Include-Exclude Optimization: A New Metaheuristic and Its Application to Handle Optimization Problems in Electrical and Mechanical Engineering

Purba Daru Kusuma, *Member IAENG*

***Abstract*—There are two challenges in the development of new metaheuristics. The first challenge is the utilization of the improving status of agents. The second challenge is the employment of stagnation avoidance strategies. This work introduces a new metaheuristic called include-exclude optimization (IEO). It provides a novel approach by combining both the quality and improving statuses of agents to construct the reference and determine the direction of the guided search or movement. IEO also proposes a new technique to avoid stagnation by accepting the best solution candidate when stagnation occurs after the agent performs three guided searches. Then, IEO is challenged to solve three use cases. The first use case is the 23 traditional functions. The second use case is ELD problem representing practical problem in electrical engineering field. The third use case is gear train design problem representing practical problem in mechanical engineering field. In this assessment, IEO is benchmarked with five new metaheuristics: golden search optimization (GSO), total interaction algorithm (TIA), dollmaker optimization algorithm (DOA), carpet weaver optimization (CWO), and hiking optimization (HO). The result shows that IEO is superior to these five metaheuristics in handling 23 traditional functions and performs the best in handling ELD problem and gear train design problem.**

***Index Terms*—optimization, metaheuristic, economic load dispatch problem, gear train problem.**

I. Introduction

OPTIMIZATION is highly related to the engineering field. Optimization studies can be found in many engineering studies, including electrical, mechanical, industrial, and so on. Optimization studies can be found in many electrical engineering studies, especially in power systems, including the economic dispatch [1], unit commitment [2], power flow [3], and so on. In mechanical engineering, optimization can be found easily in many engineering designs, such as gear train, three-bar truss, spring, welded beam, cantilever beam, tubular column, and so on [4]. In industrial engineering, optimization studies are employed in vast areas of supply chain management from the production scheduling [5], warehousing [6], shipment [7], and so on.

Manuscript received June 5, 2024; revised November 13, 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).

Metaheuristics has been widely employed in various optimization studies in engineering fields. In power system, some metaheuristics have been employed to solve optimal power flow problems, such as particle swarm optimization (PSO) [8], Archimedes optimization algorithm (AOA) [3], driving training-based optimization (DTBO) [9], marine predator algorithm (MPA) [10], and so on. Some metaheuristics also have been employed to solve economic dispatch problems, including sparrow search algorithm (SSA) [11], chameleon algorithm (CA) [12], dandelion optimizer (DO) [13], manta-ray foraging algorithm (MRFA) [14], teaching learning-based optimization (TLBO) [15], and so on. In production system, some metaheuristics have been utilized to solve scheduling problems, such as: cuckoo search algorithm (CS) [16], genetic algorithm (GA) [17], artificial bee colony (ABC) [18], and so on. In mechanical engineering design field, some metaheuristics have been employed to solve gear train design problems, such as: prairie dog optimization (PDO) [4], evolutionary algorithm (EA) [19], learning-cooking algorithm (LCA) [20], and so on.

The vast employment of metaheuristics in various optimization studies can be linked to the abundance of new metaheuristics. There are a lot of new metaheuristics introduced in the recent years, such as: golden search optimization (GSO) [21], addax optimization algorithm (AOA) [22], carpet weaver optimization (CWO) [23], total interaction algorithm (TIA) [24], dollmaker optimization algorithm (DOA) [25], hiking optimization (HO) [26], hippopotamus optimization (HO) [27], greater cane rat algorithm (GCRA) [28], group better-worse algorithm (GBWA) [29], artificial protozoa optimization (APO) [30], and so on.

In all studies proposing a new metaheuristic, the proposed technique was assessed by employing it to handle certain optimization problems. Standard use cases like 23 traditional functions or CEC series become the common or mandatory use case. Meanwhile, many of these metaheuristics were also assessed with the practical problems. The engineering design problems become the most popular ones as these problems can be found in many studies like in the first introduction of clouded leopard optimization (CLO) [31], chameleon swarm algorithm (CSA) [32], walrus optimization algorithm (WaOA) [33], deep sleep optimizer (DSO) [34], learning cooking algorithm (LCA) [35], HO [27], object-oriented programming optimization algorithm (OOPOA) [36], and so on. Meanwhile, some studies employed economic dispatch problem as their practical use case, such as: squirrel search optimization (SSO) [37], stochastic shaking algorithm (SSA) [38], technique of narrowing down area (ToNDA) [39], iteration-controlled mixture optimization (ICMO) [40], and so on. Meanwhile, studies that accommodate practical use cases from multiple fields are rare to find.

Meanwhile, in the technical aspect, many metaheuristics, especially the swarm-based ones, focus on the recent quality of the agents rather than the improving status of these agents. In some metaheuristics, the agents move toward the optimal agent [41], a randomly picked better agent [42], or the average position of better agents [43]. Meanwhile, in some cases, the agents move toward a randomly picked agent if this picked agent is better than the moving agent [31]. Otherwise, this moving agent moves away from the randomly picked agent [31]. This circumstance may lead to stagnation if the agent moves toward the reference although this reference is better than the moving agent, but it fails to improve. This movement may lead to another stagnation. Unfortunately, recent metaheuristics that give attention to the improving status of the agent to be considered as reference are also rare to find.

Based on these problems, this paper is aimed to introduce a new metaheuristic called include-exclude optimization (IEO). IEO proposes a new technique in considering both recent quality and the improving status of the agent in the construction of the reference to be used in the guided search. IEO also performs strategy to treat agents that fail to improve.

Following this objective, the scientific contribution of this paper is as follows.

- This paper proposes a new swarm-based metaheuristic called as include-exclude optimization (IEO).
- IEO introduces a new approach in considering both quality and improving status of the agents for consideration.
- IEO introduces a new approach in treating agents that fail to improve.
- This paper provides an assessment to investigate the performance of IEO by employing it to solve three use cases: 23 traditional functions, ELD problem, and an engineering design problem.
- The performance of IEO is benchmarked with five new metaheuristics.

The structure of the rest of this paper is as follows. Section two reviews the recent studies in proposing new metaheuristics, especially the use case. Section three provides a detailed description of the model of IEO including the fundamental idea, algorithm, and mathematical formulation. Section four presents the assessment of IEO in handling three use cases. Section five provides a comprehensive discussion regarding the result, findings, and limitations. Section six provides the summary of conclusion and tracks for future studies.

II. RELATED WORKS

In swarm intelligence, there are a certain number of agents that work autonomously in finding a better solution. Although these agents work autonomously, interaction among agents plays critical roles in finding the quasi-optimal solution. In the guided search which is the fundamental or primary movement of agents in the swarm intelligence, some references are needed. Meanwhile, some metaheuristics are also enriched with neighborhood search which does not need any references.

