Received: July 8, 2023. Revised: August 2, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Four Directed Search Algorithm: A New Optimization Method and Its Hyper Strategy Investigation

Purba Daru Kusuma[1]* **Ashri Dinimaharawati[1]**

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

Abstract: This paper promotes a new swarm-based metaheuristic called four directed search algorithm or FDSA. FDSA is designed as a directed search-based metaheuristic without deploying any neighbourhood search. It contains four directed searches that are carried out sequentially. Each search has its reference. These four references are the best member; the resultant of three shuffled members within the swarm; a shuffled member within the swarm; and the resultant of the best member, a shuffled member within the swarm, and the corresponding member. FDSA implements a strict acceptance procedure so that new solution is accepted only if it provides improvement. The investigation is carried out to evaluate the performance of FDSA in solving the 23 functions. FDSA is also confronted with five new metaheuristics: northern goshawk optimization (NGO), average subtraction-based optimization (ASBO), coati optimization algorithm (COA), mixed leader-based optimization (MLBO), and attack leave optimization (ALO). This work also investigates the contribution or dominance of each search in the context of finding the optimal solution. The result shows that FDSA is superior to all these confronters by consecutively outperforming NGO, ASBO, COA, MLBO, and ALO in the 17, 13, 11, 18, and 11 functions. Its superiority is mainly in the high-dimension functions. Through investigation, there is no dominant search among the four directed searches in FDSA. Meanwhile, multiple search strategy is proven to improve performance significantly. On the other hand, the contribution of neighbourhood search is not significant.

Keywords: Metaheuristic, Optimization, Directed search, Neighbourhood search.

1. Introduction

Metaheuristics has become a viral method used in optimization works. It is also has been implemented in a wide range of studies. In computer systems, grey wolf optimization (GWO) has been used in the task scheduling process in cloud computing systems [1]. In telecommunication, a mixture of the coyote and chimp optimization algorithms has been developed to optimize the neural network in the 5G network [2]. The hunter-prey optimization algorithm is implemented in robotics and automation to solve the pathfinding effort [3]. Several implementations of metaheuristics in social media are such as sentiment analysis [4] or developing collaborative teams [5]. In bio-medic engineering, African buffalo optimization has been used to improve heart disease prediction [6]. In production systems, an artificial bee colony (ABC) has been used to optimize the make-span in the flow-shop system [7]. In transportation systems, the classic variable neighborhood search is used to optimize the routing of the electric vehicle where the speed is time-dependent, and the time window is soft [8]. In the energy sector, teaching learning-based optimization (TLBO) is used to optimize the power flow in the complex power system [9].

Nature becomes the most important source of inspiration for the development of metaheuristics. Many new metaheuristics are developed by exploring nature, especially the mechanism of the animals during mating or searching for food. In other words, there are a lot of metaheuristics that use animal behavior as a metaphor. The example is modified honey badger algorithm (MHBA) [10], Komodo mlipir algorithm (KMA) [11], marine predator algorithm (MPA) [12], grey wolf optimization (GWO) [13], coati optimization algorithm (COA)

[14], zebra optimization algorithm (ZOA) [15], pelican optimization algorithm (POA) [16], clouded leopard optimization (CLO) [17], Siberian tiger optimization (STO) [18], golden jackal optimization algorithm (GJO) [19], osprey optimization algorithm (OOA) [20], northern goshawk optimization (NGO) [21], extended stochastic coati optimization (ESCO) [22], guided pelican algorithm (GPA) [23], chameleon swarm algorithm [24], and so on.

Other metaheuristics use other mechanisms as metaphors. Some metaheuristic imitates social or human activities, such as modified social forces algorithm (MSFA) [25], driving training-based optimization (DTBO) [26], chef-based optimization algorithm (CBOA) [27], election-based optimization algorithm (EBOA) [28], and so on.

Meanwhile, some metaheuristics do not use metaphors. Some metaheuristics use their core strategy for their name, such as golden search optimization (GSO) [29], attack-leave optimization (ALO) [30], average subtraction-based optimization (ASBO) [31], and so on. Some others use their reference during their directed search for their name, such as three influential member optimization (TIMBO) [32], multi-leader optimization (MLO) [33], hybrid leader-based optimization (HLBO) [34], mixed leader-based optimization (MLBO) [35], and so on.

Several things could be improved with the massive development of metaheuristics. First, many metaheuristics that use metaphors have been criticized as hiding their mere novelty [36]. These metaheuristics often used the adaptation of the metaphor as a distinct strategy [36]. Meanwhile, by abstracting the metaphor, there is a common approach in designing swarm-based metaheuristics: deploying multiple searches, where the directed search becomes the primary search and neighborhood search, or full random search becomes the secondary one. Some standard references are also used in the directed search, such as the best member, some best members, another member, the mixture of the best member with another member, and so on. On the other hand, some common random searches are also used in the iteration. Some metaheuristics implement full random search, which means the search process is carried out within an entire space. Other metaheuristics conduct neighborhood searches with local space reduction during the iteration.

The second problem concerns the investigation performed in every study promoting new metaheuristics. Performance investigation is the primary investigation due to the effort to prove that the designed metaheuristic is better than the previous ones. In general, several sets of functions were often used as theoretical problems. Meanwhile, some studies used real-world problems from several fields, such as engineering, operation research, and finance. The second investigation is the hyperparameter investigation which is carried out to evaluate which adjusted parameters play an important role in improvement. Unfortunately, the investigation related to the contribution of each search in conducting the optimization process is rare. Ironically, this investigation is essential for the future development of metaheuristics.

This second problem is highly related to the no-free-lunch theory. Due to the limitation of each metaheuristic on solving all problems [36], investigating the effectiveness of specific searches is crucial besides investigating the metaheuristic as a whole packet. This investigation is essential to determine whether a strategy is still needed in future development or if it can be disposed of with another search.

Based on this problem, this work aims to design a new swarm-based metaheuristic called FDSA that accommodates multiple directed searches. Meanwhile, the neighborhood search is absent in FDSA. Moreover, this work's secondary objective is to investigate each search's contribution or dominance in finding the optimal solution.

