Bulletin of Electrical Engineering and Informatics
Vol. 15, No. 2, April 2026, pp. 1535~1547
ISSN: 2302-9285, DOI: 10.11591/eei.v15i2.10048

Conditional toggle algorithm: an adaptive metaheuristic and its implementation on handling engineering problems

Purba Daru Kusuma[1,2], Helmy Widyantara[2,3]
[1]Computer Engineering Study Program, Faculty of Electrical Engineering, Telkom University, Bandung, Indonesia
[2]Center of Excellence of Intelligent and Embedded System, Telkom University, Surabaya, Indonesia
[3]Information Technology Study Program, Directorate of Surabaya Campus, Telkom University, Surabaya, Indonesia

Article Info

Article history:

Received Feb 10, 2025
Revised Feb 4, 2026
Accepted Mar 10, 2026

Keywords:

Economic emission dispatch
Engineering design
Metaheuristic
Optimization
Power system

ABSTRACT

There have been numerous new metaheuristic algorithms in this decade. Unfortunately, the attention on taking stagnation is still less considered so that it is difficult to find new metaheuristic algorithms that are enriched with stagnation taking mechanism. This work introduces a new method called conditional toggle algorithm (CTA). CTA is designed to be adaptive on facing enhancement and stagnation during iteration as its novelty. When enhancement occurs, the exploitation-focused look is applied. Meanwhile, the exploration-focused look is applied when stagnation occurs. The efficacy of CTA is then measured by implementing to solve three cases: 23 functions, 4 engineering design problems, and economic emission dispatch (EED) problem in Java-Bali power system in Indonesia. CTA is compared with five new metaheuristic algorithms. The evidence provides that CTA is supreme in taking high dimension functions and competing in taking fixed dimension functions. CTA is also supreme in taking pressure vessel and speed reducer design problems and the EED problem. But its performance is average in taking welded beam and spring design problems. In the future, CTA can be modified with other metaheuristic algorithms to enhance its performance and challenged to take broader problems, especially in electrical engineering fields.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Purba Daru Kusuma
Computer Engineering Study Program, Faculty of Electrical Engineering, Telkom University
Bandung, Indonesia
Email: purbodaru@telkomuniversity.ac.id

1. INTRODUCTION

Optimization is a popular subject in engineering area. This subject can be found in the power system, manufacture, supply chain, and so on. In the power system, optimization is applied in economic dispatch (ED) problem whether it is economic load dispatch (ELD) [1] or EED [2] problem, power system stabilizer [3], unit commitment (UC) problem [4], optimal power flow (OPF) [5], and voltage controller [6]. In manufacturing system, optimization is needed in assembly planning [7], job-shop scheduling [8], and picker allocation [9]. In logistics and transportation system, optimization is needed such as in multi-echelon last mile delivery system [10], and meal delivery truck routing [11].

In latest years, there are numerous new metaheuristic algorithm. Among them, swarm intelligence (SI) becomes the most popular foundation. Many of them are metaphor-based metaheuristics where animal behavior becomes the most popular inspiration, such as: pufferfish optimization algorithm (POA) [12], lyrebird optimization algorithm (LOA) [13], osprey optimization algorithm (OOA) [14], marine predator algorithm (MPA) [15], sea lion optimization (SLnO) [16], white shark optimization (WSO) [17], Komodo mlipir algorithm (KMA) [18], crayfish optimization algorithm (COA) [19], golden jackal optimization (GJO) [20], horse herd optimization algorithm (HHOA) [21], red fox optimization (RFO) [22], elk herd optimization (EHO) [23], hippopotamus optimization algorithm (HO) [24], and so on. Some of them use individual, social, or professional activity for inspiration, such as hiking optimization (HO) [25], and dollmaker optimization algorithm (DOA) [26]. Some metaheuristics use natural or mechanical behavior. Fortunately, some of them do not use any metaphor, such as golden look optimization (GSO) [27], and fully informed look algorithm (FISA) [28].

This extensive construction of metaheuristic algorithms is highly connected to the no-free-lunch (NFL) theorem. This theory states that no perfect method exists to manage all problems in a superior manner [20]. Each method has its own disadvantage as it cannot cover all mechanism. This circumstance becomes the motives of many scholars to introduce new algorithms.

Unfortunately, metaheuristic algorithm that employs adaptive approach to taking both enhancement and stagnation is rare to find. Most methods employ the same look or strategy without considering the situation that it faces during the optimization process. For example, GSO will always perform directed look following the combination of the global and local most prominent particles based on sinusoid step size during the iteration [27].

Besides, recent studies that introduced new metaheuristics becomes thicker in the number of pages. Most of the portion of the article is spent on excessive evaluation rather than the presentation of the novel approach. Meanwhile, the use of these metaheuristic algorithms in the next studies to solve various practical optimization problems relied on the novel approach and not the evaluation. For example, the article that introduces HO contains 50 pages while approximately 36 pages are consumed for the evaluation [24].

This work aims to propose a new adaptive algorithm that can take both stagnation and enhancement called conditional toggle algorithm (CTA). In general, CTA employs exploitation-focused look when it faces enhancement to intensify the trajectory. On the other hand, CTA employs exploration-focused looks when it faces stagnation to diversify and finds alternative trajectory. Moreover, below are the novelties, innovations, and contributions of this research.

- It constructs a new metaheuristic that is independent from metaphor and adaptive on taking both enhancement and stagnation.
- It introduces a novel approach where exploitation-focused look is employed during enhancement and exploration-focused look is employed during stagnation.
- Its efficacy is assessed by employing it to solve the 23 functions, four engineering problems, and EED problem.
- The submitted metaheuristic is benchmarked with five new metaheuristic algorithms.

The arrangement of the following parts of this article is described below. Section 2 summarizes the recent studies on the construction of a new algorithm, including their algorithm and the existence of the adaptive algorithm. Section 3 delivers the details of the CTA including the idea and formalization. Section 4 gives the efficacy investigation of CTA including the evaluation scenario, result, comprehensive discussion regarding the result, findings, complexity, and limitations. Section 5 gives the conclusion and the suggestion for further studies.

2. LITERATURE REVIEW

Metaheuristic algorithms have been known as a popular and advantageous tool for taking optimization problems. Its utilization can be found in numerous optimization studies. Some methods are very popular, for example: grey wolf optimization (GWO), MPA, particle swarm optimization (PSO), and so on so that they can be found anywhere. GWO has been utilized to solve traveling salesman problem (TSP) [29], routing in wireless sensor network [30], optimization in solar multigeneration system [31], task scheduling in cloud computing [32], and so on. MPA has been used to solve supercapacitor model optimization [33], task scheduling in internet of things system [34], and structural damage detection [35]. Meanwhile, OOA has been utilized to optimize the short-term power load forecasting [36], and size of hybrid renewable energy system [37].

