Best Couple Algorithm: A New Metaheuristic with Two Types of Equal Size Swarm Splits

Purba Daru Kusuma, *Member IAENG*

***Abstract*—As stated in the no-free-lunch (NFL) theory, there is not any optimizer suitable for all problems. This circumstance becomes the motivation of introducing a new swarm-based metaheuristic called best couple algorithm (BCA). BCA is constructed as a swarm-based metaheuristic where the swarm is split into two sub-swarms. There are two types of splitting. The first split is dividing the swarm into the first half and second half of swarms. The second split is dividing the swarm into the odd indexed swarm members and even indexed swarm members. There is a sub swarm leader representing the highest quality swarm member in every sub swarm. There are two sequential searches for every split: the motion toward the middle between two sub swarm leaders and the motion relative to the middle between two randomly picked sub swarm members. In the benchmark assessment, BCA is compared with total interaction algorithm (TIA), coati optimization algorithm (COA), language education algorithm (LEO), osprey optimization algorithm (OOA), and walrus optimization algorithm (WaOA). The result shows that BCA is superior to these five contenders as it is better than TIA, COA, LEO, OOA, and WaOA in 18, 18, 16, 18, and 18 functions respectively out of 23 functions.**

***Index Terms*—swarm intelligence, metaheuristic, stochastic process, random motion.**

I. Introduction

METAHEURISTIC can be found in many studies focusing on optimization. Particle swarm optimization (PSO) has been utilized to optimize the fresh product distribution [1]. PSO also has been utilized to improve the accuracy for medical image retrieval [2]. Northern goshawk optimization (NGO) has been utilized in power systems with distributed generators [3]. Ant colony optimization has been combined with independent component analysis to reduce the noise in the night vision image [4]. Grey wolf optimization (GWO) has been utilized in DC-DC converters design to minimize the operational losses [5], travelling salesman problem [6], smart grid [7], and so on. A variant of ant colony optimization (ACO) named max-min ant system (MMAS) has been utilized to solve the capacitated vehicle routing problem (CVRP) where its objective is minimizing the total operational cost [8]. The multi objective artificial bee colony (MOABC) and multi objective particle swarm optimization (MOPSO) have been utilized to develop the optimal design of low noise amplifier (LNA) [9]. Genetic algorithm (GA) has been utilized to determine the shortest path i.e., best packet route in the computer network [10]. GA was also used to recognize the Arabic name entity on the social media [11]. In the cloud system, GA has been combined with heterogeneous integrated load balancing (HILB) algorithm to optimize the task scheduling with the objective is to minimize the make-span [12].

Nowadays, there are a lot of metaheuristics already exist. Most of them are swarm-based metaheuristics. Many of them use metaphors, especially the animal behaviors like Komodo mlipir algorithm (KMA) [13], stochastic Komodo algorithm [14], coati optimization algorithm (COA) [15], zebra optimization algorithm (ZOA) [16], osprey optimization algorithm (OOA) [17], walrus optimization algorithm (WaOA) [18], whale optimization algorithm (WOA) [19], golden jackal optimization (GJO) [20], marine predator algorithm (MPA) [21], clouded leopard optimization (CLO) [22], Siberian tiger optimization (STO) [23], pelican optimization algorithm (POA) [24], northern goshawk optimization (NGO) [25], reptile search algorithm (RSA) [26], chimp optimization algorithm (ChOA) [27], squirrel search optimization (SSO) [28], green anaconda optimization (GAO) [29], and so on. Some metaheuristics are inspired by the human or social behavior, such as modified social forces algorithm (MSFA) [30], human conception optimization (HCO) [31], migration algorithm (MA) [32], language education optimization (LEO) [33], chef-based optimization algorithm (CBOA[34]), election-based optimization algorithm (EBOA) [35], drawer algorithm (DA) [36], mother optimization algorithm (MOA) [37], and so on. Some metaheuristics do not use metaphors, such as Nizar optimization algorithm (NOA) [38], total interaction algorithm (TIA) [39], average and subtraction-based optimization (ASBO) [40], golden search optimization (GSO) [41], attack leave optimization (ALO) [42], fully informed search algorithm (FISA) [43], and so on.

Despites the massive development of metaheuristics, especially the swarm-based metaheuristics, there are three considerations regarding it. First, the existence of no-free-lunch (NFL) theory becomes the main motivation of the development of a lot of metaheuristics. NFL states that there are not any generic solutions for all problems. Some metaheuristics may be superior to solve some problems but inferior in other problems [43]. This theorem also becomes the motivation to develop better metaheuristic than the previous or existing ones. Moreover, it also becomes the motivation to develop a new metaheuristic that can tackle the weakness of the previous ones. Second, almost all swarm-based metaheuristics do not perform swarm split or segregation of roles. It means that all agents perform the same

Manuscript received March 13, 2024; revised June 30, 2024. This work was financially supported by Telkom University, Indonesia.

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

search strategy or move in a single flow. Only a few metaheuristics like KMA [13] and COA [15] performs the swarm splits. In KMA [13], the swarm is split based on the quality of the swarm members while in COA [15], the swarm is split without considering the quality of the swarm members. Third, many swarm-based metaheuristics use the best swarm leader as the reference or one of its references. It can be seen in many metaheuristics like COA [15], ZOA [16], KMA [13], WaOA [18], and so on. Meanwhile, a new metaheuristic can be developed by introducing a new alternative for reference rather than the best swarm member.

This work is aimed at introducing a new swarm-based called as best-couple algorithm (BCA) to address three considerations previously explained. The innovation and novelty of this work are mainly on the swarm split mechanism and the introduction of new references used in the directed search which are the mixture of entities from each sub swarm. Moreover, the scientific contributions of this paper are as follows.

- This work introduces two approaches of swarm split where the swarm is split into two equal size sub-swarms without performing sorting based on the quality of swarm members. In both splits, the swarm members' index becomes the inly consideration.
- This work introduces two novel references for the directed search. The first reference is the mixture of sub swarm leaders while the second reference is the mixture of two randomly selected sub swarm members.
- There are three assessments to investigate the performance of BCA. The benchmark assessment is performed to compare the performance of BCA with other new metaheuristics. The individual search assessment is performed to measure the performance of each search in BCA.
- Both assessments use 23 classic functions representing the problems due to the coverage of these functions.
- The third assessment is conducted by implementing BCA to tackle the ELD problem with 13 generating units that represent the constrained practical problem.

