local optimal circumstance. Exploitation is the effort to improve the solution within the narrow search space [13]. Exploitation is important in the high precision problem to avoid the solution being thrown away from the global optimal area once it is detected.

Separation of exploration and exploitation becomes ambiguous due to the massive development of swarm-based metaheuristics. The swarm-based metaheuristic lays on the direction-based search as their main strategy. The step size of the unit depends on the distance between the unit and its reference. The farther this distance, the probability of walking effort becomes higher. Meanwhile, the iteration-controlled neighborhood search implemented in many swarm-based metaheuristics strengthens this circumstance. In the early iteration, the local search space is wide, and this space is reduced during the iteration. This strategy represents the shift between exploration to exploitation. The variety among swarm-based metaheuristics can be traced by observing the number of strategies (searches) implemented in them, the reference used in the direction-based search, the mechanism in the neighborhood search if any, and the acceptance strategy due to improvement or stagnation. The review of several latest metaheuristics is presented in Table 1. The summarized mechanic of the offered metaheuristic is presented in the last row of Table 1.

The review in Table 1 depicts the variety of strategies or combinations of strategies performed in every metaheuristic. Many metaheuristics perform multiple searches within several phases rather than a single-phase single-search strategy. The maximum number of phases is three. Meanwhile, neighborhood search becomes the secondary complement of the direction-based search. Moreover, some metaheuristics perform only a single neighborhood search while others do not perform a neighborhood search.

Based on this review, there is still an available opportunity to propose a new swarm-based metaheuristic. The position of this work is proposing a new metaheuristic with multiple searches performed in more than three (four) phases. This offered metaheuristic performs two neighborhood searches rather than only one or no neighborhood search.

3. Model

WSA is designed to become a superior metaheuristic by achieving acceptable solution fast. This motivation is then transformed into two core strategies. The first strategy is the direction-based search called "walk". The second strategy is the neighborhood search called "spread". This multiple search strategy becomes common in the latest metaheuristics because of its advantage in taking the strength of each implemented strategy and covering the weakness of each implemented strategy. In WSA, the walk is performed first before spread. Like many latest metaheuristics too, WSA also implements a strict acceptance procedure where a new solution is accepted to replace the older one only if the improvement takes place.

Table 1. Comparison among shortcoming metaheuristics that mimic animal behaviour.

No	Metaheuristic	Detailed Strategy
1	MLBO [28]	MLBO contains a single phase. It performs one direction-based search. The reference is the mixture between the best unit and a shuffled unit within the space. This dominance of this mixture is controlled by the iteration. The direction depends on the reference's quality. The strict acceptance is implemented.
2	GSO [35]	GSO contains a single phase. It performs one direction-based search. The reference is the global best unit and the local best unit. In each iteration, the worst unit is replaced by a shuffled one. The strict acceptance is not implemented.
3	POA [23]	POA contains two phases. In the first phase, the direction-based search is performed. The reference is a shuffled unit within the space. The direction depends on the reference’s quality. In the second phase, the iteration-controlled local search is performed. The local space is narrow. The strict acceptance is implemented.
4	ZOA [16]	ZOA contains two phases. In the first phase, the direction-based search toward the global best unit is performed. In the second phase, there are two possible searches. The first one is the direction-based search toward a shuffled unit among the swarm. The second one is the iteration-controlled neighbourhood search with a narrow space in the beginning. The decision between the first and the second searches is determined uniformly. The strict acceptance is implemented.
5	ALO [32]	ALO contains three phases. The first two phases are mandatory. The third phase is optional. In the first phase, the direction-based search is performed either by the walk of the corresponding unit toward the best unit or by the best unit avoiding the corresponding unit. In the second phase, the direction-based search is performed either by the walk of the corresponding unit avoiding the target or by the target avoiding the corresponding unit. The target can be the middle between the best unit and a shuffled unit in the swarm or the middle between two shuffled units in the swarm. In the third phase, a full random search is performed. The strict acceptance is implemented.
6	FFA [17]	FFA contains two phases. In the first phase, the iteration-controlled neighbourhood search with a narrow space, in the beginning, is performed. In the second phase, the direction-based search is performed where the reference is a shuffled unit within the swarm. The direction is determined by the relative quality of the reference. The strict acceptance is implemented.
7	ESCO [38]	ESCO contains three phases. In the first phase, the direction-based search toward the best unit or relative to a shuffled unit within the space is performed. In the second phase, the direction-based search of the corresponding unit relative to the reference or direction-based search of the reference relative to the corresponding unit is performed. The reference is a shuffled unit within the swarm. In the third phase, the iteration-controlled neighbourhood search with wide space, in the beginning, is performed. The strict acceptance is implemented.
8	CLO [12]	CLO contains two phases. In the first phase, the direction-based search is relative to a shuffled unit within the swarm where the direction is determined based on the relative quality of the reference. In the second phase, the iteration-controlled neighbourhood search is performed where the space declines logarithmically.
9	TIMBO [1]	TIMBO contains three phases. In the first phase, the direction-based search toward the best unit is performed. In the second phase, the direction-based search to avoid the worst unit is performed. In the third phase, the direction-based search relative to the mean unit is performed. The strict acceptance is implemented.
10	OOA [22]	OOA contains two phases. First, the direction-based search toward a shuffled unit from a set of better units within the swarm is performed. Second, the iteration-controlled neighbourhood search with wide space, in the beginning, is performed. The space declines logarithmically. The strict acceptance is implemented.
12	this work	WSA contains four phases. First, the direction-based search toward the best unit is performed. Second, the direction-based search toward the mean of two shuffled units is performed. Third, the iteration-controlled neighbourhood search with wide space, in the beginning, is performed. Fourth, the iteration-controlled neighbourhood search with a narrow space in the beginning is performed.

