Investigation on the Acceptance Strategy of Swarm-based Metaheuristics

Purba Daru Kusuma, *Member, IAENG*

***Abstract*—In general, there are two acceptance strategies that are adopted in swarm-based metaheuristics. The first strategy is a stringent acceptance strategy where a solution candidate is accepted for replacement only if it improves the current solution. On the other hand, the second strategy is a loose acceptance strategy where a solution candidate is always accepted for replacement without considering the comparative quality between the candidate and the current solution. Despites the massive development of swarm-based metaheuristics, the investigation of the acceptance strategy is hard to find as many studies focused on assessing their proposed metaheuristics compared to the existing metaheuristics. Based on this problem, this work is aimed to investigate the performance of these acceptance strategies in the relation with the performance of the metaheuristics. Moreover, some conditional acceptance strategies where the worse candidate has an opportunity to replace the current solution are introduced and assessed. These strategies are implemented in two new swarm-based metaheuristics including golden search optimization (GSO) and total interaction algorithm (TIA). The result shows that the performance difference among these five acceptance strategies is not significant so that the key finding is that the acceptance strategy is not the key determinant for the performance of metaheuristics, especially the swarm-based ones in 23 classic functions and EED problem. Meanwhile, a mixed relation is found in engineering design problems. In the future, more studies to investigate the contribution of other factors that construct the metaheuristics is important.**

***Index Terms*—swarm intelligence, metaheuristic, acceptance, golden search optimization, total interaction algorithm.**

I. Introduction

METAHEURISTIC is highly related to optimization works. Many optimization studies employed metaheuristics as optimization techniques after the model has been built due to several reasons. The first reason is that there are a lot of metaheuristics available to be utilized in various fields. The second reason is that metaheuristics are flexible [1] to be implemented in various fields of studies as they abstracts the problem formulation by focusing on the objective function and the constraints or boundaries. Third, metaheuristics utilize iterative and stochastic process [2] so that they are adaptive with environment with limited computational resources with the consequence that they do not guarantee the exact optimal solution but only the quasi-optimal one [1].

Some studies were conducted by employing, improving, or combining the existing metaheuristics to solve practical optimization problems. The examples are as follows. Grey wolf optimizer and teaching learning-based optimization were combined to optimize the security of smart grid system [3]. Ant colony optimization (ACO) has been combined with independent component analysis (ICA) algorithm to reduce the noise of night vision image [4]. Harris hawk optimization has been improved to solve the problem of the non-line-of-sight errors in the ultra-wide band light detection and ranging (LiDAR) system [5]. The popular marine predator algorithm (MPA) has been employed to solve various optimization works, such as power flow [6], structural damage detection [7], heat-power system [8], super capacitor model [9], and so on. The classic particle swarm optimization (PSO) has been employed in many optimization studies, such as timetabling [10], 5G network [11], big data clustering [12], fresh product distribution [13], image retrieval system [14], network reconfiguration [15], load shedding [16], energy management [17], and so on. grey wolf optimization (GWO) has been utilized to solve a lot of optimization works, such as task scheduling in cloud system [18], lung cancer classification [19], DC-DC boost converter, [20], and so on.

On the other hand, there are also a lot of studies that focus on introducing new metaheuristics, especially based on swarm intelligence. These metaheuristics can be called as swarm-based metaheuristics. Some metaheuristics utilized the animal behavior as metaphors, such as walrus optimization algorithm (WaOA) [21], Komodo mlipir algorithm (KMA) [22], stochastic komodo algorithm (SKA) [23], marine predator algorithm (MPA) [24], coati optimization algorithm (COA) [25], pelican optimization algorithm (POA) [26], golden jackal optimization (GJO) [27], crayfish optimization algorithm (COA) [28], and so on.. Some metaheuristics exploit physical mechanism as metaphor, like swarm magnetic optimizer [29]. Some metaheuristics do not used metaphors so that the nomenclature is based on their main strategy, such as golden search optimization (GSO) [30], group better-worse optimization (GBWO) [31], total interaction algorithm (TIA) [32], fully informed search algorithm (FISA) [33], average subtraction-based optimization (ASBO) [34], three influential members based optimization (TIMBO) [35], ransom selected leader based optimization (RSLBO) [36], multiple interaction optimizer (MIO) [37], and so on.

Manuscript received April 27, 2024; revised November 27, 2024. This work was financially supported by Telkom University, Indonesia.

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

In general, this massive studies on metaheuristics are conducted in two tracks. The first track is by introducing new metaheuristic whether it is constructed based on certain metaphor or free from metaphor. In this track, the proposed metaheuristic was assessed with standard theoretical sets of functions, such 23 classic functions or CEC series as primary optimization problem. In some studies, some additional engineering optimization problems were also added as complementary problems like mechanical engineering design problem [38]. The additional assessment is the sensitivity assessment. As an iterative based technique, an assessment with certain values of maximum iteration is conducted to observe the performance improvement due to the increase of maximum iteration.

The second track is by employing the existing metaheuristic to solve practical optimization problem. This metaheuristic can be employed in its original form, the modified form, or the hybridized form. In general, the studies in this second track consists of two models. The first model is the model of the problem that consists of objective and constraints. The second model is the metaheuristic model whether it is the original or modified form.

Despite massive development and implementation of metaheuristics, the assessment is conducted to assess the metaheuristic as a whole package. Meanwhile, in general, swarm-based metaheuristic is constructed by three aspects: searching strategy, swarm split, and the acceptance strategy. The searching strategy is constructed by three aspects: the type of searches, step size, and the random distribution. A new metaheuristic or the modification of the existing metaheuristics can be conducted by modifying one or multiple aspects. Unfortunately, studies which are dedicated to investigating these aspects in a more specific manner are hard to find.