Regarding these objectives, the leading scientific contributions of this work are as follows.

- The new swarm-based metaheuristics that contains multiple directed searches is designed.
- The reasoning, detailed description, and formalization of these searches are presented.
- This work investigates the performance of the designed metaheuristic to solve various optimization problems.
- This work investigates the comparative superiority of the designed metaheuristic with some latest metaheuristics.
- This work investigates the contribution or dominance of each search when it is deployed individually and collectively.

The remainder of this paper is arranged as follows. The review of some latest metaheuristics is carried out in section 2. The detailed description of the designed FDSA is presented in section 3, which includes the concept and formalization. The investigation of FDSA is presented in section 4. This investigation consists of the performance evaluation to solve the optimization problem, benchmarking with five new metaheuristics, and the contribution of each search that constructs the FDSA. The analysis regarding the

Table 1. The mechanics of latest metaheuristics and the theoretical test used in their first introduction

No	Algorithm	Directed Search	Random Search	References in Directed Search	Investigation
1	NGO [21]	1	1	a shuffled member within swarm	23 functions; engineering problem; hyper parameter
2	ASBO [31]	3	-	best member; resultant of best and worst members; difference between best and worst members	23 functions; hyper parameter
3	COA [14]	2	1	best member; a shuffled member within space	CEC 2011; CEC 2017; engineering problem
4	MLBO [35]	1	-	mixture of best member and a shuffled member	23 functions
5	ALO [30]	2	1	best member; resultant of best member and a shuffled member; resultant of two shuffled members	23 functions; hyper parameter
7	ZOA [15]	2	1	best member, a shuffled member	23 functions; CEC 2015; CEC 2017; engineering problem
8	GJO [19]	2	-	two best members	23 functions; engineering problem
9	OOA [20]	1	1	a better member	CEC 2017; CEC 2011
10	KMA [11]	3	1	some best members; the best member	23 functions
11	this work	4	-	best member; resultant of three members; a shuffled member; and resultant of best member; a shuffled member, and corresponding member	23 functions; hyper strategy

simulation result; strengths and weaknesses; limitations; and computational complexity of FDSA is presented in section 5. Finally, the conclusion and proposal for future studies based of this work are summarized in section 6.

2. Related works

Swarm intelligence is a popular method as a baseline for developing recent metaheuristics. Swarm intelligence is also a subset of population-based metaheuristics. As a population-based metaheuristic, a swarm contains a certain number of solutions. In several kinds of literature, these solutions have other names, such as agents, units, members, etc. Generally, each solution can be seen as an autonomous agent that performs searching independently without coordination or central command. But each agent does not work in an isolative manner. Each agent interacts with other agents to boost its search effort. The agent or solution is called a member in the remainder of this paper.

Swarm-based metaheuristics deploy directed search as their core strategy. In the directed search, a member improves the quality through interaction with other members. Based on its reference, this member moves within the search space as a boundary. In particle swarm optimization (PSO) as an early swarm-based metaheuristic, the member moves toward the mixture of the global best member and its local best member [37]. In GWO, as one popular swarm-based metaheuristic, a member moves toward the resultant of the three best members within the swarm [13]. In the marine predator algorithm (MPA), a member moves toward its local best member, or the local best member moves away from its corresponding member in the first stage [12]. Meanwhile, a member may move toward the gap of two shuffled members in the first stage, called Eddy formation [12].

Many recent swarm-based metaheuristics deploy multiple searches. In some metaheuristics, all these searches are directed inquiries. Meanwhile, in others, a neighborhood or full random search is added as a supplementary search to improve exploration effort. Some recent metaheuristics also improve the directed search by selecting more references or modifying the direction of the motion. The strategy and investigation performed in several latest swarm-based metaheuristics are reviewed in Table 1. Meanwhile, the last row exhibits the method, reference, and research of the designed metaheuristic in this work.

The summary depicted in Table 1 highlights several notes. First, it is common that the directed search becomes the primary strategy in swarm-based metaheuristics while the random search becomes the secondary one. It is seen that a swarm-based metaheuristic deploys at least one directed search. On the other hand, some swarm-based metaheuristics deploy random search during the iteration, while others do not deploy random search. Second, many swarm-based metaheuristics perform only one or two directed searches. Third, theoretical problems are commonly used for investigation, whether they are 23 functions, CEC 2011 [20], CEC 2015 [15], CEC 2017 [14], and so on. Practical optimization problems, such as engineering problems [8], are used for use cases in some studies, although it is not mandatory. Some studies also perform hyperparameters or sensitivity investigations. Finding studies proposing a new metaheuristic that conducts hyper-strategy analysis can be challenging.

These notes underscore the opportunity to propose a new metaheuristic, especially the swarm-based one. The new swarm-based metaheuristic can be designed by enriching the directed search and the reference. Moreover, the hyper strategy investigation should be promoted besides the hyperparameter investigation.

3. Model

FDSA is designed as a swarm-based metaheuristic that performs multiple search strategies. As a swarm-based metaheuristic, FDSA performs four directed searches without neighborhood searches. The motivation is returned to the basic swarm-based metaheuristic, where the directed search becomes the main strategy. As mentioned previously, the multiple search strategy is taken to tackle the weakness of every search because there is not any perfectly directed search. These four directed searches are performed sequentially by all members of the swarm. It means that there is not any role split within the swarm.

The first search is the motion toward the best member. This is the global best member, which means the best solution among the swarm from the beginning of iteration until the current iteration. This search is designed to improve the quality of the members fast because it is assumed that the best member may lead the members of the swarm to a better solution. Meanwhile, this motion does not guarantee improvement, especially when the problem is a multimodal problem with multiple optimal solutions. The motion toward the best member may lead to the local optimal. This first search is illustrated in Fig. 1.

