Multiple Interaction-Dual Leader Optimizer: A Novel Metaheuristic to Solve High Dimension Problem

Purba Daru Kusuma, *Member IAENG*, Anggunmeka Luhur Prasasti

***Abstract*—This work introduces a new metaheuristic designed to solve high-dimension problems, namely multiple interaction-dual leader optimizer (MIDLO). There are two novel mechanics regarding this proposed metaheuristic. First, there are multiple interactions and movements during the guided search carried out by every agent in every iteration. Second, there are two references in every interaction: the global best solution and a randomly chosen solution. Meanwhile, a random search within the space is carried out by any agent that fails to improve its own solution after several consecutive trials. In this work, MIDLO is tested to solve 23 classic functions so that its performance can be evaluated. In this test, MIDLO is competed with five swarm-based metaheuristics: particle swarm optimization (PSO), stochastic marine predator algorithm with multiple candidates (SMPA-MC), golden search optimizer (GSO), average and subtraction-based optimizer (ASBO), and guided pelican algorithm (GPA). The result indicates that MIDLO is better than these sparing metaheuristics, especially in optimizing the high-dimension functions. Overall, MIDLO is better than PSO, SMPA-MC, GSO, ASBO, and GPA in solving 20, 12, 15, 9, and 13 functions respectively. The result also indicates that the number of interactions has a positive relation with the performance while the effect of a maximum number of lives depends on the problem to solve.**

***Index Terms*—optimization, metaheuristic, swarm intelligence, high dimension problem.**

I. Introduction

A lot of real-world or practical problems can be categorized as optimization problems. These problems span a wide spectrum, especially in the engineering sector, such as transportation [1], energy [2], manufacture [3], electrical distribution [4], product assortment [5], aerospace design [6], and so on. This fact is highly related to the nature of human beings in achieving an objective or some objectives in the most efficient way.

Many optimizations deal with high-dimension problems. A high dimension problem can be described as a problem that consists of a lot of decision variables to adjust. Each variable can be independent or dependent among other variables. High-dimension problem is often found in many cases, especially in the operations research area. This problem can be found in a company that deals with a lot of suppliers or customers. It is also can be found in the manufacturer or production system that deals with a lot of products, for example, a manufacturer that deals with a lot of raw materials in producing some or many finished products. The other example is a department store that must order a lot of products. The purchasing manager should determine the quantity of each product that should be ordered.

Every optimization work should deal with objectives and constraints. The most common objective is minimization, such as minimizing production cost [7], make-span [8], consumed energy [9], travel distance [10], and so on. Meanwhile, some optimization works implement maximization as an objective, such as revenue or profit [11]. The constraint is often related to the limited resources used in the optimization work, such as capital, minimum or maximum order quantity, storage capacity, fleet size, and so on. Therefore, every optimization study consists of three important parts: objective, constraint, and optimization tool.

Many optimization studies used the metaheuristic method as their optimization tool. The popularity of metaheuristics comes from several aspects. First, metaheuristic uses a stochastic approach so that not all solutions in the solution space are traced [12]. The advantage is metaheuristic needs less computational resource or computational time [12]. Meanwhile, the consequence is that the global optimal solution is not guaranteed [12]. Second, the metaheuristic focuses on the objective and constraint so that the metaheuristic is flexible to solve various optimization problems with their own objective and constraint. Moreover, metaheuristic is also flexible for solving the problem where the dimension is spanned to a lot of dimensions. Third, nowadays, there are hundreds of metaheuristics ready to choose to solve any optimization problems.

The massive development of new metaheuristics in the recent decade is related to the nature of the stochastic approach. Many methods can be modified and combined to construct a new metaheuristic. Meanwhile, the searching behavior of metaheuristics is like the behavior of animals during searching for food, hunting, and foraging. This circumstance makes many shortcoming metaheuristics inspired by the behavior of many animals during hunting or

Manuscript received November 11, 2022; revised March 3, 2023. This work was financially supported by Telkom University, Indonesia.

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

Anggunmeka Luhur Prasasti is an assistant professor in computer engineering, Telkom University, Indonesia (e-mail: anggunmeka@telkomuniversity.ac.id)

foraging, such as pelican [13], leopard [14], bird [15], wolf [16], ocean predator [17], goshawk [18], Komodo [19], butterfly [20], cheetah [21], honey badger [22], squirrel [23], red deer [24], and so on. Some metaheuristics were also inspired by some stochastic processes found in some classic games, such as dart games [25], puzzles [26], ring toss [27], shell games [28], and so on. Nowadays, many new metaheuristics were developed based on swarm intelligence. The reason is that swarm intelligence is proven better than a single solution or evolution-based metaheuristics. It is common in many works to propose a new metaheuristic using a genetic algorithm (GA) as one of the sparing metaheuristics only to prove that the proposed metaheuristic in these studies was better than GA.

