Iteration Controlled Mixture Optimizer: A New Metaheuristic and Its Application to Solve Economic Load Dispatch Problem

Purba Daru Kusuma, *Member, IAENG*

***Abstract*—Metaheuristic has been utilized extensively to optimize power system. Meanwhile, the no-free-lunch (NFL) theory becomes the major consideration in the massive development of metaheuristic as there is not any ideal metaheuristic can solve all problems superiorly. Based on this problem, this work is aimed at introducing a novel metaphor-free swarm-based metaheuristic called iteration-controlled mixture optimizer (ICMO). ICMO contains three directed searches where the reference in each search is constructed by two entities. The first reference is the mixture between the finest entity and the mean of the finer entities. The second reference is the mixture between the finest entity and a randomly chosen entity. The third reference is the mixture between the finest entity and any generated entity within space. The portion between the first and second entities in each reference is controlled by the iteration. Then, ICMO is compared with five new swarm-based metaheuristics: attack leave optimization (ALO), total interaction algorithm (TIA), fully informed search algorithm (FISA), walrus optimization algorithm (WaOA), and ono-to-one based optimization (OOBO). The assessment result shows that ICMO is better than ALO, TIA, FISA, WaOA, and OOBO in 15, 13, 20, 12, and 20 functions out of 23 functions respectively. Then, ICMO is also challenged to solve the economic load dispatch (ELD) problem in the Java-Bali electricity system in Indonesia. The result shows that ICMO is competitive compared to these five metaheuristics in solving this practical problem. The result shows that the range between the best and worst metaheuristics in this problem is narrow as it represents the integer-based problem.**

***Index Terms*—economic load dispatch problem, Java-Bali electricity system, metaheuristic, swarm intelligence.**

I. Introduction

METAHEURISTIC has been extensively used in various optimization studies in power systems. There are two ways on utilizing these metaheuristics. The first one is utilizing them in their basic form. The second one is through modification. For example, various studies regarding optimal power flow (OPF) problem, where the main objective is minimizing power loss or reducing fuel cost, utilized artificial fish swarm algorithm [1], honey badger algorithm [2], hybrid crow search algorithm [3], jellyfish search algorithm [4], and so on. Crow search algorithm has been utilized to overcome the optimal reactive power dispatch (OPRD) problem [5], which is the subsequent of OPF. Meanwhile, several studies on economic load dispatch (ELD), where the objective is minimizing fuel cost, utilized bat algorithm [6], particle swarm optimization [7], Harris hawk optimizer [8], and so on. The studies on economic emission dispatch problem, in which the problem has multiple objectives in reducing the fuel cost and emission, have utilized several metaheuristics, such as simulated annealing [9], grasshopper algorithm [10], and so on.

In recent years, there are a lot of metaheuristics have been introduced. Many of them were developed based on swarm intelligence. Many of them employ the animal metaphor as inspiration, such as jellyfish search (JS) [11], honey badger algorithm (HBA) [12], pelican optimization algorithm (POA) [13], coati optimization algorithm (COA) [14], northern goshawk optimization (NGO) [15], osprey optimization algorithm (OOA) [16], walrus optimization algorithm (WaOA) [17], green anaconda optimization (GAO) [18], golden jackal optimization (GJO) [19], marine predator algorithm (MPA) [20], Komodo mlipir algorithm (KMA) [21], lyrebird optimization algorithm (LOA) [22], kookaburra optimization algorithm (KOA) [23], red fox optimization algorithm (RFO) [24], squirrel search optimization (SSO) [25], sparrow search algorithm (SSA) [26], white shark optimization (WSO) [27], graylag goose optimization (GGO) [28], and so on. There is critique regarding the use of metaphor to conceal the true novelty of the metaheuristic. Meanwhile, some metaheuristics do not use any metaphors, such as attack leave optimization (ALO) [29], total interaction algorithm (TIA) [30], mixed leader-based optimization (MLBO) [31], average subtraction-based optimization (ASBO) [32], one-to-one based optimization (OOBO) [33], quadratic interpolation algorithm (QIO) [34], geometric mean optimizer (GMO) [35], and so on.

Unfortunately, the use of these new metaheuristics to solve optimization problems in power systems is rare to find compared to the older ones. Many of these studies still used classic algorithms. On the other hand, many recent metaheuristics as presented in the previous paragraph are under-utilized to solve these problems although some of the metaheuristics have been used.

In some metaheuristics, the choice of strategy is also affected by the iteration. In this case, the early iteration is dominated by exploration while the later one is dominated by

Manuscript received January 17, 2024; revised April 29, 2024. This work was financially supported by Telkom University, Indonesia.

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

exploitation. There are various interpretations regarding this approach. Simulated annealing (SA) uses the iteration to determine whether the worse solution will be accepted to replace the current solution [36]. In MPA, the iteration represents the shifting strategy from the Brownian movement to the Levy flight through discrete split along the iteration [20]. The modification of MPA called the stochastic marine predator algorithm with multiple candidates (SMPA-MC) changes the deterministic split into stochastic manner [37]. Many metaheuristics associated with Dehghani implements the neighborhood search with declining local search space as the iteration goes as secondary search as it can be found in POA [13], zebra optimization algorithm (ZOA) [38], COA [14], NGO [15], WaOA [17], OOA [16], and so on. In MLBO [31], iteration is used to control the mixture between the finest entity and a randomly selected entity to construct the reference in the directed search during the first half of iteration. This short summary shows that there are a lot of approaches that have been used to utilize the iteration as a strategy controller. On the other hand, there is a lot of potential to construct a new approach.

Regarding this problem, this paper is aimed at proposing a new metaphor-free swarm-based metaheuristic called iteration-controlled mixture optimizer (ICMO). The two terms used for its name become the fundamental concept of ICMO. The term “mixture” refers to the mixture of two entities to construct the reference for the directed search. On the other hand, the term “controlled” refers to the iteration used to control the portion of each entity in constructing the reference. ICMO is then applied to tackle both theoretical and real-world problems. The ELD problem is chosen as the real-world problem.

The main scientific contributions of this work are listed as follows.

1) This work introduces a new metaphor free swarm-based metaheuristic called ICMO where the iteration affects the searching process by controlling the mixture of entities used as reference.
2) This paper presents the formal model of ICMO, including the fundamental concept, pseudocode, and mathematical formulation following the algorithm.
3) The performance assessment of ICMO is conducted by challenging it to solve both theoretical and practical optimization problems.
4) The set of 23 mathematical functions is chosen as the theoretical problem while the ELD problem in Java-Bali electricity system is chosen as the practical problem.
5) The performance of ICMO is confronted with five recent swarm-based metaheuristics.