Based on this problem, this work is aimed at investigating the contribution of the acceptance strategies to the performance of the swarm-based metaheuristics. This work fills the gap in the metaheuristic development subjects where in general, the studies of metaheuristics focus on the introduction of new metaheuristics, modifying the existing metaheuristics, or employing the existing metaheuristics to solve practical optimization problems.

The list of scientific contributions of this work is as follows.

- This paper investigates the contribution the acceptance strategy to the performance of the swarm-based metaheuristics in the context of the quality of the final solution.
- This paper compares five acceptance strategies including stringent acceptance strategy, loose acceptance strategy, and three conditional acceptance strategies.
- This assessment is conducted by employing these strategies on two recent swarm-based metaheuristics.
- This assessment is conducted by optimizing the theoretical optimization problems that are represented by the 23 classic functions and the practical optimization problem that is represented by the EED problem.

The following sections of this paper are organized as follows. Section two presents the model of five acceptance strategies including the stringent acceptance strategy, loose acceptance strategy, and three conditional acceptance strategies. Section three presents the assessment of these strategies which are implemented in two new swarm-based metaheuristics in solving the theoretical optimization problems. Section four provides the discussion regarding the comprehensive analysis of the assessment result, findings, and limitations. Section five provides the concluding remarks and recommended paths for future studies.

II. Models of Acceptance Strategy in Swarm-based Metaheuristic

As mentioned previously, the acceptance strategy is a strategy to determine whether the solution candidate which is produced after the searching process will be accepted or rejected. In general, a solution candidate whose quality is better than the current solution will be accepted to replace the current solution. Meanwhile, there are several approaches that can be chosen in facing a solution candidate whose quality is worse than the current solution.

There are five strategies that will be presented in this section. In the first strategy, a worse solution candidate is rejected for the replacement. In the second strategy, a worse solution candidate is still accepted for the replacement. In the next three strategies, the worse solution candidate still has opportunity for replacement based on a stochastic calculation. In the third strategy, a static threshold is set. If a generated random number is less than the threshold, then the worse solution candidate will be accepted. Otherwise, this solution candidate is rejected. In the fourth strategy, the threshold is not static but is obtained by dividing the iteration with the maximum iteration. In the fifth strategy, the threshold is obtained based on an exponential calculation of the iteration. Simulated annealing is the classic example of metaheuristic that employ conditional acceptance strategy. The summary of recent metaheuristics and their acceptance strategy is provided in Table 1.

TABLE I
LIST OF SEVERAL RECENT METAHEURISTICS AND THEIR ACCEPTANCE STRATEGY

No	Metaheuristic	Metaphor	Acceptance
1	WaOA [21]	walrus	strict
2	KMA [22]	komodo	loose
3	GWO [39]	grey wolf	loose
4	MPA [24]	marine predator	loose
5	COA [25]	coati	strict
6	POA [26]	pelican	strict
7	GJO [27]	golden jackal	loose
8	COA [28]	crayfish	strict
9	GSO[30]	-	loose
10	GBWO [31]		strict
11	FISA [33]	-	strict
12	ASBO [34]	-	strict
13	TIMBO [35]	-	strict
14	RSLBO [36]	-	strict
15	SMO [29]	magnet	strict
16	MIO [37]	-	strict

The first strategy is a worse candidate is rejected for the replacement. This strategy is called a stringent acceptance model. This strategy is designed to avoid the swarm toward the worse solution. This concept means that only a better solution candidate can replace the current solution. This strategy also plays an important role in keeping the current solution still around the optimal solution. This strategy is employed in many swarm-based metaheuristics, such as TIA [32], GBWO [31], WaOA [31], COA [25], and so on.

The second strategy is a worse solution candidate is still accepted to replace the current solution. It means that the quality of the solution candidate is not considered for the replacement. This strategy is called a loose acceptance model. This strategy may lead the swarm toward the worse solution. On the other hand, this strategy becomes the answer in facing the local optimal problem. This strategy is employed in many swarm-based metaheuristics, such as MPA [24], GWO [39], GSO [30], GJO [27], and so on.

In multimodal problems, there are multiple optimal solutions but only one global optimal solution. The other solutions are the local optimal solutions. This circumstance becomes the classic issue for metaheuristic development which employs stochastic approach so that not all solutions are traced. This issue can be called the local optimal entrapment where the solution is trapped in the local optimal solution so that the global optimal solution is never found. Sometimes, a swarm member should move to the worse solution to reach the global optimal solution or at least a better optimal solution.

By employing the first strategy, a swarm member will never move from its current solution because a worse solution is not accepted. Meanwhile, by employing the second strategy, there is an opportunity that the global optimal solution will be found. But there is also the opportunity that the swarm member is thrown away to worse area within the space. The second strategy provides better exploration capability by giving better opportunity to trace to wider space.

The next three strategies can be seen as a compromise between the first strategy and the second strategy. The worse solution candidate still can replace the current solution or swarm member. But this replacement is not guaranteed. These three strategies are conducted based on a certain stochastic calculation. In the third strategy, the iteration is not considered in the decision-making process. Meanwhile, the iteration is considered in the decision-making process in the fourth and fifth strategies.

In the third strategy, a fixed threshold is determined. The value of the threshold ranges from 0 to 1. This value is static during the iteration. Then, each time a worse solution candidate is produced, a uniform random number from 0 to 1 is generated. If the generated random number is less than the threshold, then this solution candidate is accepted. On the other hand, if the generated random number is higher than the threshold, the solution candidate is rejected. In this context, the higher the threshold means the easier the worse solution candidate to replace the current solution. In other words, a higher threshold represents the more space for exploration while a lower threshold represents the more space for exploitation. The visualization of this threshold is presented in Fig. 1.

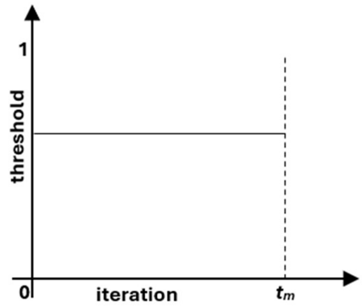

Fig. 1 The conditional acceptance strategy where the threshold is static during iteration.