WSA is developed based on swarm intelligence. It means that WSA contains a certain number of units that represent the solutions. Each unit acts autonomously to find an acceptable solution.

The walk strategy consists of two walks. The first walk is walking toward the global best unit. The global best unit is a unit that stores the best solution among all units so far during the iteration. The second

Received: June 13, 2023. Revised: July 4, 2023.

algorithm 1: walk-spread algorithm	
1	**output:** u_b
2	**begin**
3	**for all** u **in** U
4	generate initial u using Eq. (1) and Eq. (2)
5	update u_b using Eq. (3)
6	**end for**
7	**for** t=1 to t_m
8	**for all** u **in** U
9	first walk using Eq. (4)
10	update u using Eq. (5)
11	update u_b using Eq. (3)
12	second walk using Eq. (6) and Eq. (8)
13	update u using Eq. (5)
14	update u_b using Eq. (3)
15	first spread using Eq. (9)
16	find best child using Eq. (10)
17	update u using Eq. (5)
18	update u_b using Eq. (3)
19	perform the second spread using Eq. (11)
20	find the best child using Eq. (10)
21	update u using Eq. (5)
22	update u_b using Eq. (3)
23	**end for**
24	**end for**
25	**end**

walk is walking toward the target unit which is in the middle between two units shuffled from the set of units. It is different from many metaheuristics that the walking direction relative to the shuffled unit is either toward or away depending on the quality comparison between the target and the corresponding unit. Each walk generates one child only.

The first walk represents the effort to improve the quality of the corresponding unit. The assumption is that moving toward the global best unit may take the lead toward a better solution. This first walk is designed absolutely to tackle the unimodal problem because a unimodal problem has only a single optimal solution. It means that the probability of improvement is higher by moving toward a better solution.

The second walk represents the effort to diversify the search so that the second walk can be seen as diversification. The two shuffled units can be better or worse than the corresponding unit. The location of the target, which is in the middle of these two units, can be near or far from the corresponding unit. This diversification effort is different from the diversification effort performed by the spread searches which is presented later. This strategy makes the second walk still can trace inside the search space without being limited by the iteration.

The spread strategy contains two spreads. The first spread is spread inside a wider space. On the other hand, the second spread is spread inside a narrower space. The searching radius of the second spread is only one percent of the first spread. Multiple spread children are generated in every spread. Then, the best spread child is chosen to become the child in each spread.

The search space of these two spreads declines as the iteration goes on. This strategy represents the gradual shift from diversification to intensification. But the first spread provides a higher degree of diversification than the second spread due to its longer radius.

This concept is then transformed into a formal procedure. This formal procedure is represented by the algorithm of WSA which is presented in pseudocode in algorithm 1 and Eq. (1) to Eq. (11) which represents the formalization of the detail process. The annotations of the formal presentation of WSA are presented below.

b_l	lower boundary
b_u	upper boundary
b_w	width of the boundary
c	child
C	set of children
c_s	spread child
C_s	set of spread children
d	dimension
f	objective function
r	uniform random
t	iteration
t_m	maximum iteration
u	unit
U	set of units
u_b	global best unit
u_s	shuffled unit
u_t	targeted unit

WSA is divided into two stages: initialization and iteration. The initialization is presented from line 2 to line 6 in algorithm 1. On the other hand, the iteration is presented from line 7 to line 24. As common in metaheuristics, the initial unit is produced uniformly inside the search space. Meanwhile, the improvement is performed during the iteration stage. In the end, the global best unit becomes the final solution.

$$u_{i,j} = b_{l,j} + r(0,1).\, b_{w,j} \tag{1}$$

$$b_{w,j} = b_{u,j} - b_{l,j} \tag{2}$$

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

 DOI: 10.22266/ijies2023.1031.24

Eq. (1) to Eq. (3) are used in the initialization stage. Eq. (1) states that the initial unit is produced uniformly within the search space. It means that there is equal opportunity in the search space to be chosen as the initial unit. Eq. (2) is used to determine the range or width of the boundary of the certain dimension j. Eq. (3) states the strict acceptance procedure for updating the global best unit.

$$c_j = u_{i,j} + r(0,1).(u_{b,j} - 2u_{i,j}) \quad (4)$$

$$u'_{i,j} = \begin{cases} c_j, f(c_j) < f(u_{i,j}) \\ u_{i,j}, else \end{cases} \quad (5)$$

Eq. (4) shows the use of the global best unit as the reference in the first walk. Eq. (4) also shows the uniform random motion of the corresponding unit toward the global best unit. Meanwhile, Eq. (5) indicates the strict acceptance procedure of updating the corresponding unit, where the child can replace its parent only when the improvement takes place.

$$u_{s1}, u_{s2} = r(U) \quad (6)$$

$$u_{t,j} = \frac{u_{s1,j} + u_{s2,j}}{2} \quad (7)$$

$$c_j = u_{i,j} + r(0,1).(u_{t,j} - 2u_{i,j}) \quad (8)$$

Eq. (6) to Eq. (8) are used in the second walk. Eq. (6) states that two units are uniformly shuffled from the set of units (swarm). It means that every unit has an equal opportunity to be chosen. There is also a probability that both shuffled units are the same. Eq. (7) states that the targeted unit is placed in the middle between the two shuffled units. Eq. (8) states that the child of the second walk is produced based on the motion of the corresponding unit toward the targeted unit.