The structure of the rest of this paper is as follows. Section two reviews the development of swarm-based metaheuristic and presents the comparison of several recent swarm-based metaheuristics. Section three formulates the model of ICMO including the fundamental concept and formalization. Section four presents the ELD problem. Section five presents the simulation to assess the performance of ICMO and the result. Section six discusses the findings regarding the assessment result, complexity, and limitations of this work. Section seven presents the concluding remarks and the summary of future work potential.

II. Related Works

Swarm intelligence has been used for the baseline for the development of many recent metaheuristics. Swarm intelligence is a population-based metaheuristic that contains certain number of autonomous agents. Each agent traces for a better solution actively independently without any central coordination. There is a certain pattern of the agent’s movement because of the influence of the collective intelligence that is shared among agents. Moreover, there is interaction among agents that affects the movement. This interaction can be seen as a directed search where the agent moves based on a certain target or reference within the certain randomized speed.

There are various entities that are chosen as reference. The finest entity becomes the most popular reference like in COA [14] or ZOA [38]. In metaheuristics, such as POA [13] or COA [14], a randomly generated solution within the space is also utilized as a reference. A reference can also be a randomly chosen entity within swarm, such as in TIA [30], WaOA [17], and so on. In some metaheuristics like ALO [29], the reference can be a mixture between the finest entity and a randomly chosen entity. This review means that there are many ways to construct a reference for the directed search and it can be used to introduce a new swarm-based metaheuristic.

In metaheuristic, the improvement is performed through iteration or iterative process. This approach is the consequence of trial-and-error strategy which is adopted by metaheuristic. It means that the optimal solution may not be obtained in a single trial. The quality of solution may be not good in the first trial but through improvement performed in every iteration, the quality of solution may be acceptable in the end of iteration or in other words when the maximum iteration is reached. The term acceptable means that the global optimal solution may be not found but only the quasi-optimal one [38]. The maximum iteration is predetermined to limit the number of iterations that will be performed during the optimization process.

The summarized review of several recent swarm-based metaheuristics is presented in Table 1. The presented aspects include the metaphor, role of the iteration, and the use cases for the performance assessment.

Table 1 shows that some metaheuristics utilizes the iteration as a counter only while some others utilize the iteration not only as a counter but also to be involved in the searching process. There are several utilizations of the iteration to be involved in the searching process. It means that there is still potential to use iteration in other manners. Meanwhile, the use of optimization problems in power systems, such as ELD is still less popular, especially compared to the classic engineering design problems in studies introducing new metaheuristic. Based on this opportunity, this paper introduces a new metaheuristic that utilizes the iteration not only as a counter but also to determine the portion of entities that construct the reference during the directed search.

TABLE I
LIST OF SOME RECENT SWARM METAHEURISTICS

No	Metaheuristic	Metaphor	Role of the Iteration	Assessment Use Case
1	ALO [29]	-	counter	23 classic functions
2	TIA [30]	-	counter	23 classic functions
3	FISA [39]	-	counter	CEC 2014, pressure vessel design, spring design, welded beam design,
4	WaOA [17]	walrus	counter, reducing the local search space in the neighborhood search	23 classic functions, CEC 2015, CEC 2017
5	OOBO [33]	-	counter	23 classic functions, CEC 2017
6	ZOA [38]	zebra	counter, reducing the local search space in the neighborhood search	23 classic functions, CEC 2015, CEC 2017, welded beam design, pressure vessel design, spring design, speed reducer
7	SSO [25]	squirrel	counter	ELD problem
8	GJO [19]	golden jackal	counter, determining strategy whether two finest entities move toward or avoid the corresponding entity	23 classic functions, welded beam design, pressure vessel design, spring design, speed reducer, three bar truss design, ELD
9	KOA [23]	kookaburra	counter, reducing the local search space in the neighborhood search	CEC 2017, CEC 2011, pressure vessel design, speed reducer design, welded beam design, spring design
10	WSO [27]	white shark	counter, determining the speed toward the prey	CEC 2017, CEC 2011
11	this work	-	counter, determining the portion of the first and second entities that constructs the reference	23 classic functions, ELD

III. PROPOSED MODEL

The fundamental concept of ICMO can be traced back to two terms used as its name: iteration and mixture. The term iteration means that the iteration is involved not only for the stepping for improvement but also in controlling the searching process. The mixture means that there are some entities that will be mixed to construct a new entity or reference. Based on this explanation, ICMO is built as a multi-search-multi-phase metaheuristic. It means that there are several searches conducted by each entity in every iteration. Then, the multi-phase approach means that multiple searches are performed sequentially.

There are three searches employed in every iteration. All these three searches are directed searches where a reference is needed for the guidance. The reference is constructed from two entities. The main entity in every reference is the finest entity within swarm. The finest entity is chosen as its role is crucial in any swarm-based metaheuristic. In the first search, the reference is the mixture between the finest entity and the mean of finer entities. This second entity is obtained by calculating the mean of all finer entities relative to the corresponding entity plus the finest entity. First, all finer entities and the finest entity are collected into a pool. Then, the mean value is obtained by finding the average value of all entities within the pool. The first search is designed for intensification. The second reference is the mixture between the finest entity and a randomly chosen entity. The third reference is the mixture between the finest entity and a randomly generated entity within space. The second and third references are designed to improve diversification where the third reference has higher diversification degree than the second one.

The visualization of these three references is presented in Fig. 1. Fig. 1a represents the first directed search. Fig. 1b represents the second directed search. Fig. 1c represents the third directed search. The pink circle represents the entities within swarm. The blue circle represents the corresponding entity. The red circle represents the finest entity. The light green circle represents the finer entities. The dark green circle represents the mean or resultant of finer entities. The yellow circle represents a randomly generated entity within swarm. The grey circle represents the reference.

