Adaptive Balance Optimizer: A New Adaptive Metaheuristic and its Application in Solving Optimization Problem in Finance

Purba Daru Kusuma, Ashri Dinimaharawati
Computer Engineering, Telkom University, Bandung, Indonesia

***Abstract*—Adaptability becomes important in developing metaheuristic algorithms, especially in tackling stagnation. Unfortunately, almost all metaheuristics are not equipped with an adaptive approach that makes them change their strategy when stagnation happens during iteration. Based on this consideration, a new metaheuristic, called an adaptive balance optimizer (ABO), is proposed in this paper. ABO's unique strategy focuses on exploitation when improvement happens and switching to exploration during stagnation. ABO also uses a balanced strategy between exploration and exploitation by performing two sequential searches, whatever circumstance it faces. These sequential searches consist of one guided search and one random search. Moreover, ABO also deploys both a strict acceptance approach and a non-strict acceptance approach. In this work, ABO is challenged to solve a set of 23 classic functions as a theoretical optimization problem and a portfolio optimization problem as the use case for the practical optimization problem. In portfolio optimization, ABO should optimize the quantity of ten stocks in the energy and mining sector listed in the IDX30 index. In this evaluation, ABO is competed with five other metaheuristics: marine predator algorithm (MPA), golden search optimizer (GSO), slime mold algorithm (SMA), northern goshawk optimizer (NGO), and zebra optimization algorithm (ZOA). The simulation result shows that ABO is better than MPA, GSO, SMA, NGO, and ZOA in solving 21, 18, 16, 11, and 8, respectively, in solving 23 functions. Meanwhile, ABO becomes the third-best performer in solving the portfolio optimization problem.**

Keywords—Optimization; metaheuristic; adaptability; portfolio optimization; IDX30

I. Introduction

Metaheuristics is a popular method used in various optimization problems. In the cloud system, the genetic algorithm (GA) was modified in service caching and task offloading to improve resource efficiency and user satisfaction [1]. A deep convolutional neural network is enriched with a gorilla troop optimizer (GTO) to improve its capability in diagnosing skin cancer [2]. The whale optimization algorithm (WOA) was used to solve portfolio optimization based on the FTSE100 index [3]. An improved sparrow search algorithm (ISSA) was developed to improve the high-intensity focused ultrasound (HIFU) technology that is used for tumor treatment [4]. A combination of tabu search (TS) and simulated annealing (SA) is used to solve the coupled task scheduling of the heterogeneous multiple automated guided vehicles (AGV) in a manufacturing system [5]. Its popularity comes from two reasons. The first reason is that there are a huge number of metaheuristics already existing today. The second reason is that metaheuristic uses a stochastic approach to be efficient enough in solving large-scale optimization problems with limited computational resources [6]. Moreover, metaheuristic is also flexible enough to solve various kinds of problems by abstracting the problem. It focuses on the objectives and constraints of these problems. Then, it uses a trial-and-error mechanism to improve the solution through iteration. Meanwhile, this stochastic approach means that all metaheuristics do not guarantee finding the optimal global solution but only the high-quality or quasi-optimal one [7]. Besides, metaheuristic is also challenged with the optimal local issue.

One important consideration in metaheuristics is adaptability. Adaptability is important, especially in facing the circumstance of whether the current search produces a better solution or not. Each metaheuristic was developed based on a strategy for finding a better solution. This new solution is then used for the next iteration for several purposes. In some metaheuristics, new solutions are used to choose the reference for the guided search. Some metaheuristics rank the solutions and then split them into several groups where each group performs its strategy. Some other metaheuristics rank the solutions to eliminate the worst solution or some worst solutions for the next iteration.

Unfortunately, almost all metaheuristics are not adaptive enough. Many metaheuristics do not care about the quality of the new solution relative to the previous solution. The new solution still replaces the current solution, although this new solution is not better than the existing solution. This mechanism can be found in many metaheuristics, such as grey wolf optimizer (GWO) [8], MPA [9], GSO [10], SMA [11], darts game optimizer (DGO) [12], butterfly optimization algorithm (BOA) [13], chameleon swarm algorithm (CSA) [14], tunicate swarm algorithm (TSA) [15], squirrel search optimizer (SSO) [16], coronavirus optimization algorithm (COVIDOA) [17], white shark optimizer (WSO) [18], stochastic paint optimizer (SPO) [19], and so on. In some metaheuristics, a strict-acceptance approach is applied. Through this approach, a new solution is accepted to replace the current solution only if it is better than the current one. This approach can be found in many latest metaheuristics, such as the pelican optimization algorithm (POA) [20], guided pelican algorithm (GPA) [21], total interaction algorithm (TIA) [22], three-on-three optimizer (TOTO) [23], average and subtraction-based optimizer (ASBO) [24], northern goshawk optimizer (NGO) [25], zebra optimization algorithm (ZOA) [26], coati optimization algorithm (COA) [27], fennec fox optimization (FFO) [28], chef-based optimization algorithm (CBOA) [29], modified honey badger algorithm (MHBA) [30], flower pollination algorithm (FPA) [31], football game based optimizer (FBGO) [32], red fox optimization algorithm (RFO) [33], and so on.

On the other hand, few metaheuristics are adaptive enough when it fails to improve. In KMA [34], the population size increases if stagnation occurs and decreases when improvement occurs. There is a static number of the increasing or decreasing population. The population size can increase until the maximum population size, while the population size can decrease until the minimum population size. In an artificial bee colony (ABC) [35], the bee performs a full random search after it fails to improve for certain periods.

The other consideration is the use case used to evaluate the metaheuristic when it was first introduced. In general, all metaheuristics were tested by using theoretical optimization problems. This theoretical problem consists of a set of mathematical functions. The set of 23 classic functions has been widely used in the first introduction of many metaheuristics, such as in the first introduction of KMA [34]. Other functions are CEC 2015, CEC 2017, and so on. In some studies, the new metaheuristics were also challenged to solve practical problems. Some optimization problems in mechanical engineering are commonly used, such as pressure vessel design problems, speed reducer design problems, welded beam design problems, and tension-compression spring design problems. The power flow optimization problem in the energy sector is also a common use case. Unfortunately, a study that uses optimization problems in the financial sector is rare.

Based on this consideration, especially on the adaptability and use case issues, this work is aimed to develop a new metaheuristic that is adaptive enough to tackle the stagnation problem. This stagnation can be detected, especially when the optimization process fails to improve the quality of the current solution during the iteration.

The main scientific contributions presented in this work are described below:

1) A new metaphor-free metaheuristic called as adaptive balance optimizer (ABO) is presented.

2) This work presents the adaptive strategy used in ABO, especially in switching between exploration and exploitation.

3) The performance of ABO is evaluated by implementing it to solve both theoretical optimization problem (a set of 23 classic functions) and practical optimization problem (portfolio optimization problem).

4) The performance of ABO is also competed with five other metaheuristics: MPA, SMA, GSO, NGO, and ZOA.

5) The hyper parameter evaluation is performed to evaluate the performance of ABO due to the increase of maximum iteration and population size.

The rest of this paper is organized as follows. The literature review regarding the latest of metaheuristics is performed in Section II. A detailed description of the adaptive balance optimizer consisting of its main concept, algorithm, and mathematical model is presented in Section III. The evaluation regarding the performance of ABO, especially in solving the set of 23 classic functions, the hyperparameter test, and the portfolio optimization problem, is presented in Section IV. The in-depth analysis of the simulation result, the drawback regarding the theory, limitations, and the algorithm complexity is discussed in Section V. Finally, the conclusion and the potential of future studies and development are summarized in Section VI.