$$c_{s,k,j} = u_{i,j} + r(-1,1).b_{w,j}.(1 - \frac{t}{t_m}) \quad (9)$$

$$c = c_s \in C_s, \min(f(c_s)) \quad (10)$$

Eq. (9) states the neighborhood search adopted in the first spread. Variable k in Eq. (9) and Eq. (11) represents the index of the child produced in the spread search. It means that there are certain number of children produced in the spread search. Eq. (9) also indicates that the search area is reduced as the iteration goes on. Meanwhile, Eq. (10) shows that the best-spread children will be chosen as the selected child which is then compared with the corresponding unit in the replacement procedure.

$$c_{s,k,j} = u_{i,j} + 0.01.r(-1,1).b_{w,j}.\left(1 - \frac{t}{t_m}\right) \quad (11)$$

Eq. (11) is used for the second spread search. Different from the first spread, Eq. (11) shows that the radius of the search space in the second search is only one percent of the radius of the search space in the first search.

4. Simulation and result

The performance of WSA is then assessed through simulation. In this work, the 23 classic functions are chosen as the problems. These 23 functions can be split into three groups. The first group consists of seven high dimension unimodal functions. As a unimodal function, each function in the first group has only one optimal solution. The second group consists of six high-dimension multimodal functions. As a multimodal function, each function in the second group has more than one optimal solution so that the metaheuristic can be locked in the local optimal solution without being able to escape. The third group consists of ten fixed-dimension multimodal functions. Although in general the dimension of the fixed dimension multimodal functions is low, these functions are still difficult to overcome. A detailed description of these 23 functions can be found in Table 2.

In this assessment, WSA is confronted with the five latest metaheuristics. These five metaheuristics are MLBO, GSO, POA, ZOA, and ALO. In this simulation, the swarm size is set to 5 while the maximum iteration is set to 10. Both ratios in ALO are set to 0.5. The spread children size in WSA is set to 5. This setting makes the effort to find an acceptable solution difficult, especially when the dimension of the functions in the first and second group is high enough. The first simulation result of the first to third groups can be seen in Table 3 to Table 5 consecutively.

Table 3 depicts that the performance of WSA is superior compared to all these five confronting metaheuristics. WSA sits on the first rank in solving all seven high-dimension unimodal functions. Moreover, WSA can find the global optimal solution in solving Sphere and Schwefel 2.22. WSA becomes the only one metaheuristic that is placed on the first rank in solving six functions (Sphere, Schwefel 1.2, Schwefel 2.21, Rosenbrock, Step, and Quartic). Meanwhile, WSA is accompanied by ZOA, and ALO on the first rank in solving Schwefel 2.22.

Table 3 indicates that these six metaheuristics can be split into two groups based on their performance in solving high-dimension unimodal functions. The first

Table 2. A description of the 23 functions including the equation, dimension, space/boundary, and the target.

No	Function	Model	Dim	Space	Target
1	Sphere	$\sum_{i=1}^{d} x_i^2$	40	[-100, 100]	0
2	Schwefel 2.22	$\sum_{i=1}^{d}\lvert x_i\rvert + \prod_{i=1}^{d}\lvert x_i\rvert$	40	[-100, 100]	0
3	Schwefel 1.2	$\sum_{i=1}^{d}\left(\sum_{j=1}^{i} x_j\right)^2$	40	[-100, 100]	0
4	Schwefel 2.21	$\max\{\lvert x_i\rvert, 1 \le i \le d\}$	40	[-100, 100]	0
5	Rosenbrock	$\sum_{i=1}^{d-1}(100(x_{i+1} + x_i^2)^2 + (x_i - 1)^2)$	40	[-30, 30]	0
6	Step	$\sum_{i=1}^{d-1}(x_i + 0.5)^2$	40	[-100, 100]	0
7	Quartic	$\sum_{i=1}^{d} i\, x_i^4 + random\ [0,1]$	40	[-1.28, 1.28]	0
8	Schwefel	$\sum_{i=1}^{d} -x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	40	[-500, 500]	-16,759
9	Ratsrigin	$10d + \sum_{i=1}^{d}\left(x_i^2 - 10cos(2\pi x_i)\right)$	40	[-5.12, 5.12]	0
10	Ackley	$-20 \cdot exp\left(-0.2 \cdot \sqrt{\frac{1}{d}\sum_{i=1}^{d} {x_i}^2}\right) - exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos 2\pi x_i\right) + 20 + exp(1)$	40	[-32, 32]	0
11	Griewank	$\frac{1}{4000}\sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	40	[-600, 600]	0
12	Penalized	$\frac{\pi}{d}\left\{10\sin(\pi y_1) + \sum_{i=1}^{d-1}\left((y_i - 1)^2(1 + 10\, sin^2(\pi y_{i+1}))\right) + (y_d - 1)^2\right\} + \sum_{i=1}^{d} u(x_i, 10,100,4)$	40	[-50, 50]	0
13	Penalized 2	$0.1\left\{sin^2(3\pi x_1) + \sum_{i=1}^{d-1}\left((x_i - 1)^2(1 + sin^2(3\pi x_{i+1}))\right) + (x_d - 1)^2(1 + sin^2(2\pi x_d))\right\} + \sum_{i=1}^{d} u(x_i, 5,100,4)$	40	[-50, 50]	0
14	Shekel Foxholes	$\left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6}\right)^{-1}$	2	[-65, 65]	1
15	Kowalik	$\sum_{i=1}^{11}\left(a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right)^2$	4	[-5, 5]	0.0003
16	Six Hump Camel	$4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[-5, 5]	-1.0316
17	Branin	$\left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10$	2	[-5, 5]	0.398
18	Goldstein-Price	$\left(1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right).\left(30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right)$	2	[-2, 2]	3
19	Hartman 3	$-\sum_{i=1}^{4}\left(c_i exp\left(-\sum_{j=1}^{d}\left(a_{ij}\left(x_j - p_{ij}\right)^2\right)\right)\right)$	3	[1, 3]	-3.86
20	Hartman 6	$-\sum_{i=1}^{4}\left(c_i exp\left(-\sum_{j=1}^{d}\left(a_{ij}\left(x_j - p_{ij}\right)^2\right)\right)\right)$	6	[0, 1]	-3.32
21	Shekel 5	$-\sum_{i=1}^{5}\left(\sum_{j=1}^{d}\left(x_j - c_{ji}\right)^2 + \beta_i\right)^{-1}$	4	[0, 10]	-10.153
22	Shekel 7	$-\sum_{i=1}^{7}\left(\sum_{j=1}^{d}\left(x_j - c_{ji}\right)^2 + \beta_i\right)^{-1}$	4	[0, 10]	-10.402
23	Shekel 10	$-\sum_{i=1}^{10}\left(\sum_{j=1}^{d}\left(x_j - c_{ji}\right)^2 + \beta_i\right)^{-1}$	4	[0, 10]	-10.536