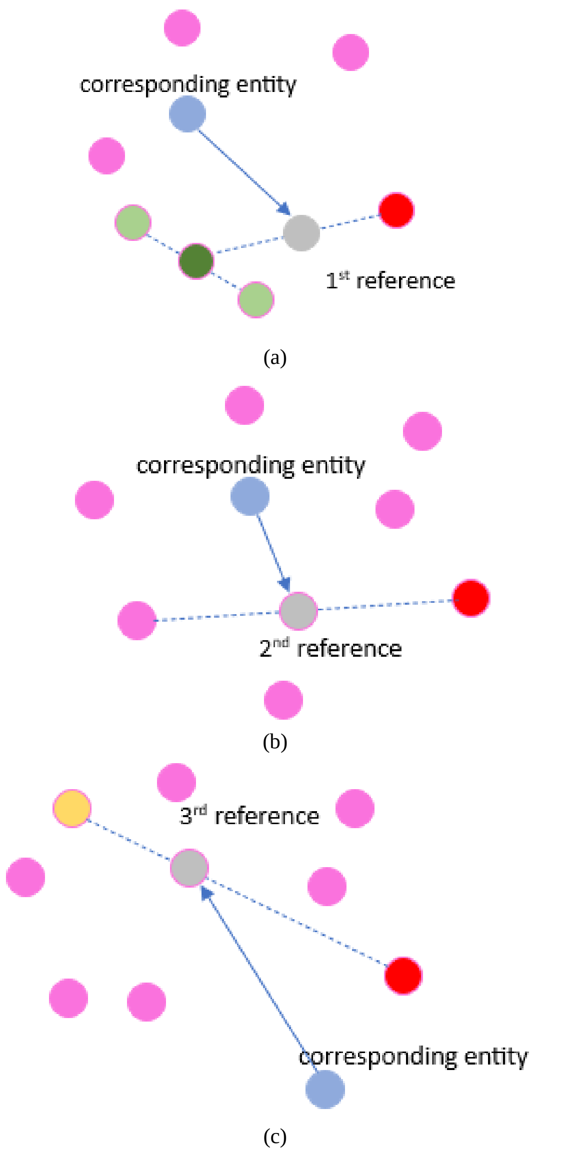
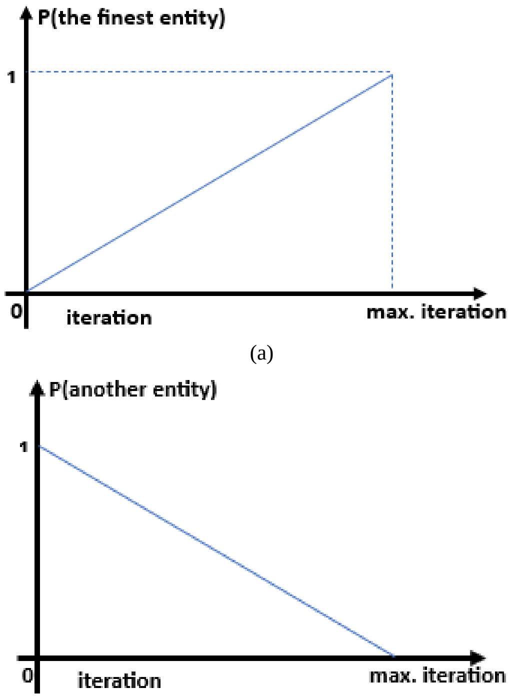


Fig. 1. Visualization of three searches in ICMO: (a) first search, (b) second search, and (c) third search

The portion between the finest entity as the first entity and another entity as the second entity is controlled by the iteration. In early iteration, the second entity is more dominant while in the later iteration, the first entity is more dominant. The dominance of the finest entity increases linearly as the iteration goes. It means that the dominance of the second entity decreases as the iteration goes on. The illustration of the change of portion between the first entity and the second entity is presented in Fig. 2.



(b)

Fig. 2. Portion change during the iteration: (a) the first entity or the finest entity, (b) the second entity or another entity

As a metaheuristic, ICMO consists of two stages. The first stage is initialization while the second stage is iteration. In the initialization, all entities within swarm are uniformly generated within the space. It means that the probability of a certain location within space chosen for initial entity is equal. The motivation is related to the abstraction of the problem. It means that in the beginning, there is not any clue regarding the location of global optimal solution. Then, each time an entity is generated, its value will be sent for the finest entity replacement. The value of the finest entity will be replaced only if the proposed value is better than the current value of the finest entity. Meanwhile, during the iteration, each entity will perform these three directed searches in every iteration until the stopping criteria is met. There are two stopping criteria. The first is the maximum iteration is reached. The second is when there is not any improvement achieved after several iterations. The first case becomes more common stopping criteria. Each search generates a solution candidate for the entity. This solution candidate will replace the current value of the entity only if the candidate is better than the current value of the entity.

The formalization of ICMO based on this fundamental concept is presented in algorithm 1. Meanwhile, the mathematical formulations that are used to describe the process in a more detailed manner are presented in (1) to (14). The annotations used in this proposed model are as follows:

e	entity
E	set of entities (swarm)
e_{fst}	the finest entity
e_{fi}	finer entity
E_{fi}	set of finer entities
e_{sel}	randomly selected entity
e_{ran}	a randomized entity within space
e_{lo}	lower boundary
e_{hi}	higher boundary
e_{r1}	the first reference
e_{r2}	the second reference
e_{r3}	the third reference
e_{s1}	the first solution candidate
e_{s2}	the second solution candidate
e_{s3}	the third solution candidate
f	objective function
t	iteration
t_m	maximum iteration
u	uniform random
ρ_1	uniform floating point random number [0,1]
ρ_2	uniform integer random number [1,2]

algorithm 1: iteration-controlled mixture optimization

1	**begin**
2	**for all** $e \in E$
3	initialize e_i using (2)
4	update e_{fst} using (3)
5	**end**
6	**for** t=1 **to** t_m
7	define μ using (4)
8	**for all** e **in** E
9	create E_{fi} using (5)
10	perform first search using (6) to (8)
11	update e_{fst} using (3)
12	perform second search using (9) to (11)
13	update e_{fst} using (3)
14	perform third search using (12) to (14)
15	update e_{fst} using (3)
16	**end for**
17	**end for**
18	**return** e_{fst}
19	**end**

Algorithm 1 can be split into two parts. The first part is the initialization which is presented from lines 2 to 5. Then, the second part is the iteration which is presented from lines 6 to 16. In the end, the finest solution becomes the final solution and the output of algorithm as presented in line 17. The generalization of the swarm is formalized using (1).

$$E = \{e_1, e_2, e_3, \ldots, e_n\} \tag{1}$$

The initialization includes two processes. The first process is generating all entities uniformly within the space as formulated in (2). Then, this process is followed by the updating of the finest entity using (3).

$$e_i = U(e_{lo}, e_{hi}) \tag{2}$$

$$e'_{fst} = \begin{cases} e_i, f(e_i) < f(e_{fst}) \\ e_{fst}, otherwise \end{cases} \quad (3)$$

