International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Mathematical Modelling of Engineering Problems
Vol. 12, No. 7, July, 2025, pp. 2362-2372
Journal homepage: http://iieta.org/journals/mmep

Adaptive Iteration Algorithm: A Metaheuristic and Its Implementation on Economic Emission Dispatch Problem

Purba Daru Kusuma[1*], Tito Waluyo Purboyo[2]

[1] Computer Engineering Study Program, Telkom University, Bandung 40258, Indonesia
[2] Biomedical Engineering Study Program, Telkom University, Bandung 40258, Indonesia

Corresponding Author Email: purbodaru@telkomuniversity.ac.id

Copyright: ©2025 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

https://doi.org/10.18280/mmep.120716

Received: 24 March 2025
Revised: 20 June 2025
Accepted: 27 June 2025
Available online: 31 July 2025

Keywords:
optimization, stochastic, power system, economic emission dispatch, metaphor

ABSTRACT

Numerous novel metaheuristic algorithms have been proposed. Unfortunately, most of them are metaphor-inspired ones. This study introduces an innovative approach called the adaptive iteration algorithm (AIA). AIA is a metaphor-free metaheuristic. It provides novel adaptive techniques and utilizes iteration for choosing the strategy. This adaptive measure is implemented through the second search during the iteration where explorative search is taken if the first search fails to produce improvement. The performance of AIA is assessed by using two use cases: the 23 standard functions and the economic emission dispatch (EED) problem in Indonesia. In this assessment, AIA is confronted with five metaheuristics: hiking optimization (HO), crayfish optimization algorithm (COA), golden search optimization (GSO), lyrebird optimization algorithm (LOA), and osprey optimization algorithm (OOA). AIA outperforms HO, COA, GSO, LOA, and OOA in 23, 21, 21, 18, and 17 functions respectively out of the 23 functions. Meanwhile, AIA is also superior to all its comparative algorithms in the EED problem. It is concluded that AIA is the best among the compared algorithms in handling the 23 standard functions. GSO is the worst technique in handling the high-dimensional functions, while HO is the worst technique in handling the fixed-dimensional functions and the EED problem.

1. INTRODUCTION

Economic emission dispatch (EED) is a popular optimization study in power systems. It is the derivative of the economic dispatch (ED) problem [1]. EED is the multi-objective version of ED problem which the goal is to reduce both the operational expenses and the environmental impact costs. [2]. There have been a lot of studies in EED problems in recent years. In these studies, metaheuristic algorithms have become popular techniques to solve this problem. For example, salp swarm algorithm has been utilized to solve EED problem that integrates wind, solar, and thermal power units [3]. Grey wolf optimization (GWO) and crow search algorithm (CSA) have been combined to solve the mixed unit commitment-economic emission dispatch (UC-EED) with the use case is IEEE-39 bus system [4]. A derivative of non-dominated sorting genetic algorithm (NSGA II) has been utilized to solve EED problem where the system consists of wind and solar power [5]. The modified version of honey badger algorithm (HBA) has been utilized to solve EED problem in hydrogen microgrid system [5]. Particle swarm optimization (PSO) has been combined using non dominated sorting which is to find the Pareto-optimal in handling EED problem [6].

Numerous innovative approaches have emerged over the past few years. Many of them use metaphors, including white shark optimization (WSO) [7], elk herd optimization [8], zebra optimization algorithm (ZOA) [9], lyrebird optimization algorithm (LOA), [10], swarm magnetic optimization (SMO) [11], osprey optimization algorithm (OOA) [12], crayfish optimization algorithm (COA) [13], prairie dog optimization algorithm (PDO) [14], golden jackal optimization (GJO) [15], horse herd optimization (HHO) [16], red fox optimization (RFO) [17], fennec fox optimization (FFO) [18], addax optimization algorithm (AOA) [19], cheetah optimization (CO) [20], hiking optimization (HO) [21], Komodo mlipir algorithm (KMA) [22], and so on. Meanwhile, certain metaheuristics operate freely of any metaphorical inspiration, including golden search optimization (GSO) [23], fully informed search algorithm (FISA) [24], average-subtraction based optimization (ASBO) [25], subtraction-average based optimization (SABO) [26], quad tournament optimizer (QTO) [27], multiple interaction optimizer (MIO) [28], modified social forces algorithm (MSFA) [29], and so on.

There are several reasons that motivate the massive development of metaheuristic algorithms. First, there are various techniques and parameters that can be exploited to construct new techniques. These techniques can be the randomized mechanism, utilization of iteration, step size calculation, condition for action, target, and so on. Secondly, according to the No Free Lunch (NFL) theorem, no single method possesses sufficient strength to effectively address every problem. It makes many new methods employ multiple search approaches by combining more than one search that is conducted sequentially, separately, or conditionally. There are a lot of practical optimization problems so that new algorithms can be assessed using various use cases whether in their first or later appearance.

Despite the extensive development and utilization of techniques, there are certain unresolved problems in this area. First, the existence of metaphors has been criticized as camouflage of mere novelty. Second, there is high motivation to outperform the existing metaheuristics by introducing the new one rather than promoting novel approach. Third, the proportion of the assessment is far higher than the proportion of explanation of the technique. It seems many studies of introducing new metaheuristics focus on the assessment rather than the algorithm itself. Fourth, many new methods do not consider the adaptability of the technique. Many of them consider the quality of recent solutions but only a few of them consider the improvement or stagnation of the process and act regarding this circumstance. Fifth, many of these metaheuristics treat the iteration just as a counter although some of them treat the iteration to determine the observation space or step size. Moreover, only a few of them treat the iteration for determining the searching method that will be conducted.

Moreover, the studies that introduce a new technique tend to be thicker in paper length. This circumstance occurs because of the excessive assessment that the study aims to evaluate the effectiveness of the suggested approach. Despite its necessity, it is better that the test is not too extensive. For example, there are multiple standard use cases that are used in a single paper, for example multiple CEC series. On the other hand, the portion explaining the concept and the formalization of the proposed metaheuristic is too little. Meanwhile, more practical use cases in broader fields can be conducted in later studies rather than in the first publication of the techniques. Moreover, excessive assessments with a lot of benchmark techniques seem to be performance competition rather than investigating the contrast and resemblance between the suggested approach and prior methodologies.

Based on these unresolved problems, this study focuses on creating an innovative method known as the adaptive iteration algorithm (AIA) with certain characteristics. First, AIA should be free from metaphors so that its novel approach can be traced easily. Second, AIA employs adaptive techniques so that it behaves differently when facing improvement or stagnation. Third, AIA utilizes iteration not only as a counter for the iterative process but also controller to make decision of choosing the searching method. The presentation of this paper is conducted in a concise but comprehensive manner and avoids excessive assessment.