The well-known metaheuristic algorithm arrives from several reasons. First, metaheuristic algorithms have advantages as flexible and simple tools so that they can be applied to broad problems with minimum customization. This advantage arrives from the nature of metaheuristics as a stochastic and iterative system, and moreover, it abstracts the problem by concentrating on the objective function and the constraints while the optimization is performed based on the trial-and-error approach [38] with certain rule. Second, there are numerous methods that can be chosen, modified, or hybridized to solve the optimization problems. In the end, this popularity motivates researchers to develop new metaheuristic algorithm. For example, GWO utilizes three best agents called wolves for the target of the motion of all wolves [39]. In GJO, the swarm moves toward two best jackals [20]. In COA, the blend of global best and local best crayfish becomes the target [19]. On the other hand, in MPA, the agents utilize the local best solution but with various strategies during the iteration [15].

One issue in the development of metaheuristic algorithms is stagnation. Stagnation can be defined as the failure of the process to find an enhanced solution than the recent solution. This situation is common due to the essence of metaheuristic as a stochastic solution. Despite numerous new methods that have been introduced in recent years, most of them do not care about stagnation. In swarm-based metaheuristics, most of them still perform same look whether the process faces enhancement or stagnation.

The second issue is that many recent articles that introduced a new metaheuristic becomes thicker in the number of pages. Many of these articles have more than 30 pages. This situation occurs because of the excessive evaluation that is carried out to measure the efficacy of the submitted metaheuristic algorithms. On the other hand, the portion for the explanation of the submitted method, including the idea, modeling, and formalization is too little. Meanwhile, the potential use of an algorithm depends on the novel approach of these metaheuristics compared to the previous or other ones while the evaluation is needed to support the submitted algorithm. Table 1 gives the existence of the adaptive approach and the thickness of the recent articles introducing new metaheuristics.

Table 1. Recent studies that introduce a new metaheuristic algorithm

No.	Algorithm	Adaptive mechanism	Number of pages	
			Evaluation	Algorithm explanation
1	POA [12]	no	42	5
2	HO [24]	no	37	6
3	HO [25]	no	18	2
4	LOA [13]	no	48	5
5	OOA [14]	no	31	4
6	DOA [26]	no	5	2
7	EHO [23]	no	41	7
8	COA [19]	no	41	9

Table 1 provides the lack of adaptive mechanism in recent metaheuristics and the excessive portion of evaluation in articles that introduce the metaheuristic. This fact gives a chance to develop a new metaheuristic that is independent from any metaphor and has ability for adaptation in facing both stagnation and enhancement rather than improving the existing ones, for example: adaptive multi-objective genetic algorithm based on greedy initialization strategy (AMoGA-GIS) [40], hybrid adaptive genetic algorithm (HAGA) [41], and chaotic elite adaptive genetic algorithm (CEAGA) [42]. Moreover, there is a chance to present this submitted method in a concise manner, but the evaluation is still appropriate and representative.

3. METHOD

The CTA is developed as swarm intelligence. It contains a collection of autonomous particles. Each particle performs looking individually without any mandatory direction from any entity. But the particle collaborates with other particles to improve the looking process.

The core concept of the CTA comes from the adaptability of metaheuristic algorithms during two situations: enhancement and stagnation. The enhancement means the solution candidate is superior to the particle's recent solution. Conversely, stagnation means that the solution candidate is not superior to the particle's recent solution. If the particle achieves enhancement in the previous look, then the particle performs the exploitation-focused look in the recent look. But if the particle faces stagnation in the previous look, then the particle performs an exploration-focused look in the recent look. In CTA, the stringent acceptance method is employed.

CTA is designed as a multiple-look algorithm. There are four looks in CTA. The 1st look is the migration following the most prominent particle. The 2nd look is the migration following the middle among the more prominent particles plus the most prominent particle. The 3rd look is the migration following a randomly taken more prominent particle plus the most prominent particle. The 4th look is the migration relative to a randomly taken particle. The visualization of these four looks is given in Figure 1. In Figure 1, the red hexagon represents the most prominent particle, the green hexagons represent the more prominent particle, the blue hexagons represent the random particle, and the grey hexagons represent the general particles.

Each particle performs two consecutive stages. In the first stage, the particle may perform the first or second look. Meanwhile, in the second stage, the particle may perform the third or fourth look. In the first stage, the particle may select the first or second look. Meanwhile, in the second stage, the particle may select between the third look or fourth look. In the first stage, the first look represents the exploitation-focused look while the second look represents the exploration-focused look. In the second stage, the third look represents the exploitation-focused look while the fourth look represents the exploration-focused look. Figure 1(a) visualizes the migration following the most prominent particle. Figure 1(b) visualizes migration following the middle among the more prominent particles plus the most prominent particle. Figure 1(c) visualizes migration following a randomly taken more prominent particle plus the most prominent particle. Figure 1(d) visualizes the migration relative to a randomly taken particle.

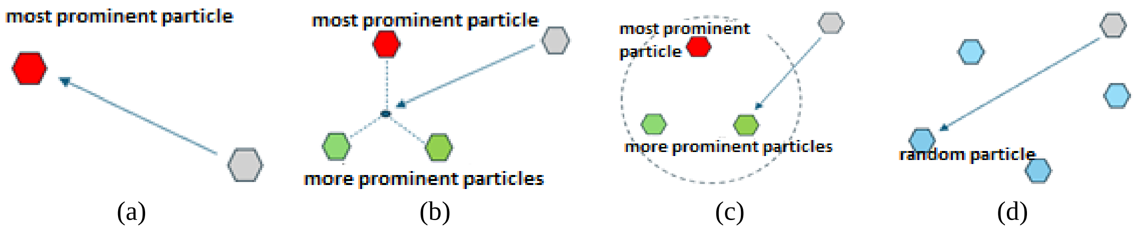

Figure 1. Visualization of four looks; (a) first look, (b) second look, (c) third look, and (d) fourth look

This adaptive mechanism is different from the existing adaptive algorithms. In AMoGA-GIS, the adaptive mechanism is performed through a non-uniform mutation during the iteration as selection strategy is recorded in every iteration [40]. In HAGA, the probability for mutation and crossover is determined based on local and global point of view, rather than the static mechanism as in the standard form [41]. In CEAGA, the adaptive mechanism is taken by determining the probability of mutation and crossover dynamically within range using the average fitness value and highest fitness value as controllers [42].