The quality of agents in the population or swarm plays important roles in the guided search. First, the quality of agents determines whether these agents will be selected to construct the reference. Some references include the optimal agent [41], a randomly picked better agent, a randomly picked agent [24], and so on. Second, the quality of agents determines the direction of the movement. In some metaheuristics, like TIA [24] or ZOA [44], an agent moves toward the reference only if the reference is better than the agent. Otherwise, this agent will move away from the reference.

Unfortunately, the improving status of the agent is not considered in most swarm-based metaheuristics. The improving status is the status that provides information whether an agent improves its quality in its last search. Improving status is also important to provide information whether a certain area still has probability to generate improvement. The opposite of the improvement is stagnation.

Many metaheuristics, especially the swarm-based metaheuristics, do not care about stagnation. In general, they focus on the motion to improve the solution. Many metaheuristics employ strict acceptance approach to avoid the worsening situation by rejecting the candidate that does not provide improvement. But it still does not guarantee facing stagnation. In some metaheuristics like ABC [45], certain penalty is introduced for units that stagnate and forces them to move anywhere in the search space.

The existence of use cases cannot be separated from the development of metaheuristics. Any new metaheuristics should be assessed to solve various optimization problems in their first introduction. This assessment is important to investigate the efficacy, strength, and weakness of the proposed metaheuristics.

In general, there are two types of use cases. The first use cases are sets of standard functions. Each set of standard functions comprises many functions where each of these functions has its own nature. In general, each function has only a single objective. Commonly, each set of functions can be split into two groups: unimodal functions and multimodal functions. The second use cases are the practical problems. The practical problems represent real-world optimization problems as they have specific objectives and constraints. Some problems are single objective problems while others are multi objective problems. Engineering design problems become the most popular practical use cases to be used in the first introduction of many new metaheuristics. The short review of the use cases which were used in the first introduction of several recent new metaheuristics is revealed in Table 1.

TABLE I
REVIEW ON USE CASES IN SEVERAL RECENT METAHEURISTICS

No	Metaheuristics	Standard Use Cases	Practical Use Cases
1	ICMO [40]	23 traditional functions	ELD
2	TIA [24]	23 traditional functions	-
3	CWO [23]	23 traditional functions	Pressure vessel, speed reducer, welded beam, spring
4	HO [26]	23 traditional functions, composite functions	I-beam, spring, gear train, NP-hard problem, traveling salesman problem, knapsack problem,
5	HOA [27]	23 traditional functions, 8 ZP functions, CEC 2019, CEC 2014,	Spring, welded beam, pressure vessel
6	COA [41]	CEC 2017, CEC 2011	Pressure vessel, speed reducer, welded beam, spring
7	WaOA	23 traditional functions, CEC 2015, CEC 2017	Spring, welded beam, speed reducer, pressure vessel
8	SSO [37]	-	ELD
9	ToNDA [39]	-	ELD
10	proposed work	23 traditional functions	ELD, gear train

Table 1 reveals that the set of 23 traditional functions dominates the standard functions for assessment of new metaheuristics. Meanwhile, CEC series are popular too. Some metaheuristics are designed to solve specific cases like SSO [37], ToNDA [41], AOA [22], and so on where SSO and ToNDA are specific to solve ELD problem while AOA is specific to solve four engineering design problems.

On the other hand, engineering design problems become the most popular practical use cases in the first introduction of new metaheuristics. Then, economic dispatch problem is also used in several studies introducing new metaheuristics. Some studies also envoys several other practical use cases. Meanwhile, some metaheuristics do not employ to solve any practical problems but only the standard functions only.

This review highlights three circumstances that motivate this work. The first circumstance is that the unpopularity of improving status of agents while most metaheuristics focus on the recent quality of the agents. The second circumstance is the unpopularity of stagnation avoidance. The third circumstance is the lack of studies that employ practical use cases from multiple fields in the introduction of new metaheuristics.

III. PROPOSED MODEL

In general, include-exclude optimization (IEO) is constructed based on the swarm intelligence method. The system of IEO consists of a certain number of autonomous agents where each of these agents represent the solution. Each agent works actively in every iteration to improve its own quality and the entire population. Each agent traces for possible better solution along the search space.

The fundamental idea of IEO is the treatment of the agent not only based on its recent quality but also its improvement. This idea is delivered into two strategies. The first strategy is that an agent will move toward a selected agent only if this selected agent is better than the related agent but also if this selected agent is able to improve itself. Otherwise, the related agent will not consider this selected agent or avoid this selected agent.

The second strategy is related to the treatment for an agent that fails to improve after it conducts all searches in an iteration. In IEO, strict acceptance is employed so that a solution candidate replaces the current solution only if this candidate is better than the current solution. This approach is needed to avoid an agent to be thrown away to the worse situation. But on the other hand, this approach may produce stagnation. To avoid this stagnation, the best solution candidate will replace the recent solution based on the best of the worse concept.

This fundamental idea is converted into three directed searches performed by every agent in every iteration. In the first search, the agent moves toward the optimal agent which is the agent whose quality is the best so far. In the second search, the agent moves toward a randomly picked better and improving agent. In the third search, the agent will move toward a randomly picked agent whose quality is better, and it is improving. Otherwise, this agent will move away from this randomly picked agent.

The formal model of IEO is described in mathematical formulation and algorithm. The mathematical formulations are provided in (1) to (16). Meanwhile, the algorithm of IEO is provided in algorithm 1 using pseudocode.

In the initialization phase, each agent is uniformly placed within the search space. It is formalized using (1) where x is the agent, i represents the index of the agent and j represents the dimension of the space. Then, x_{lb} represents the lower boundary of the space and x_{ub} represents the upper boundary of the space. U_1 is a floating-point random number that ranges from 0 to 1.

$$x_{i,j} = x_{lb,j} + U_1(x_{ub,j} - x_{lb,j}) \tag{1}$$

Each time an agent is initialized, the updating for the optimal agent is performed. Equation (2) formalizes this updating process. x_{opt} represents the optimal agent. Variable of represents the objective function.

$$x'_{opt} = \begin{cases} x_i, of(x_i) < of(x_{opt}) \\ x_{opt}, else \end{cases} \tag{2}$$

In the beginning of the searching process, the value of the agent is stored first. The storing process is formalized using (3) where $x_{prev,i}$ stores the value of agent i. This value will be used to determine whether the agent improves after it performs searches.