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

F	Parameter	MLBO [28]	GSO [35]	POA [23]	ZOA [16]	ALO [32]	WSA
1	mean	2.5530×10^{4}	3.8435×10^{4}	6.6357×10^{4}	7.0338	0.9054	0.0000
	std deviation	4.3123×10^{3}	9.6503×10^{3}	8.1432×10^{3}	3.6704	1.9713	0.0000
	mean rank	4	5	6	3	2	1
2	mean	5.9598×10^{39}	1.007×10^{51}	4.3972×10^{52}	0.0000	0.0000	0.0000
	std deviation	2.9197×10^{40}	3.480×10^{51}	2.0114×10^{53}	0.0000	0.0000	0.0000
	mean rank	4	5	6	1	1	1
3	mean	7.8350×10^{4}	9.9661×10^{4}	2.0073×10^{5}	3.0828×10^{3}	2.8545×10^{2}	1.1704
	std deviation	5.1611×10^{4}	4.6754×10^{4}	4.3396×10^{4}	1.8938×10^{3}	6.9426×10^{2}	2.1837
	mean rank	4	5	6	3	2	1
4	mean	5.4222×10^{1}	6.2524×10^{1}	8.1401×10^{1}	3.0322	0.7233	0.0059
	std deviation	7.6946	8.3612	8.0427	1.3067	1.2018	0.0045
	mean rank	4	5	6	3	2	1
5	mean	3.8558×10^{7}	6.8880×10^{7}	2.0880×10^{8}	1.6702×10^{2}	5.4588×10^{1}	3.8953×10^{1}
	std deviation	1.4816×10^{7}	2.7883×10^{7}	5.5111×10^{7}	1.0706×10^{2}	2.5250×10^{1}	0.0219
	mean rank	4	5	6	3	2	1
6	mean	2.6051×10^{4}	3.4160×10^{4}	6.2066×10^{4}	1.4566×10^{1}	1.0169×10^{1}	7.1884
	std deviation	6.5722×10^{3}	6.5626×10^{3}	1.1318×10^{4}	3.8522	1.5556	0.7056
	mean rank	4	5	6	3	2	1
7	mean	2.1142×10^{1}	5.2466×10^{1}	1.1360×10^{2}	0.0521	0.0773	0.0139
	std deviation	9.1139	3.4230×10^{1}	3.2309×10^{1}	0.0383	0.0776	0.0108
	mean rank	4	5	6	2	3	1

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

F	Parameter	MLBO [28]	GSO [35]	POA [23]	ZOA [16]	ALO [32]	WSA
8	mean	-2.5978×10^{3}	-2.3994×10^{3}	-2.7834×10^{3}	-2.2273×10^{3}	-3.0893×10^{3}	-3.8594×10^{3}
	std deviation	4.0180×10^{2}	5.7494×10^{2}	4.6250×10^{2}	4.1126×10^{2}	5.2618×10^{2}	5.4462×10^{2}
	mean rank	4	5	3	6	2	1
9	mean	3.9262×10^{2}	3.9025×10^{2}	5.4091×10^{2}	2.9173×10^{1}	0.3315	0.0000
	std deviation	3.2593×10^{1}	4.3429×10^{1}	4.1477×10^{1}	2.9996×10^{1}	0.7854	0.0000
	mean rank	5	4	6	3	2	1
10	mean	1.7737×10^{1}	1.8747×10^{1}	2.0059×10^{1}	1.0932	0.1360	0.0004
	std deviation	0.6252	0.6492	0.1974	0.4800	0.1117	0.0003
	mean rank	4	5	6	3	2	1
11	mean	2.2086×10^{2}	3.2956×10^{2}	6.0419×10^{2}	0.8420	0.1910	0.0017
	std deviation	4.8990×10^{1}	6.9290×10^{1}	1.2595×10^{2}	0.3135	0.3327	0.0791
	mean rank	4	5	6	3	2	1
12	mean	2.9860×10^{7}	6.9663×10^{7}	3.7985×10^{8}	1.1344	1.3619	0.5522
	std deviation	1.6157×10^{7}	4.1494×10^{7}	1.4446×10^{8}	0.2363	0.2235	0.1141
	mean rank	4	5	6	2	3	1
13	mean	1.0711×10^{8}	2.7642×10^{8}	9.3875×10^{8}	3.9080	3.4569	3.1317
	std deviation	6.6792×10^{7}	1.5862×10^{8}	2.8594×10^{8}	0.3538	0.3375	0.1738
	mean rank	4	5	6	3	2	1