The formalization of CTA is as follows. The overall procedure is given in Algorithm 1 while the procedure in a stage is given in Algorithm 2. Mathematical formulation is given from (1) to (14). The list of variables is given in Table 2. All random numbers follow uniform distribution.

Table 2. List of variables

Notations	Description
a	Particle
A	Set of particles
a_{hst}	The most prominent particle
a_{sel1}	A randomly taken particle from a pool of more prominent particles plus the most prominent particle
a_{sel2}	A randomly taken particle among swarm
A_{pool}	A pool that contains more prominent particles plus the most prominent particle
b_{low}	Lower boundary
b_{high}	Higher boundary
c	Solution candidate
f	Objective function
p	Previous value of particle's solution
t	Iteration
t_{max}	Maximum iteration
U_1	Random [0,1]
U_2	Random {1, 2}
U_3	Random among a population

As a metaheuristic-optimization, the procedure is split into two phases. The first phase is the initialization which lies from lines 2 to 6. Meanwhile, the second phase is the iteration which lies from lines 7 to 12. As shown in (1) provides that CTA is a swarm that contains certain particles.

The initialization phase contains three procedures. The first procedure is defining the initial particle randomly as in (2). The second procedure is setting the initial status of the particle as presented in (3). The third procedure is updating the most prominent particle as presented in (4).

$$A = \{a_1, a_2, a_3, \dots, a_n\} \tag{1}$$

$$a_{i,j} = b_{low,j} + U_1(b_{high,j} - b_{low,j}) \tag{2}$$

$$s_i = 1 \tag{3}$$

$$a'_{hst} = \begin{cases} a_i, f(a_i) < f(a_{hst}) \\ a_{hst}, else \end{cases} \tag{4}$$

Algorithm 1. Conditional toggle algorithm
```
begin
  for i=1 : n
    setup a_i
    set s_i=1
    update a_hst
  end
  for t=1: t_max
    for i=1: n
      perform stage 1
      perform stage 2
    end
  end
  return a_hst
end
```

Algorithm 2. Stage procedure
```
begin
  save previous score
  if s_i=1 then
    exploitation-focused look
  else
    exploration-focused look
  stringent acceptance
  update a_hst
  update s_i
end
```

The iteration is the most complicated phase in CTA. This phase is started by storing the quality of recent particles' solution into a certain buffer variable as provided in (5). As shown in (6) is the first look which is the migration following the most prominent particle. As shown in (7) and (8) represent the second look where (7) is used as the construction of the pool of more prominent particles plus the most prominent particle while (8) provides the migration. As shown in (9) and (10) represent the third look where (9) provides the randomly chosen particle from the more prominent pool and (10) provides the migration. As shown in (11) and (12) represent the fourth look where (11) represents the random picking and (12) provides the migration relative to the target.

$$p = f(a_i) \tag{5}$$

$$c_{i,j} = a_{i,j} + U_1(a_{hst,j} - U_2 a_{i,j}) \tag{6}$$

$$A_{pool,i} = \{\forall a_k \in A | f(a_k) < f(a_i)\} \cup a_{hst} \tag{7}$$

$$c_{i,j} = a_{i,j} + U_1\left(\frac{\sum a_{pool,i,k,j}}{n(A_{pool,i})} - U_2 a_{i,j}\right) \tag{8}$$

$$a_{sel1,i} = U_3(A_{pool,i}) \tag{9}$$

$$c_{i,j} = a_{i,j} + U_1(a_{sel1,i,j} - U_2 a_{i,j}) \tag{10}$$

$$a_{sel2,i} = U_3(A) \tag{11}$$

$$c_{i,j} = \begin{cases} a_{i,j} + U_1\left(a_{sel2,i,j} - U_2 a_{i,j}\right), f\left(a_{sel2,i}\right) < f(a_i) \\ a_{i,j} + U_1\left(a_{i,j} - a_{sel2,i,j}\right), else \end{cases} \tag{12}$$

The next two procedures are employed to complete the phase. These procedures are acceptance and updating the status. As shown in (13) provides the stringent acceptance method while (14) formalizes the updating of the status of the particle. Then, (14) is used to update the status of the particle. If the enhancement occurs, then the status is 1. It means, the next look will be the exploitation-focused look. Contrary, if stagnation occurs, then the status is 0. It means that the next look will be the exploration-focused look.

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

$$s_i = \begin{cases} 1, f(a_i) < f(p) \\ 0, else \end{cases} \quad (14)$$

4. RESULTS AND DISCUSSION

This part delivers the efficacy investigation of CTA and the discussion regarding its result. This section is divided into two sub sections. The first sub section gives the investigation scenario and its result. Meanwhile, the second sub section gives rigorous conversation regarding the result, findings, and limitations.

4.1. Experiment and result

The investigation is carried out by implementing CTA to solve three use cases. The first use case is 23 functions which represent the unconstrained problems. The second case is four engineering problems including: pressure vessel (PV), speed reducer (SR), welded beam (WB), and spring design (SD) problems. The third case is the EED problem. In this work, CTA is compared with five new metaheuristics including: POA [12], OOA [14], LOA [13], HO [25], and DOA [26]. In this investigation, the swarm size is 10 while the maximum iteration is 20.

The computational environment to conduct this experiment is as follows. The program runs on Intel Core i5 PC with 8 GB RAM. The operating system is Windows 10 Pro. The program is developed using PHP language, Apache web server, and MySQL database to store the result.

The first case is the 23 functions. It is chosen due to its coverage. It contains seven high dimension unimodal functions (HDUF), six high dimension multimodal functions (HDMF), and ten fixed dimension multimodal functions (FDMF). HDUFs are commonly used to measure exploitation capability. HDMFs are commonly used to measure exploration capability. Meanwhile, the FDMFs are used to measure the balancing capability between exploitation and exploration. This case is commonly used in various studies that submitted new algorithms, such as KMA [18], GSO [27], and so on. Detailed description of these functions including the mathematical formulation, dimension, and the optimal solution can be found in [27]. The dimension for the high dimension functions is 30. The result is presented in Tables 3 to 8. The decimal point smaller than 10^{-4} is rounded to 0.

Table 3 exhibits the result on taking HDUFs. The result provides the dominance of CTA as it becomes the best of all seven functions and the sole best of six functions. POA, OOA, LOA, and DOA also achieved the best result in taking the second function. Moreover, CTA achieves the global optimal in three functions. In this group, HO becomes the poorest algorithm in five functions. Meanwhile LOA becomes the poorest algorithm in two functions. The result also provides that the difference between the best and poorest algorithms is wide in all seven functions.