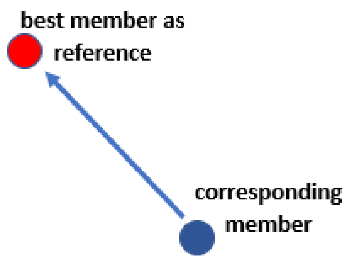

Figure. 1 First directed search

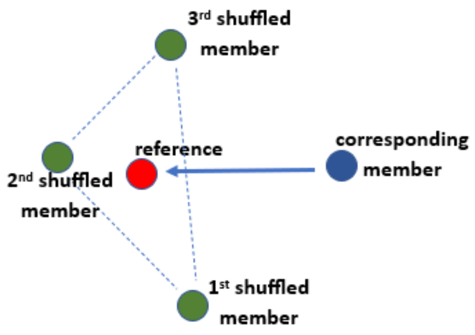

Figure. 2 Second directed search

The second search is the motion toward the resultant of three shuffled members within the swarm. This search is designed to trace possible improvement within the triangular area where three members control the edges. This search is intended as a localized diversification. The motion toward some members can also be found in some metaheuristics, such as GWO [13], KMA [11], and ASBO [31]. In GWO, the reference results from the three best members [13]. In KMA, the reference results from some better members [11]. In ASBO, the reference results from the best and worst members [31]. Unlike these metaheuristics, the reference is not deterministic, as the three members are shuffled within the swarm. This choice also reduces randomness rather than mixing members within the space. This second search is illustrated in Fig. 2.

The third search is the motion relative to the shuffled member within the swarm. This motion is designed to trace possible improvement in the area between the corresponding member and the shuffled member. There are two possible directions in this motion. The related member moves toward the shuffled member if the quality of the shuffled member is better than the corresponding member. Otherwise, the corresponding member moves away from the shuffled member. This movement is commonly implemented in various swarm-based metaheuristics, for example, in the first stage of NGO or the second stage of ZOA [15]. This third stage is illustrated in Fig. 3.

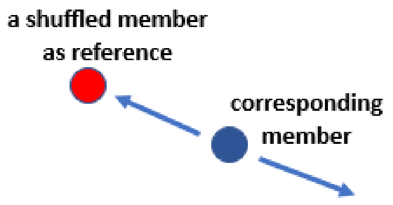

Figure. 3 Third directed search

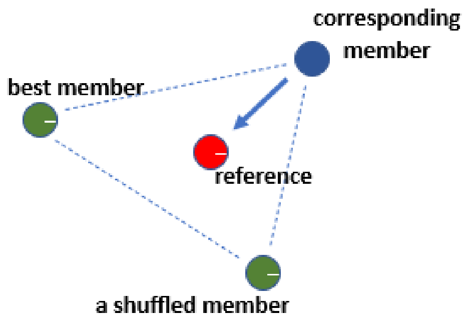

Figure. 4 Fourth directed search

The fourth search is the motion toward the resultant of the best member, a shuffled member within the swarm, and the corresponding member itself. This search is designed as a neighborhood search with a certain direction. This search can also be seen as a semi-neighborhood search. The reason is as follows. As the corresponding member becomes one of the constructors of the reference, the reference may not be far from the corresponding member. Meanwhile, different from the normal neighborhood search where the motion can be in any direction around the related member, the focus is controlled by the best member and the shuffled member in this search. This fourth search is illustrated in Fig. 4.

Formalization is then carried out by transforming this concept into an algorithm and a mathematical model. The algorithm, which is presented using pseudocode, describes the complete process performed in FDSA. Meanwhile, the mathematical model describes the detailed expression of each cycle. Some annotations used in both algorithm and mathematical model can be seen as follows. The formal process of FDSA can be seen in Algorithm 1.

The mathematical formalization is presented in Eqs. (1) to (10). Several suffices are used for representing the index or component. Suffix *i* represents the index of the swarm member. Meanwhile, suffix *d* represents the index of dimension.

As metaheuristics in general, FDSA is split into initialization and iteration phases. In Algorithm 1, lines 2 to 6 represent the initialization, while lines 7 to 21 represent the iteration. The m_b becomes the final solution.

Algorithm 1: FDSA

```
1   output: $m_b$
2   begin
3       for $i$=1: $n(M)$
4         generate initial member using Eq. (1)
5         update $m_b$ using Eq. (2)
6       end for
7       for $t$=1: $t_m$
8         for $i$=1: $n(m)$
9           1st search using Eq. (3)
10          update $m_i$ using Eq. (4)
11          update $m_b$ using Eq. (2)
12          2nd search using Eq. (5)-Eq. (7)
13          update $m_i$ using Eq. (4)
14          update $m_b$ using Eq. (2)
15          3rd search using Eq. (5) and Eq. (8)
16          update $m_i$ using Eq. (4)
17          run $m_b$ using Eq. (2)
18          4th search using Eq. (5), Eq. (9), Eq. (10)
19          update $m_b$ using Eq. (2)
20        end for
21      end for
22  end
```

b_l	lower boundary
b_u	upper boundary
c	member's candidate
f	objective function
m	swarm member
M	swarm
m_b	the best member
m_s	the shuffled member
m_r	reference
R	floating point uniform random between 0 and 1
t	iteration
t_m	maximum iteration
U	uniform random

The initialization phase consists of two processes. The first process is generating initial members uniformly within the space. This process is formalized using Eq. (1). Then, the m_b is updated based on a strict acceptance procedure, which is formalized using Eq. (2).

$$m_{i,d} = b_{l,d} + R(b_{u,d} - b_{l,d}) \tag{1}$$

$$m_b' = \begin{cases} m_i, f(m_i) < f(m_b) \\ m_b, else \end{cases} \tag{2}$$

Eq. (3) formalizes the motion toward the best member. The step range is twice the distance between the corresponding and best members. It makes the possibility that the corresponding member will surpass the best member to improve the quality of the best member in a faster way. Meanwhile, the actual step size is uniformly shuffled so that there is an equal probability of the location within this range becoming the candidate. Then this candidate will be evaluated using Eq. (4) to determine whether this candidate can be accepted for replacement of the current value of the corresponding member. In other words, Eq. (4) is the formalization of a strict acceptance procedure for the replacement of the member. Eq. (4) is used not only in the first search but in all four searches in FDSA.