$$x_{prev,i} = x_i \tag{3}$$

The formalization of the first search is provided in (4) and (5). Equation (4) formalizes the motion toward the optimal solution where $c_{1,i}$ represents the first solution candidate and U_2 represents the uniform integer random number between 1 and 2. Equation (5) formalizes the updating process of agent i based on the quality comparison with the first solution candidate. After the updating of the related agent is performed, the optimal solution is also updated using (2).

$$c_{1,i,j} = x_{i,j} + U_1(x_{opt,j} - U_2 x_{i,j}) \tag{4}$$

$$x_i' = \begin{cases} c_{1,i}, of(c_1) < of(x_i) \\ x_i, else \end{cases} \tag{5}$$

The formalization of the second search is provided in (6) to (9). Equation (6) formalizes the construction of a pond that comprises all agents whose quality is better than the related agent and whose status is improving. This pond also includes the optimal solution. Variable $X_{hi,i}$ represents the pond. Then, (7) formalizes the random picking of a selected agent from this pond. $X_{sel1,i}$ represents the first selected agent while U_3 represents the uniform random from certain population. Equation (8) formalizes the motion toward this first selected agent where $c_{2,i}$ represents the second solution candidate. Equation (9) formalizes the updating process of the related agent by comparing it with the second solution candidate. Like the first search, the updating of the optimal solution is performed after the updating of the related agent.

$$X_{hi,i} = \{\forall x | of(x) < of(x_i) \land s(x) = 1\} \cup x_{opt} \tag{6}$$

$$x_{sel1,i} = U_3(X_{hi,i}) \tag{7}$$

$$c_{2,i,j} = x_{i,j} + U_1(x_{sel1,i,j} - U_2 x_{i,j}) \tag{8}$$

$$x_i' = \begin{cases} c_{2,i}, of(c_2) < of(x_i) \\ x_i, else \end{cases} \tag{9}$$

The formalization of the third search is provided in (10) to (13). Equation (10) formalizes the second randomly picked agent from the population where $x_{sel2,i}$ represents the second picked agent. Equation (11) formalizes the motion where the direction of the motion depends on the certain circumstance. Equation (12) formalizes that the motion is toward the reference only if the random picked agent is better than the related agent and its status is improving. Equation (13) formalizes the updating process of the related agent by comparing it with the third solution candidate where $c_{3,i}$ represents the third solution candidate. Like the first and second searches, the updating of the optimal solution is performed after the updating of the related agent.

$$x_{sel2,i} = U_3(X) \tag{10}$$

$$c_{3,i,j} = \begin{cases} x_{i,j} + U_1\left(x_{sel2,i,j} - U_2 x_{i,j}\right), case = 1 \\ x_{i,j} + U_1\left(x_{i,j} - U_2 x_{sel2,i,j}\right), else \end{cases} \tag{11}$$

$$case = \begin{cases} 1, of\left(x_{sel2,i}\right) < of(x_i) \land s\left(x_{sel2,i}\right) = 1 \\ 0, else \end{cases} \tag{12}$$

$$x_i' = \begin{cases} c_{3,i}, of(c_3) < of(x_i) \\ x_i, else \end{cases} \tag{13}$$

After these three searches are performed, the next process is stagnation avoidance. First, the status of the related agent is updated by using (14). Then, the best solution candidate replaces the current value of the related agent if stagnation occurs using (15). Equation (16) provides the selection of the best solution candidate among three solution candidates.

$$s(x_i) = \begin{cases} 1, of(x_i) < of(x_{prev,i}) \\ 0, else \end{cases} \tag{14}$$

$$x_i' = \begin{cases} c_{sel,i}, s(x_i) = 0 \\ x_i, else \end{cases} \tag{15}$$

$$c_{sel,i} = c \in C_i, \min\left(of(c)\right) \tag{16}$$

algorithm 1: include-exclude optimization

```
1   start
2    for all x
3     initialize x
4    end
5    for t=1 to T
6     for all x
7      store x_prev
8      perform first search
9      perform second search
10     perform third search
11     perform stagnation avoidance
12    end
13   end
14   return x_opt
15  stop
```

The explanation of IEO based on algorithm 1 is as follows. The initialization is presented from lines 2 to 4. Meanwhile, the iteration is presented from lines 5 to 13. Line 14 states that the optimal solution becomes the final solution.

The computational complexity of IEO can be investigated from algorithm 1 based on the number of loops. In the initialization, the complexity is presented as $O(n(X).d)$. This presentation means that the complexity is linearly proportional to the population size or dimension. In the iteration, the complexity is presented as $O(T.n(X)^2.d)$ which means the complexity is linearly proportional to the maximum iteration, or dimension but quadratic proportional to population size.

IV. Simulation

This section provides the assessment of IEO to investigate its performance in handling optimization problems. There are three problems which are investigated in this work. The first problem is the set of standard functions that comprises 23 functions. The second problem is the economic load dispatch (ELD) problem representing optimization problem in electrical field. The third problem is the gear train problem representing optimization problem in mechanical field.

In this work, IEO is benchmarked with five new metaheuristics: GSO [21], TIA [24], DOA [25], CWO [23], and HO [26]. GSO [21] and HO [26] are metaheuristics that employ loose acceptance approach. On the other hand, TIA [24], DOA [25], and CWO [23] are metaheuristics that employ hard acceptance approach. In these three assessments, the population size is 5 while the maximum iteration is 10.

The first assessment is employing IEO to handle a set of 23 standard functions. These functions can be split into three groups: seven high dimension unimodal functions, six high dimension multimodal functions, and ten fixed dimension multimodal functions. The detailed description of these functions can be found in [21]. In these 13 multimodal functions, the dimension is set to 20. In this assessment, the decimal point which is less than 10^{-4} is rounded to 0.

Table 2 provides the assessment result in handling seven high dimension unimodal functions. In this assessment, IEO shows its supremacy as it becomes the best performer in all seven functions. Moreover, IEO can find the global optimal of f_2. In this function, TIA and DOA also can find the global optimal solution. The performance disparity between the best performer and the worst one is wide in all these seven functions. By comparing the range and the absolute mean of IEO in these functions, it is shown that the range is less than the absolute mean in two functions (f_5 and f_6). It means that the result in both functions is stable among the sessions.

Table 3 provides the assessment result in handling six high dimension multimodal functions. In this assessment, IEO shows its supremacy in handling four out of six functions (f_9, f_{10}, f_{11}, and f_{13}). Meanwhile, IEO becomes the second best in handling f_{12} where TIA becomes the best one. Fortunately, the disparity between IEO and TIA in f_{12} is narrow. IEO becomes the third best in handling f_8 where CWO becomes the first best and DOA becomes the second best. In this second group, the performance disparity between the best and the worst is wide in five functions (f_9 to f_{13}). Meanwhile, this disparity is narrow in f_8. The ratio between the range and the absolute means of IEO is more than 1 in only two functions (f_9 and f_{11}). Otherwise, this ratio is less than 1 representing stable quality of final solution.