Table 3. Result on taking 7 HUDFs

Function	Variable	POA [12]	OOA [14]	LOA [13]	HO [25]	DOA [26]	CTA
1	Average value	0.0924	0.0868	3.9196×10^{1}	3.9459×10^{1}	0.1852	0.0000
	Standard-dev	0.0354	0.0641	2.4578×10^{1}	2.1393×10^{1}	0.1469	0.0000
	Position	3	2	5	6	4	1
2	Average value	0.0000	0.0000	0.0000	0.0008	0.0000	0.0000
	Standard-dev	0.0000	0.0000	0.0000	0.0035	0.0000	0.0000
	Position	1	1	1	6	1	1
3	Average value	1.5170×10^{3}	4.8159×10^{2}	7.4885×10^{3}	6.3366×10^{2}	3.1350×10^{2}	0.9694
	Standard-dev	2.5554×10^{3}	5.1360×10^{2}	4.0197×10^{3}	5.4540×10^{2}	6.4690×10^{2}	2.5957
	Position	5	3	6	4	2	1
4	Average value	0.3161	0.3744	6.2914	2.5615	0.5351	0.0000
	Standard-dev	0.1151	0.1204	2.8657	0.7172	0.1968	0.0000
	Position	2	3	6	5	4	1
5	Average value	3.0448×10^{1}	3.0075×10^{1}	1.5359×10^{3}	2.8717×10^{4}	3.2409×10^{1}	2.8889×10^{1}
	Standard-dev	0.7736	0.8213	3.7238×10^{3}	2.6245×10^{4}	1.7778	0.0458
	Position	4	3	2	6	5	1
6	Average value	5.7361	5.5239	4.9501×10^{1}	5.3525×10^{1}	5.8030	4.9490
	Standard-dev	0.5087	0.5076	4.4510×10^{1}	2.2400×10^{1}	0.5453	0.3845
	Position	3	2	5	6	4	1
7	Average value	0.0267	0.0236	0.0617	1.8843×10^{2}	0.0281	0.0076
	Standard-dev	0.0157	0.0125	0.0455	9.3113×10^{1}	0.0164	0.0049
	Position	3	2	5	6	4	1

Table 4. Result on taking 6 HMFSs

Function	Variable	POA [12]	OOA [14]	LOA [13]	HO [25]	DOA [26]	CTA
8	Average value	-3.2893×10^3	-3.6152×10^3	-3.0614×10^3	-1.1293×10^2	-3.3590×10^3	-2.9727×10^3
	Standard-dev	6.0987×10^2	5.2793×10^2	5.1451×10^2	4.0907×10^1	3.5554×10^2	3.6972×10^2
	Position	3	1	4	6	2	5
9	Average value	1.4177×10^1	4.6452	1.5208×10^2	2.9480×10^2	9.4052	0.0000
	Standard-dev	1.9267×10^1	9.1606	5.1326×10^1	4.0145×10^1	2.2517×10^1	0.0000
	Position	4	2	5	6	3	1
10	Average value	0.1065	0.0748	3.0226	5.7617	0.1451	0.0000
	Standard-dev	0.0456	0.0270	0.8625	1.1121	0.0772	0.0000
	Position	3	2	5	6	4	1
11	Average value	0.0497	0.0727	1.2913	0.4583	0.2046	0.0034
	Standard-dev	0.0341	0.1429	0.2911	0.2562	0.1847	0.0107
	Position	2	3	6	5	4	1
12	Average value	0.8207	0.8561	2.5268	4.4944	0.8499	0.6501
	Standard-dev	0.1666	0.1735	1.5222	1.0396	0.1917	0.1410
	Position	2	4	5	6	3	1
13	Average value	3.3275	3.1295	7.9694	4.1274	3.1791	2.8596
	Standard-dev	0.1824	0.1837	3.2314	2.2174	0.1664	0.1755
	Position	4	2	6	5	3	1

Table 5. Result on taking 7 FDMFS

Function	Variable	POA [12]	OOA [14]	LOA [13]	HO [25]	DOA [26]	CTA
14	Average value	5.9225	6.2989	7.9789	1.2968×10^1	5.9460	6.2521
	Standard-dev	3.3294	3.4700	3.6745	0.7546	3.3828	3.8333
	Position	1	4	5	6	2	3
15	Average value	0.0045	0.0021	0.0049	0.0541	0.0026	0.0029
	Standard-dev	0.0098	0.0045	0.0039	0.0403	0.0048	0.0058
	Position	4	1	5	6	2	3
16	Average value	-1.0298	-1.0303	-1.0267	-0.1473	-1.0288	-1.0314
	Standard-dev	0.0020	0.0021	0.0064	0.7976	0.0034	0.0004
	Position	3	2	5	6	4	1
17	Average value	0.3992	0.3985	0.4111	1.2912	0.3989	0.4002
	Standard-dev	0.0014	0.0007	0.0160	1.1986	0.0014	0.0078
	Position	3	1	5	6	2	4
18	Average value	3.0317	3.0041	3.3469	3.2800×10^2	3.0536	6.8894
	Standard-dev	0.0639	0.0064	0.5209	5.3055×10^2	0.1586	9.4807
	Position	2	1	4	6	3	5
19	Average value	-0.0495	-0.0495	-0.0495	-0.0409	-0.0495	-0.0495
	Standard-dev	0.0000	0.0000	0.0000	0.0143	0.0000	0.0000
	Position	1	1	1	6	1	1
20	Average value	-3.1647	-3.1073	-2.9827	-0.8358	-3.2133	-3.0139
	Standard-dev	0.0823	0.1245	0.1572	0.5889	0.0613	0.1807
	Position	2	3	5	6	1	4
21	Average value	-5.2339	-3.0613	-3.1437	-1.3019	-5.9020	-4.3698
	Standard-dev	1.9610	1.3017	1.5709	0.9729	1.6445	1.4560
	Position	2	5	4	6	1	3
22	Average value	-5.5829	-3.4821	-3.6209	-1.1101	-6.0680	-4.0091
	Standard-dev	1.5850	1.1209	1.4544	0.5849	2.1220	1.3246
	Position	2	5	4	6	1	3
23	Average value	-5.2865	-3.4300	-3.7699	-1.2701	-6.0761	-4.4164
	Standard-dev	1.6960	1.2473	1.2632	0.4378	2.1378	2.2012
	Position	2	5	4	6	1	3

Table 4 exhibits the result of taking six HDMFs. CTA is still proven to be dominant as it achieves the best result of five functions. Moreover, CTA achieves the global optimal in two functions. Meanwhile, CTA becomes the fifth best in one function. On the other hand, HO becomes the poorest algorithm in four functions and the second poorest algorithm in two functions. LOA becomes the poorest algorithm of the two functions. The result also provides the narrow difference among the metaheuristics in taking F8 while this difference is moderate in F13. Otherwise, the difference is wide.