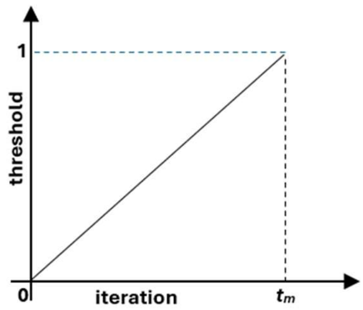

Fig. 2 The conditional acceptance strategy where the threshold increases linearly during iteration.

In the fourth strategy, the threshold is dynamic. It is determined by dividing the iteration with the maximum iteration. If a generated uniform random number from 0 to 1 is higher than the threshold, then the worse solution candidate is accepted to replace the current solution or swarm member. Otherwise, this worse solution candidate is rejected. Based on the calculation, the threshold will increase as the iteration goes on. When the iteration is equal to the maximum iteration then the value of the threshold is 1. This increase of the threshold is linear to the iteration. The visualization of the threshold relative to the iteration is presented in Fig. 2. This circumstance shows that the worse solution tends to be difficult to accept as iteration goes on. This circumstance also represents the linear shifting from exploration to exploitation which is controlled by the iteration.

In the fifth strategy, the threshold is also dynamic. The difference between the fourth and fifth strategy is that the trend of the threshold follows the sinusoidal functions. As known in sine function, the input ranges from 0 to $\pi/2$ so that the output of the sine function ranges from 0 to 1. The input from 0 to $\pi/2$ moves from the iteration 0 to the maximum iteration. But the rise in the output is not linear as it rises faster but steadily faster too. As the input which is uniformly spread from 0 to $\pi/2$, the output is not uniform from 0 to 1. As the worse solution candidate is only accepted if a uniformly generated random number should be higher than the threshold, then this strategy also represents the shifting from the exploration to exploitation as iteration goes on, but the portion of the exploitation is higher than the exploration. The visualization of this trend in threshold is presented in Fig. 3.

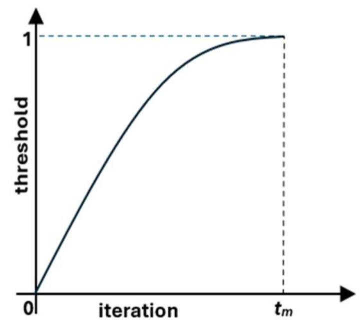

Fig. 3 The conditional acceptance strategy where the threshold follows sinusoidal trend during iteration.

This concept is then mathematically formalized using (1) to (5). In general, the current solution or swarm member is annotated using x while the solution candidate is annotated using c. The objective function is annotated using f. The iteration is annotated using t while the maximum iteration is annotated using t_{max}. In this paper, the optimization employes minimization so that the first entity is better than the second entity if its quality is less than the quality of the second entity.

The model of the first strategy is presented using (1). As shown in (1), the solution candidate is accepted for replacement only if its quality is better than the current solution. Otherwise, the current solution is still used in the next iteration.

$$x' = \begin{cases} c, f(c) < f(x) \\ x, else \end{cases} \tag{1}$$

The model of the second strategy is presented using (2). As shown in (2), the solution candidate becomes the next solution automatically. The comparative quality between the solution candidate and the current solution.

$$x' = c \tag{2}$$

The models of the third to fifth strategies are formalized using (3) to (5) consecutively. In every model, there are three cases. In the first and second cases, the solution candidate is accepted to replace the current solution. In the third case, the solution candidate is rejected. Meanwhile, in the first case, the solution candidate is accepted because it is better than the current solution. On the other hand, in the second case, the solution candidate is still accepted although its quality is worse than the current solution.

$$x' = \begin{cases} c, f(c) < f(x) \\ c, f(c) \geq f(x) \wedge U(0,1) < T \\ x, f(c) \geq f(x) \wedge U(0,1) \geq T \end{cases} \tag{3}$$

$$x' = \begin{cases} c, f(c) < f(x) \\ c, f(c) \geq f(x) \wedge U(0,1) > \frac{t}{t_m} \\ x, f(c) \geq f(x) \wedge U(0,1) \leq \frac{t}{t_m} \end{cases} \tag{4}$$

$$x' = \begin{cases} c, f(c) < f(x) \\ c, f(c) \geq f(x) \wedge U(0,1) > \sin\left(\frac{t\pi}{2t_m}\right) \\ x, f(c) \geq f(x) \wedge U(0,1) \leq \sin\left(\frac{t\pi}{2t_m}\right) \end{cases} \tag{5}$$

algorithm 1: general model of swarm-based metaheuristic

```
begin
 for all x in X
  perform random search
 end for
 for t=1 to t_m
  for all x in X
   perform search
   employ acceptance strategy
  end for
 end for
end
```

The difference between (3) to (5) is mainly on the threshold that is employed. Equation (3) shows that the threshold is static. Equation (4) shows that the threshold is the division between the iteration and the maximum iteration. Equation (5) shows that the threshold is the sine value of the division of iteration with the maximum iteration relative to π/2.

In general, the acceptance strategy is implemented during the iteration phase in the swarm-based metaheuristic. The acceptance strategy is not implemented in the initialization phase as the initial swarm member is generated for the first time in this phase. Meanwhile, in the iteration phase, the acceptance strategy is employed after a search is performed so that a solution candidate is generated. The illustration of general swarm-based metaheuristic can be seen in algorithm 1.

III. Simulation and Result

This section presents the evaluation of the five acceptance strategies. These five strategies are implemented into two new swarm-based metaheuristics: TIA and GSO. Both metaheuristics are chosen based on the reason that they employ different strategies. TIA employs stringent acceptance strategy which is the first strategy. On the other hand, GSO employs a loose acceptance strategy which is the second strategy.