The first process in the loop related to the iteration is defining the portion factor μ. This process is formulated using (4). Due to (4), it is shown that the portion value increases as the iteration goes.

$$\mu = \frac{t}{t_m} \quad (4)$$

The first process within the loop for whole swarm is defining the finer entity pool. It is formulated using (5). Due to (5) this pool consists of all finer entities related to the corresponding entity plus the finest entity. It means that this pool never becomes an empty collection due to the union of the finest entity.

$$E_{fi,i} = \{e \in E, f(e) < f(e_i)\} \cup e_{fst} \quad (5)$$

In general, each search consists of three processes. The first process is defining the reference. The second process is generating the solution candidate. The third process is updating the corresponding entity based on its solution candidate. The first search is formulated using (6) to (8). The second search is formulated using (9) to (11). The third search is formulated using (12) to (14). Each time a search is performed, then the finest entity will be updated.

$$e_{r1,i} = \frac{\mu e_{fst} + (1-\mu)\frac{\sum_{E_{fi,i}} e_{fi,i}}{n(E_{fi,i})}}{2} \quad (6)$$

$$e_{s1,i} = e_i + \rho_1(e_{r1,i} - \rho_2 e_i) \quad (7)$$

$$e'_i = \begin{cases} e_{s1,i}, f(e_{s1,i}) < f(e_i) \\ e_i, otherwise \end{cases} \quad (8)$$

$$e_{r2,i} = \frac{\mu e_{fst} + (1-\mu)U(E)}{2} \quad (9)$$

$$e_{s2,i} = e_i + \rho_1(e_{r2,i} - \rho_2 e_i) \quad (10)$$

$$e'_i = \begin{cases} e_{s2,i}, f(e_{s2,i}) < f(e_i) \\ e_i, otherwise \end{cases} \quad (11)$$

$$e_{r3,i} = \frac{\mu e_{fst} + (1-\mu)U(e_{lo}, e_{hi})}{2} \quad (12)$$

$$e_{s3,i} = e_i + \rho_1(e_{r3,i} - \rho_2 e_i) \quad (13)$$

$$e'_i = \begin{cases} e_{s3,i}, f(e_{s3,i}) < f(e_i) \\ e_i, otherwise \end{cases} \quad (14)$$

IV. ECONOMIC LOAD DISPATCH

In general, ELD can be formulated as a certain number of generators in the power system. Each generator produces certain power within its minimum and maximum power. Meanwhile, the ramp rate limits the difference between the power produced in the next hour compared to the current hour. As a power system, the total power is obtained by accumulating the power produced by all generators. This power system should meet the demand or requested power whereby neglecting the power loss, then total power produced by all generators is equal to the demand. The objective of ELD is minimizing the fuel cost produced by the generator. The fuel cost of each generator is formulated by quadratic equation. Several annotations used in ELD are as follows.

x_j	power provided by generator j
x_{min}	minimum power
x_{max}	maximum power
X	set of generators
x_d	demand or requested power
x_r	ramp rate
x_t	total power
c_t	total cost
c_j	fuel cost produced by generator j
α, β, γ	constants in cost function
h	hour

The mathematical formulation of ELD is presented in (15) to (22). Equation (15) defines that the system consists of m generators. Equation (16) states that the objective is minimizing total fuel cost. Equation (17) states that the total fuel cost is accumulated from fuel cost produced by all generators. Equation (18) formulates the fuel cost function of each generator as a quadratic function with three constants. These constants may be different among the generators. The power produced by each generator should be within its minimum and maximum power as stated in (19). Meanwhile, (20) states that total power is accumulated from the power produced by all generators. Then, the total power should be exactly matched with the demand as stated in (21). Finally, (22) states that the power difference between adjacent periods is limited to the ramp rate.

$$X = \{x_1, x_2, x_3, \ldots, x_m\} \quad (15)$$

$$objective: min(c_t) \quad (16)$$

$$c_t = \sum_{j=1}^{m} c_i \quad (17)$$

$$c_j = \alpha_j x_j^2 + \beta_j x_j + \gamma_j \quad (18)$$

$$x_{min,j} \le x_j \le x_{max,j} \quad (19)$$

$$x_t = \sum_{j=1}^{m} x_j \quad (20)$$

$$x_t = x_d \quad (21)$$

$$|x_{j,h} - x_{j,h-1}| \le x_{r,i} \quad (22)$$

V. SIMULATION

The performance evaluation for ICMO is conducted in three computational simulations. In the first simulation, ICMO is challenged to solve 23 mathematical functions. These functions represent theoretical problems. They can be split into seven high dimension unimodal functions (HDUF), six high dimension multimodal functions (HDMF), and ten fixed dimension multimodal functions (FDMF). A comprehensive description of these 23 mathematical functions is presented in Table 2. In the second simulation, the convergence assessment of ICMO is conducted. In the third simulation, ICMO is challenged to solve the real-world ELD problem which represents the real-world optimization problem. In this work, the Java-Bali electricity system is chosen as the use case.

In this evaluation, ICMO is compared with five other metaheuristics. All these five comparators are new as they were first introduced in 2023. These comparators are ALO [29], TIA [30], FISA [39], WaOA [17], and OOBO [33]. All of them are swarm-based metaheuristics. In both evaluations, the swarm size is 10 while the maximum iteration is 20. This setup represents low swarm size and low maximum iteration. It is different from many studies proposing new metaheuristics where the computational assessment is performed in high swarm size and high maximum iteration setup.

The result of the first evaluation is presented in Table 3 to Table 5. Table 3 to Table 5 present the result of the assessment evaluation of HDUF, HDMF, and FDMF consecutively. There are three parameters in Tables 3 to 5: average fitness score or mean, standard deviation, and the mean rank. Then, the result is summarized in Table 6 based on the group of functions for further investigation of the superiority of ICMO. The decimal point less than 10^{-4} is rounded to 0.

Table 3 shows the superiority of ICMO on solving the HDUFs. ICMO becomes the best performer in four functions (f_1, f_2, f_4, and f_7). Moreover, ICMO can find the global optimal solution of f_1 and f_2. Then, ICMO becomes the second best on one function (f_3) and the third best on two functions (f_5 and f_6).

There are several notes regarding the result presented in Table 3. All metaheuristics that are involved in this assessment perform similarly on solving f_2. ALO becomes the second-best performer by achieving the first best on three functions (f_1, f_2, and f_3). The performance gap between the first best and the second best is wide except on f_2.