Table 5 exhibits the result in taking FDMFs. The result proves that CTA is competing in these functions although it is not dominant. CTA becomes the sole best algorithm only on F16. Meanwhile, it becomes the third best in 5 functions, fourth best in two functions, and fifth best in F18. On the other hand, HO becomes absolute poorest algorithm as it achieves the poorest result in all ten functions. Overall, narrow efficacy differences among metaheuristics occur in all ten functions except in F18.

Table 6 provides the dominance of CTA in taking the high dimension functions. It happens in both unimodal functions and multimodal functions. Its dominance occurs compared to all five competitors. Meanwhile, the dominance of CTA in the FDMFs occurs only when CTA is compared with LOA and HO.

Table 6. Group based supremacy on taking 23 functions

Cluster	POA [12]	OOA [14]	LOA [13]	HO [25]	DOA [26]
1	6	6	6	7	6
2	5	5	5	6	5
3	2	5	8	10	1
Total	13	16	19	23	12

Table 7. Significance result using t-test based p-value on taking 23 functions

F	POA [12]	OOA [14]	LOA [13]	HO [25]	DOA [26]
1	0.000	0.000	0.000	0.000	0.000
2	1.000	1.000	1.000	0.309	1.000
3	0.010	0.000	0.000	0.000	0.029
4	0.000	0.000	0.000	0.000	0.000
5	0.000	0.000	0.065	0.000	0.000
6	0.000	0.000	0.000	0.000	0.000
7	0.000	0.000	0.000	0.000	0.000
8	0.045	0.000	0.519	0.000	0.001
9	0.001	0.017	0.000	0.000	0.058
10	0.000	0.000	0.000	0.000	0.000
11	0.000	0.011	0.000	0.000	0.000
12	0.000	0.000	0.000	0.000	0.000
13	0.000	0.000	0.000	0.012	0.000
14	0.765	0.968	0.136	0.000	0.780
15	0.519	0.619	0.190	0.000	0.897
16	0.001	0.032	0.001	0.000	0.001
17	0.551	0.309	0.006	0.001	0.445
18	0.070	0.068	0.096	0.008	0.071
19	1.000	1.000	1.000	0.007	1.000
20	0.000	0.062	0.545	0.000	0.000
21	0.108	0.003	0.011	0.000	0.002
22	0.000	0.166	0.368	0.000	0.000
23	0.143	0.077	0.240	0.000	0.015

Table 8. Sensitivity and convergence analysis

F	Increase of swarm size		Increase of maximum iteration	
	$n(A)$=20	$n(A)$=40	t_{max}=30	t_{max}=60
1	0.000	0.000	0.000	0.000
2	0.000	0.000	0.000	0.000
3	0.103	0.014	0.003	0.000
4	0.000	0.000	0.000	0.000
5	2.883×10^{1}	2.877×10^{1}	2.883×10^{1}	2.877×10^{1}
6	4.297	3.866	4.728	4.241
7	0.005	0.002	0.005	0.002
8	-3.090×10^{3}	-3.565×10^{3}	-3.155×10^{3}	-3.532×10^{3}
9	0.000	0.000	0.000	0.000
10	0.000	0.000	0.000	0.000
11	0.001	0.000	0.000	0.000
12	0.477	0.383	0.670	0.458
13	2.705	2.562	2.918	2.761
14	3.799	2.793	4.380	3.375
15	0.000	0.000	0.001	0.001
16	-1.031	-1.031	-1.031	-1.031
17	0.398	0.398	0.398	0.398
18	3.000	3.000	4.424	5.700
19	-0.049	-0.049	-0.049	-0.049
20	-3.115	-3.204	-3.166	-3.213
21	-5.261	-5.831	-5.212	-6.071
22	-5.547	-6.134	-4.690	-7.424
23	-4.001	-6.001	-4.822	-5.860

Table 7 provides that there is significant difference between CTA and other algorithms in solving 23 functions. This result is based on the p-value that is obtained by the two-tailed t-test that compares CTA and its benchmarks. Table 7 exhibits these p-values. In this context, the p-value which is less than 0.05 indicates that the difference between two results is significant. Based on this reasoning, the number of functions where CTA is significantly different compared to POA, OOA, LOA, HO, and DOA are 15, 14, 13, 22, and 16 functions consecutively. Based on this result, the difference between CTA and DO is the most significant compared to other algorithms.

Table 8 provides the convergence analysis and the sensitivity analysis of CTA. It is carried out by increasing the swarm size and the maximum iteration. There are two values of swarm size, which are 20 and 40, while there are two values of maximum iteration which are 30 and 60. The swarm size is still 20 for both values of swarm size. On the other hand, the swarm size is still 20 for both values of maximum iteration. Based on Table 8, it is shown that the convergence has been reached in the early iteration and low maximum iteration. This argument is constructed as there is not significant improvement when the swarm size increases from 20 to 40 or the maximum iteration increases from 30 to 60.

The second case is the engineering problems. There are four design problems All these problems have multiple constraints that make these problems harder to solve. These constraints make the solution cannot be put in the solution space easily in every dimension as there is complex interdependence among the dimensions. The penalty is used consequently for constraint violation. It means that a certain penalty is introduced every time the solution violates the constraints. The quadratic penalty function is chosen so that the penalty increases when the constraint violation goes wider. The detail description of these four problems can be found in various studies introducing new metaheuristics, such as in the first introduction of POA [12] and LOA [13]. The result is given in Tables 9 to 12.

Table 9. Result on taking PV problem

Algorithm	Average score	Standard deviation	Rank
POA [12]	8.5555	1.8254	3
OOA [14]	1.2155×10^{1}	4.3207	4
LOA [13]	1.2899×10^{1}	5.0713	5
HO [25]	8.2459×10^{12}	4.2202×10^{10}	6
DOA [26]	8.1738	3.1265	2
CTA	6.7021	3.6063	1

Table 10. Results on taking SR problem

Algorithm	Average score	Standard deviation	Rank
POA [12]	3.5350×10^{3}	1.0766×10^{1}	2
OOA [14]	3.5670×10^{3}	3.0100×10^{1}	5
LOA [13]	3.5453×10^{3}	1.2235×10^{1}	4
HO [25]	3.5730×10^{3}	2.4853×10^{1}	6
DOA [26]	3.5365×10^{3}	1.0342×10^{1}	3
CTA	3.5280×10^{3}	1.2856×10^{1}	1

Table 9 provides that CTA becomes the best algorithm in taking the PV problem. On the other hand, HO becomes the poorest algorithm. The difference between CTA and HO is very wide. Meanwhile, the difference between CTA and other competitors is narrow. The difference between DOA and POA is narrow. Meanwhile, the difference between OOA and LOA is also narrow.