II. Related Works

Adaptability is one important issue in the development of metaheuristics. Ironically, most of all, metaheuristics were developed without considering this issue. Many metaheuristics focus on developing strategies regarding the exploration and exploitation capability statically. It means most metaheuristics perform the same installed strategy, whether the improvement or stagnation happens.

Many metaheuristics respond to the improvement or stagnation by determining whether the new solution will be accepted to replace the previous solution or not. Some metaheuristics deploy a strict acceptance approach, meaning that a new solution replaces the previous solution only if the improvement occurs. On the other hand, some other metaheuristics deploy a non-strict acceptance approach which means that a new solution will replace the previous solution despite the improvement of stagnation. One distinct approach is introduced by simulated annealing, which uses a stochastic acceptance approach. If the improvement occurs, the new solution will replace the previous one immediately. Otherwise, the new solution may replace the previous solution based on a stochastic calculation. Ironically, metaheuristics that use improvement or stagnation circumstance to decide which strategy will be performed in the next iteration is rare to find.

Fortunately, some metaheuristics perform adaptive strategies in response to improvement or stagnation. KMA uses improvement or stagnation to determine the population size for the next iteration [34]. When the improvement occurs for two successive iterations, the population size decreases to reduce computational consumption. On the other hand, when stagnation occurs in two successive iterations, the population size increases to boost the exploration effort. However, the searches are still the same because whether the improvement or stagnation takes place, there are still three groups of agents where each group performs different searches. Meanwhile, an artificial bee colony (ABC) runs a different approach to make it adaptive. ABC generally performs neighborhood search and roulette wheel selection [35]. Meanwhile, after stagnation takes places for certain periods, full random search is performed without performing strict acceptance approach [35].

This adaptability issue is also not popular in the development of latest metaheuristics. Many metaheuristics, especially those that use a metaphor, focus on exploiting the mechanism of their metaphor as a novelty or contribution. Besides, many latest metaheuristics exploit their capability to outperform other metaheuristics as proof of their superiority.

TABLE I. SUMMARY OF SOME LATEST METAHEURISTICS

No	Metaheuristic	Metaphor	Adaptability	Acceptance Approach	Use Case
1	MHBA [30]	honey badger	no	strict	power flow
2	SPO [19]	paint	no	non strict	23 functions, CEC 2019, 52-bar planar truss structure, 120-bar dome truss structure, 3-bay 15-story frame, 3-bay 24-story frame
3	KMA [34]	komodo	population size decreases when improvement occurs and increases when stagnation occurs	non strict	23 functions
4	COVIDOA [17]	coronavirus	no	non strict	20 functions, CEC 2011
5	SSO [16]	squirrel	no	non strict	power system
6	MPA [9]	marine predator	no	non strict	30 functions, pressure vessel design, welded beam design, tension/compression spring design, operating fan schedule, building energy performance
7	GWO [8]	grey wolf	no	non strict	29 functions, tension/compression spring design, welded beam design, pressure vessel design, optical buffer design
8	BOA [13]	butterfly	no	non strict	30 functions, spring design, welded beam design, gear train design
9	POA [20]	pelican	no	strict	23 functions, pressure vessel design, speed reducer design, welded beam design, tension/compression spring design
10	TIA [22]	-	no	strict	23 functions
11	ASBO [24]	-	no	strict	23 functions
12	NGO [25]	northern goshawk	no	strict	23 functions, CEC 2015, CEC 2019, pressure vessel design, welded beam design, tension/compression spring design, speed reducer design
13	ZOA [26]	zebra	no	strict	23 functions, CEC 2017, tension/compression spring design, welded beam design, speed reducer design, pressure vessel design
14	COA [27]	coati	no	strict	CEC 2011, CEC 2017, pressure vessel design, speed reducer design, welded beam design, tension/compression spring design
15	RFO [33]	red fox	no	strict	22 functions, three bar truss, welded beam design, compression spring, pressure vessel, gear train
16	this work	-	different strategy during improvement and stagnation	strict and non-strict	23 functions, portfolio optimization problem

The other issue concerns the practical use case chosen to evaluate new metaheuristics in their first introduction. Problems in engineering are so popular; whether mechanical, civil, or electrical problems. Meanwhile, studies regarding new metaheuristic that uses problems in finance, are rare to find.

The summarized review of some latest metaheuristics is presented in Table I. There are 15 metaheuristics presented in Table I. Moreover, the proposed metaheuristic is placed in the last row to clarify its position among the existing metaheuristics.

Table I indicates that almost all metaheuristics have not considered adaptability. These metaheuristics perform the same strategy from the beginning of the iteration until the maximum iteration is reached. Some metaheuristics perform a strict acceptance approach to avoid the optimization process going to the worse solution. In comparison, some others still accept a worse solution, hoping it may lead to a better solution. Moreover, the financial sector, still not popular, became a practical use case to evaluate a new metaheuristic when it was first introduced.

Based on this circumstance, this work proposes a new metaheuristic that is adaptive enough when there is no improvement regarding the current solution. Moreover, the proposed metaheuristic gives equal treatment between two circumstances: improvement succeeds or fails. Besides, the proposed metaheuristic also performs both a strict acceptance approach and a non-strict acceptance approach.

III. PROPOSED MODEL

An adaptive balance optimizer (ABO) is designed as an adaptive metaheuristic that gives balance effort in intensifying the quality of the current solution and being adaptive when the improvement fails. Based on this objective, the reasoning for constructing ABO is as follows. ABO performs multiple searches, as many latest of metaheuristics also perform this approach. ABO performs guided search and random search explicitly. ABO performs different strategies when the improvement fails. ABO accommodates both a strict acceptance approach and a non-strict acceptance approach.

Based on this reasoning, the main concept of ABO is dividing strategy based on the circumstance it faces. There are two possible circumstances. The first circumstance is that the improvement happens. In this first circumstance, the strategy intensifies the exploitation. The second circumstance is that stagnation takes place. In this second circumstance, the strategy is deploying exploration. There are two searches in every circumstance: guided search and random search.

There are two searches performed in the exploitation mode. The first search is a guided search toward and beyond the best global solution. This guided search aims to trace possible better solutions between the corresponding solution and the global best solution. Moreover, this guided search is also designed to trace possible better solutions beyond the global best solution. As known, the global best solution is assumed as the best solution so far. It means that the quality of the best global solution is better than that of the corresponding solution. The probability of finding a better solution will increase when the corresponding solution moves closer to the global best solution. Meanwhile, it is also more probable for the global best solution to find a better solution by avoiding a worse solution. The second search is the limited random search or neighborhood search. In general, as a random search, the corresponding solution traces a new solution around its current solution. However, the search space is reduced as the iteration increases. In the exploitation mode, the strict acceptance approach is deployed in a guided search toward the global best solution and the limited random search. The searches in the exploitation mode are illustrated in Fig. 1.

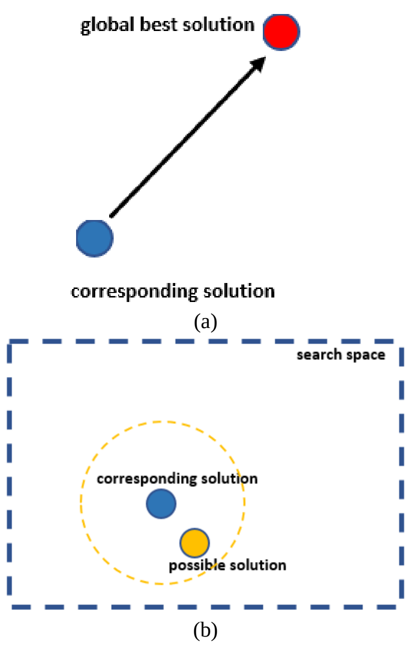

Fig. 1. Searches when improvement is achieved: (a) guided search toward the global best solution, (b) limited neighborhood search.

