Group Better-Worse Algorithm: A Superior Swarm-based Metaheuristic Embedded with Jump Search

Purba Daru Kusuma, *Member IAENG*

Abstract—**In recent years, there is massive development of new metaheuristics as stochastic methods. Meanwhile, there is not any metaheuristics is powerful to handle all problems as stated in the no-free-lunch (NFL) theory. Based on this circumstance, this paper introduces a new swarm-based metaheuristics with the main strategy moving toward the resultant of better swarm members and avoiding the resultant of worse swarm members called group better-worse algorithm (GBWA). It consists of five searches: moving toward the best swarm member, moving toward the resultant of better swarm members, moving away from the resultant of worse swarm members, searching locally, and jumping to the opposite area. GBWA is then evaluated in three ways. The first evaluation is a comparative evaluation where GBWA is compared to five recent metaheuristics: coati optimization algorithm (COA), average and subtraction-based optimization (ASBO), clouded leopard optimization (CLO), total interaction algorithm (TIA), and osprey optimization algorithm (OOA). The second evaluation is the individual search evaluation. The third evaluation is hyperparameter test. The collection of 23 classic functions is chosen as the use case in all evaluations. The result of the first evaluation shows that GBWA is better than COA, ASBO, CLO, TIA, and OOA in 20, 21, 20, 21, and 21 functions consecutively. Meanwhile, the result of the second evaluation shows the equal contribution between the motion toward the best swarm member and the motion toward the resultant of better swarm members.**

Index Terms—**stochastic optimization, metaheuristic, swarm intelligence, neighborhood search, jump search.**

I. Introduction

SWARM intelligence is a popular technique that has been used in many popular metaheuristics. Particle swarm optimization (PSO) is the early popular metaheuristic that uses swarm intelligence concept. Some popular metaheuristics, such as grey wolf optimization (GWO), slime mold algorithm (SMA), and marine predator algorithm (MPA) also use swarm intelligence. These three metaheuristics have been used extensively in many optimizations. GWO has been combined with support vector regression (SVR) to provide more accurate strip thickness prediction for raw materials [1]. PSO has been utilized to provide scheduling scheme for power system [2]. In this work, the modification of PSO was proven in reducing fuel cost, emission, and power loss [2]. PSO also has been used in the secure distributed computation in the cloud system [3]. In the operation research area, PSO has been utilized to solve vehicle routing problems for fresh product distribution with the soft time window and multi compartment issue [4]. In this study, the objective was minimizing the cost including penalty cost, damage cost, refrigeration cost, delivery cost, and vehicle cost [4]. MPA has been combined with support vector machine (SVM) to classify Alzheimer disease [5].

In the last three years, many new swarm-based metaheuristics are introduced. Some of them adopted animal behavior as inspiration, such as Komodo mlipir algorithm (KMA) [6], coati optimization algorithm (COA) [7], zebra optimization algorithm (ZOA) [8], osprey optimization algorithm (OOA) [9], golden jackal optimization (GJO) [10], clouded leopard optimization (CLO) [11], northern goshawk optimization (NGO) [12], cat and mouse based optimization (CMBO) [13], Siberian tiger optimization (STO) [14], walrus optimization algorithm (WaOA) [15], white shark optimization (WSO) [16], Tasmanian devil optimization (TDO) [17], snake optimization (SO) [18], green anaconda optimization (GAO) [19], red fox optimization (RFO) [20], and so on. Some metaheuristics adopted social or human behavior, such as modified social force algorithm (MSFA) [21], mother optimization algorithm (MOA) [22], migration algorithm (MA) [23], chef-based optimization algorithm (CBOA) [24], election-based optimization algorithm (EBOA) [25], and so on. On the other hand, some metaheuristics did not use any metaphors such as total interaction algorithm (TIA) [26], attack leave optimization (ALO) [27], multiple interaction-dual leader optimization (MIDLO) [28], average and subtraction-based optimization (ASBO) [29], golden search optimization (GSO) [30], and so on.

The best swarm member becomes the most popular reference in the directed search. This reference is used extensively in many swarm-based metaheuristics, such as ASBO [29], COA [7], ZOA [8], ALO [27], and so on. Besides the best swarm member, a randomly picked swarm member also becomes the popular reference as it is used in various metaheuristics, such as NGO [12], ZOA [8], COA [7], TIA [26], and so on. Meanwhile, some other metaheuristics uses a randomly picked better swarm members as its reference.

The neighborhood search with the reduced local search space during the iteration becomes a popular secondary search in the recent swarm-based metaheuristics. This

Manuscript received November 9, 2023; revised February 17, 2024. This work was financially supported by Telkom University, Indonesia.

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

strategy can be found in many recent metaheuristics, such as CLO [11], NGO [12], OOA [9], COA [7], ZOA [8], and so on. In the beginning, the local search space is large so that it looks like exploration. Then, the local search space declines as iteration goes on which represents the strategic shifting to exploitation.

Despite the popularity of various references in the directed search, there are opportunities to invent various other references. For example, the resultant of three best swarm members used in GWO [31], the two best swarm members in GJO [10], or the resultant of certain number of best swarm members in KMA [6] can be modified becomes the resultant of all better members in the perspective of the corresponding swarm member. On the other hand, avoiding a randomly picked swarm member whose quality is worse can be modified becomes avoiding the resultant of all worse swarm members, once again from the perspective of the corresponding swarm member.

The popularity of neighborhood search with reduced local search space during the iteration can also be challenged with other random searches. In general, the neighborhood search is an exploitation strategy. This approach can be challenged with a random search that focuses on the exploration strategy. Due to this context, rather than trying to find a better solution near the current solution, this random search tries to find in the remote area and avoids searching near the current solution.

Based on this opportunity, this work is aimed at introducing a new swarm-based metaheuristic called group better-worse algorithm (GBWA). As the name suggests, GBWA is motivated by the motion toward the group of better swarm members and avoid the group of worse swarm members. GBWA is also equipped with the jump search to improve the exploration capability. Meanwhile, GBWA is also enriched with the motion toward the best swarm member and the neighborhood search with reduced search space during the iteration.