The rest of this paper is arranged as follows. The research method is explained in section two. It consists of the description of the proposed model whether the concept and formalization. Then section three consists of the assessment result. Section four discusses in-depth analysis regarding the result, findings, limitations, and computational complexity. In the end, section five summarizes the conclusion and tracks for future studies.

II. Model

The fundamental concept of BCA starts with the swarm split strategy. As a swarm-based metaheuristic, BCA is constructed as a collection of autonomous agents called swarm. It means that these autonomous agents are also called swarm members. This swarm is split into two sub-swarms where the size of both sub-swarms is equal. In BCA there are two types of splits. The swarm member index becomes the only consideration in both splits. In the first split, the swarm is split based on the index order. The first sub swarm consists of the first half of swarm members while the second sub swarm consists of the second half of swarm members. In the second split, the swarm is split based on the odd member index and the even member index. All swarm members whose index is odd are included into the first sub swarms. On the other hand, all swarm members whose index is even are included into the second sub swarms. The illustration of these splits is in Fig. 1.

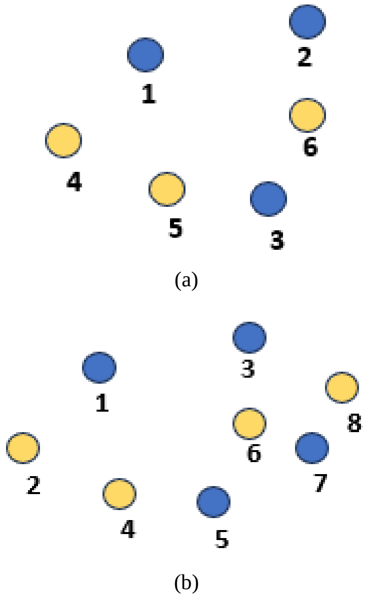

Fig. 1. Swarm split: (a) first swarm split and (b) second swarm split.

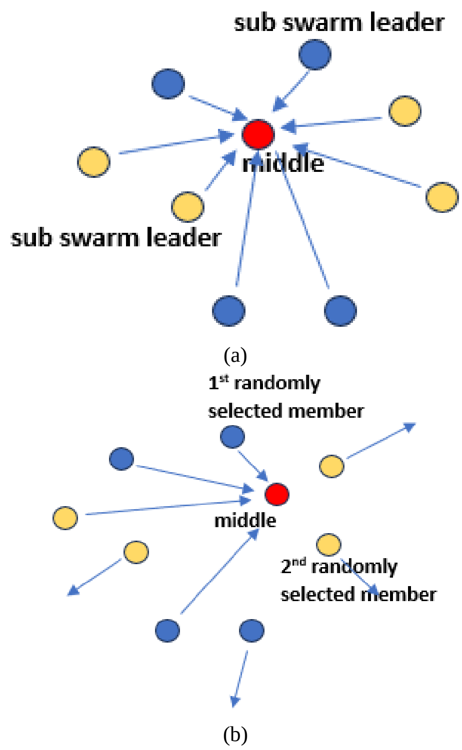

Fig. 2. Searching: (a) motion toward the middle between two sub swarm leaders and (b) motion relative to the middle between two randomly selected sub swarm members

There are two searches performed by each swarm member in every split. The first search is the motion toward the middle between two sub-swarm leaders. The sub swarm leader is the sub swarm member whose quality is the highest among the related sub swarm. The second search is the motion relative to the middle between two randomly picked sub-swarm members. In this second search a swarm member is randomly picked from each sub swarm. Then, the middle location or solution between these two randomly picked swarm members are calculated. Then, the swarm member will move relative to this reference. The direction depends on the comparative quality between the reference and the swarm member. If this reference is better than the swarm member, then the swarm member moves toward the reference. Otherwise, the swarm member avoids or moves away from this reference. The illustration of these two searches is presented in Fig. 2.

These two searches are interpreted into four sequential searches. The first search is the motion toward the middle of two sub-swarm leaders from the first split. The second search is the motion relative to the middle of two randomly picked sub swarm members from the first split. The third search is the motion toward the middle of two sub swarm leaders from the second split. The fourth search is the motion relative to the middle of two randomly picked sub swarm members from the second split.

This fundamental concept of BCA is then transformed into the algorithm of BCA. The formal presentation of BCA is found in algorithm 1. Meanwhile, the mathematical formulation following this algorithm is presented in (1) to (19). Below are the annotations used in this formalization.

a	swarm member
A	swarm
A_{11}, A_{12}	the first and second sub swarms from the first split
A_{21}, A_{22}	the first and second sub swarms from the second split
a_{be}	swarm leader
a_{be11}, a_{be12}	sub swarm leaders from the first split
a_{be21}, a_{be22}	sub swarm leaders from the second split
a_{re1}, a_{re2}, a_{re3}, a_{re4}	the first to fourth references
b_{lo}, b_{hi}	lower boundary and higher boundary
c	candidate
d	dimension
f	objective function
i, j	index for swarm member and index for dimension
r_1	real uniform random number [0,1]
r_2	integer uniform random number [1,2]
t	iteration
t_m	maximum iteration

The mathematical formulation starts with the declaration of the sub swarms. It is presented in (1) to (4). Equations (1) and (2) represent the first swarm split. Meanwhile, (3) and (4) represent the second swarm split. This process is performed before the initialization phase.

$$A_{11} = \{a_i | a_i \in A \wedge 1 \leq i \leq \tfrac{n(A)}{2}\} \tag{1}$$

$$A_{12} = \{a_i | a_i \in A \wedge \tfrac{n(A)}{2} + 1 \leq i \leq n(A)\} \tag{2}$$

$$A_{21} = \{a_i | a_i \in A \wedge i = odd\} \tag{3}$$

$$A_{22} = \{a_i | a_i \in A \wedge i = even\} \tag{4}$$

Initialization phase is presented in lines 3 to 7 in algorithm 1. Meanwhile, the mathematical formulation following the initialization phase is presented in (5) to (10). Equation (5) states that the swarm members are uniformly distributed within the search space in the initialization phase. Then, the sub-swarm leaders are updated using (6) to (9). Equations (6) and (7) are used to update the sub swarm leaders in the first split while (8) and (9) are used to update the sub swarm leaders in the second split. The swarm leader is updated by using (10) after the updating process of sub swarm leaders ends.