Table 10 provides that CTA is still the best algorithm in taking the SR problem. On the other hand, HO becomes the poorest algorithm. But the result provides that the difference between CTA and HO is very narrow. This situation indicates very tight competition among the algorithms. Moreover, the result tends to be stable as the deviation of every algorithm is also small.

Table 11 provides that CTA is competing in taking on the WB problem. It becomes the third best algorithm after DOA in the first place and POA in the second place. On the other hand, HO becomes the poorest algorithm. The difference between DOA as the best algorithm and HO as the poorest algorithm is high. The result can be grouped into three clusters. The first cluster contains DOA, POA, and CTA. The second cluster contains OOA and LOA. Finally, the third cluster contains HO. It means that the difference between DOA as the best algorithm and CTA as the third best algorithm is narrow.

Table 12 provides that CTA is competing in dealing with the SD problem. CTA is the third best after OOA as the best algorithm and POA as the second-best algorithm. HO becomes the poorest algorithm. The difference between OOA as the best algorithm and HO as the poorest algorithm is large. Meanwhile, the difference between POA, OOA, LOA, DOA, and CTA are narrow.

Table 11. Result on taking WB problem

Algorithm	Average score	Standard deviation	Rank
POA [12]	2.3111×10^{7}	1.8442×10^{7}	2
OOA [14]	1.8347×10^{8}	1.9973×10^{8}	4
LOA [13]	2.1387×10^{8}	2.4035×10^{8}	5
HO [25]	2.8781×10^{11}	4.2869×10^{11}	6
DOA [26]	2.2903×10^{7}	3.7302×10^{7}	1
CTA	9.2483×10^{7}	2.8074×10^{8}	3

Table 12. Result on taking SD problem

Algorithm	Average score	Standard deviation	Rank
POA [12]	3.7681	0.3732	2
OOA [14]	3.6881	0.2418	1
LOA [13]	4.2401	1.2797	4
HO [25]	3.5348×10^{2}	4.2405×10^{2}	6
DOA [26]	5.5644	4.7694	5
CTA	3.7782	0.4667	3

The third use case is the EED problem in Java-Bali grid system in Indonesia. This system is chosen to differentiate this study with many other studies in economic dispatch problems whose use cases are IEEE standard systems. This Java-Bali grid system is the heaviest grid system in Indonesia mostly because Java Island is the most populous and industrialized island in Indonesia. The power demand in this island is the highest compared to other regions or island in Indonesia. This system contains eight generating units where six of them are thermal power plants and two others are hydroelectric power plants [43]. This problem is a multi-objective problem where the objective is minimizing both operating cost and emission cost. This objective is presented in aggregate of weighted quadratic function. This problem contains both equality and inequality constraints. The equality constraint states that the total output power of the system should be equal to the demand. Meanwhile, the inequality constraint states that the output power of each generating unit should be within its power range. The formalization of the objective and the constraints, plus the constants for the cost function can be found in [43]. In this article, the demand is set to 12,228 MW while the weight of the function is 0.5. This situation provides the balance between the operating cost and the emission cost. The result can be found in Table 13.

The result highlights the superiority of CTA in the EED problem. CTA becomes the best algorithm while HO becomes the poorest algorithm. The result also reveals very fierce competition among the algorithms in dealing with this problem. The indication is the very low difference between the best algorithm and the poorest algorithm.

Table 13. Result on taking EED problem in Java-Bali grid system

Algorithm	Average score	Standard deviation	Rank
POA [12]	2.0908×10^{10}	8.4963×10^{7}	3
OOA [14]	2.1106×10^{10}	1.0052×10^{8}	5
LOA [13]	2.1099×10^{10}	2.0087×10^{8}	4
HO [25]	2.2562×10^{10}	6.7789×10^{8}	6
DOA [26]	2.0818×10^{10}	9.9154×10^{7}	2
CTA	2.0769×10^{10}	1.1011×10^{8}	1

4.2. Discussion

Overall, the evaluation result provides that CTA is a new method with well efficacy in taking both unconstrained and constrained problems. In general, CTA is superior in taking the high dimension functions whether they are unimodal or multimodal functions. This fact means that CTA has superior exploitation and exploration capabilities. Meanwhile, CTA has competing capability to balance exploitation and exploration as it is competing in taking the fixed dimension multimodal functions. The result of taking on the constrained problems provides that the efficacy of CTA is acceptable in taking these problems. CTA is superior in taking both PV and SR design problems. On the other hand, CTA is competing in taking on both SD and WB problems. Moreover, the superiority of CTA is seen in taking the EED problem.

HO performs as the poorest algorithm in almost all problems, whether they are unconstrained or constrained problems. The investigation can be carried out deeply on the approach that is used in HO. In general, HO is the only algorithm in the evaluation that does not employ stringent acceptance method. In HO, the worse solution candidate still replaces the existing solution. Besides, HO is the only algorithm that employs only single look while the others employ multiple looks. This fact highlights the supremacy of a stringent acceptance approach and multiple look methods to develop more capable metaheuristic algorithms. The stringent acceptance approach can prevent particles from flowing to the worsening situation while multiple look algorithm gives benefit where the strength of certain looks can cover the weakness of other looks and vice versa.

The result also highlights the no-free-lunch (NFL theory). Overall, the NFL theory highlights the imperfectness of any algorithm or algorithm that cannot solve all problems with superior results. An algorithm may perform well in some cases but inferior in other cases. This statement can be measured through the difference and the efficacy of the algorithm in taking the problems. In general, the difference among algorithms in the high dimension functions is wide but narrow in the fixed dimension functions. In the engineering problems, the difference is wide in three problems (PV, SD, and WB) but narrow in the SR. The difference is narrow in taking the EED problem. In general, DOA and POA become the second and third best algorithms. But, in some cases, OOA becomes the best algorithm.

Meanwhile, CTA still has limitations as it cannot cover all methods or approaches into a single algorithm. For example, despite the supremacy of the multiple look method proven more powerful, there is still a chance to loosen acceptance to a certain degree to prevent stagnation in several or multiple iterations as the solution candidates still fail to give enhancement in certain consecutive iterations. Meanwhile, there are numerous other adaptive approaches that can be explored to construct various adaptive algorithms in the future. There are also many adaptive mechanisms that already exist in many existing algorithms, especially the enhanced version of the standard ones. Besides, there is an approach that has been abandoned in the development of metaheuristics, which is the yellow flag or red flag mechanism. This mechanism is preventing a certain area or solution that is proven fail, which can be found in the old tabu look but now this mechanism is rare to find.