group consists of MLBO, GSO, and POA. On the other hand, the second group consists of ZOA, ALO, and WSA. The performance of metaheuristics in the second group is far better than in the first group. Moreover, the performance gap between WSA and another two metaheuristics in the second group (ZOA and ALO) is also wide in solving Schwefel 1.2.

Table 4 indicates that WSA is completely superior to the other five confronters in solving high-dimension multimodal functions. WSA is on the first rank in solving all six functions (Schwefel, Rastrigin, Ackley, Griewank, Penalized, and Penalized 2). There is not any other metaheuristic that sits on the first rank together with WSA in solving high dimension multimodal functions. Moreover, WSA can find the global optimal solution in solving Rastrigin.

The performance gap among metaheuristics in solving these functions also varies depending on the functions. In general, the performance gap between the best performer (WSA) and the worst performer (MLBO) in solving Schwefel is narrow. Different circumstance occurs in two functions (penalized and penalized 2). In these two functions, the performance

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

F	Parameter	MLBO [28]	GSO [35]	POA [23]	ZOA [16]	ALO [32]	WSA
14	mean	4.8005×10^1	2.2386×10^1	2.0911×10^1	1.0864×10^1	8.9655	5.3808
	std deviation	7.9242×10^1	3.9088×10^1	1.7540×10^1	7.3965	5.2298	2.5740
	mean rank	6	5	4	3	2	1
15	mean	0.0899	0.1812	0.0317	0.0196	0.0129	0.0062
	std deviation	0.1505	0.3072	0.0306	0.0302	0.0151	0.0009
	mean rank	5	6	4	3	2	1
16	mean	-0.7183	-0.3793	-0.6421	-0.8747	-0.9657	-1.0232
	std deviation	0.6727	1.6787	0.3858	0.2510	0.1340	0.0207
	mean rank	4	6	5	3	2	1
17	mean	1.4209	3.6310	1.4279	4.2558	0.8361	0.4429
	std deviation	1.4101	9.0083	1.7952	6.5770	0.5410	0.0600
	mean rank	3	5	4	6	2	1
18	mean	6.0902×10^1	3.2553×10^1	2.6332×10^1	4.2916×10^1	1.1362×10^1	3.5550
	std deviation	1.2649×10^2	3.3695×10^1	2.3618×10^1	4.9601×10^1	1.0468×10^1	1.0054
	mean rank	6	4	3	5	2	1
19	mean	-0.0245	-0.0028	-0.0495	-0.0495	-0.0495	-0.0495
	std deviation	0.0201	0.0050	0.0000	0.0000	0.0000	0.0000
	mean rank	5	6	1	1	1	1
20	mean	-2.2604	-1.7900	-2.2634	-1.9980	-2.1933	-2.8831
	std deviation	0.5445	0.5805	0.5852	0.5419	0.4571	0.1267
	mean rank	3	6	2	5	4	1
21	mean	-1.4763	-1.1577	-0.8632	-1.4936	-1.4590	-2.3629
	std deviation	1.3061	0.9596	0.6876	0.7550	1.0957	1.1175
	mean rank	3	5	6	2	4	1
22	mean	-1.5405	-1.9653	-1.2644	-1.3906	-1.4197	-2.3835
	std deviation	0.7052	2.0434	0.9409	0.8406	0.5944	0.6257
	mean rank	3	2	6	5	4	1
23	mean	-1.9938	-1.7958	-1.0988	-1.9236	-1.6338	-2.4865
	std deviation	1.6155	1.6259	0.4287	1.0429	0.6700	0.6488
	mean rank	2	4	6	3	5	1

Table 6. Group-based superiority of WSA.

Group	Number of Functions Where WSA is Better				
	MLBO [28]	GSO [35]	POA [23]	ZOA [16]	ALO [32]
1	7	7	7	6	6
2	6	6	6	6	6
3	10	10	9	9	9
Total	23	23	22	21	21

gap between functions in the first group (MLBO, GSO, and POA) and functions in the second group (ZOA, ALO, and WSA) is wide.

Table 5 indicates the superiority of WSA among its five confronters in solving fixed-dimension functions. WSA is placed on the first rank in solving all ten functions in this third group. WSA becomes the top performer in solving nine fixed-dimension multimodal functions. Meanwhile, WSA is accompanied by three confronters (POA, ZOA, and ALO) in solving Hartman 3.

The performance gap among metaheuristics in solving fixed-dimension multimodal functions is not so wide as in solving high-dimension functions. The wide performance gap between WSA as the best performer and the worst performer is wide enough in solving Kowalik and Goldstein Price. Otherwise, the competition among metaheuristics in solving fixed-dimension multimodal functions is fierce.