algorithm 1: best couple algorithm (BCA)

```
1   begin
2     define all sub swarms using (1) to (4)
3     for i=1 to n(A)
4       initialize swarm member using (5)
5       calculate sub swarm leaders using (6) to (9)
6       calculate swarm leader using (10)
7     end
8     for t=1 to t_m
9       perform first search using (11) to (13)
10      perform second search using (14), (15), (13)
11      perform third search using (16), (17), (13)
12      perform fourth search using (18), (19), (13)
13      update sub swarm leader using (6) to (9)
14      update swarm leader using (10)
15    end for
16  end
```

$$a_{i,j} = b_{lo,j} + r_1\left(b_{hi,j} - b_{lo,j}\right) \tag{5}$$

$$a'_{be11} = \begin{cases} a_i, f(a_i) < f(a_{be11}) \wedge 1 \leq i \leq \frac{n(A)}{2} \\ a_{be11}, otherwise \end{cases} \tag{6}$$

$$a'_{be12} = \begin{cases} a_i, f(a_i) < f(a_{be12}) \wedge \frac{n(A)}{2} + 1 \leq i \leq n(A) \\ a_{be12}, otherwise \end{cases} \tag{7}$$

$$a'_{be21} = \begin{cases} a_i, f(a_i) < f(a_{be21}) \wedge i = odd \\ a_{be21}, otherwise \end{cases} \tag{8}$$

$$a'_{be22} = \begin{cases} a_i, f(a_i) < f(a_{be22}) \wedge i = even \\ a_{be22}, otherwise \end{cases} \tag{9}$$

$$a'_{be} = \begin{cases} a_i, f(a_i) < f(a_{be}) \\ a_{be}, otherwise \end{cases} \tag{10}$$

The first and second searches are formalized using (11) to (15). Equation (11) is used to determine the middle between two sub swarm leaders of the first split. Equation (12) represents the motion toward the first reference. Equation (13) represents the strict acceptance for the updating process of the corresponding swarm member. Equation (14) represents the random selection of sub swarm members and the generation of the second reference which is in the middle between these two randomly selected sub swarm members.

$$a_{re1,j} = \frac{a_{be11,j} + a_{be12,j}}{2} \tag{11}$$

$$c_{i,j} = a_{i,j} + r_1\left(a_{re1,j} - r_2 a_{i,j}\right) \quad (12)$$

$$a_i' = \begin{cases} c_i, f(c_i) < f(a_i) \\ a_i, otherwise \end{cases} \quad (13)$$

$$a_{re2,j} = \frac{a_{ra11,j} + a_{ra12,j}}{2}, a_{ra11} = U(A_{11}) \wedge a_{ra12} = U(A_{12}) \quad (14)$$

$$c_{i,j} = \begin{cases} a_{i,j} + r_1\left(a_{re2,j} - r_2 a_{i,j}\right), f(a_{re2}) < f(a_i) \\ a_{i,j} + r_1\left(a_{i,j} - r_2 a_{re2,j}\right), otherwise \end{cases} \quad (15)$$

Equations (16) to (19) are used for the third and fourth searches. Equation (16) represents the generation of the third reference that is in the middle between two sub swarm leaders in the second split. Equation (17) represents the motion toward the third reference. Equation (18) is used to generate the fourth reference which is in the middle between two randomly selected sub swarm members from the second split. Equation (19) represents the motion relative to the fourth reference.

$$a_{re3,j} = \frac{a_{be21,j} + a_{be22,j}}{2} \quad (16)$$

$$c_{i,j} = a_{i,j} + r_1\left(a_{re3,j} - r_2 a_{i,j}\right) \quad (17)$$

$$a_{re4,j} = \frac{a_{ra21,j} + a_{ra22,j}}{2}, a_{ra21} = U(A_{21}) \wedge a_{ra22} = U(A_{22}) \quad (18)$$

$$c_{i,j} = \begin{cases} a_{i,j} + r_1\left(a_{re4,j} - r_2 a_{i,j}\right), f(a_{re4}) < f(a_i) \\ a_{i,j} + r_1\left(a_{i,j} - r_2 a_{re4,j}\right), otherwise \end{cases} \quad (19)$$

III. RESULT

The performance of BCA is assessed in three ways. The first assessment is called benchmark assessment. Its objective is to compare the performance of BCA to the performance of the existing metaheuristics. In other words, the benchmark assessment is performed to measure the improvement provided by the proposed BCA.

TABLE I
23 BENCHMARK FUNCTIONS

No	Function	Type	Dimension	Problem Space	Global Optimal
1	Sphere	HDUF	40	[-100, 100]	0
2	Schwefel 2.22	HDUF	40	[-100, 100]	0
3	Schwefel 1.2	HDUF	40	[-100, 100]	0
4	Schwefel 2.21	HDUF	40	[-100, 100]	0
5	Rosenbrock	HDUF	40	[-30, 30]	0
6	Step	HDUF	40	[-100, 100]	0
7	Quartic	HDUF	40	[-1.28, 1.28]	0
8	Schwefel	HDMF	40	[-500, 500]	-418.9 x dim
9	Ratsrigin	HDMF	40	[-5.12, 5.12]	0
10	Ackley	HDMF	40	[-32, 32]	0
11	Griewank	HDMF	40	[-600, 600]	0
12	Penalized	HDMF	40	[-50, 50]	0
13	Penalized 2	HDMF	40	[-50, 50]	0
14	Shekel Foxholes	FDMF	2	[-65, 65]	1
15	Kowalik	FDMF	4	[-5, 5]	0.0003
16	Six Hump Camel	FDMF	2	[-5, 5]	-1.0316
17	Branin	FDMF	2	[-5, 5]	0.398
18	Goldstein-Price	FDMF	2	[-2, 2]	3
19	Hartman 3	FDMF	3	[1, 3]	-3.86
20	Hartman 6	FDMF	6	[0, 1]	-3.32
21	Shekel 5	FDMF	4	[0, 10]	-10.1532
22	Shekel 7	FDMF	4	[0, 10]	-10.4028
23	Shekel 10	FDMF	4	[0, 10]	-10.5363

TABLE II
SIMULATION RESULT FOR HIGH DIMENSION UNIMODAL FUNCTIONS (HDUF)