$$c_{i,d} = m_{i,d} + R(m_{b,d} - 2m_{i,d}) \tag{3}$$

$$m_i' = \begin{cases} c_i, f(c_i) < f(m_i) \\ m_i, else \end{cases} \tag{4}$$

Eq. (5) to Eq. (7) are formalized for the second search. Eq. (5) states that a shuffled member is picked up within the swarm. This method represents the interaction of the corresponding member with another member within the swarm. In the second search, Eq. (5) is used three times because this search needs three shuffled members. Then, Eq. (6) formalized the resultant of these three shuffled members. Then, Eq. (7) formalized the motion toward this reference.

$$m_s = U(M) \tag{5}$$

$$m_{r1,d} = \frac{m_{s1,d}+m_{s2,d}+m_{s3,d}}{3} \tag{6}$$

$$c_{i,d} = m_{i,d} + R(m_{r1,d} - 2m_{i,d}) \tag{7}$$

Eq. (8) formalizes the motion relative to a shuffled member within the swarm. Meanwhile, Eq. (5) is used first to pick the shuffled member, which is unrelated to the shuffled members employed in the second search. But there is the probability that a shuffled member used in the second search is selected in the third search because there is no restriction for a member to be reelected. Then, Eq. (8) also declares the direction selection based on the quality comparison between the shuffled member and the corresponding member.

$$c_{i,d} = \begin{cases} m_{i,d} + R\left(m_{s,d} - 2m_{i,d}\right), f(m_s) < f(m_i) \\ m_{i,d} + R\left(m_{i,d} - 2m_{s,d}\right) \end{cases} \tag{8}$$

Eq. (9) and Eq. (10) are used as formalizations of the fourth search. Eq. (9) formalizes the resultant of the best, shuffled, and corresponding members. Meanwhile, Eq. (5) is used again to pick a shuffled member in this fourth search. Then, Eq. (10) formalizes this motion, and the quality comparison between the reference and the corresponding member is not considered.

$$m_{r2,d} = \frac{m_{b,d}+m_{s,d}+m_{i,d}}{3} \tag{9}$$

$$c_{i,d} = m_{i,d} + R(m_{r2,d} - 2m_{i,d}) \tag{10}$$

4. Simulation and result

The investigation of the performance of FDSA is carried out by implementing FDSA to solve theoretical optimization problems. The 23 classical functions represent these academic problems. There are two investigations carried out in this work. The first investigation is the competition between the designed FDSA and other metaheuristics. The second investigation is regarding the dominance of strategies in FDSA.

These 23 classical functions are used in both competing investigation and hyper-strategy investigation. These functions contain seven unimodal functions and seventeen high dimension functions. These seven unimodal functions and the first six multimodal functions are high-dimension functions, so the dimension ranges from very low to very high. Meanwhile, the next ten multimodal functions are fixed dimensions, so the size is static and usually quiet. The extent represents the constraints limited by the lower and upper boundaries. Meanwhile, each measurement is independent of the other dimensions. A detailed description of these functions is depicted in Table 2.

In the first investigation, FDSA faces five new metaheuristics: NGO, ASBO, COA, MLBO, and ALO. Like FDSA, all these five confronters implement strict acceptance strategies. MLBO and ASBO are the confronters that do not carry out neighborhood searches. Meanwhile, MLBO becomes the only confronter carrying out a single investigation and stage. The result is depicted in Table 3 to Table 5, representing the first group to the third group of functions consecutively. The data contains three pieces of information: the average fitness score, its related standard deviation, and the mean rank. In this investigation, the swarm size is set to 5 while the maximum iteration is set to 15.

Table 3 depicts the good performance of FDSA. It is ranked first in solving six functions (Sphere, Schwefel 2.22, Schwefel 1.2, Schwefel 2.21, Rosenbrock, and Quartic). Meanwhile, FDSA is placed on the second rank in solving Step. Meanwhile, four metaheuristics (ASBO, COA, ALO, and FDSA)

Table 2. Detail description of the set of 23 functions

No	Function	Dimension	Space	Target
1	Sphere	50	[-100, 100]	0
2	Schwefel 2.22	50	[-100, 100]	0
3	Schwefel 1.2	50	[-100, 100]	0
4	Schwefel 2.21	50	[-100, 100]	0
5	Rosenbrock	50	[-30, 30]	0
6	Step	50	[-100, 100]	0
7	Quartic	50	[-1.28, 1.28]	0
8	Schwefel	50	[-500, 500]	-2.0948x10^{4}
9	Ratsrigin	50	[-5.12, 5.12]	0
10	Ackley	50	[-32, 32]	0
11	Griewank	50	[-600, 600]	0
12	Penalized	50	[-50, 50]	0
13	Penalized 2	50	[-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]	-10.1532
22	Shekel 7	4	[0, 10]	-10.4028
23	Shekel 10	4	[0, 10]	-10.5363

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

F	Parameter	NGO [21]	ASBO [31]	COA [14]	MLBO [35]	ALO [30]	FDSA
1	mean	1.2148x10^{3}	2.9119x10^{1}	4.2395x10^{1}	2.7274x10^{4}	0.0032	0.0000
	std deviation	7.5119x10^{2}	1.0113x10^{1}	3.1203x10^{1}	2.5694x10^{3}	0.0102	0.0000
	mean rank	5	3	4	6	2	1
2	mean	2.6057x10^{12}	0.0000	0.0000	1.3369x10^{56}	0.0000	0.0000
	std deviation	1.2765x10^{13}	0.0000	0.0000	6.2706x10^{56}	0.0000	0.0000
	mean rank	5	1	1	6	1	1
3	mean	8.6739x10^{4}	1.7699x10^{4}	1.2158x10^{4}	8.7939x10^{4}	1.7109x10^{1}	0.0000
	std deviation	4.7403x10^{4}	1.6690x10^{4}	8.4953x10^{3}	2.8788x10^{4}	4.0531x10^{1}	0.0000
	mean rank	5	4	3	6	2	1
4	mean	3.2308x10^{1}	6.3633	1.0673x10^{1}	5.3075x10^{1}	0.0348	0.0000
	std deviation	1.7594x10^{1}	6.7838	3.4859	6.9979	0.0796	0.0000
	mean rank	5	3	4	6	2	1
5	mean	2.7298x10^{5}	7.7991x10^{2}	3.2379x10^{3}	3.3345x10^{7}	4.9096x10^{1}	4.8958x10^{1}
	std deviation	2.6944x10^{5}	4.7156x10^{2}	3.1540x10^{3}	1.0268x10^{7}	0.4333	0.0125
	mean rank	5	3	4	6	2	1
6	mean	1.4586x10^{3}	3.3812x10^{1}	7.7133x10^{1}	2.6067x10^{4}	1.1177x10^{1}	1.1248x10^{1}
	std deviation	8.2376x10^{2}	1.2306x10^{1}	3.5137x10^{1}	4.9236x10^{3}	0.4093	0.2713
	mean rank	5	3	4	6	1	2
7	mean	0.3565	0.1122	0.0937	2.7852x10^{1}	0.0333	0.0053
	std deviation	0.2544	0.0484	0.0523	9.1454	0.0293	0.0037
	mean rank	5	4	3	6	2	1