There is also limitation in assessing CTA in various optimization problems in a single article. There are other functions, such as CEC series that have not been accommodated in this work. Meanwhile, there are also derivatives of the economic dispatch problem, such as ELD, UC, OPF, and so on. Moreover, there are also various optimization problems in other sectors, especially in manufacturing and supply chain systems.

5. CONCLUSION

This article exhibited the concept, formalization, and efficacy investigation of the submitted CTA. The novel approach of CTA relies on the switching mechanism from exploitation-focused look to the exploration-focused look and vice versa. This mechanism is applied to create an adaptive algorithm due to the situation it faces whether it is stagnation or enhancement. Through evaluation, CTA has proven superior or competing compared to its five competitors. CTA is superior to POA, OOA, LOA, HO, and DOA in 13, 16, 19, 23, and 12 functions respectively. It means that CTA is absolute superior to HO. Overall, CTA is superior in taking the high dimension functions and competing in taking fixed dimension functions. CTA has become the best in taking pressure vessel and speed reducer design problems and the second best in taking welded beam and spring design problems. Moreover, CTA also becomes the best in taking EED problems in Java-Bali grid system. On the other hand, HO becomes the poorest algorithm in taking almost all functions, all four engineering problems, and EED problems.

The suggestion for future studies based on this work is as follows. First, the exploration to construct various adaptive methods is challenging. Then, the second track is utilizing CTA to solve numerous optimization problems, especially within the wide range of engineering sectors. The third track is hybridizing CTA with other optimization or looking algorithms or with various machine learning or deep learning algorithms to give better intelligent system.

FUNDING INFORMATION

This study was financially supported by Telkom University, Bandung, Indonesia without any specific grant.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Purba Daru Kusuma	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓		✓	
Helmy Widyantara		✓			✓	✓	✓			✓		✓		✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, [PBD], upon reasonable request.

REFERENCES

[1] 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.

[2] J. Hasudungan and J. Raharjo, “Determination of emission reduction costs through optimization of generator scheduling in Indonesia,” *International Journal of Energy Economics and Policy*, vol. 12, no. 3, pp. 395–400, May 2022, doi: 10.32479/ijeep.12929.

[3] M. A. Prakasa, I. Robandi, R. Nishimura, and M. R. Djalal, “A hybrid controlling parameters of power system stabilizer and virtual inertia using harris hawk optimizer in interconnected renewable power systems,” *IEEE Access*, vol. 12, pp. 76219–76243, 2024, doi: 10.1109/ACCESS.2024.3405994.

[4] 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, Dec. 2023, doi: 10.3390/en17010123.

[5] 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.

[6] A. Harrison *et al.*, “Probabilistic deep learning control of hybrid FACTS for real-time voltage stability in renewable-rich power grids,” *International Journal of Dynamics and Control*, vol. 14, no. 2, Feb. 2026, doi: 10.1007/s40435-025-01962-6.

[7] J. A. Muñoz-Guevara, E. Toro-Ocampo, and M. C. Vélez-Gallego, “Robotic assembly systems planning and scheduling problems: A review,” *International Journal of Industrial Engineering Computations*, vol. 15, no. 4, pp. 845–870, 2024, doi: 10.5267/j.ijiec.2024.8.001.

[8] L. Cheng, Q. Tang, L. Zhang, and Z. Li, “Inventory and total completion time minimization for assembly job-shop scheduling considering material integrity and assembly sequential constraint,” *Journal of Manufacturing Systems*, vol. 65, pp. 660–672, Oct. 2022, doi: 10.1016/j.jmsy.2022.10.013.

[9] E. E. Günay, “A two-stage stochastic model for picker allocation problem in warehouses considering the rest allowance and picker’s weight,” *International Journal of Industrial Engineering Computations*, vol. 15, no. 3, pp. 685–704, 2024, doi: 10.5267/j.ijiec.2024.5.001.

[10] S. Kır and S. E. Comert, “Improving a multi-echelon last mile delivery system by effective solution methods based on ant colony optimization,” *International Journal of Industrial Engineering Computations*, vol. 15, no. 4, pp. 977–996, 2024, doi: 10.5267/j.ijiec.2024.6.001.

[11] W. Hong, W. Yin, and S. Xu, “Collaborative truck-robot routing problem with meal delivery for the elderly on the personalized needs,” *International Journal of Industrial Engineering Computations*, vol. 15, no. 3, pp. 615–626, 2024, doi: 10.5267/j.ijiec.2024.5.005.

[12] O. Al-Baik *et al.*, “Pufferfish optimization algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems,” *Biomimetics*, vol. 9, no. 2, Jan. 2024, doi: 10.3390/biomimetics9020065.

[13] M. Dehghani, G. Bektemyssova, Z. Montazeri, G. Shaikemelev, O. P. Malik, and G. Dhiman, “Lyrebird optimization algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems,” *Biomimetics*, vol. 8, no. 6, Oct. 2023, doi: 10.3390/biomimetics8060507.

[14] M. Dehghani and P. Trojovský, “Osprey optimization algorithm: A new bio-inspired metaheuristic algorithm for solving engineering optimization problems,” *Frontiers in Mechanical Engineering*, vol. 8, Jan. 2023, doi: 10.3389/fmech.2022.1126450.

[15] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, “Marine predators algorithm: A nature-inspired metaheuristic,” *Expert Systems with Applications*, vol. 152, Aug. 2020, doi: 10.1016/j.eswa.2020.113377.

[16] R. Masadeh, A. Basel, and A. Sharieh, “Sea lion optimization algorithm,” *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 5, 2019, doi: 10.14569/IJACSA.2019.0100548.

[17] M. Braik, A. Hammouri, J. Atwan, M. A. Al-Betar, and M. A. Awadallah, “White shark optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems,” *Knowledge-Based Systems*, vol. 243, May 2022, doi: 10.1016/j.knosys.2022.108457.

[18] S. Suyanto, A. A. Ariyanto, and A. F. Ariyanto, “Komodo mlipir algorithm,” *Applied Soft Computing*, vol. 114, Jan. 2022, doi: 10.1016/j.asoc.2021.108043.

[19] H. Jia, H. Rao, C. Wen, and S. Mirjalili, “Crayfish optimization algorithm,” *Artificial Intelligence Review*, vol. 56, no. S2, pp. 1919–1979, Nov. 2023, doi: 10.1007/s10462-023-10567-4.