F	Parameter	TIA [39]	COA [15]	LEO [33]	OOA [17]	WaOA [18]	BCA
1	mean	3.8137	3.6134×10^{2}	7.6376	1.0633×10^{2}	2.5859	0.0000
	std deviation	0.9173	2.0181×10^{2}	3.9696	4.1079×10^{1}	2.3127	0.0000
	mean rank	3	6	4	5	2	1
2	mean	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	std deviation	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	mean rank	1	1	1	1	1	1
3	mean	7.1323×10^{2}	1.6017×10^{4}	3.4843×10^{3}	1.1729×10^{4}	1.1044×10^{3}	3.8542×10^{1}
	std deviation	6.3083×10^{2}	8.1915×10^{3}	2.8459×10^{3}	8.1109×10^{3}	7.4968×10^{2}	4.8111×10^{1}
	mean rank	2	6	4	5	3	1
4	mean	1.7630	1.6522×10^{1}	2.3610	7.5534	1.4771	0.0031
	std deviation	0.3904	2.7478	0.7188	1.7087	0.6077	0.0011
	mean rank	3	6	4	5	2	1
5	mean	1.1080×10^{2}	3.2822×10^{4}	1.1641×10^{2}	2.7374×10^{3}	6.2537×10^{1}	3.8926×10^{1}
	std deviation	1.5876×10^{1}	2.2998×10^{4}	8.2061×10^{1}	2.1305×10^{3}	1.1207×10^{1}	0.0255
	mean rank	3	6	4	5	2	1
6	mean	8.1432	4.0240×10^{2}	1.7381×10^{2}	1.2831×10^{2}	1.0551×10^{1}	7.6143
	std deviation	1.5876	1.4659×10^{2}	7.2356×10^{1}	5.6214×10^{1}	2.0242	0.5987
	mean rank	2	6	5	4	3	1
7	mean	0.0442	0.1888	0.0350	0.1049	0.0326	0.0095
	std deviation	0.0245	0.0831	0.0173	0.0518	0.0183	0.0059
	mean rank	4	6	3	5	2	1

TABLE III
SIMULATION RESULT FOR HIGH DIMENSION MULTIMODAL FUNCTIONS

F	Parameter	TIA [39]	COA [15]	LEO [33]	OOA [17]	WaOA [18]	BCA
8	mean	-2.1441×10^3	-3.6980×10^3	-3.4552×10^3	-3.3469×10^3	-3.3674×10^3	-3.2833×10^3
	std deviation	3.8170×10^2	4.9592×10^2	4.8126×10^2	5.3624×10^2	6.6161×10^2	4.3868×10^2
	mean rank	6	1	2	4	3	5
9	mean	3.2342×10^1	1.3497×10^2	2.0750×10^2	1.2943×10^2	1.1989×10^1	0.9084
	std deviation	2.1206×10^1	5.0442×10^1	6.5116×10^1	5.3624×10^1	9.0926	4.0301
	mean rank	3	5	6	4	2	1
10	mean	0.7889	5.3017	1.1129	3.4781	0.6211	0.0006
	std deviation	0.1514	0.8213	0.3514	0.5467	0.2595	0.0002
	mean rank	3	6	4	5	2	1
11	mean	0.6680	4.6471	0.8890	1.9218	0.4814	0.0022
	std deviation	0.1806	1.8843	0.1961	0.4108	0.2337	0.0068
	mean rank	3	6	4	5	2	1
12	mean	0.5882	6.1303	0.9989	2.1993	0.8921	0.8611
	std deviation	0.1219	1.7803	0.1881	0.7774	0.2023	0.1282
	mean rank	1	6	4	5	3	2
13	mean	2.9810	3.7357×10^2	3.7791	8.8116	3.4862	3.1208
	std deviation	0.3026	8.8267×10^2	0.3730	3.3118	0.2565	0.0960
	mean rank	1	6	4	5	3	2

In the benchmark assessment, BCA is contended with five new metaheuristics: TIA, COA, LEO, OOA, and WaOA. All these metaheuristics were first introduced in 2023. Meanwhile, the second assessment is called individual search assessment. Its objective is to evaluate the contribution of each search in BCA as BCA is a multi-search metaheuristic. In the first assessment, all searches in BCA are active. On the other hand, in the second assessment, only one search active in each session of simulation.

A set consisting of 23 functions is chosen representing the problem. These functions are clustered into three groups: seven high dimension unimodal functions (HDUF), six high dimension multimodal functions (HDMF), and ten fixed dimension multimodal functions (FDMF). These functions were chosen for two reasons. The first reason is this set of functions covers various circumstances of problems. It covers both unimodal functions and multimodal functions. The search space width in these functions also varies from the narrow ones to the very large ones. Moreover, the terrain of the search spaces also varies from smooth, ripple, to ambiguous. These varieties make solving each function with single optimizer become more challenging. The second reason is that this set of functions is popular in many studies proposing new metaheuristic like in KMA [13] or TIA [39], besides CEC 2005 [43], CEC-2011 [36], CEC-2017 [37], and so on. The detailed description of these functions can be seen in Table 1.

TABLE IV
SIMULATION RESULT FOR FIXED DIMENSION MULTIMODAL FUNCTIONS

F	Parameter	TIA [39]	COA [15]	LEO [33]	OOA [17]	WaOA [18]	BCA
14	mean	7.9810	7.2799	7.1527	8.7168	9.0781	6.4432
	std deviation	2.7256	3.4008	3.9438	4.1862	3.9976	3.6199
	mean rank	4	3	2	5	6	1
15	mean	0.0008	0.0080	0.0024	0.0040	0.0042	0.0007
	std deviation	0.0004	0.0083	0.0034	0.0049	0.0069	0.0005
	mean rank	2	6	3	4	5	1
16	mean	-1.0302	-1.0257	-1.0302	-1.0237	-1.0295	-1.0307
	std deviation	0.0043	0.0050	0.0018	0.0074	0.0040	0.0019
	mean rank	2	5	2	6	4	1
17	mean	0.4781	0.4147	0.3999	0.4136	0.3996	0.4227
	std deviation	0.2815	0.0247	0.0022	0.0185	0.0019	0.0888
	mean rank	6	4	2	3	1	5
18	mean	4.9056	5.9285	3.0352	6.8994	9.0201	3.0763
	std deviation	5.7894	7.5027	0.0436	1.6629×10^1	1.8871×10^1	0.2106
	mean rank	3	4	1	5	6	2
19	mean	-0.0495	-0.0495	-0.0495	-0.0495	-0.0495	-0.0495
	std deviation	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	mean rank	1	1	1	1	1	1
20	mean	-2.8968	-2.9422	-3.1223	-3.0238	-3.0093	-2.7805
	std deviation	0.2571	0.1652	0.1028	0.1417	0.1429	0.2261
	mean rank	5	4	1	2	3	6
21	mean	-2.9090	-3.0979	-2.6325	-2.2663	-2.6140	-3.6328
	std deviation	1.7543	1.6671	1.2102	0.9645	1.3766	1.1652
	mean rank	3	2	4	6	5	1
22	mean	-3.1173	-3.0939	-2.9231	-2.4745	-2.8865	-4.0279
	std deviation	1.6797	1.4009	1.0576	0.9287	1.6030	1.3824
	mean rank	2	3	4	6	5	1
23	mean	-2.4839	-2.5297	-3.6172	-2.6827	-2.7752	-3.4847
	std deviation	0.5039	0.5775	1.7749	0.9054	1.0683	0.9573
	mean rank	6	5	1	4	3	2