TABLE II
DETAIL DESCRIPTION OF 23 CLASSIC FUNCTIONS

No	Function	Dimension	Problem Space	Global Opt.
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]	-418.9x50
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 III
BENCHMARK SIMULATION RESULT ON SOLVING HDUFs

F	Parameter	ALO [29]	TIA [30]	FISA [39]	WaOA [17]	OOBO [33]	ICMO
1	mean	0.0000	0.0005	4.9457×10^4	0.0007	1.2948×10^2	0.0000
	standard deviation	0.0000	0.0001	1.1558×10^4	0.0006	5.8729×10^1	0.0000
	mean rank	1	3	6	4	5	1
2	mean	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	standard deviation	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	mean rank	1	1	1	1	1	1
3	mean	0.1705	5.1854	1.1083×10^5	2.7692×10^1	2.6892×10^4	0.2961
	standard deviation	0.5435	4.8942	3.0781×10^4	2.2791×10^1	1.3114×10^4	0.4421
	mean rank	1	3	6	4	5	2
4	mean	0.0035	0.0390	8.5156×10^1	0.0238	1.2171×10^1	0.0001
	standard deviation	0.0054	0.0111	1.0882×10^1	0.0089	3.0731	0.0000
	mean rank	2	4	6	3	5	1
5	mean	4.8970×10^1	4.8788×10^1	1.5178×10^8	4.8915×10^1	3.9516×10^3	4.8922×10^1
	standard deviation	0.0193	0.0525	1.3414×10^8	0.0394	4.3182×10^3	0.0190
	mean rank	4	1	6	2	5	3
6	mean	1.0803×10^1	7.5092	4.8022×10^4	9.6236	1.4402×10^2	1.0338×10^1
	standard deviation	0.5941	0.5600	9.9922×10^3	0.4441	7.8826×10^1	0.4900
	mean rank	4	1	6	2	5	3
7	mean	0.0223	0.0127	1.2059×10^2	0.0153	0.0917	0.0058
	standard deviation	0.0143	0.0089	1.1604×10^2	0.0080	0.0415	0.0043
	mean rank	4	2	6	3	5	1

TABLE IV
BENCHMARK SIMULATION RESULT ON SOLVING HDMFS

F	Parameter	ALO [29]	TIA [30]	FISA [39]	WaOA [17]	OOBO [33]	ICMO
8	mean	-4.1365×10^3	-2.4262×10^3	-3.3559×10^3	-3.9876×10^3	-3.3808×10^3	-2.8905×10^3
	standard deviation	4.1129×10^2	4.1411×10^2	5.5787×10^2	6.0287×10^2	4.9253×10^2	6.1673×10^2
	mean rank	1	6	4	2	3	5
9	mean	0.0001	0.0019	5.6491×10^2	0.0048	2.3736×10^2	0.0000
	standard deviation	0.0004	0.0011	2.8588×10^1	0.0111	8.6884×10^1	0.0000
	mean rank	2	3	6	4	5	1
10	mean	0.0006	0.0037	1.8924×10^1	0.0048	3.6689	0.0000
	standard deviation	0.0011	0.0006	0.5567	0.0017	0.8335	0.0000
	mean rank	2	3	6	4	5	1
11	mean	0.0000	0.0007	3.9695×10^2	0.0123	2.1850	0.0000
	standard deviation	0.0000	0.0032	8.3426×10^1	0.0404	0.7185	0.0000
	mean rank	1	3	6	4	5	1
12	mean	0.9823	0.5494	3.1510×10^8	0.8250	2.5233	1.0020
	standard deviation	0.1288	0.0759	3.2474×10^8	0.1084	0.8024	0.1305
	mean rank	3	1	6	2	5	4
13	mean	3.1437	2.9258	5.2933×10^8	1.9067	9.1047	3.1107
	standard deviation	0.0277	0.1733	2.4883×10^8	0.2524	2.1976	0.0221
	mean rank	4	2	6	1	5	3

TABLE V
BENCHMARK SIMULATION RESULT ON SOLVING FDMFS

F	Parameter	ALO [29]	TIA [30]	FISA [39]	WaOA [17]	OOBO [33]	IBGO
14	mean	3.8074	6.2643	7.5230	4.9987	7.0108	6.4039
	standard deviation	2.7040	2.0845	4.1639	3.3790	4.6741	3.7165
	mean rank	1	3	6	2	5	4
15	mean	0.0054	0.0005	0.0151	0.0014	0.0046	0.0009
	standard deviation	0.0077	0.0002	0.0097	0.0033	0.0047	0.0016
	mean rank	5	1	6	3	4	2
16	mean	-1.0198	-1.0312	-1.0110	-1.0316	-1.0290	-1.0300
	standard deviation	0.0220	0.0010	0.0310	0.0000	0.0026	0.0029
	mean rank	5	2	6	1	4	3
17	mean	0.5440	0.4090	0.3982	0.3981	0.4415	0.4226
	standard deviation	0.1580	0.0254	0.0001	0.0000	0.0845	0.0495
	mean rank	6	3	2	1	5	4
18	mean	4.0355	4.4914	6.8533	8.1983	4.7895	3.0486
	standard deviation	2.4172	3.9253	7.4537	1.6775×10^1	3.5897	0.0950
	mean rank	2	3	5	6	4	1
19	mean	-0.0495	-0.0495	-0.0484	-0.0495	-0.0495	-0.0495
	standard deviation	0.0000	0.0000	0.0051	0.0000	0.0000	0.0000
	mean rank	1	1	6	1	1	1
20	mean	-2.7543	-2.8452	-2.8562	-3.1634	-2.5628	-3.1310
	standard deviation	0.2561	0.3308	0.4045	0.0729	0.3179	0.1278
	mean rank	5	4	3	1	6	2
21	mean	-2.4712	-4.4067	-2.3900	-4.4539	-1.9601	-5.4880
	standard deviation	1.4618	2.1528	1.2312	1.9642	0.9086	1.6349
	mean rank	4	3	5	2	6	1
22	mean	-2.4857	-5.3279	-3.2685	-4.2286	-1.7994	-5.8986
	standard deviation	0.9401	2.3619	2.3046	1.4113	0.6030	1.9338
	mean rank	5	2	4	3	6	1
23	mean	-2.3671	-3.5929	-3.0105	-4.2019	-2.2619	-4.9400
	standard deviation	0.6727	1.4722	1.6521	1.4748	0.8231	1.9490
	mean rank	5	3	4	2	6	1

Table 4 shows that ICMO is competitive on solving HDMFs although its superiority is not so high as on solving the HDUFs. ICMO becomes the first best on solving f_9, f_{10}, and f_{11}. On these functions, ICMO also can find the global optimal solution. Meanwhile, ICMO becomes the third best on f_{13}, the fourth best on f_{12}, and the fifth best on f_8.