Table 4 shows the competitiveness of IEO in handling the fixed dimension multimodal functions. It becomes the first best in two functions (f_{19} and f_{22}), second best in f_{15}, third best in six functions (f_{14}, f_{16}, f_{17}, f_{18}, f_{21}, and f_{23}), and fourth best in f_{20}. The disparity between the best and the worst is wide in f_{18}. Otherwise, this disparity is narrow. The ratio between the range and the absolute mean of IEO is smaller than 1 only in f_{19} which indicates the stable final solution. Otherwise, this ratio is bigger than 1.

Table 5 summarizes the supremacy of IEO compared to other metaheuristics based on the group of functions. Overall, IEO is absolute superior to GSO and HO as it outperforms both in all 23 functions. Meanwhile, IEO is better than TIA, DOA, and CWO in 17, 13, and 16 functions.

TABLE II
BENCHMARK SIMULATION RESULT ON SOLVING HIGH DIMENSION UNIMODAL FUNCTIONS

F	Parameter	GSO [21]	TIA [24]	DOA [25]	CWO [23]	HO [26]	IEO
1	mean	1.5802×10^{4}	3.2028	1.1978×10^{2}	1.3246×10^{4}	1.0124×10^{2}	0.0004
	range	1.4478×10^{4}	1.2548×10^{1}	1.4581×10^{2}	7.7226×10^{3}	2.4654×10^{2}	0.0025
	mean rank	6	2	4	5	3	1
2	mean	1.7312×10^{23}	0.0000	0.0000	1.8447×10^{21}	2.2082×10^{2}	0.0000
	range	1.2039×10^{24}	0.0000	0.0000	2.9748×10^{22}	2.3605×10^{3}	0.0000
	mean rank	6	1	1	5	4	1
3	mean	3.1571×10^{4}	3.2133×10^{2}	3.3848×10^{3}	2.3129×10^{4}	9.2134×10^{2}	5.9739
	range	5.8186×10^{4}	9.1523×10^{2}	9.7469×10^{3}	2.8273×10^{4}	1.8743×10^{3}	5.1489×10^{1}
	mean rank	6	2	4	5	3	1
4	mean	5.4337×10^{1}	1.6101	9.1905	5.1060×10^{1}	4.2934	0.0598
	range	3.3162×10^{1}	1.9599	1.4997×10^{1}	2.1571×10^{1}	4.3326	0.1748
	mean rank	6	2	4	5	3	1
5	mean	2.7682×10^{7}	8.0393×10^{1}	5.7027×10^{3}	1.6132×10^{7}	8.4320×10^{4}	1.8949×10^{1}
	range	6.6276×10^{7}	1.7271×10^{2}	2.0817×10^{4}	2.0852×10^{7}	4.7671×10^{5}	0.0988
	mean rank	6	2	3	5	4	1
6	mean	1.4610×10^{4}	5.2259	6.9737×10^{1}	1.1486×10^{4}	9.1759×10^{1}	3.4568
	range	1.4292×10^{4}	6.1248	1.7829×10^{2}	7.8755×10^{3}	2.1800×10^{2}	1.2755
	mean rank	6	2	3	5	4	1
7	mean	6.3712	0.0633	0.0804	6.4945	1.1451×10^{2}	0.0182
	range	1.9700×10^{1}	0.1627	0.1720	8.3692	3.2108×10^{2}	0.0508
	mean rank	4	2	3	5	6	1

TABLE III
BENCHMARK SIMULATION RESULT ON SOLVING HIGH DIMENSION MULTIMODAL FUNCTIONS

F	Parameter	GSO [21]	TIA [24]	DOA [25]	CWO [23]	HO [26]	IEO
8	mean	-1.4667×10^{3}	-1.4947×10^{3}	-1.9597×10^{3}	-2.0882×10^{3}	-1.5823×10^{2}	-1.7713×10^{3}
	range	2.7501×10^{3}	1.6238×10^{3}	1.5566×10^{3}	1.0522×10^{3}	2.4319×10^{2}	1.4145×10^{3}
	mean rank	6	5	2	1	4	3
9	mean	1.7409×10^{2}	4.9037×10^{1}	6.3818×10^{1}	1.9683×10^{2}	2.2187×10^{2}	1.4469
	range	8.1798×10^{1}	1.0503×10^{2}	9.2964×10^{1}	5.2515×10^{1}	1.1491×10^{2}	2.5503×10^{1}
	mean rank	4	2	3	5	6	1
10	mean	1.8308×10^{1}	1.0956	3.8604	1.7701×10^{1}	6.6976	0.0041
	range	5.2981	1.3121	4.5986	2.4453	4.5398	0.0011
	mean rank	6	2	3	5	4	1
11	mean	1.4000×10^{2}	0.8888	2.0063	1.1805×10^{2}	1.0247	0.0290
	range	1.7246×10^{2}	0.5904	2.0577	6.6353×10^{1}	0.7474	0.5743
	mean rank	6	2	4	5	3	1
12	mean	2.3228×10^{7}	0.8659	3.2792	1.1723×10^{7}	7.5764	0.9393
	range	1.0759×10^{8}	1.1503	4.8871	2.1076×10^{7}	9.7004	0.8789
	mean rank	6	1	3	5	4	2
13	mean	7.8677×10^{7}	3.4491	3.8415×10^{1}	4.6308×10^{7}	1.1113×10^{1}	2.9676
	range	2.5273×10^{8}	2.5076	3.6286×10^{2}	6.4507×10^{7}	4.6060×10^{1}	0.7562
	mean rank	6	2	4	5	3	1