The main and scientific contributions of this work are briefed as follows.

1) This work introduces the motion toward the resultant of the better solutions and the motion avoiding the resultant of the worse solutions in the swarm-based metaheuristic development.
2) This work introduces the jump search as exploration activity in the development of metaheuristics.
3) This work presents a comparative evaluation to investigate the performance of GBWA compared to most recent metaheuristics.
4) This work presents an individual search evaluation to investigate the performance of each search constructing the GBWA.
5) This work presents the hyper parameter test to evaluate the impact of the adjusted parameters to the performance of GBWA.

The formulation of this paper is as follows. Section 1 is the introduction consisting of the background of this work which is followed by the problem statement, research objective, and the contribution. Then, section 2 conducts the in-depth review of the development of recent metaheuristics, including the comparison among some recent metaheuristics. Section 3 provides the fundamental concept of GBWA which is followed by the formalization of the algorithm using the pseudocode and mathematical formulation. The evaluation of GBWA is presented in section 4 consisting of the evaluation scenario and the result. The comprehensive analysis regarding the evaluation result, the drawback to the theory, limitations, and the computational complexity of GBWA is presented in section 5. In the end, the conclusion and the ground for future development is summarized in section 6.

II. Related Works

In recent years, the development of metaheuristics has been dominated by the swarm intelligence. Using swarm intelligence approach, the metaheuristic is constructed by a certain number of agents that work autonomously to find the optimal solution. Due to this autonomy, the central coordination does not exist in this system. This coordination is replaced by the common or collective intelligence shared among the agents. This intelligence is manifested by the location of certain solutions including the quality of these solutions. Through this collective intelligence, each swarm member can take better action depends on the strategy implemented in the corresponding metaheuristic.

Each metaheuristic has its own collective intelligence which is used as the reference for the directed motion conducted by each swarm member. Each swarm-based metaheuristic may use single reference or multiple references. This reference can be the best swarm member, global best solution, local best solution, a randomly picked swarm member, the mixture or resultant of certain swarm members, the worst swarm member, and so on. The reference can also be the combination among the references mentioned previously. The direction of the motion can be getting closer or avoiding the reference. In general, the step size of this motion is stochastic. Several common distributions are uniform, normal, sinusoid, and so on.

This directed search implemented in the swarm-based intelligence makes the distinction between exploration and exploitation ambiguous. The basic definition of exploitation is searching for improvement near the current solution. On the other hand, the basic definition of exploration is searching somewhere within the search space to avoid local optimal. In many metaheuristics, the exploration and exploitation are conducted by different strategies. For example, in genetic algorithm (GA), the crossover represents exploitation while mutation represents exploration [32]. The simulated annealing (SA) is a metaheuristic that focuses on the exploitation. In swarm-based metaheuristic, this directed search can be seen as exploration or exploitation depends on the distance between the swarm member and its reference. When the swarm member is close to its reference, the motion can be perceived as exploitation. On the other hand, when the swarm member is far from its reference, the motion can be perceived as exploration.

TABLE I
COMPARISON OF SOME RECENT SWARM-BASED METAHEURISTICS

No	Metaheuristics	Strategy	Acceptance
1	COA [7]	The swarm member moves toward the best swarm member. The swarm member moves toward a randomized location within space. The swarm member takes neighborhood search with reduced local space.	strict
2	ASBO [29]	The swarm member moves toward the middle location between the best and worst swarm members. The swarm member moves with the direction of the gap between the best and worst swarm members. The swarm member moves away from the best swarm member.	strict
3	CLO [11]	The swarm member moves toward or away from a randomly picked other swarm member based on the quality comparison. The swarm member takes neighborhood search with reduced local space.	strict
4	TIA [26]	The swarm member moves toward or away from all other swarm members.	strict
5	OOA [9]	The swarm member moves toward a randomly picked member from a pool consisting of all the better swarm members plus the best swarm member. The swarm member takes neighborhood search with reduced local space.	strict
6	GSO [30]	The swarm member moves toward the portion of global best member and the portion of local best member through sinusoid distribution. The worst swarm member is replaced with a randomly picked swarm member.	loose
7	NGO [12]	The swarm member moves toward or away from a randomly picked other swarm member based on the quality comparison. The swarm member takes neighborhood search with reduced local space. The local space is already narrow in the beginning of the iteration.	strict
8	GJO [12]	The first best and second-best swarm members move toward the swarm member. The first best and second-best swarm members move away the swarm member.	loose
9	STO [14]	The swarm member moves toward a randomly picked member from a pool consisting of all the better swarm members plus the best swarm member. The swarm member moves toward or away from a randomly picked other swarm member based on the quality comparison. The swarm member takes neighborhood search with reduced local space.	strict
10	WSO [16]	The swarm member moves toward the best swarm member. The swarm member performs local search.	loose
11	this work	The swarm member moves toward the best swarm member. The swarm member moves toward the resultant of the group of the better swarm members. The swarm member moves away from resultant of the group of worse swarm members. The swarm member performs the neighborhood search with reduced local search space. The swarm member jumps to the opposite area within space.	strict

As iterative-based optimization, all metaheuristics consist of two stages. The first stage is initialization while the second stage is iteration. In almost all population-based metaheuristics, especially the swarm-based metaheuristics, all swarm members are generated uniformly within the search space. This circumstance makes the tendency of exploration high in the early iteration. Meanwhile, when the iteration is close to the maximum iteration, in general, all swarm members converge in a certain small area. This circumstance makes the tendency of exploitation high in the end of iteration.

The summary of recent swarm-based metaheuristics is provided in Table 1. This presentation includes the strategy conducted in every metaheuristic and the acceptance approach regarding the seed produced in every search.

The presentation in Table 1 exhibits that there are three common searches in the recent swarm-based metaheuristics. The first search is the motion toward the best swarm member. The second search is the motion toward or away from a randomly picked swarm member. The third search is the neighborhood search with reduced local search space. Based on this explanation, the first and third searches are accommodated in the proposed GBWA.