TABLE II
FUNCTIONS

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

TABLE III
ASSESSMENT RESULT ON HIGH-DIMENSION UNIMODAL FUNCTIONS EMPLOYED IN TOTAL INTERACTION ALGORITHM

Function	Parameters	1st strategy	2nd strategy	3rd strategy	4th strategy	5th strategy
1	average	0.0007	0.0008	0.0008	0.0011	0.0009
	average rank	1	2	2	5	4
2	average	0.0000	0.0000	0.0000	0.0000	0.0000
	average rank	1	1	1	1	1
3	average	$1.6352x10^1$	0.6891	1.3463	2.0295	5.1403
	average rank	5	1	2	3	4
4	average	0.0536	0.0515	0.0480	0.0567	0.0426
	average rank	4	3	2	5	1
5	average	$2.8887x10^1$	$2.8879x10^1$	$2.8928x10^1$	$2.8902x10^1$	$2.8909x10^1$
	average rank	2	1	5	3	4
6	average	4.8238	6.1815	5.5080	4.9191	4.9320
	average rank	1	5	4	2	3
7	average	0.0246	0.2643	0.1571	0.0423	0.0291
	average rank	1	5	4	3	2

TABLE IV
ASSESSMENT RESULT ON HIGH-DIMENSION MULTIMODAL FUNCTIONS EMPLOYED IN TOTAL INTERACTION ALGORITHM

Function	Parameters	1st strategy	2nd strategy	3rd strategy	4th strategy	5th strategy
8	average	$-1.7235x10^3$	$-7.9077x10^1$	$-2.2674x10^2$	$-3.4851x10^2$	$-5.4454x10^2$
	average rank	1	5	4	3	2
9	average	0.2274	0.0024	0.4727	0.0049	0.1002
	average rank	4	1	5	2	3
10	average	0.0082	0.0070	0.0065	0.0063	0.0064
	average rank	5	4	3	1	2
11	average	0.0058	0.0003	0.0231	0.0376	0.0104
	average rank	2	1	4	5	3
12	average	0.8331	1.2699	0.9418	0.8298	0.7599
	average rank	3	5	4	2	1
13	average	2.9868	3.1963	3.1925	3.0079	3.0702
	average rank	1	5	4	2	3

TABLE V
ASSESSMENT RESULT ON FIXED-DIMENSION MULTIMODAL FUNCTIONS EMPLOYED IN TOTAL INTERACTION ALGORITHM

Function	Parameters	1st strategy	2nd strategy	3rd strategy	4th strategy	5th strategy
14	average	9.2971	$1.2494x10^1$	$1.2901x10^1$	$1.2296x10^1$	$1.1724x10^1$
	average rank	1	4	5	3	2
15	average	0.0046	0.0215	0.0201	0.0088	0.0102
	average rank	1	5	4	2	3
16	average	-1.0110	-0.8823	-0.9305	-0.9297	-0.8834
	average rank	1	5	2	3	4
17	average	3.7152	$1.0673x10^1$	4.1293	6.2493	4.3921
	average rank	1	5	2	4	3
18	average	$1.8019x10^1$	$1.5933x10^2$	$5.1323x10^1$	$4.9869x10^1$	$3.5941x10^1$
	average rank	1	5	4	3	2
19	average	-0.0495	-0.0495	-0.0495	-0.0495	-0.0495
	average rank	1	1	1	1	1
20	average	-2.3183	-0.1427	-0.6593	-0.7924	-1.2651
	average rank	1	5	4	3	2
21	average	-2.8638	-0.4071	-0.8076	-1.1624	-1.7222
	average rank	1	5	4	3	2
22	average	-3.2520	-0.4498	-0.9556	-1.2880	-2.1359
	average rank	1	5	4	3	2
23	average	-2.5871	-0.4878	-1.1415	-1.4282	-2.0815
	average rank	1	5	4	3	2

TABLE VI
ASSESSMENT RESULT ON HIGH-DIMENSION UNIMODAL FUNCTIONS EMPLOYED IN GOLDEN SEARCH OPTIMIZATION

Function	Parameters	1st strategy	2nd strategy	3rd strategy	4th strategy	5th strategy
1	average	$3.2741x10^4$	$3.0638x10^4$	$2.9921x10^4$	$3.5133x10^4$	$3.3750x10^4$
	average rank	3	2	1	5	4
2	average	$5.4966x10^{33}$	$1.9847x10^{39}$	$1.0752x10^{40}$	$1.3509x10^{37}$	$1.0121x10^{35}$
	average rank	1	4	5	3	2
3	average	$7.4495x10^4$	$6.8453x10^4$	$6.4050x10^4$	$6.2790x10^4$	$5.6410x10^4$
	average rank	5	4	3	2	1
4	average	$6.6322x10^1$	$5.9451x10^1$	$6.2827x10^1$	$6.1360x10^1$	$5.8332x10^1$
	average rank	5	2	4	3	1
5	average	$6.3658x10^7$	$5.6605x10^7$	$7.5547x10^7$	$5.6301x10^7$	$6.2597x10^7$
	average rank	4	2	5	1	3
6	average	$3.0918x10^4$	$3.0686x10^4$	$2.8142x10^4$	$2.9104x10^4$	$3.2650x10^4$
	average rank	4	3	1	2	5
7	average	$3.1716x10^1$	$3.5224x10^1$	$3.0913x10^1$	$3.2721x10^1$	$3.7816x10^1$
	average rank	2	4	1	3	5