TABLE IV
BENCHMARK SIMULATION RESULT ON SOLVING FIXED DIMENSION MULTIMODAL FUNCTIONS

F	Parameter	GSO [21]	TIA [24]	DOA [25]	CWO [23]	HO [26]	IEO
14	mean	3.0309x10[1]	1.3742x10[1]	1.0151x10[1]	9.4918	2.1643x10[1]	1.0742x10[1]
	range	3.6658x10[2]	3.1491x10[1]	1.5505x10[1]	2.1998x10[1]	1.2572x10[2]	1.4392x10[1]
	mean rank	6	4	2	1	5	3
15	mean	0.4354	0.0101	0.0275	0.0269	0.2474	0.0124
	range	5.2839	0.0446	0.0742	0.0973	1.6114	0.0883
	mean rank	6	1	5	3	4	2
16	mean	-0.6778	-1.0192	-0.9671	-0.8966	3.0957	-0.9121
	range	2.3412	0.1214	0.3437	0.7196	4.3897x10[1]	1.0314
	mean rank	5	1	2	4	6	3
17	mean	4.1276	2.3927	0.4873	0.5421	2.3867	2.1166
	range	2.1737x10[1]	8.1835	0.5160	0.4233	6.5076	1.5367x10[1]
	mean rank	6	5	1	2	4	3
18	mean	2.4641x10[1]	2.9092x10[1]	6.2122	8.1557	6.9100x10[2]	1.8662x10[1]
	range	1.2548x10[1]	1.2791x10[2]	2.7800x10[1]	2.4637x10[1]	2.5619x10[3]	8.2351x10[1]
	mean rank	4	5	1	2	6	3
19	mean	-0.0048	-0.0495	-0.0495	-0.0472	-0.0291	-0.0495
	range	0.0207	0.0000	0.0000	0.0173	0.0495	0.0000
	mean rank	6	1	1	4	5	1
20	mean	-1.7627	-2.1229	-2.8617	-2.7206	-0.7853	-1.9992
	range	2.5220	1.3723	0.8939	0.9705	2.4338	2.4594
	mean rank	5	3	1	2	6	4
21	mean	-1.1639	-1.7390	-2.5339	-1.9477	-0.6482	-1.9144
	range	2.3088	3.2959	3.3103	3.9522	1.1022	3.1388
	mean rank	5	4	1	2	6	3
22	mean	-2.0836	-2.1139	-2.4082	-1.9625	-0.9900	-2.4700
	range	5.5080	8.1125	6.5471	4.4641	2.8435	5.8247
	mean rank	4	3	2	5	6	1
23	mean	-1.6253	-1.7754	-2.5307	-2.2080	-1.1519	-1.9228
	range	5.0674	3.0456	2.0553	2.6619	1.6755	3.4290
	mean rank	5	4	1	2	6	3

TABLE V
GROUP BASED COMPARISON

Cluster	GSO [21]	TIA [24]	DOA [25]	CWO [23]	HO [26]
1	7	6	6	7	7
2	6	5	5	5	6
3	10	6	2	4	10
Total	23	17	13	16	23

TABLE VI
RESULT WITH INCREASING MAXIMUM ITERATION

Function	Average Fitness Score		Improve Significantly
	$T = 20$	$T = 40$	
1	0.0000	0.0000	no
2	0.0000	0.0000	no
3	0.0001	0.0000	no
4	0.0000	0.0000	no
5	1.8912x10[1]	1.8933x10[1]	no
6	3.5255	3.5211	no
7	0.0046	0.0020	yes
8	-1.7712x10[3]	-1.8628x10[3]	no
9	0.0000	0.0000	no
10	0.0000	0.0000	no
11	0.0000	0.0000	no
12	0.9452	0.8266	no
13	2.8116	2.8643	no
14	1.0146x10[1]	8.3496	no
15	0.0071	0.0053	no
16	-0.9868	-1.0274	no
17	1.0599	0.4545	no
18	1.4345x10[1]	1.4077x10[1]	no
19	-0.0495	-0.0495	no
20	-2.2821	-2.3260	no
21	-1.8327	-2.6322	no
22	-2.6056	-2.4514	no
23	-2.4431	-3.3457	no

Table 6 and Table 7 provide the sensitivity assessment result of IEO in handling 23 functions. Table 6 provides the performance difference due to the increasing of maximum iteration from 20 to 40. Meanwhile, Table 7 provides the performance difference due to the increasing of population or swarm size from 10 to 20.

Table 6 shows that the performance improvement due to the increasing of maximum iteration from 20 to 40 does not occur in almost all functions. The significant improvement occurs only in f_7. Meanwhile, the stagnation for eight functions (f_1 to f_4, f_9 to f_{11}, and f_{18}) occurs because the final solution is close to the global optimal solution, or the global optimal solution has been achieved.

TABLE VII
RESULT WITH INCREASING POPULATION SIZE

Function	Average Fitness Score		Improve Significantly
	$n(X) = 10$	$n(X) = 20$	
1	0.0001	0.0000	no
2	0.0000	0.0000	no
3	0.8578	0.2488	yes
4	0.0220	0.0098	yes
5	1.8918x10[1]	1.8826x10[1]	no
6	3.0968	2.5308	no
7	0.0068	0.0045	no
8	-1.9971x10[3]	-2.3863x10[3]	no
9	0.2481	0.0007	yes
10	0.0021	0.0015	no
11	0.0002	0.0028	no
12	0.7584	0.4353	no
13	2.6800	2.5066	no
14	7.6320	5.0217	no
15	0.0061	0.0018	yes
16	-1.0298	-1.0315	no
17	0.4102	0.3995	no
18	8.0512	5.8768	no
19	-0.0495	-0.0495	no
20	-2.6461	-2.8424	no
21	-2.5481	-3.8214	no
22	-2.7357	-3.2213	no
23	-2.6233	-3.1876	no

Table 7 also shows that the significant improvement regarding the increasing of population size from 10 to 20 also does not occur in most of functions. The significant improvement occurs only in four functions (f_3, f_4, f_9, and f_{15}).

The second assessment is employing IEO to handle ELD problem. ELD problem is popular in electrical engineering field as this the optimization problem in power system. The system comprises several generating units or power plants that work together to supply the demand or load. Its general objective is minimizing the operational cost while in some cases it also minimizes the emission cost. As a practical problem, it also contains the constraints which are the equality and inequality constraints. The equality constraint is that the total generated power should be equal to the demand or load. In some cases, the spinning reserve and the power loss is considered. The inequality constraint is that the power of each generating unit should be within its range. In some cases, especially in multiple timeframes, the ramp rate is introduced as an extra limitation to avoid wider power difference between adjacent timeframes.

The formal model of ELD problem in this work is provided in (17) to (21). Equation (17) states that the objective is minimizing the total cost where of_{ELD} represents the objective function and co_{total} represents the total cost. Equation (18) states that the total cost is obtained by accumulating cost from all generating units where co_i represents the cost of generating unit i. Equation (19) represents the general quadratic function of the cost where p_i represents the power of generating unit i while a_1, a_2, and a_3 represent the constants of the quadratic function. Equation (20) states the inequality constraint where $p_{o,i}$ represents the lower power limit of generating unit i while $p_{hi,i}$ represents the higher power limit of generating unit i. Equation (21) states the equality constraint where the total power of the system should meet the load. In (21), p_{load} represents the load or demand.

$$of_{ELD} = \min(co_{total}) \quad (17)$$

$$co_{total} = \sum_{\forall i} co_i \quad (18)$$

$$co_i = a_{1,i} + a_{2,i}p_i + a_{3,i}p_i^2 \quad (19)$$

$$p_{lo,i} \leq p_i \leq p_{hi,i} \quad (20)$$

$$p_{load} = \sum_{\forall i} p_i \quad (21)$$

In this work, the use case for ELD problem is a set comprising 13 generating units. The lower power limits range from 0 MW to 60 MW. Meanwhile, the higher power limits range from 120 MW to 680 MW. The power demand is 2,600 MW. The detailed specification of each generating unit including the power limit and the constants for its quadratic cost function can be found in [39]. The assessment result is provided in Table 8.