The result in Table 3 to Table 5 is then summarized in Table 6. Table 6 depicts the superior result of WSA compared to these five confronters in each group of functions. The superiority data is obtained based on the number of functions where WSA is better than the related metaheuristics.

Table 6 strengthens the fact that WSA is superior to all five confronters. WSA is superior to MLBO and GSO by outperforming these two metaheuristics in all 23 functions. Meanwhile, WSA outperforms POA, ZOA, and ALO in solving 22, 21, and 21 functions consecutively. In the functions where WSA fails to outperform the confronters, the result is drawn.

The next assessment is related to the hyperparameter assessment. Three parameters are assessed: maximum iteration, swarm size, and children size. The maximum iteration is set to 20 and

Table 7. Relation between maximum iteration and the average fitness score.

F.	Average Fitness Score		Significantly Improved?
	t_m=20	t_m=40	
1	0.0000	0.0000	no
2	0.0000	0.0000	no
3	0.0000	0.0000	no
4	0.0000	0.0000	no
5	3.8959×10^1	3.8963×10^1	no
6	6.0853	4.6018	no
7	0.0038	0.0026	no
8	-4.5603×10^3	-6.3488×10^3	no
9	0.0000	0.0000	no
10	0.0000	0.0000	no
11	0.0000	0.0000	no
12	0.2588	0.1185	yes
13	2.4593	2.0874	no
14	3.6189	2.3536	no
15	0.0027	0.0008	yes
16	-1.0316	-1.0316	no
17	0.4021	0.3981	no
18	3.0492	3.0000	no
19	-0.0495	-0.0495	no
20	-3.1005	-3.2257	no
21	-4.3438	-5.8835	no
22	-3.8914	-6.5041	no
23	-4.2232	-5.9412	no

Table 8. Relation between swarm size and the average fitness score.

F.	Average Fitness Score		Significantly Improved?
	$n(U)$=10	$n(U)$=20	
1	0.0000	0.0000	no
2	0.0000	0.0000	no
3	0.4107	0.2181	no
4	0.0054	0.0037	no
5	3.8937×10^1	3.8924×10^1	no
6	6.5346	6.1128	no
7	0.0077	0.0040	no
8	-4.1021×10^3	-4.3012×10^3	no
9	0.0000	0.0000	no
10	0.0006	0.0004	no
11	0.0222	0.0000	yes
12	0.4751	0.3746	no
13	2.9634	2.7203	no
14	2.4414	1.6020	no
15	0.0019	0.0017	no
16	-1.0316	-1.0316	no
17	0.4165	0.4048	no
18	3.1174	3.0149	no
19	-0.0495	-0.0495	no
20	-2.9684	-3.0948	no
21	-3.0133	-3.2069	no
22	-2.8901	-3.8731	no
23	-3.1177	-3.5889	no

Table. 9 Relation between children's size and the average fitness score

F.	Average Fitness Score		Significantly Improved?
	$n(C)$=10	$n(C)$=20	
1	0.0000	0.0000	no
2	0.0000	0.0000	no
3	0.4935	1.2944	yes
4	0.0084	0.0056	no
5	3.8953×10^1	3.8956×10^1	no
6	6.5793	5.7274	no
7	0.0075	0.0072	no
8	-4.2377×10^3	-4.6814×10^3	no
9	0.0000	0.0000	no
10	0.0006	0.0006	no
11	0.0000	0.0025	no
12	0.4238	0.3419	no
13	2.8921	2.5416	no
14	3.6845	2.5974	no
15	0.0041	0.0064	no
16	-1.0312	-1.0316	no
17	0.4131	0.4011	no
18	3.1893	3.0246	no
19	-0.0495	-0.0495	no
20	-3.0123	-3.1224	no
21	-3.0057	-3.7395	no
22	-2.9318	-3.2663	no
23	-3.2995	-3.9951	no

40. The swarm size is set to 10 and 20. The children's size is set to 10 and 20. The result is presented in Table 7 and Table 9.

Table 7 shows that the increase of maximum iteration from 20 to 40 does not improve the final solution significantly in almost all functions. There are only two functions where significant improvement takes place. These functions are Penalized and Kowalik. Both functions are multimodal functions. Meanwhile, there are ten functions where the global optimal solutions are found. These ten functions are Sphere, Schwefel 2.22, Schwefel 1.2, Schwefel 2.21, Rastrigin, Ackley, Griewank, Six Hump Camel, Branin, and Goldstein-Price.

Table 8 indicates that the increase of the swarm size from 10 to 20 does not improve the quality of the final solution significantly in almost all functions too. The significant improvement takes place only in one function which is Griewank. Meanwhile, there are five functions where their global optimal solution is found. These functions are Sphere, Schwefel 2.22, Rastrigin, Griewank, and Six Hump Camel.

Table 9 shows that the increase in children's size from 10 to 20 also does not improve the quality of the final solution significantly in almost all functions. The significant improvement can be found only in one function (Schwefel 1.2). There are only four functions where the global optimal solution is found. These four functions are Sphere, Schwefel 2.22, Rastrigin, and Six Hump Camel.

5. Discussion

The simulation result has shown that the multiple searches performed dedicatedly are important for any metaheuristic to perform well. Specifically, a metaheuristic should perform both direction-based search and neighbourhood search as implemented in WSA. The poor performance achieved by MLBO and GSO is highly related to their limited strategy. MLBO and GSO perform a single direction-based search only without any neighbourhood search. The difference between MLBO and GSO is only their reference. MLBO uses the mixture of the best unit and a shuffled unit where the dominance is controlled by the iteration [28]. Meanwhile, GSO uses the portion of the global best unit and the local best unit as a reference [35].