There are two searches performed in the exploration mode. The first search is the guided search relative to a randomly selected solution. This search can be viewed as a guided exploration. This search is included in the exploration because it is used a randomly selected solution among the population as the reference. The population is known to be spread within the search space, especially in the early iteration. Moving relative to one of these solutions means the corresponding solution traces any solution within the search space but based on a reference. If the reference quality is better than the quality of the corresponding solution, the corresponding solution moves toward the reference. Otherwise, the corresponding solution avoids this reference. The direction of this movement may push the corresponding solution closer to or away from the best global solution. This search is performed based on the reasoning that although the global best solution is the best solution, getting closer to the global best solution may push the corresponding solution to the local optimal entrapment. The second search is a full random search. As its name, the corresponding solution moves uniformly within the search space. This search can be viewed as a full exploration. In this exploration mode, the strict acceptance approach is not deployed. It means the new solution replaces the existing solution without considering the quality of this new solution. Moving to the worse solution may be better, which may lead to a better solution, rather than staying in the current solution without any improvement until the iteration ends. These two searches in the exploration mode are illustrated in Fig. 2.

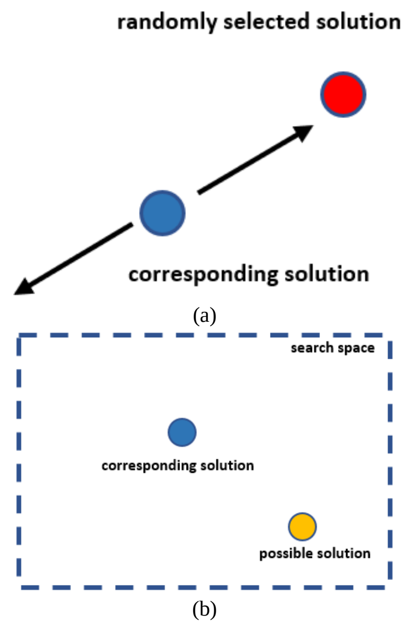

Fig. 2. Searches when improvement fails: (a) guided search relative to a randomly selected solution, (b) full random search.

This 'two-approach mechanism' needs a toggle to indicate whether the corresponding solution is in the exploitation or exploration mode. This toggle has two values updated at the end of every iteration. Suppose the corresponding solution fails to improve its quality after performing two sequential searches, whether, in the exploration or exploitation mode, the value of the toggle will be set so that exploration will be performed in the next iteration. Otherwise, the value of the toggle will be set so that exploitation will be performed in the next iteration.

The concept of ABO is then formalized using algorithm 1. As a metaheuristic, ABO consists of two phases: initialization and iteration. Lines 2 to 5 represent the initialization phase, while lines 6 to 25 represent the iteration phase. In the initialization phase, only one loop runs for the entire population. Meanwhile, two loops run in the iteration phase. The outer loop runs from the first iteration to the maximum iteration. The inner loop runs for the entire population. Lines 9 to 12 represent the exploitation mode, while lines 14 to 17 represent the exploration mode. Lines 19 to 23 represent the toggle updating process. g_s denotes the toggle value where 1 indicates the exploitation while 0 indicates the exploration.

Algorithm 1: Adaptive balance optimizer

```
begin
  for all s in S
    perform full random search using (1)
    update s_b using (2)
  end
  for t = 1 to t_m
    for all s in S
      if g_s = 1 then
        perform first guided search using (3)
        update s using (2) and s_b using (3)
        perform limited random search using (5)
        update s using (2) and s_b using (3)
      else
        perform second guided search using (6) and (7)
        update s_b using (3)
        perform full random search using (1)
        update s_b using (3)
      end if
      if improvement fail then
        g_s = 0
      else
        g_s = 1
      end if
    end for
  end for
end
output: s_b
```

The initialization phase consists of two processes. In the first process, the full random search is performed to generate initial solutions. This full random search is formalized using (1) where s denotes the solution, s_l denotes the lower boundary, s_u denotes the upper boundary, and U denotes the uniform random. The second process is the global best updating process as formalized in (2). In (2), s_b denotes the global best solution and f denotes the objective function.

$$s = U(s_l, s_u) \tag{1}$$

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

The exploration mode consists of two searches. The guided search toward the global best solution is formalized using (3) while the limited random search is formalized using (5). s_c denotes the solution candidate which is then evaluated using (4) that represents the strict acceptance approach. In (5), it is shown that the local search space gets narrow as the iteration increases.

$$s_c = s + U(0,1).(s_b - 2s) \tag{3}$$

$$s' = \begin{cases} s_c, f(s_c) < f(s) \\ s, else \end{cases} \tag{4}$$

$$s_c = s + U(-1,1).\left(1 - \frac{t}{t_m}\right).\left(\frac{s_u - s_l}{2}\right) \tag{5}$$

The guided search relative to randomly solution is formalized using (6) and (7). s_s denotes the randomly selected solution which is selected uniformly among the population S. Meanwhile, as shown in (7), the corresponding solution moves toward the reference if the quality of the reference is better than the quality of the corresponding solution. Otherwise, the corresponding solution avoids the reference.

$$s_s = U(S) \tag{6}$$

$$s = \begin{cases} s + U(0,1).(s - 2s_s), f(s) < f(s_s) \\ s + U(0,1).(s_s - 2s), else \end{cases} \tag{7}$$

Moreover, the detailed explanation regarding the influence of parameters used in this algorithm is as follows. The solution s plays an important role as autonomous agent performing the searching process. Meanwhile, the population S as the collection of s actualizes the population-based metaheuristics. The greater size of S in general improves the exploration capability although the size is not always linear to the performance quality. The s_b represents the best solution which is the basic form of collective intelligence in the swarm-based metaheuristic. Iteration t control the iterative process which is limited by the maximum iteration t_m. Uniform random U is used for the stochastic process which becomes the foundation of any metaheuristic. The boundaries s_l and s_u are used as the hard constraint in finding the possible solution. The objective function f is used for measurement of the quality of any solution.

IV. Simulation and Result

This section presents the simulation and evaluation of ABO in solving optimization problems. This evaluation can be split into three parts. The first part is a simulation regarding the theoretical optimization problem. The second part is a simulation regarding the hyperparameter test. The third part is a simulation regarding the practical optimization problem. In the first and third parts, ABO is benchmarked with five latest metaheuristics: MPA, GSO, SMA, NGO, and ZOA. MPA, GSO, and SMA are metaheuristics that do not deploy a strict acceptance approach. On the other hand, NGOs and ZOA are metaheuristics that deploy a strict acceptance approach. Meanwhile, ABO plays in the middle by deploying a strict acceptance approach and not a strict acceptance approach based on the circumstance it faces.

The first part is the simulation regarding the theoretical optimization problem. In this work, the set of 23 functions is used as the use case. This set of functions is chosen due to its broad and diverse circumstances and challenges. It can be split into three groups: high-dimension unimodal, high-dimension multimodal, and fixed-dimension multimodal functions. A detailed description of these functions is presented in Table II.

In this work, several adjusted parameters are set as follows. In general, the maximum iteration is 50 while the population size is 5. The fishing aggregate device for MPA is set 0.5 that represents balance between exploration within the search space and the guided exploration toward two randomly selected solutions. The z score for SMA is 0.5. The result in solving the high dimension unimodal functions, high dimension multimodal functions, and fixed dimension multimodal functions is presented in Table III, Table IV, and Table V respectively.