get the same average fitness score and are placed on the first rank in solving Schwefel 2.22. Moreover, FDSA can find the optimal global solution of four functions (Sphere, Schwefel 2.22, Schwefel 1.2, and Schwefel 2.21). This result indicates the superiority of FDSA in solving high-dimension unimodal functions. Besides, FDSA can solve unimodal functions with narrow spaces like Quartic.

In this first group of functions, the performance gap between the best and worst performers is significant. MLBO becomes the first worst performer, while NGOs always becomes the second worst performer. ALO becomes the second-best performer

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

F	Parameter	NGO [21]	ASBO [31]	COA [14]	MLBO [35]	ALO [30]	FDSA
8	mean	-3.1809x10^3	-3.7907x10^3	-4.5420x10^3	-3.2581x10^3	-3.6500x10^3	-2.7990x10^3
	std deviation	7.0541x10^2	6.1682x10^2	6.9911x10^2	7.7897x10^2	4.4754x10^2	7.3615x10^2
	mean rank	5	2	1	4	3	6
9	mean	3.5093x10^2	2.3192x10^1	4.0614x10^1	4.6768x10^2	0.0005	0.0000
	std deviation	9.5580x10^1	3.7899	2.5754x10^1	4.6811x10^1	0.0011	0.0000
	mean rank	5	3	4	6	2	1
10	mean	7.0629	3.7616	2.0623	1.7083x10^1	0.0186	0.0000
	std deviation	2.0648	0.8731	0.7214	0.6547	0.0530	0.0000
	mean rank	5	4	3	6	2	1
11	mean	1.0260x10^1	1.3055	1.5516	2.5075x10^2	0.0016	0.0000
	std deviation	6.4522	0.0717	0.4901	4.8027x10^1	0.0042	0.0000
	mean rank	5	3	4	6	2	1
12	mean	9.3091x10^3	0.7101	2.0415	2.4159x10^7	1.1681	1.1456
	std deviation	3.8334x10^4	0.4424	0.7824	1.9696x10^7	0.1141	0.1292
	mean rank	5	1	4	6	3	2
13	mean	4.0829x10^5	1.5597x10^1	7.9737	7.0312x10^7	3.1417	3.1260
	std deviation	1.6051x10^6	2.7729	2.5471	1.9508x10^7	0.0060	0.0314
	mean rank	5	4	3	6	2	1

in five functions. The performance gap between FDSA and ALO is insignificant except in Schwefel 1.2. Meanwhile, COA and ASBO are often on the third or fourth rank.

Table 4 depicts the good performance of FDSA in solving high-dimension multimodal functions. FDSA FDSA is on the first rank in four parts (Rastrigin, Ackley, Griewank, and Penalized 2), second rank in solving Penalized, and sixth rank in solving Schwefel. Moreover, FDSA can find the global optimal solution of three functions (Rastrigin, Ackley, and Griewank). This result indicates that FDSA performs well in solving functions with various search spaces, from the narrow ones, such as Rastrigin, to the large ones, such as Griewank.

The performance gap between the best and worst performers in the second group of functions. This performance gap in solving Schwefel is narrow. It means that FDSA is still competitive in solving Schwefel, although FDSA is the worst performer in this function. Its average fitness score is less than half of COA, which becomes the best performer. On the other hand, the performance gap in different tasks in this second group of parts is wide. Although FDSA is only the second-best performer in solving Penalizes, its performance is not from ASBO, which becomes the first-best performer. It means that FDSA is very competitive in solving high-dimension multimodal functions.

Table 4 also depicts the stable rank among these six competing metaheuristics. ALO is the second-best performer in four functions and the third-best performer in two parts. ASBO becomes the first-best performer in solving Penalized and the second-best performer in solving Schwefel. Otherwise, ASBO is placed in the third or fourth rank. On the other hand, COA becomes the first-best performer in solving SChwefel. Otherwise, COA is placed in the third or fourth rank. NGO is the second-worst performer in all functions in this second group. MLBO is the first-worst performer in solving five tasks in this group.

Table 5 depicts the fierce competition in the third group of functions, which contains ten fixed-dimension multimodal parts. In general, the performance gap among these metaheuristics is narrow, and this circumstance occurs in ten functions in this third group. FDSA becomes the first best performer only once in solving Hartman 3. But there are four confronters whose performance is also equal: NGO, ASBO, COA, and ALO. Besides, FDSA is placed in the third rank three times, in the fourth rank, in the fifth rank twice, and in the sixth rank twice.

The result in Table 3 to Table 5 is then summarized in Table 6 to investigate the superiority of FDSA relative to its confronters in every group of functions. Data in Table 6 represents how many times FDSA is better than the related confronter.

Table 6 depicts that overall, FDSA is superior to its five confronters. FDSA is better than NGO, ASBO, COA, MLBO, and ALO in 17, 13, 11, 18, and 11 functions consecutively. This data means that MLBO becomes the easiest confronters to beat. On the other hand, COA and ALO become the most difficult confronters to beat. Fierce competition occurs between FDSA and the other two confronters (ASBO and ALO). Compared to these two confronters, FDSA outperforms only eleven times and draws twice. It means that FDSA is beaten only in ten functions by both metaheuristics. The superiority of FDSA mostly occurs in solving high-dimension functions, whether unimodal or multimodal.