In both first and second assessments, both swarm size and maximum iteration are set to 10. Due to very low maximum iteration, it becomes more difficult to find the acceptable solution or quasi optimal solution. This scenario is different from many studies where the maximum iteration is set high.

The result of the first assessment is presented in Table 2 to Table 5. Meanwhile, the result of the second assessment is presented in Table 6. In Table 6, the best result in every function is written in bold font. The decimal point smaller than 10^{-4} is rounded.

Table 2 indicates that BCA performs well in overcoming the high dimension unimodal functions. BCA becomes the best performer in all seven HDUFs. It is also the distinct best performer for six functions (f_1, f_3, f_4, f_5, f_6, and f_7). Meanwhile, all contestants achieve the same result in solving f_2. BCA achieves the global optimal solution in solving f_1 and f_2. Except in f_2, the performance gap between the best performer and the worst performer is wide. The gap performance between BCA as the best contestant and TIA as the second-best contestant is narrow in solving f_4. The gap performance between BCA as the best contestant and WaOA as the second-best contestant is narrow in solving f_5. Otherwise, the performance gap between the first best contestant and the second-best contestant is wide in five functions (f_1, f_3, f_6, and f_7).

Table 3 indicates that BCA still performs well and is superior enough in solving HDMFs. BCA becomes the first best contestant in three functions (f_9, f_{10}, and f_{11}), second best contestant in two functions (f_{12} and f_{13}), and fifth best contestant in f_8. In f_{12} and f_{13}, TIA becomes the best contestant and better than BCA. Meanwhile, in f_8, TIA is the only contestant where BCA is better than it. The competition in solving f_8 is fierce and it can be seen through the narrow performance gap between the best contestant and the worst contestant. On the other hand, the performance gap between the best contestant and the worst contestant is wide in four functions (f_9, f_{10}, f_{11}, and f_{13}).

Table 4 indicates the good performance of BCA in solving FDMFs. BCA becomes the distinct best contestant in five functions (f_{14}, f_{15}, f_{16}, f_{21}, and f_{22}). All contestants achieved the same result in f_{19}. Meanwhile, BCA becomes the second-best contestant in two functions (f_{18} and f_{23}), the fifth-best contestant in f_{17}, and sixth-best contestant in f_{20}. Different from in HDUFs and HDMFs, the performance gap among contestants in FDMFs is narrow. All contestants achieved the same result in f_{19}.

TABLE V
SUPERIORITY SUMMARY OF BCA

Cluster	Number of functions where BCA is better				
	TIA [39]	COA [15]	LEO [33]	OOA [17]	WaOA [18]
1	6	6	6	6	6
2	4	5	5	5	5
3	8	7	5	7	7
Total	18	18	16	18	18

The result in Table 2 to Table 4 is then summarized in Table 5. Table 5 presents the superiority of BCA among other contestants based on the number of functions in which BCA is better than the related contestant in every cluster of functions. This exhibition shows that overall, BCA is better than TIA, COA, LEO, OOA, and WaOA in 18, 18, 16, 18, and 18 functions respectively. This distribution shows that LEO is the toughest competitor of BCA. Moreover, except for LEO, the superiority of BCA among other contestants takes place in all groups of functions.

The result presented in Table 6 indicates the superiority of the third search compared to other searches. The third search achieves the best result in 14 functions. Its superiority takes place in all groups of functions. Meanwhile, the first search, second search, and third search achieve the distinct best result in 2, 2, and 3 functions respectively. Overall, the performance gap between the first search and third search is narrow. A narrow performance gap also occurs between the second search and fourth search.

The third assessment is conducted to assess the proposed BCA to tackle a practical problem. The practical problem is also needed as it consists of constraints so that the practical problem can be seen as a constrained problem. In this work, the economic load dispatch (ELD) problem is chosen as the practical one.

The ELD problem is known as a popular optimization problem in power systems. The system consists of a set of power generating units which can be generators or power plants. Each generating unit produces power within its own power range (minimum and maximum powers) where this case can be seen as inequality constraint. Then, the power that is produced by each generating unit will be accumulated as total power. This total power should meet the power demand where in this case can be seen as an equality constraint. The operation of each generating unit produces cost where the produced power becomes variable. This operational cost function is presented as a quadratic equation in its basic form. The operational cost of each generating unit is then accumulated as total cost and it should be minimized.

Based on this general explanation, it can be summarized that as an optimization problem, ELD consists of three parts: the system, objective, and constraint. The system is a set of generating units. The objective is to minimize the total operational cost. There are two constraints: the equality constraint and inequality constraint. The inequality constraint is that the power of each generating unit should be within its power range. The equality constraint is that the total power should be equal to the power demand.

In this paper, the use case is a set of power grids that consists of 13 generating units. There are three scenarios for the power demand: 1,000 MW; 1,800 MW; and 2,600 MW. The detail description of the system including the power range and the constants of the cost function can be found in [44] and in [45]. The result is presented in Table 7.

Result in Table 7 shows the fierce confrontation among metaheuristics in solving the ELD problem. This fierce confrontation occurs in all scenarios, whether the demand is 1,000 MW; 1,800 MW; or 2,600 MW. This fierce confrontation can be seen from the total cost range between the best performer and the worst one.

The positioning of BCA among its confronters in this third assessment is as follows. In the first scenario, BCA is the best performer, but the result is the same as TIA, LEO, and WaOA. In the second scenario, BCA is in the fourth place after LEO, OOA, and WaOA. In the third scenario, BCA is in the second place after WaOA.