Table 4 also indicates different circumstances rather than HDUFs. ALO is also very competitive in this second group of functions. It becomes the first best of two functions (f_8 and f_{11}). The competition on f8 is fierce as the difference between the best and worst performers is narrow. The performance gap between the best and worst performers is wide on three functions (f_9, f_{10}, and f_{11}) and very wide on two functions (f_{12} and f_{13}). Fierce competition takes place among four functions (ALO, TIA, WaOA, and ICMO). Meanwhile, FISA becomes the worst performer.

Table 5 indicates the superiority of ICMO on solving ten FDMFs. ICMO becomes the first best of five functions (f_{18}, f_{19}, f_{21}, f_{22}, f_{23}). ICMO becomes the second best on two functions (f_{15} and f_{20}), third best on one function (f_{16}), and fourth best on two functions (f_{14} and f_{17}). It means that ICMO is never on the fifth or sixth rank.

TABLE VI
GROUP BASED COMPARISON

Cluster	ALO [29]	TIA [30]	FISA [39]	WaOA [17]	OOBO [33]
1	4	4	6	4	6
2	3	4	5	3	5
3	8	5	9	5	9
Total	15	13	20	12	20

The competition among metaheuristics on ten FDMFs is fierce. This circumstance takes place on all ten functions. The performance gap between the best performer and the worst performer is narrow. Besides, five metaheuristics (ALO, TIA, WaOA, OOBO, and ICMO) achieve similar result on f_{19}. In some functions, the final solution is near the global optimal solution.

Table 6 indicates the superiority of ICMO among all its competitors. As summary, ICMO is better than ALO, TIA, FISA, WaOA, and OOBO on 15, 13, 20, 12, and 20 functions. Table 6 also indicates that ICMO is superior compared to FISA and OOBO while its superiority takes places on all groups of functions. Meanwhile, TIA and WaOA become the toughest competitors as the number of functions where ICMO is better than TIA or OOBO is the lowest. ICMO is superior to ALO especially on solving FDMFs.

The second simulation is conducted to assess the convergence behavior of ICMO. In this simulation, there are four iteration values which are observed: 5, 15, 15, and 20. In this simulation, the swarm size does not change. The result is presented in Table 7. In this simulation, the convergence of ICMO is not compared to the confronters as it focuses on ICMO. The result shows the average fitness score of ICMO in every chosen iteration and every function.

TABLE VII
CONVERGENCE ASSESSMENT RESULT

F	*t*=5	*t*=10	*t*=15	*t*=20
1	1.2093×10^{2}	0.0435	0.0000	0.0000
2	0.0000	0.0000	0.0000	0.0000
3	6.6244×10^{3}	3.7674×10^{2}	1.9092×10^{1}	0.8938
4	5.6009	0.1839	0.0040	0.0001
5	2.4216×10^{3}	4.9606×10^{1}	4.8937×10^{1}	4.8922×10^{1}
6	1.0812×10^{2}	1.0293×10^{1}	1.0180×10^{1}	1.0338×10^{1}
7	0.0917	0.0166	0.0101	0.0058
8	-2.7579×10^{3}	-2.8126×10^{3}	-2.8430×10^{3}	-2.8905×10^{3}
9	1.6864×10^{2}	1.9555	0.0007	0.0000
10	3.1447	0.0521	0.0006	0.0000
11	1.8687	0.0297	0.0000	0.0000
12	1.8919	1.0290	1.0034	1.0020
13	7.8307	3.3543	3.1423	3.1107
14	1.0201×10^{1}	7.0450	6.6139	6.4039
15	0.0056	0.0013	0.0006	0.0006
16	-1.0032	-1.0253	-1.0298	-1.0300
17	0.5071	0.4617	0.4417	0.4226
18	7.0134	3.2932	3.2164	3.0486
19	-0.0495	-0.0495	-0.0495	-0.0495
20	-2.7902	-3.0376	-3.1098	-3.1310
21	-2.5399	-3.5891	-4.5543	-5.4880
22	-2.8092	-3.7614	-4.2944	-5.8986
23	-3.0302	-3.9187	-4.8932	-4.9400

Table 7 indicates that convergence is achieved when the iteration is less than or equal to 20. This fast convergence is found in 20 functions. Five functions are unimodal functions while the rest ones are multimodal functions. It means that fast convergence is achieved in all multimodal functions and most unimodal functions.

The Java-Bali electricity system case is specified at time 18.00 where the demand is 13,096 MW. This grid consists of eight generator buses. The detailed specification of this grid system can be seen in [9]. As obtained from [9], the specification of the eight generators in the power system is presented in Table 8 and Table 9 where Table 8 presents the constants for the cost function of each generator while Table 9 presents the power constraints of each generator. Table 10 presents the result.

Table 10 indicates fierce competition among the metaheuristics on solving the ELD problem in electricity system. ICMO becomes the fifth best. FISA becomes the first best while ALO becomes the worst best. But the range between the best performer and the worst performer is very narrow.

TABLE VIII
CONSTANTS FOR THE COST FUNCTION OF EACH GENERATORS

Gen.	α	β	γ
1	-400.0	3,332,794.0	57,543,208.0
2	691.0	3,047,098.0	519,353,767.1
3	0.0	400.0	0.0
4	0.0	660.0	0.0
5	-80.0	2,828,349.0	133,177,025.6
6	218.0	2,104,640.0	133,177,025.6
7	203.0	2,545,832.0	140,621,312.5
8	-73.0	5,877,235.0	112,522,922.1

TABLE IX
POWER CONSTRAINTS

Gen.	x_{min} (MW/hour)	x_{max} (MW/hour)	x_r (MW/hour)
1	1,610.0	4,200.0	300
2	934.0	2,308.0	510
3	404.0	1,008.0	930
4	208.0	700.0	660
5	848.0	2,400.0	337
6	1,080.0	4,714.0	420
7	360.0	900.0	240
8	305.0	1,610.0	420

TABLE X
PERFORMANCE ASSESSMENT ON SOLVING ELD PROBLEM IN JAVA-BALI ELECTRICITY SYSTEM

Metaheuristic	Cost (rupiah/hour)
ALO [29]	31,501,795,690
TIA [30]	29,964,404,197
FISA [39]	29,495,245,703
WaOA [17]	29,520,347,525
OOBO [33]	29,665,828,720
ICMO	30,062,030,553

VI. DISCUSSION

In general, the performance of ICMO is acceptable as it becomes the best performer on eleven functions out of 23 functions. Due to the competitive result on solving the HDUFs, ICMO has good intensification capability. Meanwhile, the competitiveness of ICMO on solving the HDMFs, ICMO has good diversification capability. Finally, the superiority of ICMO on solving the FDMFs means that ICMO has good balance between intensification and diversification capabilities. Moreover, this performance is achieved in the low swarm size and low maximum iteration setup. This result means that ICMO can perform well in the environment where the computational resource is limited.