The absence of the global best unit or best unit in the direction-based search can be seen as the reason for the poor performance of POA [23]. Although POA implements both direction-based search and neighbourhood search, a shuffled unit inside the search space becomes the only reference [23]. Meanwhile, the presence of the best unit as a reference is important to exploit the location near this best unit and improve the solution fast. Even PSO as the early swarm-based metaheuristic uses both the global best unit and local best unit as its reference.

The close performance gap among WSA, ALO, and ZOA proves that multiple searches are critical to produce superior performance. These three metaheuristics perform a direction-based search toward the best unit explicitly as the first search. All these metaheuristics also perform neighbourhood or random searches. The difference is as follows. In ZOA, the probability of performing a neighbourhood search is shared with the direction-based search toward a shuffled unit [16]. In ALO, the random search is optional only if the improvement fails. Meanwhile, the direction-based search toward the shuffled unit is performed as the second search.

The neighbourhood search with narrow space in the beginning also becomes the difference between WSA and the two other metaheuristics (ALO and ZOA). In ZOA, the search space in the neighbourhood search is wide. On the other hand, the search space of ALO in the random search is always wide along the iteration because it traces uniformly inside the problem boundaries [32]. Meanwhile, the narrow search space in the neighbourhood search can be seen as additional intensification as it is also performed in NGO [21], POA [23], and FFA [17].

The algorithm complexity of WSA is different between the initialization stage and the iteration stage. In the initialization stage, the algorithm complexity of WSA can be presented as $O(n(U).d)$. In this stage, there are two loops where the outer loop is regarding the swarm size while the inner loop is regarding the dimension of the problem. On the other hand, the algorithm complexity during the iteration phase can be presented as $O(t_m.n(U).d.(2(1+n(C))))$. In this stage, there are three core loops regarding the maximum iteration, swarm size, and the dimension of the problem. Meanwhile, there are four searches performed by each unit. In these two spread searches, there is a loop regarding the spread of children's size.

Although WSA is proven as a fast and superior metaheuristic confronted to the five other metaheuristics, it is still not perfect. There are functions where WSA still fails to find their quasi-optimal solution in the low swarm size and maximum iteration circumstance. WSA still fails to find the quasi-optimal solution in solving Rosenbrock and Step in the first group although the performance of the other metaheuristics is much worse. WSA also fails to find the quasi-optimal solution in solving Schwefel in the second group. The final solution found by WSA in solving Hartman Hartman 3, Shekel 5, Shekel 7, and Shekel 10 in the third group is also not quasi-optimal. It means that the improvement of the current form of WSA is still open.

Moreover, there is a limitation regarding the use case for assessment in this work. Several other theoretical problems have not been used for the assessment, such as CEC 2015, CEC 2017, and so on. WSA also has not been tested to solve practical optimization problems, either numerical problems or combinatorial ones.

6. Conclusion

This work has offered a new swarm-based metaheuristic called as walk-spread algorithm (WSA). WSA is a metaphor-free metaheuristic as it does not use any metaphors and adopts its core strategy (walk and spread) as its name. It consists of four searches that are performed sequentially. Through assessment, WSA is proven superior compared to the five new metaheuristics as its confronters. WSA outperforms MLBO, GSO, POA, ZOA, and ALO in solving 23, 23, 22, 21, and 21 functions consecutively. Moreover, WSA is proven fast due to its superior achievement taken in the low maximum iteration circumstance. In the higher maximum iteration, WSA can find the global optimal solution of ten functions.

There are several tracks proposed for future work. The modification or hybridization of WSA is still open to improve its performance, especially to solve some functions where the quasi-optimal solution has been achieved in this current work. Moreover, the challenge to implement WSA to solve various practical optimization problems is still widely open.

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

The funding of this work is provided by Telkom University, Indonesia.

References

[1] 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.

[2] H. S. Dakheel, Z. B. Abdullah, and S. W. Shneen, "Advanced Optimal GA-PID Controller for BLDC Motor", *Bulletin of Electrical Engineering and Informatics*, Vol. 12, No. 4, pp. 2077-2086, 2023.

[3] M. Zadehbagheri, T. Sutikno, M. J. Kiani, and M. Yousefi, "Designing a Power System Stabilizer Using a Hybrid Algorithm by Genetics and Bacteria for the Multi-machine Power System", *Bulletin of Electrical Engineering and Informatics*, Vol. 12, No. 3, pp. 1318-1331, 2023.

[4] Y. Zahraoui, I. Alhamrouni, S. Mekhilef, T. Korõtko, A. Jusoh, and T. Sutikno, "A Chameleon Algorithm for Solving Economic Dispatch Problem in Microgrid System", *Bulletin of Electrical Engineering and Informatics*, Vol. 12, No. 4, pp. 1982-1992, 2023.

[5] G. S. Kishore and P. C. Sekhar, "Deep Convolutional Spiking Neural Network Optimized with Coyote Chimp Optimization Algorithm for Imperfect Channel Estimation in MIMO-f-OFDM/FQAM Based 5G Network", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 3, pp. 115-125, 2023, doi: 10.22266/ijies2023.0630.09.