Moreover, the presentation in Table 1 exposes the distinct searches performed in GBWA. The first distinct search is the motion toward the resultant of the group of better swarm members. The second distinct search is the motion away from the resultant of the worse swarm members. The third distinct search is the jump search to the opposite side of the space. These three searches also become the novel strategy proposed in this work regarding the continuous development of metaheuristics.

III. PROPOSED MODEL

GBWA is constructed based on the swarm intelligence concept. It means that GBWA consists of a set of autonomous agents called swarm. The fundamental concept is that each swarm member tries to move toward the group of better swarm members and avoids the group of worse better members. These two groups are based on the perspective of the corresponding swarm member. It means that the members of the better swarm members will be different between one swarm member and the others. This circumstance is also applied for the group of worse swarm members. As the corresponding swarm member will move once for each group, then the swarm member will move to the central point of the group. This central point is determined based on the resultant or average location of the group without considering the quality of each swarm members within the group.

GBWA is also enriched with other searches besides moving toward the resultant of better swarm members and away from the resultant of the worse swarm members. The first enrichment is the motion toward the best swarm member. This search is adopted due to its commonality in many recent swarm-based metaheuristics. The second enrichment is the jump search to the opposite area within the search space. This search is adopted because it is a new concept in the studies regarding the development of metaheuristics. The third enrichment is the neighborhood search with the reduction of local search space during the iteration. This search is adopted because it is commonly found in many recent metaheuristics.

These five searches are accommodated into four sequential steps. The first step is the motion toward the best swarm member. The second step is the motion toward the resultant of better swarm members. The third step is the motion away from the worse swarm members. The fourth step is the jump search or neighborhood search. The determination of choosing between the jump search or neighborhood search is conducted stochastically. In the early iteration, the probability of performing jump search is high. Then, this probability declines as the iteration goes on. On the other hand, the probability of performing neighborhood search grows as the iteration goes on. The motivation of this approach in the fourth step is to force the swarm members juggle on both sides of the search space that represents exploration. Then, the swarm member will focus on the local search with narrow search space in the end of the iteration to avoid being thrown away from the current location.

The strict acceptance approach is implemented in GBWA. It means that the seed generated in every search can replace its parent (current solution) only if the seed is better than the parent. Moreover, the best swarm member will be replaced with the corresponding swarm member only if the corresponding swarm member is better than the current best swarm member.

The formalization of GBWA is presented in algorithm 1. The mathematical formulation is presented in (1) to (11). Meanwhile, below are the annotations used in this formalization.

d	dimension
e	seed
f	objective function
i	index for swarm member
j	index for dimension
s	swarm member
S	swarm
s_l	lower boundary of space
s_u	upper boundary of space
s_m	middle of the space
s_b	the best swarm member
S_{be}	group of better swarm members
S_{wo}	group of worse swarm members
s_{rb}	resultant of the group of better swarm members
s_{rw}	resultant of the group of worse swarm members
t	iteration
t_m	maximum iteration
U_{re1}	real uniform random number between 0 and 1
U_{re2}	real uniform random number between -1 and 1
U_{in1}	integer uniform random number between 1 and 2

algorithm 1: group better-worse algorithm

```
1   begin
2    for i=1: n(S)
3     initialization of s using (1)
4     update s_b using (2)
5    end for
6    for t=1: t_m
7     for i=1: n(S)
8      first motion using (3)
9      set up better and worse groups using (4) to (7)
10     second motion using (8)
11     third motion using (9)
12     if U_re1 < (t/t_m) then
13      fourth motion using (10)
14     else
15      fifth motion using (11)
16     update s_b using (2)
17    end for
18   end for
19   return s_b
20  end
```

The initialization step appears from lines 2 to 5 in algorithm 1. In this initialization, all swarm members are generated uniformly within the search space so that the probability of each location within the space becomes the initial solution is uniform or equal. This process is formalized using (1). Then, each time a swarm member is initialized, the best swarm member is updated based on the strict acceptance regulation as presented in (2).

$$s_{i,j} = s_{l,j} + U_{re1}.(s_{u,j} - s_{l,j}) \tag{1}$$

$$s_b' = \begin{cases} s_i, f(s_i) < f(s_b) \\ s_b, else \end{cases} \tag{2}$$

The iteration stage appears from lines 6 to 18 in algorithm 1. The first motion is the motion toward the best swarm member. This process is formalized using (3).

$$e_{1,j} = s_{i,j} + U_{re1}.\left(s_{b,j} - U_{in1}.s_{i,j}\right) \tag{3}$$

The grouping process is performed before the second and third motions. The grouping of the better swarm members is formalized using (4). Meanwhile, the grouping of the worse swarm members is formalized using (5). Then the resultant of the better swarm members is formalized using (6) while the resultant of the worse swarm members is formalized using (7).

$$S_{be,i} = \{s \in S | f(s) < f(s_i)\} \tag{4}$$

$$S_{wo,i} = \{s \in S | f(s) > f(s_i)\} \tag{5}$$

$$s_{rb,i,j} = \frac{\sum_{n(S_{be,i,j})} s_{be,i,j}}{n(S_{be,i,j})} \tag{6}$$

$$s_{wo,i,j} = \frac{\sum_{n(S_{wo,i,j})} s_{wo,i,j}}{n(S_{wo,i,j})} \tag{7}$$

After these two resultants are determined, the next processes are performing the second motion and the third motion. The second motion is formalized using (8) while the third motion is formalized using (9).

$$e_{2,j} = s_{i,j} + U_{re1}.\left(s_{rb,i,j} - U_{in1}.s_{i,j}\right) \tag{8}$$

$$e_{3,j} = s_{i,j} + U_{re1}(0,1).\left(s_{i,j} - U_{in1}.swo_{,i,j}\right) \tag{9}$$

The last search is the neighborhood search or the jump search. The neighborhood search is formalized using (10). Meanwhile, the jump search is formalized using (11).