TABLE VIII
RESULT FOR ELD PROBLEM WITH 13 UNITS

Metaheuristic	Mean (USD)	Range (USD)
GSO	24,849	82
TIA	24,815	43
DOA	24,797	50
CWO	24,812	72
HO	24,905	70
IEO	24,797	50

Table 8 exhibits that ELD problem is a highly competitive problem. The total cost disparity among metaheuristics is very narrow where the minimum average total cost is USD 24,797 while the maximum average total cost is USD 24,905. It means the disparity between the best result and the worst result is USD 108. IEO and DOA become the best performers in handling this problem. The minimum range is USD 43 which is obtained by TIA while the maximum range is USD 82 which is obtained by GSO. This result shows that the result fluctuation is verry low.

The third assessment is employing IEO to handle gear train problems. The gar train problem is a classic optimization problem in mechanical engineering field. The system consists of several number of gears which are connected to each other. The number of gears represents the dimension of the problem. Meanwhile, the number of teeth of each gear represents the value of the solution. It means that the value of the solution will be discrete.

In this work, the system consists of four gears as it can be found in [4]. The number of teeth ranges from 12 to 600. The objective function is minimizing the ratio of the output/input shaft angular velocity as presented in (22) where of_{GT} represents the objective function and g represents the number of teeth. Equation (23) states that the number of teeth ranges from 12 to 600 units. The result is provided in Table 9. The decimal point less than 10^{-4} is rounded to 0.

$$of_{GT} = \min\left(\left(\frac{1}{6.931} - \frac{g_2 g_3}{g_1 g_4}\right)^2\right) \quad (22)$$

$$12 \leq g_i \leq 600 \quad (23)$$

TABLE IX
RESULT FOR GEAR TRAIN DESIGN PROBLEM

Metaheuristic	Mean	Range
GSO	0.0252	0.3555
TIA	0.0000	0.0003
DOA	0.0000	0.0001
CWO	0.0000	0.0002
HO	0.0524	0.4299
IEO	0.0000	0.0000

Table 9 shows that IEO, TIA, CWO, and DOA perform better than GSO and HO. All these four metaheuristics can find the global optimal solution while the final solution of GSO and HO is much worse. Meanwhile, based on the range, IEO is better than TIA, DOA, and CWO as its range is lower than the others. It means that IEO produces result with lower fluctuation.

V. DISCUSSION

Overall, the assessment result shows the competitiveness of IEO in handling all three use cases: 23 standard functions, ELD problem, and gear train problem. The supremacy of IEO in handling all high dimension unimodal functions highlights the capability of IEO to perform exploitation as each of these functions has only one optimal solution. The supremacy of IEO in most high dimension multimodal functions highlights the capability of IEO to perform exploration as each of these functions has multiple optimal solutions so that the ability to avoid local entrapment plays a critical role. Meanwhile, the competitiveness of IEO in handling all fixed dimension multimodal functions highlights its capability in balancing the exploration and exploitation as each of these functions have multiple optimal solutions but fewer than the high dimension multimodal ones. But the terrain of these functions is ambiguous which provides different levels of entrapment.

The competitiveness of IEO is also shown in handling ELD problem. IEO still performs the best although with a close gap as the fierce competition in ELD problem. In general, a quadratic function is a unimodal function as it has only one optimal solution, the combination of several quadratic functions with various constants makes this problem turn to multimodal. The more difficult circumstance comes from the equality constraint that makes the search space narrower as the minimum power is difficult to reach.

The competitiveness of IEO also be found in the gear train problem. In general, gear train problem does not have any equality constraint so that the solution can be generated anywhere in the search space. But the gear train problem provides a specific characteristic which is a discrete problem. The solution should be presented in integer numbers rather than floating point so that the disparity among metaheuristics is difficult to achieve.

The result shows that strict acceptance approach is proven better than the loose acceptance approach. This strict acceptance is employed in TIA, DOA, CWO and IEO where they achieve equal results. Meanwhile, IEO is proven more stable than TIA, DOA, and CWO as its range is the lowest.

The sensitivity assessment result shows that IEO performs well in circumstances where the maximum iteration and population size are low. The increasing of maximum iteration or population size does not improve the result significantly in almost all functions. Meanwhile, in some functions, the increase of maximum iteration is better than the increase of the population size. On the other hand, in some other functions, the increase of population size gives higher yield rather than the increase of iteration. This circumstance is relevant to the no-free-lunch theory whereas the nature of the problem affects the quality of the produced solution as there is not any technique the is superior to solve all problems.

This work still has limitations despites it has provided acceptable result in providing the quasi-optimal solution and IEO is proven competitive compared to the existing metaheuristics. There are three limitations in this work that can be split into three aspects: technique, investigation, and use case. There are still a lot of available methods including the searching processes, role division, and the stagnation avoidance, but it is impossible to accommodate all these technique into single metaheuristic. This searching process can be a guided search or random direction search. IEO also does not employ role division whether it is the quality-based split or not while some others employ this technique. There are also many stagnations avoidance techniques such as full random search as in artificial bee colony (ABC) algorithm or transforming loose to strict acceptance approach as iteration goes like in simulated annealing (SA). This work also has not performed more complex assessment, such as the variety in dimension. This work also has not performed a single search investigation or missing search investigation to observe the significance of the existence of a search. In the use case aspect, this work also assesses IEO using three use cases. Meanwhile, there are a lot of optimization problems from various sectors that can be employed. There are also several common engineering design problems, such as pressure vessel, welded beam, spring, and speed reducer [46]. Besides, there are other standard functions such as CEC series which are also commonly employed in the studies proposing new metaheuristic.

The implementation of IEO in the optimization in farming system is also challenging as the optimization studies in farming are too few compared to in the manufacturing system. In the farming system, especially in the context of smart or precision farming, optimization is needed in three segments: the upstream, farm, and downstream. The upstream segment is highly related with the suppliers that provide seed, food, and other supporting stuff. In the farm, optimization is important to achieve efficiency of the resource and keep the operational cost low. In the downstream segment, optimization is needed to distribute the harvested products in more efficient and effective ways.