The result also indicates the need of applying multiple strategy approach in any metaheuristics. OOBO and FISA perform worst on solving theoretical problems. Meanwhile, OOBO [33] and FISA [39] are the only metaheuristics that employ only single search. Meanwhile, although TIA [30] also employs single search, an entity interacts with all other entities within the swarm. Meanwhile, WaOA [17] and ALO [29] are metaheuristics that employ multiple strategy. In WaOA [17], the neighborhood search becomes the secondary search while full random search after stagnation takes place is employed in ALO [29].

The result difference between the theoretical problem represented by the set of 23 mathematical functions and the practical problem represented by the ELD for the Java-Bali electricity system can be linked to NFL theory. First, a metaheuristic that is superior in some problems may lose its superiority in other problems. Superiority of ICMO on solving 23 functions is not followed by superiority on ELD problem. On the other hand, FISA becomes the first best performer on ELD problem although its performance is poor on solving the theoretical problems. This circumstance can also be linked to the nature of the floating-point based problem in the mathematical functions and the integer-based problem in the ELD problem. The wide performance gap in the theoretical problems is highly related to the precision level in the floating-point number. Meanwhile, in the integer-based problem, the range of performance or the difference between the best and worst performer becomes narrower. As the difference is not significant, it can be said that there is not any dominant metaheuristics on solving the ELD problem.

The computational complexity of ICMO is highly related to the number of loops in the algorithm. As a population-based metaheuristic, the swarm size and the maximum iteration affects the complexity. Meanwhile, the complexity during the initialization is different from the iteration. Apart from the dimensions, the loop during the initialization is the loop for whole swarm members. It means that the complexity during the initialization is presented as $O(n)$. Meanwhile, there are several aspects that should be considered in determining the complexity during the iteration. During the iteration, the process runs from the first iteration until the stopping criteria is achieved with the worst scenario is the maximum iteration. Then, there is a loop for whole swarm members to accommodate the searching process performed by each member. Then, there is also a loop for whole swarm members in the first search to collect the better entity within swarm. Based on this explanation, the complexity during the iteration is presented as $O(t_m.n^2)$.

There are several limitations in this work, especially in the proposed ICMO. First, ICMO has been applied to solve both theoretical and practical problems. But this paper only accommodates the ELD problems with the specific locus in Java-Bali electricity system. Meanwhile, there are broader cases in the power system, and moreover in the engineering sector, such as in handling congestion management in power transmission as in [40], managing voltage stability as in [41], optimizing the reactive power [42], and so on. In this paper, the fuel cost is the only parameter that is considered so that it can be seen as a single objective optimization problem. On the other hand, as the environmental issue becomes more crucial, it is better that parameters related to the environmental issue, such as emission cost are also considered. It means that this single objective problem can be transformed into multi objective problem where several methods can be chosen, such as weighted sum, non-dominated sorting, and so on.

VII. Conclusion

A new metaheuristic called iteration-controlled mixture optimization (ICMO) has been presented in this paper. Its strategy relies on three directed searches performed by every entity in the swarm in every iteration. Meanwhile the iteration controls the portion of two entities constructing the reference during the directed search. The performance of ICMO has been conducted by confronting ICMO with five new swarm-based metaheuristics to solve 23 mathematical functions and ELD problems. The result shows that ICMO is still superior to its five contenders. ICMO is better than ALO, TIA, FISA, WaOA, and OOBO in 15, 13, 20, 12, and 20 functions respectively. Meanwhile, ICMO is competitive enough in solving the ELD problems as the range of performance among these metaheuristics are narrow. This result strengthens the statement of NFL theory as ICMO cannot be the best performer in all 23 functions.

In the future, ICMO can be utilized to solve broader optimization problems in power systems. The environmental aspect can be considered too as the optimization objective, for example to minimize emission. A multi-objective optimization problem like economic emission dispatch (EED) is an interesting challenge. Moreover, more generators that are involved in the power system are needed to provide higher power to meet the increasing demand.

References

[1] G. Guo, J. Qian, and S. Li, “Optimal power flow based on novel multi-objective artificial fish swarm algorithm”, *Engineering Letters*, vol. 28, no. 2, pp. 542-550, 2020.

[2] 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.

[3] G. Chen, X. Wang, S. Mo, J. Zhang, W. Xong, H. Long, and M. Zou, “Multi-objective power flow optimization based on improved hybrid crow search algorithm”, *Engineering Letters*, vol. 30, no. 4, pp. 1417-1435, 2022.

[4] T. T. Nguyen, H. D. Nguyen, and M. Q. Duong, "Optimal power flow solutions for power system considering electric market and renewable energy", *Applied Sciences*, vol 13, ID. 3330, 2023.

[5] S. R. Salkuti, “Solving reactive power scheduling problem using multi-objective crow search algorithm”, *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, pp. 42-48, 2021.

[6] F. Tariq, S. Alelyani, G. Abbas, A. Qahmash, and M. R. Hussain, "Solving renewables-integrated economic load dispatch problem by variant of metaheuristic bat-inspired algorithm", *Energies*, vol. 13, ID. 6225, 2020.

[7] N. Singh, T. Chakrabarti, P. Chakrabarti, M. Margala, A. Gupta, S. P. Praveen, S. B. Krishnan, and B. Unhelkar, "Novel heuristic optimization technique to solve economic load dispatch and economic emission load dispatch problems", *Electronics*, vol. 12, no. 2921, 2023.

[8] M. A. Al-Betar, M. A. Awadallah, S. N. Makhadmeh, I. A. Doush, R. A. Zitar, S. Alshathri, and M. A. Elaziz, "A hybrid Harris hawks optimizer for economic load dispatch problems", *Alexandria Engineering Journal*, vol. 64, pp. 365-389, 2023.