In the meantime, the contributions presented in this paper can be outlined as follows:

- This study presents an innovative technique that is free from metaphor and called adaptive iteration algorithm.
- AIA provides novel techniques in creating adaptability and iteration-controlled decision making.
- AIA is assessed by using two use cases: the 23 standard functions representing the unconstrained problem and the EED problem in Indonesia representing the constrained problem.
- AIA is confronted with five novel optimization algorithms inspired by metaheuristic principles: HO, COA, GSO, LOA, and OOA.

Below is the organization of the remainder of this paper. Section two provides discussion regarding the recent development of techniques including the searching method, the use of iteration, and the existence of the adaptive method. Section three provides a detailed description of the proposed AIA including the idea, pseudocode, flowchart, and mathematical formulation. Section four explains the assessment to investigate the performance of AIA including the use case; the scenario and outcomes are discussed. The fifth section elaborates on this aspect and provides a thorough analysis of the outcomes, discoveries, and constraints is presented. The sixth section outlines the conclusion and potential directions for future research.

2. RELATED WORKS

Numerous innovative approaches have emerged over the past few years. Most of them use metaphors as inspiration, especially animal behavior. Besides, swarm intelligence becomes the favorite framework although some of them still use the evolution system as a framework. Both frameworks are population-based systems. It means that the system consists of several entities. The basic difference is that in swarm intelligence, all entities are active and autonomous on finding improvement in every iteration based on the movement with certain direction within the space. On the other hand, in evolution-based methods, the improvement is conducted by cross overing certain number of entities to each other. In most cases, not all entities perform crossover. In many cases, crossover is conducted by the high-quality members. Then, population selection is conducted to determine which members will survive to the next iteration.

By abstracting the metaphor, swarm intelligence-based metaheuristic algorithms have similarities and differences. In general, targeted exploration serves as the fundamental framework for search methodologies. This approach has four components: the moving entity or agent, reference, direction, and step size. The moving entity is the entity that moves during the searching process. The reference is the entity that becomes the guidance for the moving entity to move. The direction is the vector that is taken by the moving entity relative to the target. Step size is the length of motion that is taken by the moving entity.

There are several common references in swarm-based metaheuristics. The best agent becomes the most popular reference. This best agent can be the global-best agent like in coati optimization algorithm (COA) [30] or KMA [22], local best agent, or combination of both like in marine predator algorithm (MPA) [31] or GSO [23]. This global best agent can be the global best agent in the recent iteration, global best agent so far until the recent iteration, or the combination of both like in COA [13]. The other references can also be several best agents where the number of best agents is static like in grey wolf optimization (GWO) [32] and GJO [15] or manually determined like in MIO [28]. There are also other common references like a randomly selected agent or other agent like in zebra optimization algorithm (ZOA) [9], a randomly solution within space, like in COA [30], and so on. Each of these references has strengths and weaknesses where some references tend to be exploitation-oriented reference while the others tend to be exploration-oriented reference.

Certain metaheuristic algorithms utilize a combination of search strategies, whereas others rely on a singular search method. The multiple search approach is chosen in many recent methods to accommodate the strength and weakness of each searching method. This method of conducting multiple searches can be implemented in various manners, including sequential execution, optional selection, iterative processes, or division into swarm segments.

KMA becomes the example of multiple search-based metaheuristic that employs multiple searches in two ways: swarm-split and optional execution. The swarm split is conducted by splitting the swarm into three groups: the high-quality agents called big males, the moderate quality agents called females, and the low-quality agents called small males [22]. Meanwhile, the optional mechanism is conducted as each female has two options with equal probability: crossover with highest quality agent or move anywhere within space [22].

Regarding this explanation, there are several notes in developing a new technique due to the shortcomings of many recent techniques. First, this proposed technique should be free from metaphor so that its true approach can be clearly investigated and understood. As mentioned previously, there are common methods in many techniques where each common method has many names or terms in the metaphor-inspired techniques. Second, it is challenging to create a new technique that has adaptive mechanisms to handle both improvement and stagnation as the adaptive approach is rare to find in many techniques so that the exploration and exploitation are conducted in the fixed rule where the improvement or stagnation is neglected in the decision-making process. Third, it is interesting to exploit iteration not only as a counter but also part of decision making as also commonly found in many techniques. Fourth, it's important to challenge this technique not only using standard or classic use cases but also the practical and contemporary ones as many techniques were assessed using standard use cases in their first appearance.

3. PROPOSED MODEL

The proposed adaptive iteration algorithm (AIA) is constructed based on swarm intelligence framework. AIA comprises multiple self-governing agents that operate independently, striving to identify the optimal solution in each iteration. But collaboration among agents is conducted to boost its performance. As previously mentioned, AIA is developed as a multiple search-based technique. This approach is conducted sequentially and optionally.

The novel approach of AIA relies on the adaptive and iteration terms. Regarding the first term, AIA performs exploitation-oriented search when it faces improvement and performs exploration-oriented search when it faces stagnation. Enhancement occurs when the agent discovers a superior solution compared to its previous search. Conversely, if the agent is unable to identify a more optimal solution during the prior search, it is considered unsuccessful. Regarding the second term, AIA also utilizes iteration to determine the searching method.

Every agent performs two stages that are performed sequentially in every iteration. In both stages, iteration controls the searching method that is taken. Meanwhile, the adaptive mechanism is taken only in the second stage.

In the first stage, there are two options. The initial choice involves moving in the direction of a randomly selected agent from a group comprising the better agents along with the best agent. The second option is the motion toward the best agent. In the early iteration, the agent tends to choose the first option. Then the probability of the first option to be chosen declines as iteration goes. Then, in the late iteration, the agent tends to choose the second option. This strategy remarks on the shift from exploration to exploitation where iteration becomes the controller.

In the second stage, there are three options. The initial choice involves moving in the direction of the best agent. The next option entails the best agent shifting away from its associated counterpart. Lastly, the third alternative represents movement in relation to a randomly selected agent. When improvement occurs in the first stage, the agent may choose the first or second option. At the beginning of the process, the likelihood of selecting the first option is considerable. However, as the iterations progress, this probability gradually decreases. Then, this probability of the second option is high in the late iteration. When stagnation occurs in the first stage, the agent chooses the third option.

Stringent acceptance is applied in both stages. A solution candidate is generated in every stage. Only when a superior solution candidate is found can it take the place of the agent's current value.

The AIA framework is defined through both pseudocode and mathematical expressions, with the pseudocode detailed in Algorithm 1 and Algorithm 2. Algorithm 1 shows the process of the whole optimization process while algorithm 2 shows the process in the second stage. Meanwhile, the mathematical formulation is presented from Eq. (1) to Eq. (11). The list of notations that are used in this paper is provided in nomenclatures.