$$e_{4,j} = s_{i,j} + U_{re2}\left(\frac{s_{l,j}}{t} + U_{re1}.\frac{s_{u,j} - s_{l,j}}{t}\right) \tag{10}$$

$$e_{4,j} = \begin{cases} s_{m,j} + U_{re1}.\left(s_{u,j} - s_{m,j}\right), s_{i,j} < s_{m,j} \\ s_{l,j} + U_{re1}.\left(s_{m,j} - s_{l,j}\right), else \end{cases} \tag{11}$$

IV. Simulation

There are three evaluations provided in this paper to evaluate the performance of GBWA. The first evaluation is called a comparative evaluation. In this paper, GBWA is compared with five new swarm-based metaheuristics: COA, ASBO, CLO, TIA, and OOA. Three of these metaheuristics (COA, TIA, and OOA) were first introduced in 2023. On the other hand, the two others (ASBO and CLO) were first introduced in 2022. There are several reasons why these five metaheuristics are chosen as comparison. First, there are a lot of new metaheuristics introduced in the last three years. This circumstance pushes the introduction of any new metaheuristics should be compared with any existing metaheuristic as new as possible. Second, all these comparators are swarm-based metaheuristics and deploy strict acceptance rule. Meanwhile, except TIA, the comparators implement multi-strategy and multi-stage approaches. COA, CLO, and OOA are comparators that implement neighborhood search. Third, all these comparators do not contain any adjusted parameters except the maximum iteration and swarm size. It means that the performance of these comparators will always be in its default status.

The second evaluation is called an individual search evaluation. As GBWA is a multi-strategy metaheuristic, it is important to assess the contribution of each search. The objective of this evaluation is to find the contribution or dominance of each search compared to the other search. Moreover, as stated in NFL theory, it is also important to assess the performance of each search to overcome various problems. Moreover, the strength and weakness of each search can be investigated.

The third evaluation is called hyperparameter test. This test is conducted to assess the impact of the adjusted parameters on the performance of GBWA. There are two adjusted parameters assessed in this paper: the maximum iteration and the swarm size.

The set of 23 classic functions is used as the theoretical use case in both evaluations. It consists of seven high dimension unimodal (HDU) functions, six high dimension multimodal (HDM) functions, and fixed dimension multimodal (FDM) functions. This use case is popular so that it was used in the first introduction of many new metaheuristics, such as KMA [6], TIA [26], ALO [27], and so on. Its popularity comes from the variety of circumstances it covers. It consists of unimodal functions that have only one optimal solution. On the other hand, some functions are multimodal functions that have multiple optimal solutions so that it is challenging to avoid the local optimal entrapment. There is variety in the search space from the narrow ones to the large ones. The shape or terrain is also various, from smooth, ripple, to a flat shape with narrow steep hole.

In this evaluation, the swarm size is set to 5 while the maximum iteration is set to 10. The dimensions for the high dimension functions are set to 60. Tables 2 to 4 exhibit the detailed result of the first evaluation, including the average fitness score, standard deviation, and the mean rank. Then, this result is summarized based on the functions class and provided in Table 5. The result of the second evaluation is provided in Table 6. The value more precise than 10^{-4} is rounded to 0.

The result in Table 2 indicates the superiority of GBWA compared to its confronters in the first group of functions. GBWA becomes the first best in solving all seven functions (Sphere, Schwefel 2.22, Schwefel 1.2, Schwefel 2.21, Rosenbrock, Step, and Quartic). For additional note, all metaheuristics in this evaluation achieve the same result in solving Schwefel 2.22. The wide performance gap between the best and worst performers can be seen in five functions (Sphere, Schwefel 1.2, Schwefel 2.21, Rosenbrock, and Step). Meanwhile this gap is narrow in solving Quartic.

The result in Table 3 indicates that GBWA is still superior to its comparators in solving the second group of functions. GBWA becomes the best performer in all six functions (Schwefel, Rastrigin, Ackley, Griewank, Penalized, and Penalized 2). The performance gap between the best and worst performer is wide in five functions where GBWA becomes the best performer (Rastrigin, Ackley, Griewank, Penalized, and Penalized 2). Meanwhile, the performance gap in solving Schwefel is narrow.

Table 4 indicates that GBWA is competitive compared to its comparators in solving the third group of functions. GBWA becomes the best performer in solving nine functions (Shekel Foxholes, Kowalik, Six Hump Camel, Branin, Hartman 3, Hartman 6, Shekel 5, Shekel 7, and Shekel 10). GBWA becomes the third best in solving Goldstein-Price behind COA and CLO. Meanwhile, all metaheuristics in this evaluation performs equally in solving Hartman 3.

TABLE II
PERFORMANCE EVALUATION REGARDING SEVEN HDU FUNCTIONS

F	Parameter	COA [7]	ASBO [29]	CLO [11]	TIA [26]	OOA [9]	GBWA
1	mean	$1.1701x10^3$	$9.1475x10^2$	$2.1589x10^3$	$5.1486x10^1$	$2.9559x10^2$	1.2624
	std deviation	$4.6962x10^2$	$3.1434x10^2$	$1.4090x10^3$	$1.1224x10^1$	$1.1659x10^2$	2.0914
	mean rank	5	4	6	2	3	1
2	mean	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	std deviation	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	mean rank	1	1	1	1	1	1
3	mean	$5.3145x10^4$	$6.8943x10^4$	$1.0325x10^5$	$1.1664x10^4$	$3.3104x10^4$	$4.3104x10^3$
	std deviation	$4.1827x10^4$	$3.4362x10^4$	$3.8338x10^4$	$7.8975x10^3$	$1.7512x10^4$	$5.3334x10^3$
	mean rank	4	5	6	2	3	1
4	mean	$2.4593x10^1$	$2.3075x10^1$	$5.0583x10^1$	4.6889	$1.3862x10^1$	0.9801
	std deviation	5.0695	$1.6315x10^1$	$1.5177x10^1$	0.7244	4.6121	0.5262
	mean rank	5	4	6	2	3	1
5	mean	$1.9685x10^5$	$6.6069x10^4$	$9.5053x10^5$	$1.0713x10^3$	$1.7701x10^4$	$6.9518x10^1$
	std deviation	$1.6510x10^5$	$3.2273x10^4$	$8.0527x10^5$	$3.9091x10^2$	$1.3345x10^4$	9.2275
	mean rank	5	4	6	2	3	1
6	mean	$1.0882x10^3$	$8.4527x10^2$	$2.2405x10^3$	$5.5244x10^1$	$3.0685x10^2$	$1.3627x10^1$
	std deviation	$4.0103x10^2$	$3.4945x10^2$	$1.2400x10^3$	$1.0442x10^1$	$1.3415x10^2$	0.9724
	mean rank	5	4	6	2	3	1
7	mean	0.5658	0.3067	0.9324	0.1154	0.1917	0.0377
	std deviation	0.2683	0.1356	0.6095	0.0599	0.1053	0.0164
	mean rank	5	4	6	2	3	1