TABLE VII
ASSESSMENT RESULT ON HIGH-DIMENSION MULTIMODAL FUNCTIONS EMPLOYED IN GOLDEN SEARCH OPTIMIZATION

Function	Parameters	1st strategy	2nd strategy	3rd strategy	4th strategy	5th strategy
8	average	-2.2367x10^{3}	-2.4113x10^{3}	-2.2784x10^{3}	-2.5789x10^{3}	-2.6385x10^{3}
	average rank	5	3	4	2	1
9	average	2.8053x10^{2}	2.6911x10^{2}	2.6911x10^{2}	2.7281x10^{2}	2.9245x10^{2}
	average rank	4	1	1	3	5
10	average	1.9174x10^{1}	1.9086x10^{1}	1.8545x10^{1}	1.8762x10^{1}	1.8979x10^{1}
	average rank	5	4	1	2	3
11	average	2.9647x10^{2}	3.0877x10^{2}	2.9908x10^{2}	3.1871x10^{2}	3.1614x10^{2}
	average rank	1	3	2	5	4
12	average	1.2575x10^{8}	1.0350x10^{8}	1.0867x10^{8}	1.1136x10^{8}	8.6555x10^{7}
	average rank	5	2	3	4	1
13	average	2.8087x10^{8}	2.1910x10^{8}	2.6117x10^{8}	2.2338x10^{8}	2.5367x10^{8}
	average rank	5	1	4	2	3

TABLE VIII
ASSESSMENT RESULT ON FIXED-DIMENSION MULTIMODAL FUNCTIONS EMPLOYED IN GOLDEN SEARCH OPTIMIZATION

Function	Parameters	1st strategy	2nd strategy	3rd strategy	4th strategy	5th strategy
14	average	3.6605x10^{1}	1.2402x10^{1}	2.2731x10^{1}	2.3156x10^{1}	1.3922x10^{1}
	average rank	5	1	3	4	2
15	average	0.0849	0.4188	0.0379	0.5161	0.1364
	average rank	2	4	1	5	3
16	average	-0.7724	-0.6922	-0.6621	-0.8358	-0.9173
	average rank	3	4	5	2	1
17	average	2.2027	1.7329	2.3434	2.1649	1.8252
	average rank	4	1	5	3	2
18	average	9.0030x10^{1}	2.1156x10^{1}	1.0705x10^{2}	2.6793x10^{1}	7.6161x10^{1}
	average rank	5	2	1	3	4
19	average	-0.0082	-0.0114	-0.0157	-0.0115	-0.0060
	average rank	4	3	1	2	5
20	average	-1.8883	-2.1027	-2.2732	-2.0803	-1.8349
	average rank	4	2	1	3	5
21	average	-1.5597	-1.8033	-2.1171	-1.4151	-1.7751
	average rank	4	2	1	5	3
22	average	-1.1257	-2.3757	-2.2582	-1.5889	-1.6753
	average rank	5	1	2	4	3
23	average	-1.5287	-2.4679	-2.1369	-2.1447	-2.0804
	average rank	5	1	3	2	4

In the first evaluation, the set of 23 classic functions is chosen as the optimization problems. This set is chosen as it covers various circumstances of optimization problems. It contains seven high dimension unimodal functions (HDU), six high dimension multimodal functions (HDM), and ten fixed dimension multimodal functions (FDM). A detailed description of these functions is presented in Table 2. The swarm size is 5 and the maximum iteration is 20. The threshold is set to 0.5.

The result of the assessment is presented in Table 3 to Table 8. Table 3 to Table 5 exhibit the result where the metaheuristic is TIA. On the other hand, Table 6 to Table 8 exhibit the result where the metaheuristic is GSO. There are two parameters observed in each table: the average fitness score and the average rank.

Table 3 to Table 5 show that the stringent acceptance strategy performs the best in the first evaluation. It produces the best result in sixteen functions which can be distributed in four HDUs, two HDMs, and ten FDMs. On the other hand, the loose acceptance strategy produces the best result in six functions which can be distributed in three HDUs, two HDMs, and one FDM.

In general, the performance difference among strategies is narrow. This narrow performance difference can be found in six HDUs, four HDMs, and ten FDMs. The wide performance difference can be found in f_7, f_9, and f_{11}.

The result of the second evaluation also shows that the stringent acceptance strategy is not so dominant as in the first evaluation. The first to fifth strategies provide the best result in 2, 6, 9, 1, and 5 functions respectively. Moreover, the performance difference among the five acceptance strategies is narrow in 22 functions. The wide performance difference occurs only in f_{15}.

TABLE IX
ASSESSMENT RESULT ON EED PROBLEM IN TOTAL INTERACTION ALGORITHM

Strategy	Total Cost (IDR/hour)	
	p_d = 12,228 MW	p_d = 13,108 MW
1st strategy	21,130,047,367	23,456,590,460
2nd strategy	22,049,290,048	24,208,747,198
3rd strategy	21,810,359,021	24,000,454,685
4th strategy	21,427,325,392	23,649,622,928
5th strategy	21,317,966,725	23,699,068,075

TABLE X
ASSESSMENT RESULT ON EED PROBLEM IN GOLDEN SEARCH OPTIMIZATION

Strategy	Total Cost (IDR/hour)	
	p_d = 12,228 MW	p_d = 13,108 MW
1st strategy	22,083,441,218	24,141,855,544
2nd strategy	21,917,531,019	24,034,430,994
3rd strategy	21,966,035,832	24,381,204,827
4th strategy	22,370,867,986	24,045,912,344
5th strategy	22,341,984,211	24,311,585,049

TABLE XI
ASSESSMENT RESULT ON ENGINEERING DESIGN PROBLEMS USING TOTAL INTERACTION ALGORITHM

Strategy	Optimal Score			
	Pressure Vessel	Speed Reducer	Welded Beam	Spring
1st strategy	1.1526×10^{1}	3.5797×10^{3}	7.7874×10^{9}	1.8387×10^{1}
2nd strategy	3.8225×10^{12}	3.6660×10^{3}	8.0026×10^{12}	8.9294×10^{2}
3rd strategy	5.9537×10^{11}	3.6050×10^{3}	2.6313×10^{11}	1.4363×10^{2}
4th strategy	4.1380×10^{12}	3.6711×10^{3}	9.1442×10^{12}	9.8231×10^{2}
5th strategy	4.2430×10^{12}	3.6544×10^{3}	9.2001×10^{12}	1.5546×10^{3}