Algorithm 1: Adaptive iteration algorithm

start
 for i=1 to $n(S)$
  initialize s_i
  update s_{best}
 end for
 for t=1 to t_m
  for i=1 to $n(S)$
   $s_{prev} \leftarrow s_i$
   perform first stage and update s_{best}
   update *imp*
   perform second stage and update s_{best}
  end for
 end for
 return s_{best}
stop

Algorithm 2: Second stage process

start
 if *imp* = 1 then
  perform improving search
 else
  perform stagnation search
 end if
stop

The optimization process begins with the initialization phase. During this stage, agents are initially distributed uniformly across the defined space, as described in Eq. (1). Subsequently, with each new agent generated, the optimal agent is continuously updated following Eq. (2). The rationale of choosing uniformity is to provide equal opportunity in the beginning of the optimization process as the location of the optimal solution or the trend of it is still unknown. The s_{best}' in Eq. (2) represents the best agent after updating process.

$$s_{i,j} = b_{low,j} + r_1(b_{up,j} - b_{low,j}) \quad (1)$$

$$s_{best}' = \begin{cases} s_i, f(s_i) < f(s_{best}) \\ s_{best}, else \end{cases} \quad (2)$$

The first stage is formalized using Eq. (3) to Eq. (6). Eq. (3) formalizes the construction of a pool that comprises all better agents plus the best agent. Then, Eq. (4) formalizes the random selection from this pool. Eq. (5) illustrates the process of generating the initial solution candidate, which is derived from both the first and second options, with stochastic control applied through iterative computation. The stringent acceptance of the agent based on the first solution candidate is formalized using Eq. (6). The s_i' in Eq. (6) represents the agent after updating process using the first candidate.

$$S_{bet,i} = \{s_k \in S \wedge f(s_k) < f(s_i)\} \cup s_{best} \quad (3)$$

$$s_{sel1} = r_3(S_{bet,i}) \quad (4)$$

$$c_{1,j} = \begin{cases} s_{i,j} + r_1(s_{sel1,j} - r_2 s_{i,j}), r_1 > \frac{t}{t_m} \\ s_{i,j} + r_1(s_{best,j} - r_2 s_{i,j}), else \end{cases} \quad (5)$$

$$s_i' = \begin{cases} c_1, f(c_1) < f(s_i) \\ s_i, else \end{cases} \quad (6)$$

$$imp = \begin{cases} 1, f(s_i) < f(s_{prev}) \\ 0, else \end{cases} \quad (7)$$

The improving status is updated by using Eq. (7). Based on algorithm 1, the current solution of the agent is stored before performing the first stage. Then, the quality of this stored value is compared with the quality of the agent after performing the first stage. The status is set to 1 if improvement takes place. Otherwise, the status is set to 0.

$$c_{2,j} = \begin{cases} s_{i,j} + r_1(s_{best,j} - r_2 s_{i,j}), r_1 > \frac{t}{t_m} \\ s_{best,j} + r_1(s_{best,j} - s_{i,j}), else \end{cases} \quad (8)$$

$$s_{sel2,i} = r_3(S) \quad (9)$$

$$c_{2,j} = \begin{cases} s_{i,j} + r_1(s_{sel2,j} - r_2 s_{i,j}), f(s_{sel2}) < f(s_i) \\ s_{i,j} + r_1(s_{i,j} - s_{sel2,j}), else \end{cases} \quad (10)$$

$$s_i' = \begin{cases} c_2, f(c_2) < f(s_i) \\ s_i, else \end{cases} \quad (11)$$

The methodology in the second phase is structured through Eq. (8) and Eq. (9). The enhancement of the search process is defined by Eq. (8), where the second solution candidate is generated either by moving toward the best agent or by the best agent moving away from another agent. The exploration during stagnation is described by Eq. (9) and Eq. (10). Specifically, Eq. (9) represents the selection of a random agent, while Eq. (10) outlines the agent's movement—either approaching the chosen agent if it exhibits superior performance or distancing itself if the selected agent is inferior. Lastly, Eq. (11) establishes the strict acceptance criteria for the agent based on the second solution candidate. The s_i' in Eq. (11) represents the agent after updating process using the second candidate.

According to this clarification, the intricacy of AIA is demonstrated as $O(n(S).d)$ during the initialization and $O(n(S)^2.d.t_m)$ throughout the repetitive process. The complexity scales linearly with either the swarm size or the dimensionality during the initialization phase. Meanwhile, the complexity is linear to the maximum iteration and dimension but quadratic proportional to the swarm size.

4. RESULTS

This section provides an assessment of the performance of AIA. In this paper, AIA is tested by using two use cases: standard use case and practical use case. The group of 23 standard functions serves as the typical application scenario, whereas the EED problem within the Java-Bali power grid exemplifies real-world implementation. The first use case represents the unconstrained problem where the solution can be placed anywhere within the solution space. On the other hand, the second use case represents the constrained problem where the solution cannot be placed anywhere within space as the solution in certain dimension depends on the solution in other dimensions. In this paper, the formalization of the use cases is not presented as they can be found in many previous works so that presenting the model and formalization may produce redundancy.

In this evaluation, AIA encounters five novel metaheuristic algorithms: HO, COA, GSO, LOA, and OOA. These five metaheuristics are chosen as they represent various techniques or approaches. HO, COA, and GSO are metaheuristics that do not employ stringent acceptance while LOA and OOA are the metaheuristics that employ stringent acceptance. HO, COA, and GSO are metaheuristics that employ only directed search while LOA and OOA employ both directed search and local search. HO uses single reference which is the best agent. COA uses two references: the best agent so far and the middle between the best agent so far and the best agent in the current iteration. GSO uses two references: the global-best agent and the local best agent. LOA uses a single reference which is a randomly chosen better agent. COA also uses a single reference: a randomly chosen agent from the pool that consists of all the better agents plus the best agent.

In the first case, AIA is challenged to solve the set of 23 standard functions. This case is popular and widely used as standard functions to assess new techniques, such as KMA, GSO, TIA, and so on. The widespread adoption of this application arises from the diverse conditions present within these functions. It consists of 13 high-dimensional functions and 10 fixed dimensional functions. It also consists of 7 unimodal functions and 16 multimodal functions. A unimodal function refers to a function possessing a single optimal solution. In contrast, a multimodal function contains multiple optimal solutions, with one being the global optimum while the rest are local optimum. A comprehensive explanation of these functions is available in reference [23] including the mathematical formulation, optimal value, and the solution space.

In this case, the dimension of the high dimensional functions is set to 50. In the meantime, the maximum iteration is defined as 30, whereas the swarm size is configured to 10. The result for this first uses case can be found from Tables 1-5. In this evaluation, any decimal value smaller than 10^{-4} is approximated as zero. There are 20 independent runs for each function and each technique in this first case. This number is justified as the result shows stability in this low number of independent runs.