TABLE II. DETAIL DESCRIPTION OF 23 FUNCTIONS

No	Function	Model	Dimension	Problem Space	Global Opt.
1	Sphere	$\sum_{i=1}^{d} x_i^2$	50	[-100, 100]	0
2	Schwefel 2.22	$\sum_{i=1}^{d} \lvert x_i \rvert + \prod_{i=1}^{d} \lvert x_i \rvert$	50	[-100, 100]	0
3	Schwefel 1.2	$\sum_{i=1}^{d} \left(\sum_{j=1}^{i} x_j \right)^2$	50	[-100, 100]	0
4	Schwefel 2.21	$\max\{\lvert x_i \rvert, 1 \le i \le d\}$	50	[-100, 100]	0
5	Rosenbrock	$\sum_{i=1}^{d-1} (100(x_{i+1} + x_i^2)^2 + (x_i - 1)^2)$	50	[-30, 30]	0
6	Step	$\sum_{i=1}^{d-1} (x_i + 0.5)^2$	50	[-100, 100]	0
7	Quartic	$\sum_{i=1}^{d} i\, x_i^4 + random\, [0,1]$	50	[-1.28, 1.28]	0
8	Schwefel	$\sum_{i=1}^{d} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	50	[-500, 500]	-418.9x50
9	Ratsrigin	$10d + \sum_{i=1}^{d} \left(x_i^2 - 10cos(2\pi x_i)\right)$	50	[-5.12, 5.12]	0
10	Ackley	$-20 \cdot exp\left(-0.2 \cdot \sqrt{\frac{1}{d}\sum_{i=1}^{d} {x_i}^2}\right) - exp\left(\frac{1}{d}\sum_{i=1}^{d} \cos 2\pi x_i\right) + 20 + exp(1)$	50	[-32, 32]	0
11	Griewank	$\frac{1}{4000}\sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	50	[-600, 600]	0
12	Penalized	$\frac{\pi}{d}\left\{10\sin(\pi y_1) + \sum_{i=1}^{d-1}\left((y_i - 1)^2\left(1 + 10\, sin^2(\pi y_{i+1})\right)\right) + (y_d - 1)^2\right\} + \sum_{i=1}^{d} u(x_i, 10,100,4)$	50	[-50, 50]	0
13	Penalized 2	$0.1\left\{sin^2(3\pi x_1) + \sum_{i=1}^{d-1}\left((x_i - 1)^2\left(1 + sin^2(3\pi x_{i+1})\right)\right) + (x_d - 1)^2\left(1 + sin^2(2\pi x_d)\right)\right\} + \sum_{i=1}^{d} u(x_i, 5,100,4)$	50	[-50, 50]	0
14	Shekel Foxholes	$\left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2}\left(x_i - a_{ij}\right)^6}\right)^{-1}$	2	[-65, 65]	1
15	Kowalik	$\sum_{i=1}^{11} \left(a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right)^2$	4	[-5, 5]	0.0003
16	Six Hump Camel	$4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	[-5, 5]	-1.0316
17	Branin	$\left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10$	2	[-5, 5]	0.398
18	Goldstein-Price	$\left(1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)\right).\left(30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2)\right)$	2	[-2, 2]	3

19	Hartman 3	$-\sum_{i=1}^{4}\left(c_i exp\left(-\sum_{j=1}^{d}\left(a_{ij}\left(x_j-p_{ij}\right)^2\right)\right)\right)$	3	[1, 3]	-3.86
20	Hartman 6	$-\sum_{i=1}^{4}\left(c_i exp\left(-\sum_{j=1}^{d}\left(a_{ij}\left(x_j-p_{ij}\right)^2\right)\right)\right)$	6	[0, 1]	-3.32
21	Shekel 5	$-\sum_{i=1}^{5}\left(\sum_{j=1}^{d}\left(x_j-c_{ji}\right)^2+\beta_i\right)^{-1}$	4	[0, 10]	-10.1532
22	Shekel 7	$-\sum_{i=1}^{7}\left(\sum_{j=1}^{d}\left(x_j-c_{ji}\right)^2+\beta_i\right)^{-1}$	4	[0, 10]	-10.4028
23	Shekel 10	$-\sum_{i=1}^{10}\left(\sum_{j=1}^{d}\left(x_j-c_{ji}\right)^2+\beta_i\right)^{-1}$	4	[0, 10]	-10.5363

TABLE III. BENCHMARK RESULT IN SOLVING HIGH DIMENSION UNIMODAL FUNCTIONS

F	Paramater	MPA [9]	GSO [10]	SMA [11]	NGO [25]	ZOA [26]	ABO
1	mean	$4.3320x10^3$	$5.6223x10^4$	$7.4626x10^4$	0.0286	0.0000	0.0000
	st dev	$1.9644x10^3$	$1.3632x10^4$	$1.1712x10^4$	0.0306	0.0000	0.0000
	min	$1.4711x10^3$	$3.1528x10^4$	$4.1722x10^4$	0.0006	0.0000	0.0000
	max	$8.5016x10^3$	$8.1858x10^4$	$9.7149x10^4$	0.0940	0.0000	0.0000
	mean rank	4	5	6	3	1	1
2	mean	0.0000	$2.9343x10^{67}$	0.0000	0.0000	0.0000	0.0000
	st dev	0.0000	$1.0545x10^{68}$	0.0000	0.0000	0.0000	0.0000
	min	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	max	0.0000	$4.8598x10^{68}$	0.0000	0.0000	0.0000	0.0000
	mean rank	1	6	1	1	1	1
3	mean	$2.8487x10^4$	$1.3073x10^5$	$2.1393x10^5$	$5.4226x10^3$	0.0266	$9.3981x10^1$
	st dev	$1.9967x10^4$	$5.5564x10^4$	$4.0595x10^4$	$5.4283x10^3$	0.1163	$3.2324x10^2$
	min	$3.0737x10^3$	$4.8925x10^4$	$1.3394x10^5$	$1.0971x10^2$	0.0000	0.0000
	max	$8.5281x10^4$	$2.5705x10^5$	$2.6921x10^5$	$2.3616x10^4$	0.5470	$1.7244x10^3$
	mean rank	4	5	6	3	1	2
4	mean	$2.9387x10^1$	$5.3229x10^1$	$8.6454x10^1$	0.4646	0.0000	0.0000
	st dev	5.5277	5.9824	2.5769	0.2558	0.0000	0.0000
	min	$1.6824x10^1$	$4.0608x10^1$	$8.2000x10^1$	0.1033	0.0000	0.0000
	max	$4.3961x10^1$	$6.4096x10^1$	$9.1000x10^1$	1.0545	0.0000	0.0000
	mean rank	4	5	6	3	1	1
5	mean	$1.1760x10^6$	$1.4327x10^8$	$2.5991x10^8$	$4.9230x10^1$	$4.8938x10^1$	$4.8948x10^1$
	st dev	$6.8289x10^5$	$5.4495x10^7$	$4.5292x10^7$	0.3290	0.0199	0.0300
	min	$1.8459x10^4$	$5.1647x10^7$	$1.9294x10^8$	$4.8997x10^1$	$4.8907x10^1$	$4.8844x10^1$
	max	$2.6106x10^6$	$2.4223x10^8$	$3.4018x10^8$	$5.0357x10^1$	$4.8970x10^1$	$4.8982x10^1$
	mean rank	4	5	6	3	1	2
6	mean	$4.8607x10^3$	$5.5470x10^4$	$7.3622x10^4$	$1.0499x10^1$	$1.0452x10^1$	$1.0797x10^1$
	st dev	$1.5346x10^3$	$9.7258x10^3$	$1.1486x10^4$	0.5606	0.6440	0.4690
	min	$2.4904x10^3$	$3.9545x10^4$	$4.2292x10^4$	9.3723	8.9883	9.6110
	max	$7.7010x10^3$	$7.5723x10^4$	$8.6172x10^4$	$1.1632x10^1$	$1.1331x10^1$	$1.1453x10^1$
	mean rank	4	5	6	2	1	3
7	mean	1.4044	$1.0796x10^2$	$4.9953x10^2$	0.0341	0.0064	0.0181
	st dev	0.8089	$4.7866x10^1$	$6.3307x10^1$	0.0176	0.0047	0.0139
	min	0.2819	$4.2202x10^1$	$3.9856x10^2$	0.0047	0.0007	0.0015
	max	3.4301	$2.5038x10^2$	$6.0442x10^2$	0.0756	0.0201	0.0454
	mean rank	4	5	6	3	1	2

Table III indicates that ABO performs well in solving high-dimension unimodal functions. ABO becomes the first best in solving three functions (Sphere, Schwefel 2.22, and Schwefel 2.21), second best in solving three functions (Schwefel 1.2, Rosenbrock, and Quartic), and third best in solving one function (Step). Table III also indicates a significant gap between three metaheuristics (ABO, ZOA, and NGO) and the three others (SMA, MPA, and GSO), where the first group's performance is far better than the second group.