The implementation of IEO to optimize computational systems is also interesting, especially in the cloud system era. There are many studies that utilized metaheuristics to solve optimization problems in the cloud system. For example, the adaptive walrus optimization algorithm has been utilized to optimize the intrusion detection system while minimizing the computational complexity to reduce the time consumption [47]. GA and PSO has been utilized to optimize the scheduling process in mobile edge computing [48]. The black widow optimization algorithm has been hybridized with the chaos theory to optimize the load balancing process in the cloud computing system [49]. This example becomes the motivation of implementing IEO in the cloud computing in the future.

VI. Conclusion

A new metaheuristic called as include-exclude optimization (IEO) has been introduced in this paper. It proposes a novel technique in considering not only the quality of an agent but also its improving status. This technique is taken with the idea as an agent that still improves gives more probability for other agents to improve. On the other hand, it will be hard to improve by following other agents that fail to improve. The assessment to investigate the performance of IEO has been conducted by employing IEO to solve 23 standard functions, ELD problem, and gear train problem. The result shows the absolute supremacy of IEO in handling 23 standard functions compared to GSO and HO. Meanwhile, IEO is still superior to TIA, CWO, and DOA, especially in handling high dimension functions. The result also shows that IEO is superior to other metaheuristics with narrow gap in handling both ELD and gear train problems. In the future, studies can be conducted in three tracks: modifying the technique, performing more comprehensive investigation, and employing more various use cases, especially the farming system to achieve profitability and in the end the sustainability.

References

[1] H. M. Rosli, S. A. Halim, L. J. Awalin, and S. M. Mustaza, "Economic-emission load dispatch for power system operation using enhanced sunflower optimization," *Indonesian Journal of Electrical*

Engineering and Computer Science, vol. 27, no. 1, pp. 1–10, Jul. 2022, doi: 10.11591/ijeecs.v27.i1.pp1-10.
[2] V. K. Kamboj and O. P. Malik, "Optimal Unit Commitment and Generation Scheduling of Integrated Power System with Plug-In Electric Vehicles and Renewable Energy Sources," *Energies (Basel)*, vol. 17, no. 1, p. 123, Dec. 2023, doi: 10.3390/en17010123.
[3] M. H. Ali, Ahmed. M. A. Soliman, and S. K. Elsayed, "Optimal power flow using archimedes optimizer algorithm," *International Journal of Power Electronics and Drive Systems (IJPEDS)*, vol. 13, no. 3, pp. 1390–1405, Sep. 2022, doi: 10.11591/ijpeds.v13.i3.pp1390-1405.
[4] A. E. Ezugwu, J. O. Agushaka, L. Abualigah, S. Mirjalili, and A. H. Gandomi, "Prairie Dog Optimization Algorithm," *Neural Comput Appl*, vol. 34, no. 22, pp. 20017–20065, Nov. 2022, doi: 10.1007/s00521-022-07530-9.
[5] G. Ni and L. Chen, "Improved Scheduling for the Three-Machine Proportionate Open Shop and Mixed Shop Minimum Makespan Problems," *IEEE Access*, vol. 8, pp. 186805–186812, 2020, doi: 10.1109/ACCESS.2020.3030694.
[6] P. D. Kusuma, "Weighted Round Robin (WRR) Based Replenishment Model in Vendor Managed Inventory (VMI) System," *International Journal of Integrated Engineering*, vol. 14, no. 1, pp. 191–202, Mar. 2022, doi: 10.30880/ijie.2022.14.01.017.
[7] B. Ma, D. Hu, X. Chen, Y. Wang, and X. Wu, "The vehicle routing problem with speed optimization for shared autonomous electric vehicles service," *Comput Ind Eng*, vol. 161, p. 107614, Nov. 2021, doi: 10.1016/j.cie.2021.107614.
[8] A. Khan, H. Hizam, N. I. bin Abdul Wahab, and M. Lutfi Othman, "Optimal power flow using hybrid firefly and particle swarm optimization algorithm," *PLoS One*, vol. 15, no. 8, p. e0235668, Aug. 2020, doi: 10.1371/journal.pone.0235668.
[9] C. Paul, T. Sarkar, S. Dutta, and P. K. Roy, "Integration of optimal power flow with combined heat and power dispatch of renewable wind energy based power system using chaotic driving training based optimization," *Renewable Energy Focus*, vol. 49, p. 100573, Jun. 2024, doi: 10.1016/j.ref.2024.100573.
[10] R. A. Swief, N. M. Hassan, H. M. Hasanien, A. Y. Abdelaziz, and M. Z. Kamh, "Multi-Regional Optimal Power Flow Using Marine Predators Algorithm Considering Load and Generation Variability," *IEEE Access*, vol. 9, pp. 74600–74613, 2021, doi: 10.1109/ACCESS.2021.3081374.
[11] G. Xie, M. Zhang, M. Yang, and D. Wang, "Economic Dispatch of Isolated Microgrids Based on Enhanced Sparrow Search Algorithm," *Engineering Letters*, vol. 32, no. 4, pp. 753–760, 2024.
[12] Y. Zahraoui, I. Alhamrouni, S. Mekhilef, T. Korõtko, A. Jusoh, and T. Sutikno, "A Chameleon algorithm for solving economic dispatch problem in microgrid system," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 4, pp. 1982–1992, Aug. 2023, doi: 10.11591/eei.v12i4.4700.
[13] H. D. Nguyen and L. H. Pham, "Solutions of economic load dispatch problems for hybrid power plants using Dandelion optimizer," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 5, pp. 2569–2576, Oct. 2023, doi: 10.11591/eei.v12i5.5245.
[14] S. R. Spea, "Optimizing economic dispatch problems in power systems using manta ray foraging algorithm: an oppositional-based approach," *Computers and Electrical Engineering*, vol. 117, p. 109279, Jul. 2024, doi: 10.1016/j.compeleceng.2024.109279.
[15] T. Khobaragade and K. T. Chaturvedi, "Enhanced Economic Load Dispatch by Teaching–Learning-Based Optimization (TLBO) on Thermal Units: A Comparative Study with Different Plug-in Electric Vehicle (PEV) Charging Strategies," *Energies (Basel)*, vol. 16, no. 19, p. 6933, Oct. 2023, doi: 10.3390/en16196933.
[16] L. Zhang, Y. Yu, Y. Luo, and S. Zhang, "Improved cuckoo search algorithm and its application to permutation flow shop scheduling problem," *J Algorithm Comput Technol*, vol. 14, p. 174830262096240, Jan. 2020, doi: 10.1177/1748302620962403.
[17] M. S. Umam, M. Mustafid, and S. Suryono, "A hybrid genetic algorithm and tabu search for minimizing makespan in flow shop scheduling problem," *Journal of King Saud University - Computer and Information Sciences*, Sep. 2021, doi: 10.1016/j.jksuci.2021.08.025.
[18] D. Lei and H. Yang, "Scheduling unrelated parallel machines with preventive maintenance and setup time: Multi-sub-colony artificial bee colony," *Appl Soft Comput*, vol. 125, p. 109154, Aug. 2022, doi: 10.1016/j.asoc.2022.109154.
[19] Q. Liu, Y. Feng, B. Liu, J. Yang, and Z. Dong, "Optimal Sizing, Gear Ratios, and Shifting Schedule of Battery-Electric Mining Haul Trucks to Enhance Energy Efficiency," *Energy Technology*, vol. 12, no. 5, p. 2301123, May 2024, doi: 10.1002/ente.202301123.
[20] S. Gopi and P. Mohapatra, "Opposition-based Learning Cooking Algorithm (OLCA) for solving global optimization and engineering problems," *International Journal of Modern Physics C*, vol. 35, no. 05, p. 2450051, May 2024, doi: 10.1142/S0129183124500517.
[21] 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.
[22] T. Hamadneh *et al.*, "Addax Optimization Algorithm: A Novel Nature-Inspired Optimizer for Solving Engineering Applications," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 3, pp. 732–743, Jun. 2024, doi: 10.22266/ijies2024.0630.57.
[23] S. Alomari *et al.*, "Carpet Weaver Optimization: A Novel Simple and Effective Human-Inspired Metaheuristic Algorithm," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 4, pp. 230–242, Aug. 2024, doi: 10.22266/ijies2024.0831.18.
[24] 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.
[25] S. Al-Omari *et al.*, "Dollmaker Optimization Algorithm: A Novel Human-Inspired Optimizer for Solving Optimization Problems," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 3, pp. 816–828, Jun. 2024, doi: 10.22266/ijies2024.0630.63.
[26] S. O. Oladejo, S. O. Ekwe, and S. Mirjalili, "The Hiking Optimization Algorithm: A novel human-based metaheuristic approach," *Knowl Based Syst*, vol. 296, no. 5, p. 111880, Jul. 2024, doi: 10.1016/j.knosys.2024.111880.
[27] M. H. Amiri, N. Mehrabi Hashjin, M. Montazeri, S. Mirjalili, and N. Khodadadi, "Hippopotamus optimization algorithm: a novel nature-inspired optimization algorithm," *Sci Rep*, vol. 14, no. 1, p. 5032, Feb. 2024, doi: 10.1038/s41598-024-54910-3.
[28] J. O. Agushaka, A. E. Ezugwu, A. K. Saha, J. Pal, L. Abualigah, and S. Mirjalili, "Greater cane rat algorithm (GCRA): A nature-inspired metaheuristic for optimization problems," *Heliyon*, vol. 10, no. 11, p. e31629, Jun. 2024, doi: 10.1016/j.heliyon.2024.e31629.
[29] P. D. Kusuma, "Group Better-Worse Algorithm: A Superior Swarm-based Metaheuristic Embedded with Jump Search," *IAENG International Journal of Applied Mathematics*, vol. 54, no. 4, pp. 614–622, 2024.
[30] X. Wang, V. Snášel, S. Mirjalili, J.-S. Pan, L. Kong, and H. A. Shehadeh, "Artificial Protozoa Optimizer (APO): A novel bio-inspired metaheuristic algorithm for engineering optimization," *Knowl Based Syst*, vol. 295, p. 111737, Jul. 2024, doi: 10.1016/j.knosys.2024.111737.
[31] 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.
[32] M. S. Braik, "Chameleon Swarm Algorithm: A bio-inspired optimizer for solving engineering design problems," *Expert Syst Appl*, vol. 174, p. 114685, Jul. 2021, doi: 10.1016/j.eswa.2021.114685.
[33] 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.
[34] S. O. Oladejo, S. O. Ekwe, L. A. Akinyemi, and S. A. Mirjalili, "The Deep Sleep Optimizer: A Human-Based Metaheuristic Approach," *IEEE Access*, vol. 11, pp. 83639–83665, 2023, doi: 10.1109/ACCESS.2023.3298105.
[35] S. Gopi and P. Mohapatra, "Learning cooking algorithm for solving global optimization problems," *Sci Rep*, vol. 14, no. 1, p. 13359, Jun. 2024, doi: 10.1038/s41598-024-60821-0.
[36] K. M. Hosny, A. M. Khalid, W. Said, M. Elmezain, and S. Mirjalili, "A novel metaheuristic based on object-oriented programming concepts for engineering optimization," *Alexandria Engineering Journal*, vol. 98, pp. 221–248, Jul. 2024, doi: 10.1016/j.aej.2024.04.060.
[37] 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.
[38] 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.