[9] K. M. D. Puspitasari, J. Raharjo, A. S. Sastrosubroto, and B. Rahmat, "Generator scheduling optimization involving emission to determine emission reduction costs", *International Journal of Engineering, Transactions B: Applications*, vol. 35, no. 8, pp. 1468-1478, 2022.

[10] H. Lotfi, "A multiobjective evolutionary approach for solving the multi-area dynamic economic emission dispatch problem considering reliability concerns", *Sustainability*, vol. 15, ID. 442, 2023.

[11] J. -S. Chou, and D. -N. Truong, "A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean", *Applied Mathematics and Computation*, vol. 389, ID. 125535, 2021.

[12] F. A. Hashim, E. H. Houssein, K. Hussain, M. S. Mabrouk, and W. Al-Atabany, "Honey badger algorithm: new metaheuristic algorithm for solving optimization problems", *Mathematics and Computers in Simulation*, vol. 192, pp. 84-110, 2022.

[13] 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.

[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] 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.

[16] 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.

[17] P. Trojovsky and M. Dehghani, “A new bio-inspired metaheuristic algorithm for solving optimization problems based on walruses behavior”, *Scientific Reports*, vol. 13, ID. 8775, pp. 1-32, 2023.

[18] M. Dehghani, P. Trojovsky, and O. P. Malik, “Green anaconda optimization: a new bio-inspired metaheuristic algorithm for solving optimization problems”, *Biomimetics*, vol. 8, ID. 121, pp. 1-60, 2023.

[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] 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.

[21] Suyanto, A. A. Ariyanto, and A. F. Ariyanto, “Komodo mlipir algorithm”, *Applied Soft Computing*, vol. 114, ID: 108043, 2022.

[22] M. Dehghani, G. Bektemyssova, Z. Montazeri, G. Shaikemelev, O. M. Malik, and G. Dhiman, “Lyrebird optimization algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems”, *Biomimetics*, vol. 8, ID. 507, pp. 1-62, 2023.

[23] M. Dehghani, Z. Montazeri, G. Bektemyssova, O. P. Malik, G. Dhiman, and A. E. M. Ahmed, “Kookaburra optimization algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems”, *Biomimetics*, vol. 8, ID. 470, pp. 1-54, 2023.

[24] D. Polap and M. Wozniak, “Red fox optimization algorithm”, *Expert Systems with Applications*, vol. 166, ID. 114107, pp. 1-21, 2021.

[25] M. Suman, V. P. Sakthivel, and P. D. Sathya, “Squirrel search optimizer: nature inspired metaheuristic strategy for solving disparate economic dispatch problems”, *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 5, pp. 111–121, 2020.

[26] J. Xue and B. Shen, “A novel swarm intelligence optimization approach: sparrow search algorithm”, *Systems Science & Control Engineering*, vol. 8, no. 1, pp. 22-34, 2020.

[27] M. Braik, A. Hammouri, J. Atwan, M. A. Al-Betar, and M. A. Awadallah, “White shark optimizer: a novel bio-inspired meta-heuristic algorithm for global optimization problems”, *Knowledge-Based Systems*, vol. 243, ID. 108457, pp. 1-29, 2022.

[28] E.-S. M. El-kenawy, N. Khodadadi, S. Mirjalili, A. A. Abdelhamid, M. M. Eid, and A. Ibrahim, “Greylag goose optimization: nature-inspired optimization algorithm”, *Expert Systems with Applications*, vol. 238, part E, ID. 122147, 2024.

[29] 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.

[30] P. D. Kusuma and A. Novianty, “Total interaction algorithm: a metaheuristic in which each agent interacts with all other agents”, *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 1, pp. 224-234, 2023.

[31] 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.

[32] 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.

[33] M. Dehghani, E. Trojovska, P. Trojovsky, and O. P. Malik, “OOBO: a new metaheuristic algorithm for solving optimization problems”, *Biomimetics*, vol. 8, ID. 468, pp. 1-48, 2023.

[34] W. Zhao, L. Wang, Z. Zhang, S. Mirjalili, N. Khodadadi, and Q. Ge, “Quadratic interpolation optimization (QIO): a new optimization algorithm based on generalized quadratic interpolation and its applications to real-world engineering problems”, *Computer Methods in Applied Mechanics and Engineering*, vol. 417, part A, 116446, 2023.

[35] F. Rezaei, H. R. Safavi, M. A. Elaziz, S. Mirjalili, “GMO: geometric mean optimizer for solving engineering problems”, *Soft Computing*, vol. 27, pp. 10571–10606, 2023.

[36] P. Gonzalez-Ayala, A. Alejo-Reyes, E. Cuevas, and A. Mendoza, “A modified simulated annealing (MSA) algorithm to solve the supplier selection and order quantity allocation problem with non-linear freight rates”, *Axioms*, vol. 12, ID. 459, pp. 1-17, 2023.

[37] P. D. Kusuma and R. A. Nugraheni, “Stochastic marine predator algorithm with multiple candidates”, *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 4, pp. 241-251, 2022.

[38] 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.

[39] M. Ghasemi, A. Rahimnejad, E. Akbari, R. V. Rao, P. Trojovský, E. Trojovská, and S. A. Gadsden, "A new metaphor-less simple algorithm based on Rao algorithms: a fully informed search algorithm (FISA)", *PeerJ Computer Science*, vol. 9, ID. e1431, pp. 1-24, 2023.

[40] B. N. Bhukya, P. R. Chinda, S. R. Rayapudi, and S. R. Bondalapati, "Advanced control with an innovative optimization algorithm for congestion management in power transmission networks", *Engineering Letters*, vol. 31, no.1, pp. 194-205, 2023.

[41] J. -H. Zhu, J. -S. Wang, and X. -Y. Zhang, "Solving optimal power flow problem of power system based on Archimedes optimization algorithm", *IAENG International Journal of Computer Science*, vol. 50, no.1, pp. 63-70, 2023.

[42] H. Long, S. Liu, T. Chen, H. Tan, J. Wei, C. Zhang, and W. Chen, "Optimal reactive power dispatch based on multi-strategy improved aquila optimization algorithm", *IAENG International Journal of Computer Science*, vol. 49, no.4, pp. 1249-1267, 2022.

Purba Daru Kusuma is an assistant professor in computer engineering in Telkom University, Indonesia. He received bachelor and master 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 include artificial intelligence, machine learning, and operational research.