TABLE XII
ASSESSMENT RESULT ON ENGINEERING DESIGN PROBLEMS USING GOLDEN SEARCH OPTIMIZATION

Strategy	Optimal Score			
	Pressure Vessel	Speed Reducer	Welded Beam	Spring
1st strategy	1.6805×10^{6}	3.8554×10^{3}	4.8900×10^{9}	4.0148×10^{1}
2nd strategy	2.1342×10^{6}	3.8595×10^{3}	7.8645×10^{10}	4.6500×10^{1}
3rd strategy	1.7974×10^{6}	3.8605×10^{3}	1.4328×10^{10}	4.3449×10^{1}
4th strategy	2.5664×10^{6}	3.8800×10^{3}	2.9640×10^{10}	3.6004×10^{1}
5th strategy	2.3248×10^{6}	3.8779×10^{3}	7.2907×10^{9}	3.9250×10^{1}

In the second use cases, the economic emission dispatch (EED) problem represents the constrained problem. EED problem is an optimization problem which is the variance of economic load dispatch (ELD) problem. Its objective is to minimize both operational cost and emission reduction cost of a power system that consists of a certain number of power plants or generating units. Each generating unit provides power within its own power range while these generating units work collectively to meet the power demand where the accumulated power of all generating units should be equal to the power demand.

In this work, the use case is a grid system in Indonesia named Java-Bali electricity system which is the biggest grid system in Indonesia. This system comprises eight generating units where six of them are thermal units while the two others are hydro-power units. The specification of this system including the power range, cost function constants, and the cost function can be found in [40]. There are two power demands that should be met, which are 12,228 MW and 13,108 MW. The result is presented in Table 9 and Table 10. Table 9 provides the result where TIA is chosen as metaheuristic while Table 10 provides the result where GSO is chosen as metaheuristic.

Table 9 shows the tough competition among strategies in both power demand scenarios when TIA is chosen. In the first power, the first strategy becomes the best performer in both power demand scenarios. On the other hand, the second strategy becomes the worst performer in both power demand scenarios.

The tough competition also occurs when GSO is chosen as the metaheuristic as presented in Table 10. In the first power demand scenario, the second strategy becomes the best strategy while the fourth strategy becomes the worst strategy. Meanwhile, in the second power plant scenario, the second strategy becomes the best strategy while the third strategy becomes the worst strategy.

The third use case is four engineering design problems. These problems include pressure vessel design problem, speed reducer design problem, welded beam design problem, and spring design problem. These four engineering design problems are constrained optimization problems. Meanwhile, in this third use case, the quadratic penalty function is implemented so that certain penalty is added to the objective function as consequence for constraint violation. A detailed description of these four engineering design problems can be found in [25]. Meanwhile the result of this assessment is provided in Table 11 for TIA and Table 12 for GSO.

Table 11 reveals that the first acceptance strategy becomes the best option for TIA in handling four engineering design problems. This first strategy provides the best result compared to other strategies in four problems. The performance gap between the first strategy and the other four strategies in the pressure vessel design problem is very wide. Meanwhile, the gap among the other strategies in this problem is narrow. It can be said that there are two clusters in this pressure vessel design problem. The first cluster consists of the first strategy while the second cluster consists of the second to fifth strategies. Contrast circumstance occurs in the speed reducer design problem. In this second problem, the performance gap among the five strategies is narrow. The performance gap between the first strategy and other strategies is wide in welded beam design problem. This circumstance is like the first problem. But the distance between the first cluster and the second cluster is not so far as the first problem. The performance gap among the five strategies is wide in handling the spring design problem. But the performance gap between the best strategy and the worst strategy is not so wide as the first problem. Moreover, the distribution of the performance of these five strategies cannot be clustered as in previous problems.

Overall, the result in Table 11 reveals that the stringent acceptance strategy plays a critical role in TIA in handling four engineering design problems with different critical levels. Meanwhile, accepting worse solution candidate with various scenarios as in the second to fifth strategies is not suitable for TIA. This significance is high in handling pressure vessel design problem, low or almost zero in handling speed reducer design problem, and moderate in handling both welded beam and spring design problems.

Different circumstances are found in GSO as provided in Table 12. The performance gap between the best and worst strategies is not significant in all four engineering design problems. Meanwhile, the significance of these strategies is little bit moderate in handling welded beam design where the first strategy becomes the best strategy while the second strategy becomes the worst strategy. In this case, stringent acceptance is moderately better than loose acceptance.

There are different levels of gap between TIA and GSO in handling engineering design problems. The best result of TIA is far better than the best result in GSO in handling pressure vessel design. Meanwhile, the worst result of TIA is far worse than the worst result in GSO. The performance of TIA and GSO tends to be equal in handling the speed reducer problem. In the welded beam design problem, the best result in TIA is a little bit worse than GSO while the worst result in TIA is worse than GSO. In the spring design problem, the best result of TIA is better than the best result in GSO while the worst result of TIA is worse than the worst result in GSO.

IV. Discussion

The evaluation result provides an important or key finding that the acceptance strategy is not the determinant factor of swarm-based metaheuristic. This result occurs in all assessments, whether the use case is a set of 23 functions or the EED problem. In other words, the contribution of the acceptance strategy in affecting the performance of swarm-based metaheuristic is not significant. Moreover, by comparing the result in TIA and GSO, there are other factors that affect the performance difference between these two metaheuristics rather than the acceptance strategy.

This result also raises a second finding regarding the shifting strategy from exploration to exploitation during iteration. As previously mentioned, the third, fourth, and fifth strategies are designed for this purpose. The objective of this shifting is to give opportunity for the optimization process to explore as wide as possible in the early iteration.