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

F	Parameter	NGO [21]	ASBO [31]	COA [14]	MLBO [35]	ALO [30]	FDSA
14	mean	$1.5813\text{x}10^1$	6.3241	7.1172	$1.7320\text{x}10^1$	6.3921	9.8512
	std deviation	$1.6333\text{x}10^1$	4.9530	4.3306	$2.1951\text{x}10^1$	3.9449	3.3327
	mean rank	5	1	3	6	2	4
15	mean	0.0303	0.1075	0.0092	0.0434	0.0077	0.0098
	std deviation	0.0336	0.0403	0.0104	0.0300	0.0108	0.0171
	mean rank	4	6	2	5	1	3
16	mean	-1.0076	-0.0836	-1.0281	-0.9186	-0.9735	-0.9376
	std deviation	0.0378	0.2274	0.0047	0.1444	0.1018	0.1518
	mean rank	2	6	1	5	3	4
17	mean	0.7828	1.2891	0.4259	0.5514	0.9934	2.2561
	std deviation	0.7494	1.2903	0.0818	0.2687	0.6483	1.8387
	mean rank	3	5	1	2	4	6
18	mean	$1.8744\text{x}10^1$	$1.5500\text{x}10^1$	6.5842	$2.5355\text{x}10^1$	8.8191	$3.7243\text{x}10^1$
	std deviation	$2.5815\text{x}10^1$	$5.8630\text{x}10^1$	$1.0047\text{x}10^1$	$3.2161\text{x}10^1$	8.5119	$4.1133\text{x}10^1$
	mean rank	4	3	1	5	2	6
19	mean	-0.0495	-0.0495	-0.0495	-0.0392	-0.0495	-0.0495
	std deviation	0.0000	0.0000	0.0000	0.0151	0.0000	0.0000
	mean rank	1	1	1	6	1	1
20	mean	-2.2164	-0.7612	-2.9541	-2.5143	-2.4354	-1.7453
	std deviation	0.5275	0.6314	0.2557	0.5174	0.4021	0.5905
	mean rank	4	6	1	2	3	5
21	mean	-1.0357	-2.6518	-3.3184	-1.6126	-1.6368	-1.9907
	std deviation	0.6182	2.9233	1.7410	0.9068	1.0119	1.0055
	mean rank	6	2	1	5	4	3
22	mean	-1.5039	-2.4312	-3.4216	-1.8828	-1.9788	-1.8060
	std deviation	1.3372	2.9365	1.4043	0.8615	1.0263	1.0661
	mean rank	6	2	1	4	3	5
23	mean	-1.7516	-2.8488	-3.5380	-1.9457	-2.0567	-2.6013
	std deviation	0.8701	2.8853	1.9692	1.0084	0.9372	1.0442
	mean rank	6	2	1	5	4	3

Table 6. Group based superiority of FDSA.

Group	Number of Functions Where FDSA is Better				
	NGO [21]	ASBO [31]	COA [14]	MLBO [35]	ALO [30]
1	7	6	6	7	5
2	5	4	5	5	5
3	5	3	0	6	2
Total	17	13	11	18	11

Meanwhile, FDSA maintains its superiority in solving the fixed dimension multimodal functions only to NGOs and MLBO.

The second investigation is the hyper-strategy investigation. This investigation is carried out to evaluate the contribution or dominance of each search in finding the optimal solution. This investigation is divided into two parts. The first part is a single search investigation. The second part is the missing search investigation. The first part is done by activating only one search while the three others are inactive. This part is designed to find the superiority of certain searches for a better result. On the other hand, the second part is carried out by deactivating one search while the three others remain active. In this second part, the dominance of a search is investigated by measuring the drop in performance when this certain strategy is missing. The result of the first part is depicted in Table 7, while the second part is shown in Table 8. The best result is written in bold font.

Table 7 depicts that the fourth search becomes the most dominant strategy in the single search investigation. Meanwhile, no dominant search exists in solving Schwefel 2.22 and Hartman 3. Neglecting these two functions makes the fourth search the best performer in solving 13 functions. These functions are distributed among all groups of parts. Meanwhile, the first, second, and third searches consecutively become the best performers in solving one role, one position, and six functions. Most processes where the third search becomes the dominant search occur in the third group of parts. Fortunately, the performance gap among these four searches is narrow enough.

Table 8 depicts that the performance gap when one search is deactivated is more distributed. There are eight functions where the average fitness score is

Table 7. Single search investigation

Function	Average Fitness Score			
	First Search	**Second Search**	**Third Search**	**Fourth Search**
1	0.0150	0.0433	3.6681	**0.0002**
2	**0.0000**	**0.0000**	**0.0000**	**0.0000**
3	3.7998	$1.0667\text{x}10^1$	$2.2476\text{x}10^3$	**1.4001**
4	0.0965	0.0997	2.9611	**0.0087**
5	$4.9183\text{x}10^1$	$4.9059\text{x}10^1$	$1.6214\text{x}10^2$	**$4.8980\text{x}10^1$**
6	$1.1240\text{x}10^1$	**$1.1203\text{x}10^1$**	$1.5694\text{x}10^1$	$1.1443\text{x}10^1$
7	0.0237	0.0327	0.0493	**0.0228**
8	$-1.9348\text{x}10^3$	$1.9882\text{x}10^3$	**$-2.9341\text{x}10^3$**	$-1.8332\text{x}10^3$
9	0.1057	0.0122	8.2561	**0.0001**
10	0.0234	0.0314	0.6404	**0.0027**
11	0.0419	0.0495	0.4780	**0.0079**
12	**1.1937**	1.1946	1.3391	1.2533
13	3.1690	3.1692	3.8462	**3.1383**
14	$1.1351\text{x}10^1$	$1.0504\text{x}10^1$	**$1.0228\text{x}10^1$**	$1.2160\text{x}10^1$
15	0.0254	0.0281	0.0230	**0.0171**
16	-0.9403	-0.8958	**-0.9667**	-0.7486
17	7.3172	2.6958	**2.5224**	6.3156
18	$5.2365\text{x}10^1$	$3.5693\text{x}10^1$	**$1.9610\text{x}10^1$**	$9.6933\text{x}10^1$
19	**-0.0495**	**-0.0495**	**-0.0495**	**-0.0495**
20	-1.1988	-1.3571	**-1.6013**	-1.3662
21	-0.9380	-1.0900	-1.1304	**-2.1250**
22	-1.1781	-1.1678	-1.4506	**-1.9257**
23	-1.0688	-1.3489	-1.3828	**-1.9976**