Table IV indicates that ABO performs well in solving high-dimension multimodal functions. ABO becomes the first best in solving two functions (Rastrigin and Ackley), second best in solving three functions (Griewank, Penalized, and Penalized 2), and fourth best in solving one function (Schwefel). Moreover, ABO can find the optimal global solution in solving Rastrigin and Ackley. Meanwhile, ZOA can also find the optimal global solution for these two functions.

The result in Table IV also divides these metaheuristics into two groups. The first group consists of MPA, GSO, and SMA. On the other hand, the second group consists of NGO, ZOA, and ABO. Metaheuristics in the second group perform better than metaheuristics in the first group, while the performance gap is significant. This circumstance takes place in almost all high-dimension multimodal functions except Schwefel.

Table V indicates fierce competition among these six metaheuristics in solving fixed-dimension multimodal functions. Fortunately, ABO also performs well in solving these functions. Among these ten functions, ABO performs as the second best in six functions (Shekel Foxholes, Kowalik, Branin, Goldstein-Price, Hartman 3, and Hartman 6), third best in one function (Six Hump Camel), fourth best in one function (Shekel 5), and fifth best in two functions (Shekel 7 and Shekel 10). Different from the high-dimension unimodal and high-dimension multimodal functions, fierce competition happens for all metaheuristics in all ten functions. In general, the performance gap among the metaheuristics is narrow.

TABLE IV. BENCHMARK RESULT IN SOLVING HIGH DIMENSION MULTIMODAL FUNCTIONS

F	Parameters	MPA [9]	GSO [10]	SMA [11]	NGO [25]	ZOA [26]	ABO
8	mean	-3.4011×10^3	-4.6121×10^3	-4.9782×10^3	-4.2472×10^3	-2.8643×10^3	-3.9093×10^3
	st dev	6.5297×10^2	1.2521×10^3	5.6572×10^2	6.4312×10^2	5.6143×10^2	5.8738×10^2
	min	-5.3489×10^3	-7.2897×10^3	-6.0276×10^3	-5.3898×10^3	-3.7800×10^3	-5.2110×10^3
	max	-2.3316×10^3	-2.6163×10^3	-3.7595×10^3	-3.1568×10^3	-1.8763×10^3	-2.9836×10^3
	mean rank	5	2	1	3	6	4
9	mean	3.5962×10^2	5.1678×10^2	2.0979×10^2	1.5365	0.0000	0.0000
	st dev	7.6955×10^1	5.6595×10^1	4.0802×10^1	4.6372	0.0000	0.0000
	min	1.3328×10^2	3.9772×10^2	1.3101×10^2	0.0096	0.0000	0.0000
	max	4.4327×10^2	6.3509×10^2	2.7401×10^2	2.1832×10^1	0.0000	0.0000
	mean rank	5	6	4	3	1	1
10	mean	9.9728	1.9884×10^1	1.7682×10^1	0.0984	0.0000	0.0000
	st dev	1.5062	0.5195	0.3827	0.2589	0.0000	0.0000
	min	5.6136	1.7694×10^1	1.6002×10^1	0.0061	0.0000	0.0000
	max	1.1753×10^1	2.0513×10^1	1.8082×10^1	1.1222	0.0000	0.0000
	mean rank	4	6	5	3	1	1
11	mean	4.3770×10^1	4.8578×10^2	6.0186×10^2	0.0592	0.0000	0.0209
	st dev	1.6561×10^1	1.0101×10^2	1.0956×10^2	0.1247	0.0000	0.0805
	min	2.2313×10^1	3.4842×10^2	3.8985×10^2	0.0004	0.0000	0.0000
	max	7.8995×10^1	6.7935×10^2	7.9722×10^2	0.5502	0.0000	0.3806
	mean rank	4	5	6	3	1	2
12	mean	9.8780×10^4	2.1018×10^8	5.3614×10^8	1.0144	1.0824	1.0486
	st dev	1.6112×10^5	1.6467×10^8	1.4009×10^8	0.1346	0.1164	0.1113
	min	3.1218×10^2	4.6630×10^7	2.0839×10^8	0.7859	0.8031	0.8918
	max	6.2116×10^5	6.4261×10^8	7.7637×10^8	1.2334	1.2492	1.2878
	mean rank	6	4	5	1	3	2
13	mean	1.5887×10^6	5.5397×10^8	1.0300×10^9	3.3180	3.0916	3.1319
	st dev	2.1136×10^6	3.4988×10^8	2.1375×10^8	0.1249	0.0393	0.0176
	min	9.1574×10^3	1.8162×10^8	6.7786×10^8	3.1396	2.9894	3.0821
	max	1.0123×10^7	1.5915×10^9	1.4917×10^9	3.5531	3.1363	3.1835
	mean rank	4	5	6	3	1	2