[20] N. Chopra and M. M. Ansari, “Golden jackal optimization: A novel nature-inspired optimizer for engineering applications,” *Expert Systems with Applications*, vol. 198, Jul. 2022, doi: 10.1016/j.eswa.2022.116924.

[21] R. N. Hasanah *et al.*, “A novel horse herd optimization algorithm for optimal economic dispatch in Sulbagsel electricity system,” *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 6, pp. 1059–1069, Dec. 2024, doi: 10.22266/ijies2024.1231.78.

[22] D. Połap and M. Woźniak, “Red fox optimization algorithm,” *Expert Systems with Applications*, vol. 166, Mar. 2021, doi: 10.1016/j.eswa.2020.114107.

[23] M. A. Al-Betar, M. A. Awadallah, M. S. Braik, S. Makhadmeh, and I. A. Doush, “Elk herd optimizer: a novel nature-inspired metaheuristic algorithm,” *Artificial Intelligence Review*, vol. 57, no. 3, Feb. 2024, doi: 10.1007/s10462-023-10680-4.

[24] M. H. Amiri, N. M. Hashjin, M. Montazeri, S. Mirjalili, and N. Khodadadi, “Hippopotamus optimization algorithm: A novel nature-inspired optimization algorithm,” *Scientific Reports*, vol. 14, no. 1, Feb. 2024, doi: 10.1038/s41598-024-54910-3.

[25] S. O. Oladejo, S. O. Ekwe, and S. Mirjalili, “The hiking optimization algorithm: A novel human-based metaheuristic approach,” *Knowledge-Based Systems*, vol. 296, Jul. 2024, doi: 10.1016/j.knosys.2024.111880.

[26] 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.

[27] 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.

[28] M. Ghasemi *et al.*, “A new metaphor-less simple algorithm based on rao algorithms: A fully informed search algorithm (FISA),” *PeerJ Computer Science*, vol. 9, Aug. 2023, doi: 10.7717/peerj-cs.1431.

[29] Z. Xu and X. Zhang, “An improve grey wolf optimizer algorithm for traveling salesman problems,” *IAENG International Journal of Computer Science*, vol. 51, no. 6, pp. 602–612, 2024.

[30] S. Sennan, S. S, R. Somula, D. Pandey, and Y. Cho, “A multi-objective grey wolf optimization algorithm for energy-efficient cluster-based routing in IoT-enabled WSNs,” *Scientific Reports*, vol. 16, no. 1, Dec. 2025, doi: 10.1038/s41598-025-28950-2.

[31] Y. Li *et al.*, “Dynamic modeling and techno-economic ANN-MOGWO optimization of a solar multigeneration system for hydrogen production and liquefaction,” *Fuel*, vol. 414, Jun. 2026, doi: 10.1016/j.fuel.2026.138254.

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

[33] D. Yousri, A. Fathy, and H. Rezk, "A new comprehensive learning marine predator algorithm for extracting the optimal parameters of supercapacitor model," *Journal of Energy Storage*, vol. 42, Oct. 2021, doi: 10.1016/j.est.2021.103035.

[34] M. Abdel-Basset, R. Mohamed, M. Elhoseny, A. K. Bashir, A. Jolfaei, and N. Kumar, "Energy-aware marine predators algorithm for task scheduling in IoT-based fog computing applications," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 5068–5076, Jul. 2021, doi: 10.1109/TII.2020.3001067.

[35] L. V. Ho *et al.*, "A hybrid computational intelligence approach for structural damage detection using marine predator algorithm and feedforward neural networks," *Computers & Structures*, vol. 252, Aug. 2021, doi: 10.1016/j.compstruc.2021.106568.

[36] J. Liu, Z. Hou, and T. Yin, "Short-term power load forecast using OOA optimized bidirectional long short-term memory network with spectral attention for the frequency domain," *Energy Reports*, vol. 12, pp. 4891–4908, Dec. 2024, doi: 10.1016/j.egyr.2024.10.050.

[37] M. A. Mohamed, M. Shadoul, H. Yousef, R. Al-Abri, and H. M. Sultan, "Multi-agent based optimal sizing of hybrid renewable energy systems and their significance in sustainable energy development," *Energy Reports*, vol. 12, pp. 4830–4853, Dec. 2024, doi: 10.1016/j.egyr.2024.10.051.

[38] O. Alsayyed *et al.*, "Giant armadillo optimization: A new bio-inspired metaheuristic algorithm for solving optimization problems," *Biomimetics*, vol. 8, no. 8, Dec. 2023, doi: 10.3390/biomimetics8080619.

[39] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[40] W. Zhao, X. Bian, and X. Mei, "An adaptive multi-objective genetic algorithm for solving heterogeneous green city vehicle routing problem," *Applied Sciences*, vol. 14, no. 15, Jul. 2024, doi: 10.3390/app14156594.

[41] Y.-B. Li, H.-B. Sang, X. Xiong, and Y.-R. Li, "An improved adaptive genetic algorithm for two-dimensional rectangular packing problem," *Applied Sciences*, vol. 11, no. 1, Jan. 2021, doi: 10.3390/app11010413.

[42] H. Fei, B. Zhang, Y. Liu, M. Yan, Y. Lu, and J. Zhou, "A novel chaotic elite adaptive genetic algorithm for task allocation of intelligent unmanned wireless sensor networks," *Applied Sciences*, vol. 13, no. 17, Aug. 2023, doi: 10.3390/app13179870.

[43] K. M. D. Puspitasari, J. Raharjo, A. S. Sastrosubroto, and B. Rahmat, "Generator scheduling optimization involving emission to determine emission reduction costs," *International Journal of Engineering*, vol. 35, no. 8, pp. 1468–1478, 2022, doi: 10.5829/IJE.2022.35.08B.02.

BIOGRAPHIES OF AUTHORS

Purba Daru Kusuma received his bachelor's and master's degrees in electrical engineering from Bandung Institute of Technology, Indonesia, in 2002 and 2009, respectively. He received his doctoral degree in computer science from Gadjah Mada University, Indonesia, in 2017. Recently, he is an associate professor at Telkom University, Indonesia. His research area is artificial intelligence, machine learning, and operational research. He can be contacted at email: purbodaru@telkomuniversity.ac.id.

Helmy Widyantara recently he is an associate professor in Telkom University, Indonesia. He received his bachelor's degree from STMIK Surabaya in 2002. He received master's degree from Gadjah Mada University in 2010 and doctoral degree from Institut Teknologi Sepuluh Nopember in 2020. The field of research interests are but are not limited to embedded systems, internet of things, and smart farming. He can be contacted at email: helmywidyantara@telkomuniversity.ac.id.