On the other hand, the exploitation focus during the late iteration is designed to avoid the swarm member to be thrown away from the achievement so far so that the optimization should be restarted. This approach is conducted by several swarm-based metaheuristics such as MPA but in a different way. In MPA [24], this shifting is conducted by splitting the iteration into three equal frames and a specific search is performed in each frame.

This third finding is that the type of problem does not have a relation with the chosen acceptance strategy in handling the 23 standard functions. It is shown that all acceptance strategies perform similarly in general, whether the functions are unimodal or multimodal. It is also shown that the performance is similar whether the dimension is low as it is found in fixed dimension functions or high as it is found in high dimension functions.

The fourth finding is that there is a mixed relation between the type of the problem and the chosen acceptance strategy in handling the engineering design problems. This relation depends on the metaheuristics that is chosen. In TIA, the stringent acceptance strategy performs the best in handling these engineering problems. The significance is high in handling pressure vessel, welded beam, and spring design problems. But the significance is low in handling speed reducer problem. On the other hand, the acceptance strategy plays a less significant role in GSO in all engineering design problems.

There are several limitations in this study. First, this study employs only five acceptance strategies. Meanwhile, there are many unexplored conditional acceptance strategies. Second, there are only two swarm-metaheuristics chosen in this paper. Meanwhile, there are a lot of swarm-based metaheuristics that already exist and can be chosen as implementors. For example, future studies can be performed by changing the stringent acceptance strategy in WaOA [21], COA [25], and POA [26] with the loose or conditional acceptance strategy. Third, there are a lot of optimization problems, whether theoretical or practical ones, so that they cannot be accommodated in a single paper. There are other standard functions such as CEC series that can be used for additional evaluation. Moreover, there are a huge number of practical optimization problems where some of the problems are common, such as economic load dispatch problem (ELD) [41] and optimal power flow [42] which common in power system field, four engineering designs [38] which are common in mechanical field, and so on. Besides, there are also specific practical optimization problems that can be observed like vehicle routing problem [43], traveling salesman problem [44], vehicle scheduling [45], and so on.

Future studies can also be conducted by combining acceptance with expanded adaptability. In general, adaptability can be defined as the ability to act based on the current condition or trend. The third to fifth acceptance strategies represent this adaptability. But there are a lot of various adaptive mechanisms that can be chosen which are related to the acceptance strategy. For example, in this paper, the decision is taken based on the solution candidate compared to the current solution. It is better if the trend is also concerned. Another action may be taken if the progress is too slow although improvement still takes place.

Moreover, the observation can also be taken to the performance of the swarm contingent. In general, the acceptance strategy is taken based on the condition of only the related agent. It will be better if the decision is also taken based on the condition or performance of all agents, some agents, or other agents that are near the related agent. This approach can also be taken in future studies.

V. Conclusion

An investigation regarding various acceptance strategies in swarm-based metaheuristics has been performed in this paper. There are five acceptance strategies evaluated in this paper including the stringent acceptance strategy, loose acceptance strategy, and five conditional acceptance strategies. Two swarm-based metaheuristics which are total interaction algorithm and golden search optimization have been chosen as implementors. The 23 classic functions have been chosen as the unconstrained optimization problems. Meanwhile, the EED problem in Java-Bali power system has been chosen as the constrained optimization problem. The result shows that in general, the performance difference among the strategies is narrow i.e., not significant. This key finding highlights that the acceptance strategy is not the determinant factor in the performance of swarm-metaheuristics in handling 23 classic functions and EED problem. Meanwhile, the acceptance strategy plays a significant factor for TIA in handling pressure vessel, welded beam, and spring design problems. Meanwhile, the acceptance strategy is not significant for TIA in handling the speed reducer problem. On the other hand, the acceptance strategy is not a significant factor for GSO in handling all four engineering design problems.

There are four recommendations for future studies based on this work. First, future studies can be conducted by implementing these five acceptance strategies into more and various swarm-based metaheuristics. This track is important to provide a more comprehensive investigation regarding the relation between the acceptance to the performance of the swarm-based metaheuristics. Second, future studies can be conducted by investigating other factors that construct the metaheuristics in the relation with the performance of the metaheuristics. This study is important to find the significance of these factors to the performance of metaheuristics. Third, future studies can also be taken by combining acceptance with the expanded adaptive strategy. Fourth, investigating or developing acceptance strategy that considers the circumstances of not only the related agent but the swarm or some agents is also interesting.

REFERENCES

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

[2] M. A. Akbari, M. Zare, R. Azizipanah-abarghooee, S. Mirjalili, and M. Deriche, "The cheetah optimizer: a nature-inspired metaheuristic algorithm for large-scale optimization problems," *Sci Rep*, vol. 12, no. 1, p. 10953, Jun. 2022, doi: 10.1038/s41598-022-14338-z.

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

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

[5] X. Wang *et al.*, "Indoor Localization Method Enhanced by an Improved Harris's Hawk Optimization-Based Particle Filter," *IAENG Int J Comput Sci*, vol. 51, no. 3, pp. 267–275, 2024.

[6] R. A. Swief, N. M. Hassan, H. M. Hasanien, A. Y. Abdelaziz, and M. Z. Kamh, "Multi-Regional Optimal Power Flow Using Marine Predators Algorithm Considering Load and Generation Variability," *IEEE Access*, vol. 9, pp. 74600–74613, 2021, doi: 10.1109/ACCESS.2021.3081374.

[7] L. V. Ho *et al.*, "A hybrid computational intelligence approach for structural damage detection using marine predator algorithm and feedforward neural networks," *Comput Struct*, vol. 252, p. 106568, Aug. 2021, doi: 10.1016/j.compstruc.2021.106568.