Table 1 shows the assessment result on handling seven high dimensions unimodal functions. This result reveals the supremacy of AIA in handling these functions. AIA produces the best result of all seven functions. Moreover, it attains the global optimum in both F1 and F2. Additionally, three other metaheuristic algorithms including COA, LOA, and OOA are also capable of identifying the global optimum for F2. Conversely, GSO performs the poorest, yielding the lowest accuracy across six functions (F1 to F6) and ranking as the second least effective method for F7. HO becomes the second worst technique. This result also reveals the wide disparity between the best and worst technique.

Table 2 shows the assessment result on handling six high dimension multimodal functions. In these functions, AIA is still superior compared to its confronters. AIA produces the best result in fixed functions (F9 to F13). Meanwhile, it becomes the fourth best in handling F8. AIA successfully attains the global optimum for two functions (F9 and F10). Conversely, GSO remains the least effective method, yielding the poorest performance across four functions (F10 to F13). The performance gap between the most effective and least effective techniques is substantial in five functions but relatively small in one function (F8).

Table 3 shows the assessment result on handling ten fixed dimension multimodal functions. In these functions, AIA is also still superior as it achieves the best result in five functions (F15 to F17, F21, and F23) and the second-best result in four functions (F14, F19, F20, and F22). AIA also achieves the global optimal in two functions (F16 and F17). On the other hand, HO becomes the worst techniques as it achieves the worst result in eight functions (F14-F16, F18, and F20-F23) and second worst result in two functions (F17 and F19). The result also reveals the narrow disparity between the best technique and the worst technique in almost all ten functions.

Table 1. Result on high dimension unimodal functions

F	Parameter	HO	COA	GSO	LOA	OOA	AIA
1	mean	5.8729×10^{1}	0.1190	2.5744×10^{4}	3.9040	0.0003	0.0000
	standard deviation	3.7494×10^{1}	0.5176	5.9146×10^{3}	2.4535	0.0002	0.0000
	mean rank	5	3	6	4	2	1
2	mean	0.0177	0.0000	6.2169×10^{23}	0.0000	0.0000	0.0000
	standard deviation	0.0812	0.0000	2.7676×10^{64}	0.0000	0.0000	0.0000
	mean rank	5	1	6	1	1	1
3	mean	1.4899×10^{3}	3.2969×10^{1}	6.3120×10^{4}	1.1218×10^{4}	3.0434×10^{2}	1.2412×10^{1}
	standard deviation	1.0568×10^{3}	1.3534×10^{2}	2.4666×10^{4}	6.7414×10^{3}	4.3642×10^{2}	2.6859×10^{1}
	mean rank	4	2	6	5	3	1
4	mean	2.9604	0.0126	4.2960×10^{1}	2.3279	0.0427	0.0010
	standard deviation	0.6251	0.0248	4.3740	1.1228	0.0183	0.0008
	mean rank	5	2	6	4	3	1
5	mean	6.0544×10^{4}	4.8991×10^{1}	2.4887×10^{7}	8.8788×10^{1}	4.8928×10^{1}	4.8811×10^{1}
	standard deviation	5.2482×10^{4}	0.0147	9.8078×10^{6}	3.5022×10^{1}	0.0239	0.0966
	mean rank	5	3	6	4	2	1
6	mean	1.0235×10^{2}	1.5650×10^{1}	2.1274×10^{4}	1.3874×10^{1}	1.0085×10^{1}	8.5402
	standard deviation	4.9019×10^{1}	1.6518×10^{1}	5.7468×10^{3}	2.7474	0.5201	0.4606
	mean rank	5	4	6	3	2	1
7	mean	3.8201×10^{2}	0.0370	1.4995×10^{1}	0.0462	0.0183	0.0079
	standard deviation	2.3212×10^{2}	0.0448	4.3672	0.0368	0.0081	0.0045
	mean rank	6	3	5	4	2	1

Table 2. Result on high dimension multimodal functions

F	Parameter	HO	COA	GSO	LOA	OOA	AIA
8	mean	-1.6074×10^{3}	-2.8215×10^{3}	-4.2115×10^{3}	-4.1872×10^{3}	-5.1989×10^{3}	-3.7511×10^{3}
	standard deviation	6.5121	1.2485×10^{3}	9.5979×10^{2}	6.7849×10^{2}	5.1375×10^{2}	4.7292×10^{2}
	mean rank	6	5	2	3	1	4
9	mean	5.5057×10^{2}	0.4605	3.6017×10^{2}	4.7749×10^{1}	0.0014	0.0000
	standard deviation	5.0791×10^{1}	2.1487	2.5652×10^{1}	5.3420×10^{1}	0.0020	0.0000
	mean rank	6	3	5	4	2	1
10	mean	6.1579	0.0052	1.6660×10^{1}	1.1530	0.0035	0.0000
	standard deviation	1.2128	0.0107	0.7982	0.7530	0.0009	0.0000
	mean rank	5	3	6	4	2	1
11	mean	0.4216	0.0876	2.3043×10^{2}	0.4940	0.0071	0.0007
	standard deviation	0.1662	0.3189	4.5628×10^{1}	0.3274	0.0297	0.0034
	mean rank	4	3	6	5	2	1
12	mean	5.3729	1.3559	1.4930×10^{7}	1.0706	0.8470	0.6623
	standard deviation	1.9022	0.1406	1.2733×10^{7}	0.1748	0.1052	0.0793
	mean rank	5	4	6	3	2	1
13	mean	4.4155	3.3280	6.9271×10^{7}	4.0510	3.1684	2.9412
	standard deviation	3.4172	0.6004	3.1828×10^{7}	0.3688	0.0698	0.0568
	mean rank	5	3	6	4	2	1