TABLE V. BENCHMARK RESULT IN SOLVING FIXED DIMENSION MULTIMODAL FUNCTIONS

F	Parameters	MPA [9]	GSO [10]	SMA [11]	NGO [25]	ZOA [26]	ABO
14	mean	1.1070x10^{1}	1.0182x10^{1}	5.4396	6.9609	9.3836	6.8624
	st dev	4.1377	4.8001	3.5089	4.6037	3.7321	3.0608
	min	3.0579	1.9920	0.9980	0.9980	0.9980	2.0116
	max	1.7717x10^{1}	2.1073x10^{1}	1.2671x10^{1}	1.6441x10^{1}	1.2670x10^{1}	1.3619x10^{1}
	mean rank	6	5	1	3	4	2
15	mean	0.0254	0.0910	0.1326	0.0046	0.0076	0.0048
	st dev	0.0158	0.3541	0.0236	0.0068	0.0139	0.0045
	min	0.0050	0.0016	0.0778	0.0004	0.0003	0.0010
	max	0.0666	1.6747	0.1484	0.0206	0.0462	0.0165
	mean rank	4	5	6	1	3	2
16	mean	-0.9560	-1.0198	-0.0367	-1.0316	-0.9480	-1.0159
	st dev	0.0620	0.0315	0.1083	0.0000	0.2207	0.0189
	min	-1.0218	-1.0316	-0.4578	-1.0316	-1.0316	-1.0315
	max	-0.7795	-0.8923	0.0000	-1.0316	-0.1680	-0.9609
	mean rank	4	2	6	1	5	3
17	mean	3.4078	1.3439	0.6438	0.3982	7.4391	0.4691
	st dev	2.5163	2.7178	0.0000	0.0003	1.0158x10^{1}	0.0688
	min	0.5003	0.3981	0.6438	0.3981	0.3981	0.3981
	max	8.7875	1.0341x10^{1}	0.6438	0.3993	3.5964x10^{1}	0.6438
	mean rank	5	4	3	1	6	2
18	mean	3.0989x10^{1}	1.3393x10^{1}	3.0000	2.9077x10^{1}	4.7903x10^{1}	4.0246
	st dev	2.2928x10^{1}	2.4219x10^{1}	0.0000	3.5703x10^{1}	7.8849x10^{1}	1.3238
	min	4.5662	3.0000	3.0000	3.0000	2.9999	3.0000
	max	8.1122x10^{1}	8.4200x10^{1}	3.0000	8.4824x10^{1}	3.3648x10^{2}	7.2993
	mean rank	5	3	1	4	6	2
19	mean	-3.3279	-0.0357	-0.0495	-0.0495	-0.0495	-0.0495
	st dev	0.4178	0.0174	0.0000	0.0000	0.0000	0.0000
	min	-3.8536	-0.0495	-0.0495	-0.0495	-0.0495	-0.0495
	max	-2.2160	-0.0056	-0.0495	-0.0495	-0.0495	-0.0495
	mean rank	1	6	2	2	2	2
F	**Parameters**	**MPA [9]**	**GSO [10]**	**SMA [11]**	**NGO [25]**	**ZOA [26]**	**ABO**
20	mean	-1.3750	-2.5835	-0.9122	-2.7680	-2.1337	-2.6242
	st dev	0.5288	0.6656	0.6304	0.3178	0.4571	0.1370
	min	-2.2961	-3.2141	-2.5003	-3.2903	-3.0922	-2.7921
	max	-0.5046	-0.9227	-0.1587	-2.1049	-1.4533	-2.3383
	mean rank	5	3	6	1	4	2
21	mean	-0.8384	-3.7757	-2.1840	-3.0163	-3.7476	-2.4582
	st dev	0.2820	2.5466	2.8701	2.3674	2.3361	0.8451
	min	-1.4088	-9.2109	-1.0153x10^{1}	-1.0084x10^{1}	-9.1107	-5.0406
	max	-0.4575	-0.5200	-0.5090	-0.8561	-0.7072	-1.2512
	mean rank	6	1	5	3	2	4
22	mean	-0.8586	-3.0051	-3.1134	-2.8936	-3.4404	-2.5657
	st dev	0.3301	1.8833	3.6634	1.9308	2.1138	0.9282
	min	-1.5584	-8.5430	-1.0403x10^{1}	-8.5422	-8.3691	-5.0053
	max	-0.3876	-0.8972	-0.6342	-0.8590	-0.9487	-1.5357
	mean rank	6	3	2	4	1	5
23	mean	-1.2082	-3.9409	-2.5317	-3.6734	-3.1585	-2.3391
	st dev	0.4895	2.8005	2.2583	1.8173	1.4812	0.80007
	min	-2.4122	-1.0195x10^{1}	-1.0536x10^{1}	-7.6571	-7.2028	-4.3752
	max	-0.5738	-0.8049	-0.7951	-1.4377	-1.0123	-1.2768
	mean rank	6	1	4	2	3	5

TABLE VI. CLUSTER BASED SUPERIORITY OF ABO COMPARED TO OTHER METAHEURICTICS

Group	MPA [9]	GSO [10]	SMA [11]	NGO [25]	ZOA [26]
1	6	7	6	5	0
2	6	5	5	4	2
3	9	6	5	2	6
Total	21	18	16	11	8

Table VI summarizes the performance comparison between ABO and the five other metaheuristics. The comparison represents the superiority of ABO compared to other metaheuristics based on the group of functions. This group-based comparison is needed because of the distinct characteristics among the groups of functions. It is assumed that some metaheuristics may be better in a certain group but mediocre in another group. The last row represents the total number of functions where ABO outperforms a metaheuristic.

Table VI indicates that ABO is superior to MPA, GSO, and SMA and still competitive compared with NGO and ZOA. ABO is better than MPA, GSO, SMA, NGO, and ZOA in solving 21, 18, 16, 11, and 8 functions, respectively. Meanwhile, ABO equals NGO in solving two functions and ZOA in solving six functions. It means ABO is worse than NGO in solving ten functions and ZOA in solving nine functions. ABO is superior to MPA in all groups of functions. Meanwhile, ABO is superior to GSO and SMA in solving unimodal and multimodal functions. On the other hand, ABO is still competitive compared to GSO and SMA in solving fixed-dimension multimodal functions. ABO is superior to NGO in solving unimodal and high-dimension multimodal functions but inferior in solving fixed-dimension multimodal functions. Contrary, ABO is superior to ZOA in solving fixed-dimension multimodal functions but inferior in solving high-dimension functions. Based on this summary, ABO has fierce competition with NGO and ZOA, where ABO is better than NGO in high-dimension functions while ABO is better than ZOA in fixed-dimension functions.

The second part of the simulation is regarding the hyper-parameter evaluation. In this work, two parameters are observed. The first parameter is the population size, while the second is the maximum iteration. The set of 23 functions is still used in this evaluation. The result of the population size evaluation is presented in Table VII, while the result of the maximum iteration is presented in Table VIII.

Table VII indicates that the increase in population size does not improve the quality of the solution in almost all functions. The improvement takes place only in three functions. Among these three functions, two functions are high dimension unimodal functions, and one function is a fixed dimension multimodal function. Meanwhile, there are two reasons why the improvement fails. The first reason is that the global optimal or quasi-optimal solution has been achieved. This reason can be found in twelve functions. The second reason is that ABO fails to find the quasi-optimal solution after reaching the maximum iteration. This reason can be found in eight functions.

TABLE VII. RELATION BETWEEN POPULATION SIZE AND AVERAGE FITNESS SCORE

Function	Average Fitness Score		Significantly Improved?
	$n(X) = 10$	$n(X) = 40$	
1	0.0000	0.0000	no
2	0.0000	0.0000	no
3	4.2181×10^1	0.0030	yes
4	0.0000	0.0000	no
5	4.8935×10^1	4.8880×10^1	no
6	1.0556×10^1	9.8178	no
7	0.0187	0.0056	yes
8	-4.2131×10^3	-4.7692×10^3	no
9	0.0000	0.0000	no
10	0.0000	0.0000	no
11	0.0000	0.0000	no
12	0.9883	0.7949	no
13	3.1324	3.1151	no
14	2.8768	1.5147	no
15	0.0043	0.0012	yes
16	-1.0258	-1.0290	no
17	0.4353	0.4057	no
18	3.5492	3.1038	no
19	-0.0495	-0.0495	no
20	-2.6985	-2.9387	no
21	-2.5507	-3.5834	no
22	-2.4601	-3.3685	no
23	-2.9200	-3.3965	no

TABLE VIII. RELATION BETWEEN MAXIMUM ITERATION AND AVERAGE FITNESS SCORE

Function	Average Fitness Score		Significantly Improved?
	t = 60	t = 120	
1	0.0000	0.0000	no
2	0.0000	0.0000	no
3	1.9130×10^1	0.0638	yes
4	0.0000	0.0000	no
5	4.8975×10^1	4.8939×10^1	no
6	1.0700×10^1	1.0506×10^1	no
7	0.0253	0.0164	no
8	-4.0184×10^3	-4.2073×10^3	no
9	0.0000	0.0000	no
10	0.0000	0.0000	no
11	0.0000	0.0000	no
12	1.0667	0.9978	no
13	3.1436	3.1255	no
14	4.2826	2.6412	no
15	0.0043	0.0034	no
16	-1.0224	-1.0248	no
17	0.4671	0.4316	no
18	3.6447	3.2859	no
19	-0.0495	-0.0495	no
20	-2.7096	-2.7824	no
21	-2.7736	-2.5821	no
22	-2.1184	-2.9625	no
23	-2.2307	-3.0198	no

Table VIII indicates that the increase of the maximum iteration does not improve the quality of the solution in almost all functions. The improvement occurs only in one high-dimension unimodal function (Schwefel 1.2). There are twelve functions where the global optimal or quasi-optimal solution has been achieved in the low maximum iteration circumstance so that there is no improvement anymore due to the increase of maximum iteration.