equal. There are 5, 5, 2, and 4 functions with the best average fitness score when the first, second, third, and fourth search is missing consecutively without considering the eight functions. Fortunately, the performance gap among the missing search scenarios is not wide. It means that random search during the iteration is less significant in improving the performance of the metaheuristic in the third group of functions.

5. Discussion

The in-depth analysis carried out in this section is divided into four parts. The first part concerns the competition between the designed FDSA and the confronters. The second part is related to the dominance of searches implemented in FDSA. The third part is related to the weakness or limitations of FDSA that can be used as the baseline for future development. The fourth part is associated with the computational complexity of the FDSA.

In general, the superiority of FDSA in almost all high-dimension functions, whether unimodal or multimodal, can be linked to the number of directed searches implemented in FDSA. Among all metaheuristics in this simulation, FDSA is the one with the highest number of directed searches. Moreover, the FDSA is also the one with the most diverse references in its directed searches. FDSA guarantees four directed searches performed by its member in every iteration. Meanwhile, as depicted in Table 1, the number of directed searches guaranteed in every member in every iteration for NGO, ASBO, COA, MLBO, and ALO is 1, 3, 1, 1, and 2 consecutively. Meanwhile, FDSA uses four references for its directed searches. On the other hand, the number of references used in NGO, ASBO, COA, MLBO, and ALO is 1, 3, 2, 1, and 3 consecutively.

The poor performance of MLBO and NGO in solving high-dimension functions can be linked to their limitation in conducting the directed search. MLBO deploys a single search which is the directed search [35]. Its reference is the mixture of the best member and shuffled members within the space. On the other hand, NGO deploys a single-directed search in its first phase, where its reference is the shuffled member within the swarm [21]. ALO performs better than ASBO can be linked to the diversity of the references used in them. All references to ASBO contain the best member [31]. In the first two stages, the best member is combined with the worst member, while in the third stage, the reference is the best member itself. On the other hand, the references of ALO are the best member, the best member, the shuffled member, and the mixture of two shuffled members [30]. Although COA deploys only two references, these references are the best and shuffled

Table 8. Single missing search investigation

Function	Average Fitness Score			
	First Search	Second Search	Third Search	Fourth Search
1	**0.0000**	**0.0000**	**0.0000**	**0.0000**
2	**0.0000**	**0.0000**	**0.0000**	**0.0000**
3	**0.0000**	**0.0000**	**0.0000**	**0.0000**
4	**0.0000**	**0.0000**	**0.0000**	**0.0000**
5	**$4.8950x10^1$**	$4.8960x10^1$	$4.8960x10^1$	$4.8954x10^1$
6	$1.1356x10^1$	**$1.1037x10^1$**	$1.1082x10^1$	$1.1182x10^1$
7	0.0074	0.0059	**0.0035**	0.0071
8	$-2.4702x10^3$	**$-2.7571x10^3$**	$-2.0766x10^3$	$-2.7470x10^3$
9	**0.0000**	**0.0000**	**0.0000**	**0.0000**
10	**0.0000**	**0.0000**	**0.0000**	**0.0000**
11	**0.0000**	**0.0000**	**0.0000**	**0.0000**
12	1.1689	1.1914	1.1752	**1.1517**
13	3.1336	3.1309	3.1328	**3.1283**
14	**9.7311**	9.8640	$1.1732x10^1$	$1.1064x10^1$
15	**0.0107**	0.0123	0.0228	0.0122
16	-0.8630	-0.8985	-0.8703	**-0.9027**
17	3.9166	**1.5844**	3.2249	3.0077
18	**$2.1881x10^1$**	$2.6214x10^1$	$3.0389x10^1$	$2.9601x10^1$
19	**-0.0495**	**-0.0495**	**-0.0495**	**-0.0495**
20	-1.6208	-1.2830	-1.1273	**-1.7568**
21	**-2.0505**	-1.3893	-2.0115	-1.3111
22	-1.8704	**-2.5283**	-1.7757	-1.5549
23	-1.9948	**-2.1755**	-2.0062	-1.5374

members [14]. This analysis indicates the necessity of the presence of the shuffled member.

The poor performance of FDSA in solving fixed-dimension multimodal functions can be linked to the absence of random search. This full random search can be transformed into the neighbourhood search, directed search whose reference is the shuffled member within the space, or full random search. The superiority of COA in this third group of functions can be linked to its better strategy in randomization. COA implements random search partially in its first stage and fully in its second stage [14]. On the other hand, ALO deploys a full random search conditionally [30]. Fortunately, the absence of a random search in the iteration is unnecessary. Although FDSA is not as superior as in the first and second group of functions, the performance gap between FDSA and the best performer in each role in the third group is narrower.

There are two findings regarding the strategy dominance investigation in the directed search. The first finding is that the mixture of the best member and the shuffled member is the strongest reference compared to the best member only, some shuffled members, or a shuffled member. But the dominance of this reference is not significant compared to other concerns. The second finding is that implementing multiple directed searches provides considerable improvement rather than developing a single-directed search strategy. The result in Table 8 becomes the baseline for this statement. Table 8 depicts that FDSA still can find the optimal global solution for nine functions, although it activates only three searches, and the missing search needs to be considered. When FDSA starts all its searches, the number of functions where the optimal global solution is found is also the same.