TABLE III
PERFORMANCE EVALUATION REGARDING SIX HDM FUNCTIONS

F	Parameter	COA	ASBO	CLO	TIA	OOA	GBWA
8	mean	-4.1944×10^3	-3.7869×10^3	-4.0912×10^3	-2.1995×10^3	-3.4490×10^3	-4.3019×10^3
	std deviation	6.6039×10^2	5.6699×10^2	6.8842×10^2	5.1639×10^2	7.2869×10^2	7.1444×10^2
	mean rank	2	4	3	6	5	1
9	mean	1.8095×10^2	4.1047×10^1	4.0321×10^2	8.4084×10^1	1.6430×10^2	4.8890
	std deviation	4.1437×10^1	1.0241×10^1	6.0488×10^1	2.5798×10^1	5.0772×10^1	6.3793
	mean rank	5	2	6	3	4	1
10	mean	6.4208	7.0680	8.0799	2.3669	4.0514	0.1634
	std deviation	1.2437	1.6667	1.1876	0.2709	0.6600	0.0991
	mean rank	4	5	6	2	3	1
11	mean	1.0078×10^1	9.6762	1.6941×10^1	1.4357	3.7119	0.2529
	std deviation	3.6954	2.7918	5.7041	0.1610	1.2336	0.2251
	mean rank	5	4	6	2	3	1
12	mean	1.6864×10^2	1.4487×10^1	1.9487×10^4	1.3326	4.0603	1.0396
	std deviation	7.2867×10^2	2.8425×10^1	5.0775×10^4	0.2593	1.8224	0.0910
	mean rank	5	4	6	2	3	1
13	mean	2.5787×10^4	2.1288×10^4	4.7040×10^5	5.3925	1.2007×10^2	3.5035
	std deviation	2.7943×10^4	4.3858×10^4	7.06831×10^5	1.1211	3.1206×10^2	0.1712
	mean rank	5	4	6	2	3	1

TABLE IV
PERFORMANCE EVALUATION REGARDING TEN FDM FUNCTIONS

F	Parameter	COA [7]	ASBO [29]	CLO [11]	TIA [26]	OOA [9]	GBWA
14	mean	9.8117	7.7597	8.5985	1.7542×10^1	9.8587	6.9324
	std deviation	5.6911	4.3037	4.1532	2.7773×10^1	5.2495	4.1514
	mean rank	4	2	3	6	5	1
15	mean	0.0146	0.1114	0.0216	0.0109	0.0158	0.0054
	std deviation	0.0187	0.0369	0.0158	0.0143	0.0166	0.0105
	mean rank	3	6	5	2	4	1
16	mean	-1.0103	-0.0545	-1.0162	-1.0086	-1.0065	-1.0287
	std deviation	0.0235	0.1927	0.0159	0.0427	0.0272	0.0060
	mean rank	3	6	2	4	5	1
17	mean	0.5203	1.5819	0.6105	1.6286	0.4661	0.4347
	std deviation	0.3669	2.3113	0.6234	2.2443	0.15932	0.0971
	mean rank	3	5	4	6	2	1
18	mean	7.6897	1.5500×10^1	7.8656	2.4145×10^1	1.3955×10^1	1.2998×10^1
	std deviation	8.3094	5.7282×10^1	9.5166	2.4549×10^1	2.2565×10^1	1.3142×10^1
	mean rank	1	5	2	6	4	3
19	mean	-0.0495	-0.0495	-0.0495	-0.0495	-0.0495	-0.0495
	std deviation	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	mean rank	1	1	1	1	1	1
20	mean	-2.8134	-0.8035	-2.7745	-1.9985	-2.7845	-2.8728
	std deviation	0.1815	0.4663	0.3036	0.4795	0.2421	0.2293
	mean rank	2	6	4	5	3	1
21	mean	-2.2640	-1.8789	-1.9033	-1.5961	-1.6516	-3.1369
	std deviation	1.2584	1.7740	0.8282	1.1761	1.0794	1.7915
	mean rank	2	4	3	6	5	1
22	mean	-1.9182	-2.1539	-1.8842	-1.8682	-2.0006	-3.7708
	std deviation	0.9708	1.6965	0.6596	0.8550	1.0165	1.7586
	mean rank	4	2	5	6	3	1
23	mean	-2.4329	-2.4678	-2.1912	-1.8718	-1.9591	-2.8361
	std deviation	1.0192	2.1375	0.7356	0.9275	0.7060	1.1751
	mean rank	3	2	4	6	5	1

TABLE V
CLUSTER BASED COMPARISON RESULT OF GBWA

Cluster	Number of Functions Beaten by GBWA				
	COA [7]	ASBO [29]	CLO [11]	TIA [26]	OOA [9]
1	6	6	6	6	6
2	6	6	6	6	6
3	8	9	8	9	9
Total	20	21	20	21	21

Table 5 shows the superiority of GBWA compared to its comparators. This superiority is based on the number of functions where GBWA is better than the related comparator. In Table 5, the superiority is grouped based on the group of functions so that there are three groups for each comparator.

Table 5 indicates that in general, GBWA is still superior to its comparators. GBWA is better than COA, ASBO, CLO, TIA, and OOA in 20, 21, 20, 21, and 21 functions consecutively. Overall, GBWA is superior in solving all three groups of functions. The superiority of GBWA compared to its comparators is followed by a significant performance gap in the first and second groups of functions. Meanwhile a close performance gap takes place in the third group of functions although GBWA is still superior.