Table 3. Result on fixed dimension multimodal functions

F	Parameter	HO	COA	GSO	LOA	OOA	AIA
14	mean	1.2769×10^1	9.0907	1.0504×10^1	6.0583	4.9225	5.0757
	standard deviation	0.2095	3.8382	4.3744	3.1060	2.4433	3.3486
	mean rank	6	4	5	3	1	2
15	mean	0.0566	0.0357	0.0192	0.0017	0.0020	0.0009
	standard deviation	0.0377	0.0366	0.0233	0.0012	0.0043	0.0011
	mean rank	6	5	4	2	3	1
16	mean	-0.1084	-0.6309	-1.0316	-1.0292	-1.0310	-1.0316
	standard deviation	0.9511	0.3652	0.0000	0.0036	0.0007	0.0000
	mean rank	6	5	1	4	3	1
17	mean	0.7573	1.5738	0.4047	0.4016	0.3981	0.3980
	standard deviation	0.2580	2.9144	0.0277	0.0036	0.0000	0.0000
	mean rank	5	6	4	3	2	1
18	mean	2.5750×10^2	2.1141×10^1	8.2033	4.0363	3.0016	5.4545
	standard deviation	2.3161×10^2	2.8664×10^1	1.8155×10^1	4.7204	0.0035	7.7619
	mean rank	6	5	4	2	1	3
19	mean	-0.0428	-2.2143	-0.0342	-0.0495	-0.0495	-0.0495
	standard deviation	0.0116	0.1123	0.0181	0.0000	0.0000	0.0000
	mean rank	5	1	6	2	2	2
20	mean	-0.7481	-1.7841	-2.8434	-3.1064	-3.2137	-3.1487
	standard deviation	0.5285	0.6855	0.3589	0.1446	0.0522	0.1219
	mean rank	6	5	4	3	1	2
21	mean	-0.7662	-0.8965	-5.7165	-5.3179	-3.7228	-6.2757
	standard deviation	0.3822	0.9895	2.6717	1.7993	1.7221	1.9302
	mean rank	6	5	2	3	4	1
22	mean	-1.0591	-1.4411	-4.8725	-5.0581	-4.7277	-5.0523
	standard deviation	0.5072	0.9057	3.0798	1.7269	1.8102	2.0958
	mean rank	6	5	3	1	4	2
23	mean	-1.1842	-1.8559	-3.9834	-4.7913	-4.4489	-6.2953
	standard deviation	0.5544	0.9512	1.9768	1.7919	1.8072	2.5109
	mean rank	6	5	4	2	3	1

Table 4. p-value score based on t-test of the comparison between AIA and its benchmarks

F	HO	COA	GSO	LOA	OOA
1	0.000	0.314	0.000	0.000	0.000
2	0.312	1.000	0.314	1.000	1.000
3	0.000	0.500	0.000	0.000	0.003
4	0.000	0.034	0.000	0.000	0.000
5	0.000	0.000	0.000	0.000	0.000
6	0.000	0.050	0.000	0.000	0.000
7	0.000	0.004	0.000	0.000	0.000
8	0.000	0.002	0.060	0.020	0.000
9	0.000	0.313	0.000	0.000	0.001
10	0.000	0.025	0.000	0.000	0.000
11	0.000	0.301	0.000	0.000	0.322
12	0.000	0.000	0.000	0.000	0.000
13	0.064	0.004	0.000	0.000	0.000
14	0.000	0.000	0.000	0.323	0.865
15	0.000	0.000	0.000	0.041	0.264
16	0.000	0.000	0.181	0.004	0.001
17	0.000	0.065	0.295	0.002	0.000
18	0.000	0.017	0.525	0.468	0.146
19	0.012	0.000	0.000	1.000	1.000
20	0.000	0.000	0.000	0.299	0.027
21	0.000	0.000	0.439	0.096	0.000
22	0.000	0.000	0.826	0.992	0.587
23	0.000	0.000	0.001	0.027	0.007

Table 5. Group based supremacy of AIA

Cluster	HO	COA	GSO	LOA	OOA
1	7	6	7	6	6
2	6	6	5	5	5
3	10	9	9	7	6
Total	23	21	21	18	17

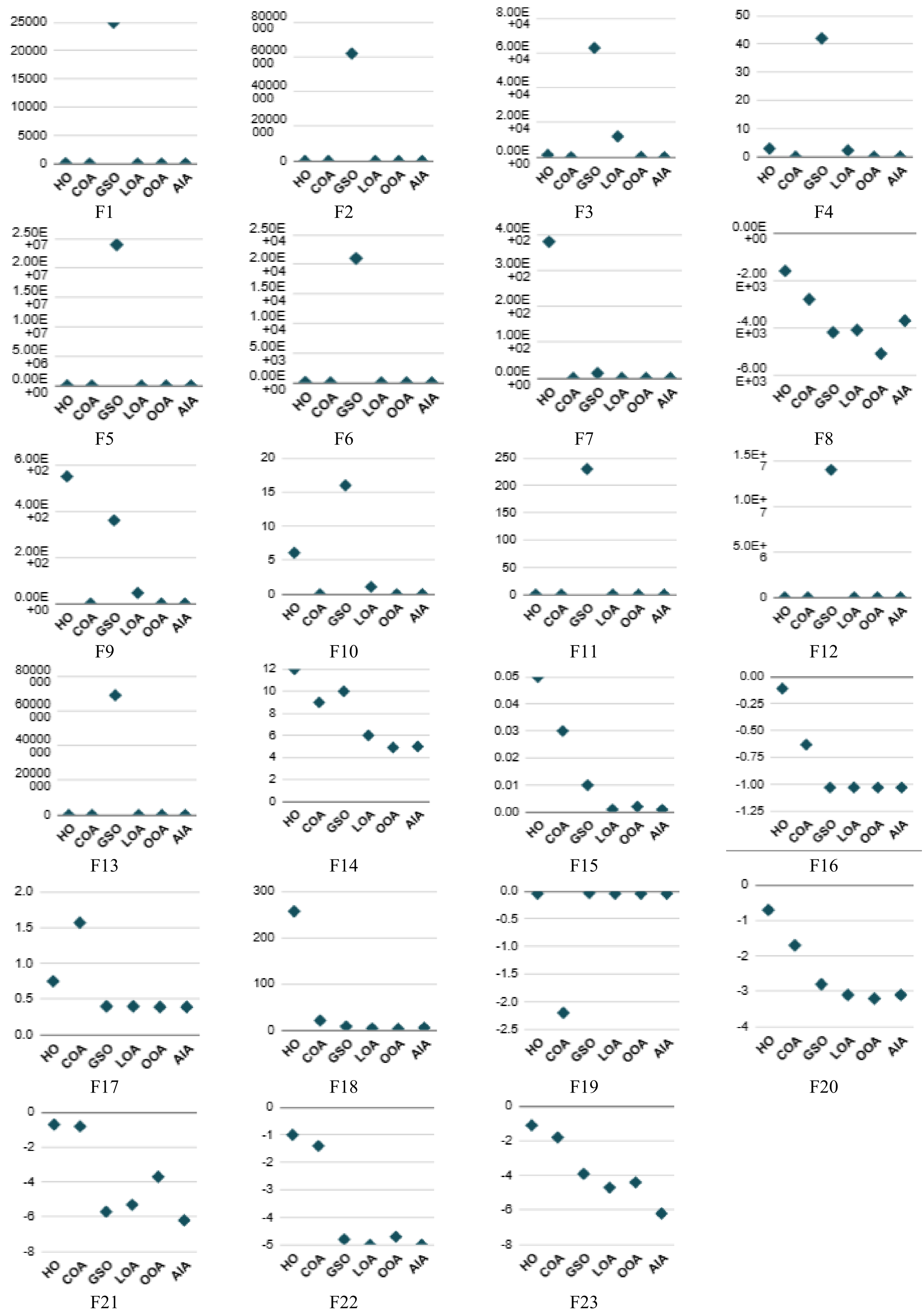


Figure 1. Mean on handling 23 standard functions

Table 4 provides the significance result based on the p-value that is acquired using t-test. In this t-test, AIA is compared with its benchmarks. The p-value which is less than 0.05 remarks the significant difference between AIA and its benchmark. Meanwhile, the p-value which is equal to or higher than 0.05 remarks the non-significant difference between AIA and its benchmark. Based on this explanation, AIA is different significantly compared to HO, COA, GSO, LOA, and OOA in 21, 16, 17, 16, and 16 functions.