[6] B. E. Soualah, A. Chemsa, and R. Ajgou, "Enhancement of Spectral Efficiency in Massive MIMO System Availing Eagle Crow Optimization-Based Scheduling Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 3, pp. 139-150, 2023, doi: 10.22266/ijies2023.0630.11.

[7] H. K. Bella and V. Sanjeevulu, "Intrusion Detection Using Pareto Optimality Based Grasshopper Optimization Algorithm with Stacked Autoencoder in Cloud and IoT Networks", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 3, pp. 269-279, 2023, doi: 10.22266/ijies2023.0630.21.

[8] D. Sunaryono, J. Siswantoro, A. B. Raharjo, R. Ridho, R. Sarno, S. I. Sabilla, and R. I. Susilo, "Optimized One-Dimension Convolutional Neural Network for Seizure Classification from EEG Signal based on Whale Optimization Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 3, pp. 310-322, 2023, doi: 10.22266/ijies2023.0630.25.

[9] H. D. Praveena, C. Subhas, and K. R. Naidu, "Improved Artificial Bee Colony Based Feature Selection for Epileptic Seizure Detection", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 3, pp. 332-344, 2023, doi: 10.22266/ijies2023.0630.27.

[10] A. Elnawasany, M. A. A. Makhlouf, B. Tawfik, and H. Nassar, "Feature Selection of Unbalanced Breast Cancer Data Using Particle Swarm Optimization", *International Journal of Electrical and Computer Engineering*, Vol. 12, No. 5, pp. 4951-4959, 2022.

[11] M. Dehghani, S. Hubalovsky, and P. Trojovksy. "Cat and Mouse Based Optimizer: A New Nature-Inspired Optimization Algorithm", *Sensors*, Vol. 21, ID. 5214, pp. 1-30, 2021.

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

[13] 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.

[14] M. A. Akbari, M. Zare, R. A. Abarghooee, S. Mirjalili, and M. Deriche, "The Cheetah Optimizer: A Nature-inspired Metaheuristic Algorithm for Large-scale Optimization

Problems", *Scientific Reports*, Vol. 12, ID. 10953, pp. 1-20, 2022.

[15] 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.

[16] 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.

[17] E. Trojovska, M. Dehghani, and P. Trojovsky, "Fennec Fox Optimization: A New Nature-Inspired Optimization Algorithm", *IEEE Access*, Vol. 10, pp. 84417-84443, 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] Suyanto, A. A. Ariyanto, and A. F. Ariyanto, "Komodo Mlipir Algorithm", *Applied Soft Computing*, Vol. 114, ID: 108043, 2022.

[20] P. Coufal, S. Hubalovsky, M. Hubalovska, and Z. Balogh, "Snow Leopard Optimization Algorithm: A New Nature-Based Optimization Algorithm for Solving Optimization Problems", *Mathematics*, Vol. 9, ID. 2832, pp. 1-26, 2021.

[21] M. Dehghani, S. Hubalovsky, and P. Trojovsky, "Northern Goshawk Optimization: A New Swarm-Based Algorithm for Solving Optimization Problems", *IEEE Access*, pp. 162059-162080, 2021.

[22] 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.

[23] 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.

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

[25] L. Abualigah, M. A. Elaziz, P. Sumari, Z. W. Geem, and A. H. Gandomi, "Reptile Search Algorithm (RSA): A Nature-Inspired Meta-Heuristic Optimizer", *Expert Systems with Applications*, Vol. 191, ID. 116158, pp. 1-33, 2022.

[26] M. Suman, V. P. Sakthivel, and P. D. Sathya, "Squirrel Search Optimizer: Nature Inspired Metaheuristic Strategy for Solving Disparate Economic Dispatch Problems", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 5, pp. 111–121, 2020, doi: 10.22266/ijies2020.1031.11.

[27] 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.

[28] F. A. Zeidabadi, S. A. Doumari, M. Dehghani, and O. P. Malik, "MLBO: Mixed Leader Based Optimizer for Solving Optimization Problems", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 4, pp. 472-479, 2021, doi: 10.22266/ijies2021.0831.41.

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

[30] 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.

[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] 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.

[34] 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

Received: June 13, 2023. Revised: July 4, 2023.

Optimization Problems", *PeerJ Computer Science*, Vol. 8, ID: e910, pp. 1-29, 2022.

[35] 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.

[36] A. Aboud, N. Rokbani, B. Neji, Z. A. Barakeh, S. Mirjalili, and A. M. Alimi, "A Distributed Bi-Behaviors Crow Search Algorithm for Dynamic Multi-Objective Optimization and Many-Objective Optimization Problems", *Applied Sciences*, Vol. 12, ID. 9627, pp. 1-42, 2022.

[37] S. A. Yasear and H. M. A. Ghanimi, "A Modified Honey Badger Algorithm for Solving Optimal Power Flow Optimization Problem", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 4, pp. 142-155, 2022, doi: 10.22266/ijies2022.0831.14.

[38] P. D. Kusuma and A. Dinimaharawati, "Extended Stochastic Coati Optimizer", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 3, pp. 482-494, 2023, doi: 10.22266/ijies2023.0630.38.

[39] J. Swan, S. Adriaensen, A. E. I. Brownlee, K. Hammond, C. G. Johnson, A. Kheiri, F. Krawiec, J. J. Merelo, L. L. Minku, E. Ozcan, G. L. Pappa, P. G. Sanchez, K. Sorensen, S. Vob, M. Wagner, and D. R. White, "Metaheuristics in the Large", *European Journal of Operational Research*, Vol. 297, pp. 393-406, 2022.