IV. Discussion

The result of benchmark assessment proofs that BCA has good both exploitation and exploration capabilities. Theoretically, the high dimension unimodal functions are designed to measure the exploitation capability due to its single optimal solution [37]. It means that any search should be deployed to find this global optimal solution as fast as possible.

The high dimension multimodal functions are designed to measure the exploration capability [37]. As it is known that multimodal functions consist of multiple optimal solutions, avoiding the local optimal solution becomes a priority. Once the searching process is entrapped in the local optimal region, there should be a mechanism to explore other regions to keep the possibility of a better optimal solution would be found.

The fixed dimension multimodal functions are designed to measure the balance between exploration and exploitation [37]. These functions are well-known for their ambiguity. In some functions, the terrain of the search space is flat with some narrow holes consisting of optimal solutions. The flat terrain makes the improvement becomes more difficult. Meanwhile, the narrow holes make the optimal solution more difficult to find. Short step means longer iteration while long step means jumping over the region consisting of the optimal solution. A small region of the optimal solution makes the agent easy to pull out from this region.

The result of the benchmark assessment also shows three other findings. The first finding is that the neighborhood search is not important. COA [15], LEO [33], OOA [17], and WaOA [18] are enriched with the neighborhood search with reduced search space during the iteration besides the directed search as primary strategy. This neighborhood search is first introduced in MPA [21]. Meanwhile, both BCA and TIA [39] are not enriched with this neighborhood search. While TIA has a single search which is the motion relative to all other agents [39], BCA performs only two strategies in four searches. The second finding is that splitting the swarm into sub-swarms and the existence of sub-swarm leaders improves the exploration capability. The first sub swarm leader is the global leader while second sub swarm leader is not always the second global leader. It becomes the alternative for COA [15], LEO [33], OOA [17], and WaOA [18] that choose only the global leader.

The result of individual search assessment indicates that the searching toward the best solution gives better results than the searching relative to other solution within swarm. The performance gap between these two searches is wide in the high dimension functions. Meanwhile, this performance gap is narrow in the fixed dimension functions. Moreover, searching toward the best solution produces the same result as searching relative to a randomly picked solution.

The result in the ELD problem shows that solving the constrained problems is more difficult than the unconstrained ones. In the constrained problem, the solution space is narrower because there is a collective boundary regarding the equality between the total output power and the demand power. It makes the circumstance where the power of a generating unit is not so independent to be set anywhere within its power range. The power of a generating unit reduces the power range of other generating units as a collective system.

TABLE VI
INDIVIDUAL SEARCH ASSESSMENT RESULT

F	Average Fitness Score			
	First Search	Second Search	Third Search	Fourth Search
1	2.5268×10^{1}	9.9516×10^{1}	**1.9103×10^{1}**	7.5497×10^{1}
2	**0.0000**	**0.0000**	**0.0000**	**0.0000**
3	**2.2577×10^{3}**	9.0606×10^{3}	2.5615×10^{3}	8.8716×10^{3}
4	2.9764	6.2356	**2.6427**	5.6459
5	3.8174×10^{2}	2.5243×10^{3}	**2.9830×10^{2}**	2.3748×10^{3}
6	2.8996×10^{1}	1.0298×10^{2}	**2.5858×10^{1}**	8.8772×10^{1}
7	**0.0438**	0.1005	0.0482	0.0962
8	-2.3049×10^{3}	-2.8655×10^{3}	-2.2616×10^{3}	**-2.9678×10^{3}**
9	7.9553×10^{1}	2.8210×10^{2}	**5.6839×10^{1}**	2.7359×10^{2}
10	2.1016	3.2243	**1.8270**	3.1980
11	1.1433	1.8289	**1.0770**	1.7351
12	1.2170	1.9098	**1.2169**	2.1463
13	4.7401	7.5917	**4.6668**	7.0547
14	1.1254×10^{1}	1.1839×10^{1}	1.3536×10^{1}	**7.8287**
15	0.0106	0.0133	**0.0008**	0.0170
16	-0.9763	**-1.0064**	-0.9734	-0.9957
17	1.0644	0.7259	**2.0747**	0.6859
18	2.9608×10^{1}	9.4076	2.3568×10^{1}	**7.5801**
19	**-0.0495**	**-0.0495**	**-0.0495**	**-0.0495**
20	-2.3506	**-2.4851**	-2.3888	-2.4173
21	-2.0084	-1.5282	**-2.0270**	-1.9980
22	-1.9698	-1.6343	**-2.2030**	-1.7648
23	-2.0354	-1.8388	**-2.2754**	-1.6421

TABLE VII
ASSESSMENT RESULT ON HANDLING ELD PROBLEM

Metaheuristic	Total Cost (USD/hour)		
	1000 MW	1800 MW	2600 MW
TIA	11,297	17,945	24,795
COA	11,298	17,947	24,789
LEO	11,297	17,938	24,783
OOA	11,298	17,939	24,786
WaOA	11,297	17,940	24,778
BCA	11,297	17,942	24,779

The computational complexity of BCA is presented as $O(n(A).d)$ during the initialization phase and $O(4t_m.n(A).d)$ during the iteration phase. There are two factors that make the computational complexity of BCA during the initialization and iteration different. First, BCA performs only a single search which is the full random search during initialization. On the other hand, there are four searches in BCA during the iteration. Second, there is not any loop until the maximum iteration in the initialization phase while in the iteration phase, the outer loop is the loop from the first iteration to maximum iteration.

Meanwhile, there are three limitations regarding this work. The first limitation is that BCA cannot accommodates all variances of swarm split whether they are quality-based swarm split or no quality-based swarm split. The second limitation is that BCA cannot accommodate all references and step size for the directed searches. The third limitation is that this work cannot accommodate all use cases for assessment whether they the unconstrained or constrained ones.

This presentation triggers some potentials for improvement, or in general, the future development of metaheuristics. The first potential is the splitting of the swarm into sub-swarms. As previously mentioned, almost all metaheuristics do not implement swarm splitting or segregation of roles. On the other hand, some metaheuristics, like in KMA [13] or COA [15] performs swarm splitting.

The quality-based swarm splitting may produce better performance by the reasoning that there should be different strategy between higher quality swarm members and lower quality swarm members. But quality-based swarm splitting brings consequence in adding complexity due to performing sorting in every iteration. Meanwhile, the swarm splitting without considering the quality is designed to accommodate more strategies without increasing the complexity.