Table 5 strengthens the supremacy of AIA compared to other metaheuristics in handling the 23 standard functions. It is better than HO, COA, GSO, LOA, and OOA in 23, 21, 21, 18, and 17 functions consecutively. This result means AIA is absolute better than HO and almost absolute better than COA and GSO.

Figure 1 shows that there is a significant gap between the worst technique and the other techniques in the high dimension functions. The gap between GSO and the other techniques in F1 to F6 and F10 to F13 is high. On the other hand, Figure 2 also shows that the gap among techniques is narrow in handling the fixed dimension functions.

The second use case is the EED problem in Java-Bali electricity system in Indonesia. This system is the biggest one in Indonesia as it provides electricity for the most populous and industrialized region in Indonesia. It contains eight power plants, six of them are thermal power plants and the rest two are hydroelectric power plants.

The EED problem is a multi-objective problem where its objective is reducing both operational cost and emission cost. It contains two constraints: the equality constraint and the inequality constraint. The equality constraint states that the total power that is produced by the system should be equal to the power demand. Meanwhile, the inequality constraint states that the power that is produced by each power plant should be within the power range of the power plant. The model, mathematical formulation, and the cost constant of this EED problem can be found in reference [2].

In this assessment, there are four power demand scenarios: 12,228 MW; 12,863 MW; 13,108 MW; and 13,096 MW. These scenarios refer to [2] where they represent the actual load of peak hours from 18.00 to 21.00 obtained on June 14, 2014. The result is provided in Tables 6-9. Like in the first case, the swarm size is set to 10 and maximum iteration is set to 30. Like in the first case, there are 20 independent runs for every scenario and every technique with the same reasoning as in the first case.

Table 6. Mean and standard deviation on handling EED with 12,228 MW demand

Metaheuristic	Mean	Standard Deviation	Rank
HO	22,773,462,758	844,563,562	6
COA	22,150,790,659	593,851,032	5
GSO	21,453,604,830	440,608,764	4
LOA	21,138,064,112	172,145,219	3
OOA	21,057,809,930	122,554,283	2
AIA	20,654,579,368	72,824,954	1

Table 7. Mean and standard deviation on handling EED with 12,863 MW demand

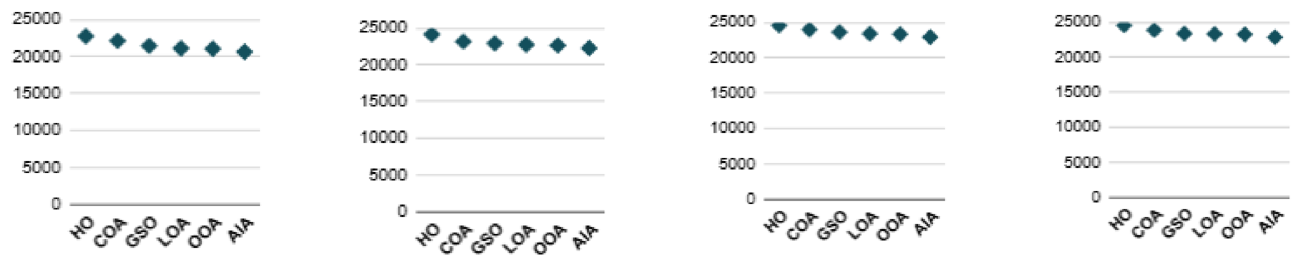
Metaheuristic	Mean	Standard Deviation	Rank
HO	24,150,497,082	700,743,707	6
COA	23,207,933,717	602,064,090	5
GSO	22,939,966,295	253,533,730	4
LOA	22,762,388,426	221,139,144	3
OOA	22,679,233,377	138,863,621	2
AIA	22,305,494,174	65,435,999	1

Table 8. Mean and standard deviation on handling EED with 13,108 MW demand

Metaheuristic	Mean	Standard Deviation	Rank
HO	24,599,876,167	672,548,588	6
COA	24,013,353,210	709,170,702	5
GSO	23,663,797,782	362,394,912	4
LOA	23,401,829,892	162,566,648	3
OOA	23,363,039,718	119,900,158	2
AIA	22,942,691,779	49,450,402	1

Table 9. Mean and standard deviation on handling EED with 13,096 MW demand

Metaheuristic	Mean	Standard Deviation	Rank
HO	24,571,829,121	629,955,718	6
COA	23,910,355,011	591,248,630	5
GSO	23,422,580,577	420,128,554	4
LOA	23,359,607,355	161,379,789	3
OOA	23,283,689,221	113,294,632	2
AIA	22,900,490,278	30,310,629	1



(a) 12,228 MW demand (b) 12,863 MW demand (c) 13,108 MW demand (d) 13,096 MW demand

Figure 2. Mean on handling EED problem

General motion is employed to handle the inequality constraint. This inequality constraint is like the independent solution space in the first case. It means that when the value of a generating unit is above the maximum power, then this value is set to its maximum power. On the other hand, when the value of a generating unit is below the minimum power, then this value is set to its minimum power. After the inequality constraint is handled, the next fixation is to handle the equality constraint.

The round robin mechanism is employed to handle the equality constraint in this EED problem. If the total power of the system is above the power demand, then the iterative decrease is applied. On the other hand, if the total power of the system is below the power demand, the iterative increase is applied. A it uses round robin; the circular increase or decrease is applied where the circulation starts from the first generating unit. The maximum increase/decrease is 10 MW for every turn and the circulation stops after the equality constraint is met. Besides, the power range for each generating unit is considered. If the decreasing mechanism is applied, the generating unit whose produced power is equal to its minimum power is skipped. On the other hand, if the increasing mechanism is applied, the generating unit whose produced power is equal to its maximum power is skipped.

Tables 6-9 reveal the supremacy of AIA in handling the EED problem. It achieves the best result in all four scenarios. On the other hand, HO is consistent as the worst technique in this EED problem. The result also shows the very narrow disparity between the best and the worst techniques which means the competition among techniques is fierce. The result also shows the very low standard deviation representing the stability of the result by every technique. This very narrow gap among the techniques is also shown in Figure 2.

5. DISCUSSION

The findings indicate that AIA demonstrates strong performance in managing both the 23 benchmark functions and the EED problem, marking it as an initial discovery. The result in the first use case shows that AIA has good exploitation capability, exploration capability, and balancing capability between exploitation and exploration. This capability can be traced based on the result on handling the high dimension unimodal functions for exploitation capability, high dimension multimodal functions for exploration capability, and fixed dimension multimodal functions for exploration/exploitation balancing capability. The result in the second case reveals that AIA also performs well in tackling the constrained problem with fierce competition.