The third part is the evaluation of ABO in solving a practical optimization problem: portfolio optimization. This problem is chosen because most works that introduced new metaheuristics chose problems in mechanical engineering or power flow distribution as their use case. The portfolio optimization problem is an important optimization work in the finance sector. In general, two objectives can be chosen for the portfolio optimization problem. The portfolio optimization problem can be defined as an effort to arrange financial assets (stock, bond, gold, and so on) to maximize the return or control the risk [36].

In this work, the assets are stocks of the energy or mining companies listed in IDX30. IDX30 is a list published by the Indonesian stock exchange that consists of 30 very liquid stocks with high market capitalization and strong fundamentals. There are ten stocks, and the list is presented in Table IX. These stocks have three important attributes: stock index, market price, and year-on-year capital gain. The market price and capital gain are presented in rupiah per share. The information regarding these stocks was obtained on February 22, 2023.

This portfolio optimization is taken based on some scenario. The objective is maximizing the total capital gain which is calculated by accumulating the capital gain of all shares that are held. The quantity of each stock ranges from 200 lots to 1,000 lots where each lot represents 100 shares. The maximum total investment is five billion rupiah. This problem can be seen as a unimodal problem where the dimension is 10.

In this portfolio optimization, ABO is also competed with five metaheuristics like in the first part: MPA, GSO, SMA, NGO, and ZOA. The population size is 10 where the maximum iteration is 30. The result is presented in Table X.

TABLE IX. STOCK INFORMATION

No	Stock Index	Price	Capital Gain
1	ADRO	2,850	530
2	ANTM	2,050	-150
3	BRPT	910	-55
4	ESSA	940	295
5	INCO	6,800	1,860
6	ITMG	35,575	11,600
7	MDKA	4,610	853
8	MEDC	1,060	500
9	PGAS	1,540	135
10	PTBA	3,540	530

Table X indicates that ABO is competitive in solving this portfolio optimization problem, although it is not the best performer. ABO becomes the third best after SMA and NGO. On the other hand, ZOA becomes the worst metaheuristic in solving this portfolio optimization problem, although it is very competitive in solving the set of 23 functions.

TABLE X. PORTFOLIO OPTIMIZATION RESULT

No	Metaheuristic	Total Capital Gain
1	ABO	1,461,870,054
2	ZOA [26]	1,353,039,265
3	NGO [25]	1,478,676,650
4	SMA [11]	1,479,375,445
5	GSO [10]	1,461,377,050
6	MPA [9]	1,377,967,381

V. DISCUSSION

The simulation result shows that ABO is competitive enough as a swarm-based metaheuristic. ABO can find an acceptable solution in both theoretical and practical optimization problems. ABO can find the optimal global solution in solving five functions. ABO is superior to MPA, SMA, and GSO and competitive to NGO and ZOA in solving the set of 23 functions. Meanwhile, ABO becomes the third best in solving the portfolio optimization problem.

Solving the theoretical optimization problem shows that a strict-acceptance approach is important to achieve good performance, especially for high-dimension functions. In these functions, ZOA [26] and NGO [25] are metaheuristics that implement a strict-acceptance approach, while ABO implements both strict-acceptance and non-strict-acceptance approaches. ABO, ZOA, and NGO are the best of the three in solving almost all functions in the big dimension problems, while GSO [10], MPA [9], and SMA [11] do not implement a strict-acceptance approach. This circumstance indicates that metaheuristics implementing a strict-acceptance approach significantly achieves better results than others. On the other hand, circumstance becomes more dynamic in solving fixed-dimension multimodal functions where the gap among metaheuristics is narrow whether these metaheuristics implement a strict-acceptance approach. It means avoiding a worse solution is important in solving high-dimension functions, while this strategy is not important in solving fixed-dimension multimodal functions.

The strict-acceptance approach also does not significantly affect solving the portfolio optimization problem. Table X shows the narrow gap between the best and worst metaheuristics. On the other hand, ZOA as the worst performer is a metaheuristic that adopts a strict-acceptance approach. On the contrary, MPA, as a metaheuristic that does not adopt a strict acceptance approach, becomes the second worst performer. As a metaheuristic that does not adopt a strict-acceptance approach, SMA becomes the best performer. Meanwhile, NGO becomes the second-best performer as a metaheuristic that adopts a strict-acceptance approach.

The simulation result also shows that the competition among latest metaheuristics becomes tougher. It is common for many latest metaheuristics to deploy multiple searches and enrich the guided search with the random search. Many metaheuristics can find the global optimal in several functions and quasi-optimal in many other functions. Meanwhile, many metaheuristics still need help finding the quasi-optimal solution, especially in the low maximum iteration and low population size. This circumstance strengthens the no-free-lunch theory that a metaheuristic cannot solve all problems with superior results.

There are three loops conceptually performed during the iteration. The outer loop is the iteration from the first iteration to the maximum iteration. The intermediate loop is the iteration for the entire population. The inner loop is the iteration for the entire dimension because all dimensions are calculated in every search. Meanwhile, there are two searches performed by every agent in every iterations. Based on this explanation, the algorithm complexity of ABO is presented as $O(2t_{max}.n(X).n(D))$. This complexity is normal for the population-based metaheuristic. Moreover, this complexity is achieved because ABO does not implement a sorting process in every iteration.

There are limitations regarding this work and its proposed metaheuristics, even though the proposed ABO performs well in solving both theoretical and practical optimization works. In ABO, the adaptability in tackling the local optimal is performed by choosing a non-strict acceptance approach. Meanwhile, this non-strict acceptance approach is implemented in any iteration. It differs from simulated annealing, where accepting a worse solution becomes more difficult as iteration increases. Meanwhile, different metaheuristic, such as tabu search, uses tabu list to restrict the repetition of a similar solution. This circumstance shows that there are various adaptive approaches that can be explored in the future. At the same time, a single metaheuristic such as ABO cannot adopt various adaptive strategies into a single metaheuristic.

This work has presented the use of optimization problem in financial sector, which is the portfolio optimization problem in the introduction of a new metaheuristic. This work also proves that ABO is competitive enough in solving this problem which is an integer-based problem. Meanwhile, there are various kinds of other optimization problems in the financial sector, such as credit risk assessment, investment planning, debtor analysis, and refinancing problems. These problems can also be addressed in future work.

There is also a limitation in choosing a practical optimization problem as a use case to evaluate the performance of the new metaheuristic. This work chooses a portfolio optimization problem as the use case, with its characteristics being integer-based and unimodal. On the other hand, there are various practical optimization problems, whether common or not, in many studies introducing new metaheuristics. These problems can be used for future studies, especially proposing an improved or modified version of ABO.

The future studies can also be performed by implementing ABO to solve various sustainable development goals (SDGs) related issues. SDG has become the global issue and consideration for developing sustainable society and environment. For example, efficient energy consumption becomes the main and important issue related to climate change, renewable and affordable energy. Besides, optimization plays an important role in the operation of industry, transportation, and many other economic activities.