The computational complexity of FDSA can be investigated based on the looping process performed in the initialization and iteration. The complexity of FDSA in the initialization stage is $O(n(M).d)$ because there are only two loops in the initialization stage. The outer one is looping for a while swam, and the inner one is looping for all dimensions. Meanwhile, the complexity of FDSA in the iteration stage is $O(4t_m.n(M).d)$. There are three loops in the nested loop during the iteration stage. Meanwhile, the are four searches performed by each member in every iteration.

The main limitation of this work is the construction of FDSA as a fully directed search metaheuristic. Meanwhile, two searches have yet to be explored. The first is a neighbourhood search, while the second is a crossover-based search. Although this work indicates the less significant contribution of the neighbourhood search, improving this search is still challenging. The study to enhance the performance of crossover-based search is also difficult because GA, whose main strategy is based on the crossover-based search, is still widely used in many studies conducting optimization work. This improvement can also be made by combining the neighbourhood or crossover-based search with the directed search due to the superiority of the required investigation. This combination of hybridization is also important to cover the weakness of directed search, for example, in some functions.

6. Conclusion

This paper has presented the description and investigation of the designed metaheuristic called as four-directed-search algorithm (FDSA). FDSA is a metaheuristic that performs multiple directed searches and does not conduct neighborhood searches during iteration. By investigating the simulation result, FDSA performs well in solving the 23 classic functions. FDSA can find the optimal global solution of nine functions. Moreover, FDSA is superior to its five confronters. FDSA is better than NGO, ASBO, COA, MLBO, and ALO in 17, 13, 11, 18, and 11 functions consecutively. Its superiority is mostly in solving high-dimension functions. Meanwhile, FDSA is not so superior in solving fixed-dimension multimodal functions. By investigating the strategy dominance, implementing multiple directed searches improves performance significantly. Meanwhile, the neighborhood search is less significant in contributing the performance improvement.

In the future, studies in developing better neighborhood search and crossover-based search will be challenging. These new neighborhood searches or crossover-based searches can be combined with the directed search, where the directed search becomes the core strategy. This proposal comes from the fact that the performance of FDSA, a fully executed search metaheuristic, is still mere in some functions.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

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

Acknowledgments

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

References

[1] M. Nanjappan, P. Krishnadoss, J. Ali, G. Natesan, and B. Ananthakrishnan, "Task Scheduling Based on Cost and Execution Time Using Ameliorate Grey Wolf Optimizer Algorithm in Cloud Computing", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 3, pp. 417-427, 2023, doi: 10.22266/ijies2023.0630.33.

[2] 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.

[3] J. A. Abdulsaheb and D. J. Kadhim, "Multi-Objective Robot Path Planning Using an Improved Hunter Prey Optimization Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 2, pp. 215-227, 2023, doi: 10.22266/ijies2023.0430.18.

[4] G. Bompem, N. Chiluka, and D. Pandluri, "Effective Twitter Sentiment Analysis Using Deep Belief Network with Enhanced Dragonfly Optimization Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 2, pp. 64-73, 2023, doi: 10.22266/ijies2023.0430.06.

[5] M. A. Salam, N. N. Ahmed, A. Elshamy, A. W. S. Hassan, and M. Sami, "Enhanced Jellyfish Search Optimizer for Collaborative Team Formation in Social Network", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 2, pp. 74-91, 2023, doi: 10.22266/ijies2023.0430.07.

[6] P. Muthulakshmi and M. Parveen, "Z-Score Normalized Feature Selection and Iterative African Buffalo Optimization for Effective Heart Disease Prediction", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 1, pp. 25-37, 2023, doi: 10.22266/ijies2023.0228.03.

[7] Y.-Z. Li, K. Gao, L.-L. Meng, X.-L. Jing, and B. Zhang, "Heuristics and Metaheuristics to Minimize Makespan for Flowshop with Peak Power Consumption Constraints",

International Journal of Industrial Engineering Computations, Vol. 14, No. 2, pp. 221-238, 2023.

[8] L. Matijević, "General Variable Neighborhood Search for Electric Vehicle Routing Problem with Time-Dependent Speeds and Soft Time Windows", *International Journal of Industrial Engineering Computations*, Vol. 14, No. 2, pp. 275-292, 2023.

[9] P. Jangir, P. Manoharan, S. Pandya, and R. Sowmya, "MaOTLBO: Many-Objective Teaching-Learning-Based Optimizer for Control and Monitoring the Optimal Power Flow of Modern Power Systems", *International Journal of Industrial Engineering Computations*, Vol. 14, No. 2, pp. 293-308, 2023.

[10] 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.

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

[12] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine Predators Algorithm: A Nature-inspired Metaheuristic", *Expert System with Applications*, Vol. 152, ID: 113377, 2020.

[13] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer", *Advances in Engineering Software*, Vol. 69, pp. 46–61, 2014.

[14] 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.

[15] 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.

[16] 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.

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

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] P. D. Kusuma and A. L. Prasasti, "Guided Pelican Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 6, pp. 179-190, 2022, doi: 10.22266/ijies2022.1231.18.

[24] 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.

[25] P. D. Kusuma and D. Adiputra, "Modified Social Forces Algorithm: from Pedestrian Dynamic to Metaheuristic Optimization", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 3, pp. 294–303, 2022, doi: 10.22266/ijies2022.0630.25.

[26] 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.

[27] 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.

[28] P. Trojovský and M. Dehghani, "A New Optimization Algorithm based on Mimicking the Voting Process for Leader Selection", *Peer J Computer Science*, Vol. 8, ID. e976, pp. 1-40, 2022.

[29] M. Noroozi, H. Mohammadi, E. Efatinasab, A. Lashgari, M. Eslami, and B. Khan, "Golden Search Optimization Algorithm", *IEEE Access*, Vol. 10, pp. 37515-37532, 2022.

[30] 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.

[31] M. Dehghani, S. Hubalovsky, and P. Trojovsky, "A New Optimization Algorithm based on Average and Subtraction of the Best and Worst Members of the Population for Solving Various Optimization Problems", *PeerJ Computer Science*, Vol. 8, ID: e910, pp. 1-29, 2022.

[32] 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.

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

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

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

[36] 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. Garcia-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.

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