The result also reveals that the multiple search approach is better than the single search one as the second finding. GSO and HO are the metaheuristic that employ single search approach while COA, LOA, OOA, and AIA are metaheuristic that employ multiple search approach. COA and LOA employ multiple searches based on options. OOA employs multiple searches sequentially. AIA employs multiple searches based on option and sequence.

The third finding is that stringent acceptance still performs better than loose acceptance. This circumstance occurs in almost all cases, especially in 23 standard functions. AIA, OOA, and LOA tend to be better than HO, COA, and GSO.

The fourth finding is that the NFL theory is still relevant. AIA is proven as superior metaheuristics compared to its confronters. But its performance is not superior in all problems. In some functions like F8, its performance is moderate. By investigating the terrain of F8, this function has a lot of global optimal, and these global optimal solutions are distributed across the solution space. It is different with many other multimodal functions that consist of multiple optimal solutions but there is a trend for the global optimal solution. On the other hand, GSO can find the global optimal in handling F16 although its performance is poor in many other functions. The NFL also influences the degree of variation in performance gaps among different techniques, with some functions exhibiting significant disparities while others show only moderate or minimal differences.

Moreover, the superiority of AIA cannot be concluded that it comes from single reason. This superiority comes from a whole package of the approach including the multiple searching mechanism, stringent acceptance, selection of the reference, step size, and the adaptive mechanism. As also relevant to the NFL theory, a single approach may perform superior in some cases but expose inferiority in other cases.

The analysis regarding the exploitation and exploration of AIA can be traced back to the formalization. In general, the first stage is exploitation in its nature. But the second option is more exploitative rather than the first option because the best agent is used in the second option while a randomized better agent is used in the first option. In the second stage, the first and second option is also exploitation in its nature as both options employ the best agent as the reference. But the second option is more exploitative as the best agent becomes the moving agent. On the other hand, the third option is exploitation. Due to this analysis, it can be said that AIA tends to be an exploitative technique.

Overall, there is a trade-off between exploitation and exploration in developing a new technique. In general, a balance between both exploitation and exploration is needed. More exploitative techniques are better in handling unimodal problems while more exploring techniques are better in handling multimodal problems. But the real problem in the practical optimization problems is the constraint as shown in EED problem. In this problem, the superiority of AIA is narrow compared to GSO and HO whose performance is inferior in handling the theoretical problems. This result also becomes the reason why old techniques like genetic algorithms, simulated annealing, particle swarm optimization, and so on; are still widely used in many studies related to practical optimization problems.

Despite its supremacy, there is still limitation regarding this method and this work. There exist additional conventional applications, including the CEC series and four well-known engineering design challenges, which are frequently utilized to evaluate the effectiveness of emerging methodologies. However, evaluating a novel metaheuristic across all standard cases within a single paper is unfeasible. There are also a lot of practical optimization problems that can be used as additional use cases, whether these problems are still in the power system, such as ELD problem or OPF problem, or outside the power system, such as production scheduling, vehicle routing problem, and so on. Moreover, there are also a lot of other searching methods that are impossible to accommodate in a single metaheuristic.

6. CONCLUSIONS

This study introduces the adaptive iteration algorithm (AIA), a novel metaheuristic approach that operates without relying on metaphors. This presentation comprises the concept, formalization, assessment, and discussion. According to the evaluation outcomes, AIA has demonstrated its effectiveness as a robust method, capable of identifying near-optimal solutions to various problems. Furthermore, it outperforms its competitors in both 23 standard benchmark functions and the EED problem in Indonesia. AIA outperforms HO, COA, GSO, LOA, and OOA by effectively managing 23, 21, 21, 18, and 17 functions out of a total of 23, respectively. Additionally, AIA achieves the optimal performance across all four scenarios in the ELD problem.

This study can be expanded in the future by modifying this proposed AIA or employing AIA to solve more practical optimization problems. The modification of AIA can be done by combining AIA with other techniques, implementing other adaptive methods, or changing the random method. At the same time, various optimization challenges continue to arise in power systems as well as in other domains like manufacturing and operations research.

ACKNOWLEDGMENT

This work was funded by Telkom University, Indonesia.

REFERENCES

[1] Zein, H., Raharjo, J., Mardiyanto, I.R. (2022). A method for completing economic load dispatch using the technique of narrowing down area. IEEE Access, 10: 30822-30831. https://doi.org/10.1109/ACCESS.2022.3158928

[2] Puspitasari, K.M.D., Raharjo, J., Sastrosubroto, A.S., Rahmat, B. (2022). Generator scheduling optimization involving emission to determine emission reduction costs. International Journal of Engineering, 35(8): 1468-1478. https://doi.org/10.5829/IJE.2022.35.08B.02

[3] Gbadega, P.A., Sun, Y. (2024). Synergistic integration of renewable energy and HVDC technology for enhanced multi-objective economic emission dispatch using the salp swarm algorithm. In International Conference on Neural Computing for Advanced Applications, pp. 232-249. https://doi.org/10.1007/978-981-97-7004-5_17

[4] Syama, S., Ramprabhakar, J., Anand, R., Guerrero, J.M. (2024). An integrated binary metaheuristic approach in dynamic unit commitment and economic emission dispatch for hybrid energy systems. Scientific Reports, 14(1): 23964. https://doi.org/10.1038/s41598-024-75743-0

[5] Zou, D., Ma, L., Li, C., Ouyang, H. (2024). Integration of the wind and solar power for the dynamic economic emission dispatch with the charging and discharging of plug-in electric vehicles. Engineering Applications of Artificial Intelligence, 138: 109293. https://doi.org/10.1016/j.engappai.2024.109293

[6] Khalil, M.I.K., Rahman, I.U., Zakarya, M., Zia, A., et al. (2024). A multi-objective optimisation approach with improved pareto-optimal solutions to enhance economic and environmental dispatch in power systems. Scientific Reports, 14(1): 13418. https://doi.org/10.1038/s41598-024-62904-4

[7] Braik, M., Hammouri, A., Atwan, J., Al-Betar, M.A., Awadallah, M.A. (2022). White shark optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems. Knowledge-Based Systems, 243: 108457. https://doi.org/10.1016/j.knosys.2022.108457

[8] Al-Betar, M.A., Awadallah, M.A., Braik, M.S., Makhadmeh, S., Doush, I.A. (2024). Elk herd optimizer: A novel nature-inspired metaheuristic algorithm. Artificial Intelligence Review, 57(3): 48. https://doi.org/10.1007/s10462-023-10680-4