VI. Conclusion

The introduction of a new adaptive metaheuristic, namely adaptive balance optimizer (ABO), has been presented in this paper. This proposed model is designed to make a metaheuristic adaptive, especially when facing optimal local circumstances. This paper also presents the competitiveness of ABO in solving both theoretical and practical optimization problems. ABO is better than MPA, GSO, SMA, NGO, and ZOA in solving 21, 18, 16, 11, and 8 functions, respectively, in solving 23 functions. It means ABO is superior to MPA, GSO, and SMA and still competitive with NGO and ZOA in solving 23 functions. Meanwhile, ABO is still competitive in solving portfolio optimization problems, although ABO is not the best performer in solving this problem.

Adaptability can be used for future studies in metaheuristics. Various strategies have yet to be explored to make metaheuristics more adaptive, especially in tackling the local optimal entrapment. Besides, developing a superior metaheuristic that can solve the optimization problem in the low maximum iteration and low population size becomes challenging too. Moreover, future work can be conducted by addressing several common issues, such as scalability and more practical recent and future use cases. The scalability issue is related to wider boundaries and higher dimensions of the problem. Meanwhile, there are various recent and future optimization problems, such as in the green and blue economy, climate change, renewable energy, and many more.

Acknowledgment

This work was financially supported by Telkom University, Indonesia.

References

[1] L. Li, Y. Sun, and B. Wang, “A hybrid genetic algorithm for service caching and task offloading in edge-cloud computing”, International Journal of Advanced Computer Science and Applications, vol. 13, no. 11, pp. 761-765, 2022.

[2] A. Damarla and D. Sumathi, “An approach for optimization of features using gorilla troop optimizer for classification of melanoma”, International Journal of Advanced Computer Science and Applications, vol. 13, no. 10, pp. 275-286, 2022.

[3] Q. H. Zhai, T. Ye, M. X. Huang, S. L. Feng, and H. Li, “Whale optimization algorithm for multiconstraint second-order stochastic dominance portfolio optimization”, Computational Intelligence and Neuroscience, vol. 2020, ID. 8834162, pp. 1-19, 2020.

[4] Y. Yang, J. Tao, J. Zhou, Y. Wang, and X. Guo, “An improved sparrow search algorithm and its application in HIFU sound field”, Computational Intelligence and Neuroscience, vol. 2023, ID. 1228685, pp. 1-19, 2023.

[5] X. Wang, W. Wu, and Z. Xing, “Multi-decision points model to solve coupled-task scheduling problem with heterogeneous multiAGV in manufacturing systems”, International Journal of Industrial Engineering Computations, vol. 14, no. 1, pp. 49-64, 2023.

[6] H. R. Moshtaghi, A. T. Eshlaghy, and M. R. Motadel, “A comprehensive review on meta-heuristic algorithms and their

classification with novel approach", Journal of Applied Research on Industrial Engineering, vol. 8, no. 1, pp. 63-89, 2021.

[7] J. Swan, S. Adriaensen, A. E. I. Brownlee, K. Hammond, C. G. Johnson, A. Kheiri, F. Krawiec, J. J. Merelo, L. L. Minku, E. Ozcan, G. L. Pappa, P. Garcia-Sanchez, K. Sorensen, S. Vob, M. Wagner, and D. R. White, "Metaheuristics in the large", European Journal of Operational Research, Vol. 297, pp. 393-406, 2022.

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

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

[10] 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.

[11] 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.

[12] M. Dehghani, Z. Montazeri, H. Givi, J. M. Guerrero, and G. Dhiman, "Darts game optimizer: a new optimization technique based on darts game", International Journal of Intelligent Engineering and Systems, vol. 13, no. 5, pp. 286–294, 2020.

[13] S. Arora and S. Singh, "Butterfly optimization algorithm: a novel approach for global optimization", Soft Computing, vol. 23, no. 3, pp. 715–734, 2019.

[14] M. S. Braik, "Chameleon swarm algorithm: a bio-inspired optimizer for solving engineering design problems", Expert Systems with Applications, vol. 174, ID. 114685, pp. 1-25, 2021.

[15] S. Kaur, L. K. Awasthi, A. L. Sangal, and G. Dhiman, "Tunicate swarm algorithm: a new bio-inspired based metaheuristic paradigm for global optimization", Engineering Applications of Artificial Intelligence, vol. 90, ID. 103541, 2020.

[16] M. Suman, V. P. Sakthivel, and P. D. Sathya, "Squirrel search optimizer: nature inspired metaheuristic strategy for solving disparate economic dispatch problems", International Journal of Intelligent Engineering and Systems, vol. 13, no. 5, pp. 111–121, 2020.

[17] A. M. Khalid, K. M. Hosny, and S. Mirjalili, "COVIDOA: a novel evolutionary optimization algorithm based on coronavirus disease replication lifecycle", Neural Computing and Applications, vol. 34, pp. 22465-22492, 2022.

[18] 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, ID. 108457, pp. 1-29, 2022.

[19] A. Kaveh, S. Talatahari, and N. Khodadadi, "Stochastic paint optimizer: theory and application in civil engineering", Engineering with Computers, vol. 38, pp. 1921-1952, 2022.

[20] P. Trojovsky and M. Dehghani, "Pelican optimization algorithm: a novel nature-inspired algorithm for engineering applications", Sensors, Vol. 22, ID. 855, pp. 1-34, 2022.

[21] P. D. Kusuma and A. L. Prasasti, "Guided pelican algorithm", International Journal of Intelligent Engineering and Systems, vol. 15, no. 6, pp. 179-190, 2022.

[22] 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.

[23] P. D. Kusuma and A. Dinimaharawati, "Three on three optimizer: a new metaheuristic with three guided searches and three random searches", International Journal of Advanced Science and Applications, vol. 14, no. 1, pp. 420-429, 2023.

[24] M. Dehghani, S. Hubalovsky, and P. Trojovsky, "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, ID: e910, pp. 1-29, 2022.

[25] 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.

[26] 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.

[27] M. Dehghani, Z. Montazeri, E. Trojovska, and P. Trojovsky, "Coati optimization algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems", Knowledge-Based Systems, vol. 259, ID. 110011, pp. 1-43, 2023.

[28] E. Trojovska, M. Dehghani, and P. Trojovsky, "Fennec fox optimization: a new nature-inspired optimization algorithm", IEEE Access, vol. 10, pp. 84417-84443, 2022.

[29] E. Trojovska and M. Dehghani, "A new human-based metaheuristic optimization method based on mimicking cooking training", Scientific Reports, vol. 12, ID. 14861, pp. 1-24, 2022.

[30] S. A. Yasear and H. M. A. Ghanimi, "A modified honey badger algorithm for solving optimal power flow optimization problem", International Journal of Intelligent Engineering and Systems, vol. 15, no. 4, pp. 142–155, 2022.

[31] M. I. A. Latiffi, M. R. Yaakub, and I. S. Ahmad, "Flower pollination algorithm for feature selection in tweets sentiment analysis", International Journal of Advanced Computer Science and Applications, vol. 13, no. 5, pp. 429-436, 2022.

[32] M. Dehghani, M. Mardaneh, J. S. Guerrero, O. P. Malik, and V. Kumar, "Football game based optimization: an application to solve energy commitment problem", International Journal of Intelligent Engineering and Systems, vol. 13, no. 5, pp. 514-523, 2020.

[33] D. Polap and M. Wozniak, "Red fox optimization algorithm", Expert Systems with Applications, vol. 166, ID. 114107, pp. 1-21, 2021.

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

[35] A. Banharnsakun, "A new approach for solving the minimum vertex cover problem using artificial bee colony", Decision Analytics Journal, vol. 6, ID. 100175, pp. 1-6, 2023.

[36] Y. Chen, X. Zhao, and J. Yuan, "Swarm intelligence algorithms for portfolio optimization problems: overview and recent advances", Mobile Information Systems, vol. 2022, ID. 4241049, pp. 1-15, 2022.