The result of the second evaluation is presented in Table 6. As there are four searches in GBWA, then there are four individual searches evaluated in this work. In this second evaluation the parameter is the average fitness score. The best result for each function is written in bold font.

TABLE VI
SINGLE SEARCH ASSESSMENT RESULT

F	Average Fitness Score			
	1st search	2nd search	3rd search	4th search
1	1.7736×10^{2}	$\mathbf{1.0377 \times 10^{2}}$	1.7321×10^{5}	1.5048×10^{5}
2	**0.0000**	**0.0000**	1.0750×10^{89}	0.0000
3	2.2269×10^{4}	$\mathbf{1.8944 \times 10^{4}}$	9.7830×10^{5}	5.3092×10^{5}
4	$\mathbf{1.1916 \times 10^{1}}$	3.1211×10^{1}	9.7000×10^{1}	9.1906×10^{1}
5	5.0221×10^{3}	$\mathbf{4.3189 \times 10^{3}}$	7.8769×10^{8}	6.3055×10^{8}
6	1.9544×10^{2}	$\mathbf{1.0595 \times 10^{2}}$	1.7640×10^{5}	1.4319×10^{5}
7	**0.1429**	0.1450	7.9055×10^{2}	5.8870×10^{2}
8	-2.6620×10^{3}	-2.2227×10^{3}	-2.9418×10^{3}	$\mathbf{-3.7681 \times 10^{3}}$
9	$\mathbf{1.1796 \times 10^{2}}$	1.5064×10^{2}	1.0264×10^{3}	9.0765×10^{2}
10	3.4059	**2.9504**	2.0078×10^{1}	2.0781×10^{1}
11	**0.1719**	0.3014	1.8989×10^{3}	1.3566×10^{3}
12	2.4510	**2.1046**	1.9976×10^{9}	1.4164×10^{9}
13	8.7248	**7.4720**	4.0673×10^{9}	2.7336×10^{9}
14	$\mathbf{2.2107 \times 10^{1}}$	2.6632×10^{1}	2.1912×10^{2}	3.0855×10^{1}
15	0.0414	**0.0263**	1.2586	0.0809
16	**-0.8804**	-0.8154	1.1690×10^{2}	-0.0495
17	4.2884	3.9010	6.1218	**3.7443**
18	4.3698×10^{1}	7.5125×10^{1}	4.3272×10^{2}	$\mathbf{3.0364 \times 10^{1}}$
19	**-0.0495**	**-0.0495**	**-0.0495**	-0.0117
20	-2.0379	**-2.0410**	-1.1048	-1.7446
21	**-1.1765**	-1.0175	-0.3298	-0.7786
22	**-1.7324**	-1.4462	-0.4604	-0.9565
23	**-1.5400**	-1.2761	-0.6314	-1.3927

Table 6 shows the dominance of the first and second searches in GBWA. In Schwefel 2.22, the first and second can find the global optimal solution. Meanwhile, in Hartman 3, the first, second, and third searches achieve the same result. The first search is the distinct best of nine functions. The second search is the distinct best of nine functions. The fourth search is the distinct best of only three functions. Meanwhile, the third search never becomes the distinct best. This result shows the equal contribution of the first and second searches.

TABLE VII
SINGLE SEARCH ASSESSMENT RESULT

F	Average Fitness Score	
	$t_m = 20$	$t_m = 40$
1	0.0000	0.0000
2	0.0000	0.0000
3	9.5572×10^{1}	0.0036
4	0.0028	0.0000
5	5.8917×10^{1}	5.8919×10^{1}
6	1.3076×10^{1}	1.2947×10^{1}
7	0.0147	0.0056
8	-5.0217×10^{3}	-5.6280×10^{3}
9	0.0040	0.0000
10	0.0002	0.0000
11	0.0075	0.0003
12	0.9889	0.6881
13	3.1097	3.0889
14	6.5781	4.8327
15	0.0042	0.0053
16	-1.0313	-1.0316
17	0.4120	0.3981
18	1.0755×10^{1}	6.2401
19	-0.0495	-0.0495
20	-2.9981	-3.0324
21	-5.3959	-7.1506
22	-5.9673	-7.7913
23	-4.8683	-7.5229

There are two tests regarding the hyperparameter test. The first test is conducted to assess the impact of the increase of maximum iteration on the average fitness score. There are two values on the maximum iteration: 20 and 40. The result is presented in Table 7. Meanwhile, the second test is conducted to assess the impact of the increase of swarm size on the average fitness score. There are two values on swarm size: 10 and 20. The result is provided in Table 8.

Table 7 shows that there are only six functions where the average fitness score improves significantly due to the increase of maximum iteration from 20 to 40. These functions are Schwefel 1.2, Schwefel 2.21, Quartic, Rastrigin, Ackley, and Penalized 2. All these functions are high dimension functions where the first three functions are unimodal, and the rest are multimodal. Meanwhile, in some cases where there is not any improvement, the final solution has reached or near to the global optimal like in Sphere, Schwefel 2.22, Six Hump Camel, Branin, and so on.

TABLE VIII
SINGLE SEARCH ASSESSMENT RESULT

F	Average Fitness Score	
	$n(S) = 10$	$n(S) = 20$
1	0.0084	0.0006
2	0.0000	0.0000
3	5.6633×10^{2}	3.9174×10^{1}
4	0.1414	0.0302
5	5.9021×10^{1}	5.8896×10^{1}
6	1.2474×10^{1}	1.1701×10^{1}
7	0.0164	0.0061
8	-4.7970×10^{3}	-4.8957×10^{3}
9	0.0530	0.0020
10	0.0162	0.0039
11	0.0187	0.0013
12	0.9208	0.8509
13	3.2653	3.1789
14	6.8505	3.8001
15	0.0044	0.0024
16	-1.0313	-1.0316
17	0.3989	0.3982
18	5.3554	3.0032
19	-0.0495	-0.0495
20	-3.0014	-3.1010
21	-4.4230	-5.2858
22	-4.4545	-4.9468
23	-3.4306	-4.6758

Table 8 shows that there are only seven functions where the average fitness score improves significantly due to the increase of swarm size from 10 to 20. These functions are Sphere, Schwefel 1.2, Schwefel 2.21, Quartic, Rastrigin, Ackley, and Griewank. Like in the first hyperparameter test, all these seven functions are high dimension. Four of them are unimodal, and the rest of three are multimodal.