[39] 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.

[40] P. D. Kusuma, "Iteration Controlled Mixture Optimizer: A New Metaheuristic and Its Application to Solve Economic Load Dispatch Problem," *IAENG International Journal of Applied Mathematics*, vol. 54, no. 6, pp. 1073–1082, 2024.

[41] 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.

[42] O. Alsayyed *et al.*, "Giant Armadillo Optimization: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems," *Biomimetics*, vol. 8, no. 8, p. 619, Dec. 2023, doi: 10.3390/biomimetics8080619.

[43] 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.

[44] 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.

[45] E. E. Hassan, H. I. M. Noor, M. R. Bin Hashim, M. F. Sulaima, and N. Bahaman, "Optimal economic environmental power dispatch by using artificial bee colony algorithm," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 13, no. 2, pp. 1469–1478, Jun. 2024, doi: 10.11591/ijai.v13.i2.pp1469-1478.

[46] X.-R. Zhao, X.-H. Chen, and J.-S. Wang, "An Improved Wild Horse Optimizer Incorporating Dual Weight Starvation Strategy and Randomized Convergence Factor," *Engineering Letters*, vol. 32, no. 9, pp. 1794–1813, 2024.

[47] K. V. K. Chithanya and V. L. Reddy, "Automatic intrusion detection model with secure data storage on cloud using adaptive cyclic shift transposition with enhanced ANFIS classifier," *Cyber Security and Applications*, vol. 3, p. 100073, Dec. 2025, doi: 10.1016/j.csa.2024.100073.

[48] Safaa Nouman Alali and Abdulkarim Assalem, "New Methods for Optimal Power Allocation and Joint Resource Scheduling in 5G Network which Use Mobile Edge Computing," *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 47, no. 2, pp. 237–265, Jun. 2024, doi: 10.37934/araset.47.2.237265.

[49] V. Hayyolalam and Ö. Özkasap, "CBWO: A Novel Multi-objective Load Balancing Technique for Cloud Computing," *Future Generation Computer Systems*, vol. 164, p. 107561, Mar. 2025, doi: 10.1016/j.future.2024.107561.

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.