[9] Trojovská, E., Dehghani, M., Trojovský, P. (2022). Zebra optimization algorithm: A new bio-inspired optimization algorithm for solving optimization algorithm. IEEE Access, 10: 49445-49473. https://doi.org/10.1109/ACCESS.2022.3172789

[10] Dehghani, M., Bektemyssova, G., Montazeri, Z., Shaikemelev, G., Malik, O.P., Dhiman, G. (2023). Lyrebird optimization algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems. Biomimetics, 8(6): 507. https://doi.org/10.3390/biomimetics8060507

[11] Kusuma, P.D., Hasibuan, F.C. (2023). Swarm magnetic optimizer: A new optimizer that adopts magnetic behaviour. International Journal of Intelligent Engineering & Systems, 16(4): 264-275. https://doi.org/10.22266/ijies2023.0831.22

[12] Dehghani, M., Trojovský, P. (2023). Osprey optimization algorithm: A new bio-inspired metaheuristic algorithm for solving engineering optimization problems. Frontiers in Mechanical Engineering, 8: 1126450. https://doi.org/10.3389/fmech.2022.1126450

[13] Jia, H., Rao, H., Wen, C., Mirjalili, S. (2023). Crayfish optimization algorithm. Artificial Intelligence Review, 56(Suppl 2): 1919-1979. https://doi.org/10.1007/s10462-023-10567-4

[14] Ezugwu, A.E., Agushaka, J.O., Abualigah, L., Mirjalili, S., Gandomi, A.H. (2022). Prairie dog optimization algorithm. Neural Computing and Applications, 34(22): 20017-20065. https://doi.org/10.1007/s00521-022-

07530-9

[15] Ragunathan, R., Ramadoss, B. (2024). Golden jackal optimization for economic load dispatch problems with complex constraints. Bulletin of Electrical Engineering and Informatics, 13(2): 781-793. https://doi.org/10.11591/eei.v13i2.6572

[16] Hasanah, R.N., Robandi, I., Guntur, H.L., Lystianingrum, V., Djalal, M.R., Prakasa, M.A., Himawari, W. (2024). A novel horse herd optimization algorithm for optimal economic dispatch in Sulbagsel electricity system. International Journal of Intelligent Engineering & Systems, 17(6): 1059-1069. https://doi.org/10.22266/ijies2024.1231.78

[17] Połap, D., Woźniak, M. (2021). Red fox optimization algorithm. Expert Systems with Applications, 166: 114107. https://doi.org/10.1016/j.eswa.2020.114107

[18] Trojovska, E., Dehghani, M., Trojovský, P. (2022). Fennec fox optimization: A new nature-inspired optimization algorithm. IEEE Access, 10: 84417-84443. https://doi.org/10.1109/ACCESS.2022.3197745

[19] Hamadneh, T., Kaabneh, K., Alssayed, O., Eguchi, K., Gochhait, S., Leonova, I., Dehghani, M. (2024). Addax optimization algorithm: A novel nature-inspired optimizer for solving engineering applications. International Journal of Intelligent Engineering & Systems, 17(3): 732-743. https://doi.org/10.22266/ijies2024.0630.57

[20] Akbari, M.A., Zare, M., Azizipanah-Abarghooee, R., Mirjalili, S., Deriche, M. (2022). The cheetah optimizer: A nature-inspired metaheuristic algorithm for large-scale optimization problems. Scientific Reports, 12(1): 10953. https://doi.org/10.1038/s41598-022-14338-z

[21] Oladejo, S.O., Ekwe, S.O., Mirjalili, S. (2024). The hiking optimization algorithm: A novel human-based metaheuristic approach. Knowledge-Based Systems, 296: 111880. https://doi.org/10.1016/j.knosys.2024.111880

[22] Suyanto, S., Ariyanto, A.A., Ariyanto, A.F. (2022). Komodo mlipir algorithm. Applied Soft Computing, 114: 108043. https://doi.org/10.1016/j.asoc.2021.108043

[23] Noroozi, M., Mohammadi, H., Efatinasab, E., Lashgari, A., Eslami, M., Khan, B. (2022). Golden search optimization algorithm. IEEE Access, 10: 37515-37532. https://doi.org/10.1109/ACCESS.2022.3162853

[24] Ghasemi, M., Rahimnejad, A., Akbari, E., Rao, R.V., Trojovský, P., Trojovská, E., Gadsden, S.A. (2023). A new metaphor-less simple algorithm based on Rao algorithms: A Fully Informed Search Algorithm (FISA). PeerJ Computer Science, 9: e1431. https://doi.org/10.7717/peerj-cs.1431

[25] Dehghani, M., Hubálovský, Š., Trojovský, P. (2022). 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, 8: e910. https://doi.org/10.7717/peerj-cs.910

[26] Trojovský, P., Dehghani, M. (2023). Subtraction-average-based optimizer: A new swarm-inspired metaheuristic algorithm for solving optimization problems. Biomimetics, 8(2): 149. https://doi.org/10.3390/biomimetics8020149

[27] Kusuma, P.D., Kallista, M. (2023). Quad tournament optimizer: A novel metaheuristic based on tournament among four strategies. International Journal of Intelligent Engineering & Systems, 16(2): 268-278. https://doi.org/10.22266/ijies2023.0430.22

[28] Kusuma, P.D., Novianty, A. (2023). Multiple interaction optimizer: A novel metaheuristic and its application to solve order allocation problem. International Journal of Intelligent Engineering & Systems, 16(2): 440-453. https://doi.org/10.22266/ijies2023.0430.35

[29] Kusuma, P.D., Adiputra, D. (2022). Modified social forces algorithm: From pedestrian dynamic to metaheuristic optimization. International Journal of Intelligent Engineering & Systems, 15(3): 294-303. https://doi.org/10.22266/ijies2022.0630.25

[30] Dehghani, M., Montazeri, Z., Trojovská, E., Trojovský, P. (2023). Coati optimization algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems. Knowledge-Based Systems, 259: 110011. https://doi.org/10.1016/j.knosys.2022.110011

[31] Faramarzi, A., Heidarinejad, M., Mirjalili, S., Gandomi, A.H. (2020). Marine predators algorithm: A nature-inspired metaheuristic. Expert Systems with Applications, 152: 113377. https://doi.org/10.1016/j.eswa.2020.113377

[32] Mirjalili, S., Mirjalili, S.M., Lewis, A. (2014). Grey wolf optimizer. Advances in Engineering Software, 69: 46-61. https://doi.org/10.1016/j.advengsoft.2013.12.007

NOMENCLATURE

s	agent
S	set of agents / swarm
s_{best}	the best agent
s_{bet}	better agent
S_{bet}	set of better agents
s_{sel}	selected agent
c_1, c_2	solution candidate
imp	improving status
r_1	floating point uniform random [0,1]
r_2	integer uniform random [1,2]
r_3	uniform random within population
t	iteration
t_m	maximum iteration
f	objective function