V. Discussion

The result of the first evaluation proofs that GBWA is a competitive metaheuristic that can perform well during the low-swarm size and low-maximum iteration scenario. In general, GBWA is superior in solving unimodal and multimodal functions. Moreover, GBWA is also superior in solving whether high dimension functions and fixed dimension functions. Its superiority in solving high dimension unimodal functions shows that GBWA has good exploitation capability while its superiority in solving high dimension multimodal functions shows that GBWA has good exploration capability. Meanwhile, the superiority of GBWA in solving the fixed dimension multimodal functions shows that GBWA has balanced exploration and exploitation capability.

The superiority of GBWA among its comparators shows that GBWA can be a breakthrough in the development of metaheuristics. All these metaheuristics are developed based on swarm intelligence. Except TIA [26], all of them deploy multiple search strategies. Moreover, all these metaheuristics implement a strict acceptance approach. This approach is successful in beating other metaheuristics without strict acceptance approach, such as GWO [31], MPA [33], SMA [34], GSO [30], and so on. But this approach can be evaluated or modified to overcome the recent metaheuristics that use this approach. The existence of neighborhood search with reduction of the search space during the iteration can also be evaluated.

The directed motion toward the best swarm member and the resultant of better swarm members contributes equally in GBWA. The contribution of these two searches is also higher than the third and fourth searches. This result strengthens the importance of the best swarm member and the resultant of better swarm members as the references in the directed search. Meanwhile, the result in the individual search evaluation also shows that moving toward the better place is more effective than avoiding the worse solution for the improvement effort.

The jump search to the opposite area within space is designed to improve the exploration capability of GBWA where this search is designed to avoid the swarm member search near its current location. Unfortunately, its contribution is least significant. It means that the modification of this jump search is needed to make this search more competitive. In GBWA, the jump search is performed in the fourth search as an alternative to the neighborhood search. The probability of jump search is higher in the early iteration while the probability of neighborhood search is higher in the end of iteration. The modification can be performed in many ways, such as equal opportunity between two searches or making these two searches as dedicated searches. The modification can also be taken by performing the jump search as an alternative when the stagnation occurs.

The hyper parameter test shows that the increase of maximum iteration gives more significant impact rather than the increase of swarm size. This result can be observed by comparing Table 7 and Table 8. Meanwhile, significant improvement occurs mostly in high dimension functions, whether they are unimodal or multimodal due to the increase of swarm size or maximum iteration. The improvement in the fixed dimension multimodal functions is less significant. Meanwhile the significant improvement occurs only half of the high dimension functions.

The computation complexity of GBWA can be traced from the number of loops in its process. In this context, there are different number of loops performed during the initialization and iteration. In the initialization, there is a nested loop consisting of two loops. The outer loop is the loop for whole swarm members. The inner loop is the loop for whole dimension. Based on this explanation, the complexity of the initialization can be presented as $O(n(S).d)$. Meanwhile, in the iteration, there is a nested loop consisting of four loops. The sequence of the loops from outer to inner is as follows. The first loop is the loop until the maximum iteration. The next loop is the loop for whole swarm members in the context of performing the searches. Once again, the loop for whole swarm members is performed to find the better swarm members and worse swarm members where this loop is performed by each swarm member. The last loop is the loop for whole dimension. Meanwhile, there are four sequential searches performed by each swarm member in each iteration. Based on this explanation, the computational complexity in the iteration can be presented as $O(t_m.n(S).(4+n(S)).d)$.

There are limitations regarding this work or the proposed GBWA although this metaheuristic has shown the acceptable performance through superiority among the recent swarm-based metaheuristics. First, the contribution of the third and fourth searches are still minimal. It makes avoiding the resultant of the worse swarm members, as third search, should be evaluated or replaced with more powerful search. Meanwhile, the existence of the neighborhood search where the local search space declines through the iteration can also be evaluated. Second, GBWA is still inferior to other recent metaheuristics in the third group of functions although it is superior in the high dimension functions. This circumstance can be used as a new path for further improvement for the modification of GBWA so that it is superior in all groups of functions. Meanwhile, as stated in the NFL theory, superiority in all groups does not mean that the future metaheuristic will be superior in all functions. Third, there are five searches accommodated in GBWA. Meanwhile, there are a lot of other searches already exist in many recent metaheuristics. The improvement of GBWA can be used by adopting some of these searches or hybridizing the GBWA with ASBO as its strongest comparator. Fourth, many metaheuristics focus on the searching but do not give appropriate attention to the worst swarm member and stagnation. In the future, adding mechanisms to handle the worst swarm member or improving the exploration during the stagnation can also be chosen as alternative.

VI. Conclusion

This paper has presented a novel swarm-based metaheuristics with the main strategy is moving toward the resultant of the better swarm members and avoiding the resultant of the worse swarm members called group better-worse algorithm (GBWA). Besides, GBWA is also enriched with the motion toward the best swarm member, neighborhood search, and jumping to opposite location within the space. Through the comparative evaluation, GBWA is proven superior to its comparators with fierce competition. GBWA is better than COA, ASBO, CLO, TIA, and OOA in 20, 21, 20, 21, and 21 functions consecutively in solving the set of 23 classic functions. This superiority occurs in all groups of functions. Meanwhile, through the individual search evaluation, there is equal contribution between the motion toward the best swarm member and the motion toward the resultant of better swarm members. The contribution of the random search as fourth search is less significant.

This work, especially the GBWA, has opened several tracks for possible future studies. Inventing a new kind of search, whether it is based on the swarm intelligence or random search is still challenging due to the fierce competition in developing new metaheuristics. More use cases, especially the practical ones are important to evaluate various new metaheuristics more comprehensively. It is also challenging to hybridize the GBWA with various optimization methods from the deterministic ones to the stochastic ones.