The second potential is the construction of more powerful references for the directed search. This work has shown that a reference constructed from multiple leaders is better than a reference constructed from only one leader. But the scheme does not change along the iteration. In the future, constructing an iteration-considered reference is also challenging.

The third potential is the implementation of BCA in wider constrained practical problems. In the optimization problems in power system, there are several other cases, such as economic emission dispatch problem, unit commitment problem, optimal power flow problem, and so on, with many other considerations, such as power loss can also be added to enrich the decision-making process.

These constrained practical problems are not limited to the problems in the power system. There are a lot of problems in manufacturing system that ranges from selecting suppliers, deciding the products, allocating resources, whether machines or people, to the storage system. Expanding to the logistic system, the problems range from fleet management, warehouse or inventory, and so on. Some of these problems are numerical problems while the others are combinatorial ones.

V. CONCLUSION

A new stochastic optimization called best couple algorithm (BCA) has been presented in this paper. The swarm-splitting into two equal size sub swarms becomes the fundamental concept of BCA. This paper has also presented that there are two splitting strategies in BCA. Then, the highest quality sub swarm members become the sub swarm leader. The searching process is then performed by moving toward the middle between two sub swarm leaders and moving relative to the middle between two randomly picked sub swarm members. The set consisting of 23 functions is employed representing the problems in the benchmark assessment where BCA is challenged with five metaheuristics: TIA, COA, LEO, OOA, and WaOA. The result shows that BCA is better than TIA, COA, LEO, OOA, and WaOA respectively. Meanwhile, the individual search assessment result indicates that the motion toward the middle of two sub-swarm leaders performs better than the motion relative to the middle between two randomly picked sub swarm members. The third assessment exhibits the tight competition among metaheuristics in handling ELD problem.

The potential for future studies can be traced back to the superiority of the reference consisting of the middle between two sub swarm leaders. It means that constructing new reference for the directed search becomes the window of opportunity for proposing a new metaheuristic. Besides the swarm slitting mechanism can also be used for further improvement. Moreover, utilizing BCA to solve many kinds of practical problems can be used for more comprehensive assessment.

REFERENCES

[1] Y. Hou, C. Wang, W. Dong, and L. Dang, "An Improved Particle Swarm Optimization Algorithm for the Distribution of Fresh Products," *Engineering Letters*, vol. 31, no. 2, pp. 494–503, 2023.

[2] A. Bassel, M. Jameel, and M. A. Saad, "Jumping particle swarm optimization algorithm framework for content-based image retrieval system," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 6, pp. 3665–3676, Dec. 2023, doi: 10.11591/eei.v12i6.5024.

[3] G. Chen, J. Li, Y. Xu, B. Peng, H. Tan, and H. Long, "Optimal Configuration of Renewable Energy DGs Based on Improved Northern Goshawk Optimization Algorithm Considering Load and Generation Uncertainties," *Engineering Letters*, vol. 31, no. 2, pp. 511–530, 2023.

[4] J. Ren, "A Reliable Night Vision Image De-Noising Based on Optimized ACO-ICA Algorithm," *IAENG Int J Comput Sci*, vol. 51, no. 3, pp. 169–177, 2024.

[5] B. Jyoti Saharia and N. Sarmah, "Grey wolf optimizer for the design optimization of a DC-DC boost converter," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 6, pp. 3263–3270, Dec. 2023, doi: 10.11591/eei.v12i6.5846.

[6] Z. Xu and X. Zhang, "An Improve Grey Wolf Optimizer Algorithm for Traveling Salesman Problems," *IAENG Int J Comput Sci*, vol. 51, no. 6, pp. 602–612, 2024.

[7] M. S. Wankhade and S. V. Kottur, "HBSCPG: Design of a Hybrid Bioinspired Model for Optimization of existing Security & Control Parameters of Cyber-Physical Smart Grids," *IAENG Int J Comput Sci*, vol. 51, no. 4, pp. 355–366, 2024.

[8] Y. Hou, C. Wang, C. Zhang, L. Dang, and C. Xiao, "A Hybrid Max-Min Ant System Algorithm for Electric Capacitated Vehicle Routing Problem," *IAENG Int J Comput Sci*, vol. 51, no. 3, pp. 195–203, 2024.

[9] H. Bouali, B. Benhala, and M. Guerbaoui, "Multi-objective optimization of CMOS low noise amplifier through nature-inspired swarm intelligence," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 5, pp. 2824–2836, Oct. 2023, doi: 10.11591/eei.v12i5.5512.

[10] Fahad. A. Alghamdi, A. Younes Hamed, A. M. Alghamdi, A. Ben Salah, T. H. Farag, and W. Hassan, "A genetic algorithm for shortest

path with real constraints in computer networks," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 1, p. 435, Feb. 2023, doi: 10.11591/ijece.v13i1.pp435-442.

[11] B. A. Benali, S. Mihi, I. El Bazi, and N. Laachfoubi, "New approach for Arabic named entity recognition on social media based on feature selection using genetic algorithm," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 2, pp. 1485–1497, Apr. 2021, doi: 10.11591/ijece.v11i2.pp1485-1497.

[12] W. Saber, W. Moussa, A. M. Ghuniem, and R. Rizk, "Hybrid load balance based on genetic algorithm in cloud environment," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 3, pp. 2477–2489, Jun. 2021, doi: 10.11591/ijece.v11i3.pp2477-2489.

[13] S. Suyanto, A. A. Ariyanto, and A. F. Ariyanto, "Komodo Mlipir Algorithm," *Appl Soft Comput*, vol. 114, pp. 1–17, Jan. 2022, doi: 10.1016/j.asoc.2021.108043.

[14] P. D. Kusuma and M. Kallista, "Stochastic Komodo Algorithm," *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 4, pp. 156–166, Aug. 2022, doi: 10.22266/ijies2022.0831.15.

[15] M. Dehghani, Z. Montazeri, E. Trojovská, and P. Trojovský, "Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems," *Knowl Based Syst*, vol. 259, p. 110011, Jan. 2023, doi: 10.1016/j.knosys.2022.110011.

[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, doi: 10.1109/ACCESS.2022.3172789.