[8] X. Sun, G. Wang, L. Xu, H. Yuan, and N. Yousefi, "Optimal performance of a combined heat-power system with a proton exchange membrane fuel cell using a developed marine predators algorithm," *J Clean Prod*, vol. 284, p. 124776, Feb. 2021, doi: 10.1016/j.jclepro.2020.124776.

[9] D. Yousri, A. Fathy, and H. Rezk, "A new comprehensive learning marine predator algorithm for extracting the optimal parameters of supercapacitor model," *J Energy Storage*, vol. 42, p. 103035, Oct. 2021, doi: 10.1016/j.est.2021.103035.

[10] G. Christopher and A. Wicaksana, "Particle swarm optimization for solving thesis defense timetabling problem," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, no. 3, pp. 762–769, Jun. 2021, doi: 10.12928/telkomnika.v19i3.18792.

[11] A. G. Gad, "Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review," *Archives of Computational Methods in Engineering*, vol. 29, no. 5, pp. 2531–2561, Aug. 2022, doi: 10.1007/s11831-021-09694-4.

[12] S. I. Boushaki, B. H. Mahammed, O. Bendjeghaba, and M. Mosbah, "Big data clustering based on spark chaotic improved particle swarm optimization," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 34, no. 1, pp. 419–429, Apr. 2024, doi: 10.11591/ijeecs.v34.i1.pp419-429.

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

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

[15] Y. Siregar, T. S. Jaya Tambun, S. P. Panjaitan, K. Tanjung, and S. Yana, "Distribution network reconfiguration utilizing the particle swarm optimization algorithm and exhaustive search methods," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 2, pp. 821–831, Apr. 2024, doi: 10.11591/eei.v13i2.6206.

[16] N. Trong Le, T. Trieu Phung, Anh. Huy Quyen, B. L. Phung Nguyen, and A. Ngoc Nguyen, "A hybrid approach of artificial neural network-particle swarm optimization algorithm for optimal load shedding strategy," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 4, pp. 4253–4263, Aug. 2022, doi: 10.11591/ijece.v12i4.pp4253-4263.

[17] N. Aqilah Mahmud and N. Yenita Dahlan, "Optimum energy management strategy with enhanced time of use tariff for campus building using particle swarm optimization," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 28, no. 2, pp. 644–653, Nov. 2022, doi: 10.11591/ijeecs.v28.i2.pp644-653.

[18] 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.

[19] R. Mothkur and V. Budhihal Nagendrappa, "An optimal model for classification of lung cancer using grey wolf optimizer and deep hybrid learning," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 30, no. 1, pp. 406–413, Apr. 2023, doi: 10.11591/ijeecs.v30.i1.pp406-413.

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

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

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

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

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

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

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

[27] R. Ragunathan and B. Ramadoss, "Golden jackal optimization for economic load dispatch problems with complex constraints," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 2, pp. 781–793, Apr. 2024, doi: 10.11591/eei.v13i2.6572.

[28] H. Jia, H. Rao, C. Wen, and S. Mirjalili, "Crayfish optimization algorithm," *Artif Intell Rev*, vol. 56, no. S2, pp. 1919–1979, Nov. 2023, doi: 10.1007/s10462-023-10567-4.

[29] P. D. Kusuma and F. C. Hasibuan, "Swarm Magnetic Optimizer: A New Optimizer that Adopts Magnetic Behaviour," *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 4, pp. 264–275, Aug. 2023, doi: 10.22266/ijies2023.0831.22.

[30] 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.

[31] P. D. Kusuma, "Group Better-Worse Algorithm: A Superior Swarm-based Metaheuristic Embedded with Jump Search," *IAENG International Journal of Applied Mathematics*, vol. 54, no. 4, pp. 614–622, 2024.

[32] P. D. Kusuma and A. Novianty, "Total Interaction Algorithm: A Metaheuristic in which Each Agent Interacts with All Other Agents," *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 1, pp. 224–234, Feb. 2023, doi: 10.22266/ijies2023.0228.20.

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

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

[35] F. Zeidabadi, M. Dehghani, and O. Malik, "TIMBO: Three Influential Members Based Optimizer," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 5, pp. 121–128, Oct. 2021, doi: 10.22266/ijies2021.1031.12.

[36] F. Zeidabadi, M. Dehghani, and O. P. Malik, "RSLBO: Random Selected Leader Based Optimizer," *International Journal of*

Intelligent Engineering and Systems, vol. 14, no. 5, pp. 529–538, Oct. 2021, doi: 10.22266/ijies2021.1031.46.
[37] P. D. Kusuma and A. Novianty, “Multiple Interaction Optimizer: A Novel Metaheuristic and Its Application to Solve Order Allocation Problem,” *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 2, pp. 440–453, Feb. 2023, doi: 10.22266/ijies2023.0430.35.
[38] M. Dehghani and P. Trojovský, “Osprey optimization algorithm: A new bio-inspired metaheuristic algorithm for solving engineering optimization problems,” *Front Mech Eng*, vol. 8, p. 1126450, Jan. 2023, doi: 10.3389/fmech.2022.1126450.
[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] 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.
[41] 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.
[42] 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.
[43] Y. Hou, C. Wang, C. Zhang, L. Dang, and C. Xiao, “A Hybrid Max-Min Ant System Algorithm for Electric Capacitated Vehicle Routing Problem,” *IAENG Int J Comput Sci*, vol. 51, no. 3, pp. 195–203, 2024.
[44] Z. Xu and X. Zhang, “An Improve Grey Wolf Optimizer Algorithm for Traveling Salesman Problems,” *IAENG Int J Comput Sci*, vol. 51, no. 6, pp. 602–612, 2024.
[45] S. Tian, C. Dai, X. Wang, R. Wu, K. Bi, and T. Wang, “Robust Optimization of Vehicle Scheduling for the Secondary Distribution of Refined Oil Products Under an Active Distribution Model,” *IAENG International Journal of Applied Mathematics*, vol. 54, no. 8, pp. 1560–1573, 2024.

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