REFERENCES

[1] L. Sun, J. Li, X. Xiao, L. Zhang, and J. Li, "Improved differential gray wolf algorithm optimized support vector regression strip thickness prediction method," *Engineering Letters*, vol. 29, no.4, pp. 1462-1469, 2021.

[2] J. Qian and G. Chen, "Improved multi-goal particle swarm optimization algorithm and multi-output BP network for optimal operation of power system," *IAENG International Journal of Applied Mathematics*, vol. 52, no.3, pp. 576-588, 2022.

[3] H. Miyajima, N. Shigei, H. Miyajima, and N. Shiratori, "Securely distributed computation with divided data and parameters for hybrid particle swarm optimization," *IAENG International Journal of Applied Mathematics*, vol. 52, no.3, pp. 541-549, 2022.

[4] 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.

[5] P. S. Topannavar and D. M. Yadav, "An effective feature selection using improved marine predators algorithm for Alzheimer's disease classification," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 5, pp. 5126-5134, 2023.

[6] S. Suyanto, A. A. Ariyanto, and A. F. Ariyanto, "Komodo mlipir algorithm," *Applied Soft Computing*, vol. 114, pp. 1–17, 2022.

[7] M. Dehghani, Z. Montazeri, E. Trojovská, and P. Trojovský, "Coati optimization algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems," *Knowledge-based Systems*, vol. 259, p. 110011, 2023.

[8] E. Trojovska, M. Dehghani, and P. Trojovsky, "Zebra optimization algorithm: a new bio-inspired optimization algorithm for solving optimization algorithm," *IEEE Access*, vol. 10, pp. 49445–49473, 2022.

[9] 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, p. 1126450, 2023.

[10] N. Chopra and M. Mohsin Ansari, "Golden jackal optimization: A novel nature-inspired optimizer for engineering applications," *Expert Systems with Applications*, vol. 198, p. 116924, 2022.

[11] E. Trojovska and M. Dehghani, "Clouded leopard optimization: a new nature-inspired optimization algorithm," *IEEE Access*, vol. 10, pp. 102876–102906, 2022.

[12] M. Dehghani, S. Hubalovsky, and P. Trojovsky, "Northern goshawk optimization: a new swarm-based algorithm for solving optimization problems," *IEEE Access*, vol. 9, pp. 162059–162080, 2021.

[13] M. Dehghani, Š. Hubálovský, and P. Trojovský, "Cat and mouse based optimizer: a new nature-inspired optimization algorithm," *Sensors*, vol. 21, no. 15, p. 5214, 2021.

[14] P. Trojovsky, M. Dehghani, and P. Hanus, "Siberian tiger optimization: a new bio-inspired metaheuristic algorithm for solving engineering optimization problems," *IEEE Access*, vol. 10, pp. 132396–132431, 2022.

[15] P. Trojovsky and M. Dehghani, "A new bio-inspired metaheuristic algorithm for solving optimization problems based on walruses behavior," *Science Reports*, vol. 13, no. 1, p. 8775, 2023.

[16] 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, p. 108457, 2022.

[17] M. Dehghani, S. Hubalovsky, and P. Trojovsky, "Tasmanian devil optimization: a new bio-inspired optimization algorithm for solving optimization algorithm," *IEEE Access*, vol. 10, pp. 19599–19620, 2022.

[18] F. A. Hashim and A. G. Hussien, "Snake optimizer: a novel meta-heuristic optimization algorithm," *Knowledge-based Systems*, vol. 242, p. 108320, 2022.

[19] M. Dehghani, P. Trojovsky, and O. P. Malik, "Green anaconda optimization: a new bio-inspired metaheuristic algorithm for solving optimization problems," *Biomimetics*, vol. 8, p. 121, 2023.

[20] D. Polap and M. Wozniak, "Red fox optimization algorithm," *Expert Systems with Applications*, vol. 166, p. 114107, 2021.

[21] P. D. Kusuma and D. Adiputra, "Modified social forces algorithm: from pedestrian dynamic to metaheuristic optimization," *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 3, pp. 294–303, 2022.

[22] I. Matoušová, P. Trojovský, M. Dehghani, E. Trojovska, and J. Kostra, "Mother optimization algorithm: a new human-based metaheuristic approach for solving engineering optimization," *Science Reports*, vol. 13, p. 10312, 2023.

[23] P. Trojovský and M. Dehghani, "Migration algorithm: a new human-based metaheuristic approach for solving optimization problems," *Computer Modeling in Engineering & Sciences*, vol. 137, no. 2, pp. 1695-1730, 2023.

[24] E. Trojovska and M. Dehghani, "A new human-based metaheuristic optimization method based on mimicking cooking training," *Scientific Reports*, vol. 12, p. 14861, 2022.

[25] P. Trojovský and M. Dehghani, "A new optimization algorithm based on mimicking the voting process for leader selection," *PeerJ Computer Science*, vol. 8, p. e976, 2022.

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

[27] P. D. Kusuma and F. C. Hasibuan, "Attack-leave optimizer: a new metaheuristic that focuses on the guided search and performs random search as alternative," *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 3, pp. 244–257, 2023.

[28] P. D. Kusuma and A. L. Prasasti, "Multiple interaction-dual leader optimizer: a novel metaheuristic to solve high dimension problem," *IAENG International Journal of Computer Science*, vol. 50, no.2, pp. 440-448, 2023.

[29] 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 Computer Science*, vol. 8, pp. 1–40, 2022.

[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.

[31] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer", *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[32] S. Katoch, S. S. Chauhan, and V. A. Kumar, "A review on genetic algorithm: past, present, and future," *Multimedia Tools and Applications*, vol. 80, pp. 8091–8126, 2021.

[33] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine predators algorithm: a nature-inspired metaheuristic", *Expert Systems with Applications*, vol. 152, p. 113377, 2020.

[34] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: a new method for stochastic optimization", *Future Generation Computer Systems*, vol. 111, pp. 300-323, 2020.

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