[17] M. Dehghani and P. Trojovský, "Osprey optimization algorithm: A new bio-inspired metaheuristic algorithm for solving engineering optimization problems," *Front Mech Eng*, vol. 8, p. 1126450, Jan. 2023, doi: 10.3389/fmech.2022.1126450.

[18] P. Trojovský and M. Dehghani, "A new bio-inspired metaheuristic algorithm for solving optimization problems based on walruses behavior," *Sci Rep*, vol. 13, no. 1, p. 8775, May 2023, doi: 10.1038/s41598-023-35863-5.

[19] N. Rana, M. S. A. Latiff, S. M. Abdulhamid, and H. Chiroma, "Whale optimization algorithm: a systematic review of contemporary applications, modifications and developments," *Neural Comput Appl*, vol. 32, no. 20, pp. 16245–16277, Oct. 2020, doi: 10.1007/s00521-020-04849-z.

[20] N. Chopra and M. Mohsin Ansari, "Golden jackal optimization: A novel nature-inspired optimizer for engineering applications," *Expert Syst Appl*, vol. 198, p. 116924, Jul. 2022, doi: 10.1016/j.eswa.2022.116924.

[21] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine Predators Algorithm: A nature-inspired metaheuristic," *Expert Syst Appl*, vol. 152, p. 113377, Aug. 2020, doi: 10.1016/j.eswa.2020.113377.

[22] E. Trojovska and M. Dehghani, "Clouded Leopard Optimization: A New Nature-Inspired Optimization Algorithm," *IEEE Access*, vol. 10, pp. 102876–102906, 2022, doi: 10.1109/ACCESS.2022.3208700.

[23] 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, doi: 10.1109/ACCESS.2022.3229964.

[24] P. Trojovský and M. Dehghani, "Pelican Optimization Algorithm: A Novel Nature-Inspired Algorithm for Engineering Applications," *Sensors*, vol. 22, no. 3, pp. 1–34, Jan. 2022, doi: 10.3390/s22030855.

[25] 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, doi: 10.1109/ACCESS.2021.3133286.

[26] 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 Syst Appl*, vol. 191, p. 116158, Apr. 2022, doi: 10.1016/j.eswa.2021.116158.

[27] M. Khishe and M. R. Mosavi, "Chimp optimization algorithm," *Expert Syst Appl*, vol. 149, p. 113338, Jul. 2020, doi: 10.1016/j.eswa.2020.113338.

[28] M. Suman, V. Sakthivel, and P. 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, Oct. 2020, doi: 10.22266/ijies2020.1031.11.

[29] M. Dehghani, P. Trojovský, and O. P. Malik, "Green Anaconda Optimization: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems," *Biomimetics*, vol. 8, no. 1, p. 121, Mar. 2023, doi: 10.3390/biomimetics8010121.

[30] 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.

[31] D. Acharya and D. K. Das, "A novel Human Conception Optimizer for solving optimization problems," *Sci Rep*, vol. 12, no. 1, p. 21631, Dec. 2022, doi: 10.1038/s41598-022-25031-6.

[32] P. Trojovsky and M. Dehghani, "Migration Algorithm: A New Human-Based Metaheuristic Approach for Solving Optimization Problems," *Computer Modeling in Engineering & Sciences*, vol. 0, no. 0, pp. 1–36, 2023, doi: 10.32604/cmes.2023.028314.

[33] P. Trojovsky, M. Dehghani, E. Trojovska, and E. Milkova, "Language Education Optimization: A New Human-Based Metaheuristic Algorithm for Solving Optimization Problems," *Computer Modeling in Engineering & Sciences*, vol. 136, no. 2, pp. 1527–1573, 2023, doi: 10.32604/cmes.2023.025908.

[34] E. Trojovská and M. Dehghani, "A new human-based metahurestic optimization method based on mimicking cooking training," *Sci Rep*, vol. 12, no. 1, p. 14861, Sep. 2022, doi: 10.1038/s41598-022-19313-2.

[35] P. Trojovský and M. Dehghani, "A new optimization algorithm based on mimicking the voting process for leader selection," *PeerJ Comput Sci*, vol. 8, p. e976, May 2022, doi: 10.7717/peerj-cs.976.

[36] E. Trojovská, M. Dehghani, and V. Leiva, "Drawer Algorithm: A New Metaheuristic Approach for Solving Optimization Problems in Engineering," *Biomimetics*, vol. 8, no. 2, p. 239, Jun. 2023, doi: 10.3390/biomimetics8020239.

[37] I. Matoušová, P. Trojovský, M. Dehghani, E. Trojovská, and J. Kostra, "Mother optimization algorithm: a new human-based metaheuristic approach for solving engineering optimization," *Sci Rep*, vol. 13, no. 1, p. 10312, Jun. 2023, doi: 10.1038/s41598-023-37537-8.

[38] S. E. Khouni and T. Menacer, "Nizar optimization algorithm: a novel metaheuristic algorithm for global optimization and engineering applications," *J Supercomput*, Sep. 2023, doi: 10.1007/s11227-023-05579-4.

[39] 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, Feb. 2023, doi: 10.22266/ijies2023.0228.20.

[40] M. Dehghani, Š. Hubálovský, and P. Trojovský, "A new optimization algorithm based on average and subtraction of the best and worst members of the population for solving various optimization problems," *PeerJ Comput Sci*, vol. 8, pp. 1–40, Mar. 2022, doi: 10.7717/peerj-cs.910.

[41] 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, doi: 10.1109/ACCESS.2022.3162853.

[42] 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, Jun. 2023, doi: 10.22266/ijies2023.0630.19.

[43] M. Ghasemi *et al.*, "A new metaphor-less simple algorithm based on Rao algorithms: a Fully Informed Search Algorithm (FISA)," *PeerJ Comput Sci*, vol. 9, p. e1431, Aug. 2023, doi: 10.7717/peerj-cs.1431.

[44] H. Zein, J. Raharjo, and I. R. Mardiyanto, "A Method for Completing Economic Load Dispatch Using the Technique of Narrowing Down Area," *IEEE Access*, vol. 10, pp. 30822–30831, 2022, doi: 10.1109/ACCESS.2022.3158928.

[45] P. D. Kusuma and A. L. Prasasti, "Stochastic Shaking Algorithm: A New Swarm-Based Metaheuristic and Its Implementation in Economic Load Dispatch Problem," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 3, pp. 276–289, Jun. 2024, doi: 10.22266/ijies2024.0630.23.

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