This massive development of metaheuristics is also related to the nature of metaheuristics that does not trace all possible solutions. It makes there is not any metaheuristic suitable to solve all optimization problems as it was stated in the no-free-lunch theory [29]. The strength or weakness of any metaheuristic also depends on the problem it tries to solve [29].

A new metaheuristic can also be developed by hybridizing several existing metaheuristics. This development can be found in many optimization studies. The example is as follows. Yulmaz, Altun, and Koklu [30] hybridized simulated annealing (SA) and GA to improve the training process in the artificial neural network system. Dursun and Ozger [31] combined GA and variable neighborhood search in the multi-depot vehicle routing problem in the aviation industry. Adamuthe and Kagwade [32] combined the harmony search (HS) and SA to optimize the virtual machine placement in the cloud system.

Proposing a new metaheuristic is still interesting and challenging even though a lot of metaheuristics already exist. There are a lot of strategies that have not been explored yet. Meanwhile, many approaches can be combined or modified to build a new metaheuristic. For example, the core of any swarm intelligence-based metaheuristics is the existence of the reference used in the guided search. This reference can be the global best solution [15], the local best solution [17], several best solutions [16], the best solution in the current population, a randomly chosen solution, and so on. Meanwhile, the behaviors of each agent in the guided search are also various. Some metaheuristics are enriched with a strategy to escape from the worse solution. Moreover, proposing a new metaheuristic should not be purposed to outperform the existing metaheuristic.

The objective of this study is to propose a new metaheuristic to solve the high-dimension problem. This metaheuristic is built based on swarm intelligence, so this metaheuristic consists of a set of agents where each agent represents the solution. This agent tries to find the optimal solution autonomously so that there is not any centralized coordination. Meanwhile, collective intelligence and interaction among agents are needed to speed up the search process.

This proposed metaheuristic is called a multiple interaction-dual leader optimizer (MIDLO). This name comes from the distinct strategy adopted in this work. In MIDLO, each solution carries out multiple uniform interaction-based guided searches in every iteration. Meanwhile, two references are needed to construct the guided search in every interaction. The global best solution becomes the first reference while a randomly chosen solution becomes the second one. MIDLO is also enriched with a random search which is carried out if the guided search fails to improve the quality of the solution after several tries.

Based on the previous explanation, the contribution of this work is as follows.

1) This work proposes a new swarm-based metaheuristic which is designed to solve high dimension problems.
2) This work proposes distinct mechanics where each solution carries out multiple interactions in every iteration and each interaction is followed by a guided search.
3) This work proposes the adoption of the global best solution and a randomly chosen solution as the references in the guided search.
4) This work proposes a novel mechanism in random search where the random search is not carried out in every iteration but carried out after a solution fails to improve several times and a strict acceptance-rejection policy is implemented on it.

This work uses the classic 23 functions to evaluate the performance of MIDLO. These functions are chosen because they have represented various problems. These functions consist of 13 high-dimension functions and 10 fixed-dimension functions. These functions also cover both unimodal problems and multimodal problems. Moreover, these functions have been chosen in many studies proposing a new metaheuristic.

This paper is organized as follows. Section one explains the background, problem statement, research objective, and contribution of this work. Section two draws back the development of shortcoming metaheuristics, especially the ones that are based on swarm intelligence. Section three presents the concept and formalization of the proposed metaheuristic. Section four presents the test carried out in this work to evaluate the performance of the proposed metaheuristic. Section five discusses the in-depth and more profound analysis of the result and its relationship with the theory. Section six summarizes the conclusion and future studies regarding this current work.

II. Related Works

The development of many metaheuristics can be traced back to the core strategy of any metaheuristic: exploration and exploitation. Exploration can be defined as an effort to find the area where the global optimal solution exists within the solution space [12]. Meanwhile, exploitation can be defined as an effort to find a better solution around the current solution [12]. Both activities are crucial to tackle any problem faced by any metaheuristic. Exploration plays important role in tackling the local optimal entrapment while exploitation plays important role in finding the quasi-optimal solution. Moreover, some metaheuristics are proven in finding the global optimal solution.

The development of metaheuristics can also be traced back from the evolution of the single solution-based metaheuristic, population-based metaheuristic, to the swarm intelligence-based metaheuristic. Single solution-based metaheuristic, such as tabu search or simulated annealing is proven simple

and has been used extensively in many studies in optimization. Then, the population-based metaheuristic was introduced to speed up the convergence as the optimization works become more complex. Moreover, the population-based metaheuristic is useful to diversify the solutions and provide better exploration rather than the single solution-based metaheuristic which usually depends on the neighborhood or local search. Genetic algorithm (GA) is an example of a population-based metaheuristic that is so popular and extensively used, combined, and modified.

TABLE I
LIST OF SHORTCOMING METAHEURISTICS AND THEIR MECHANICS

Func.	Metaheuristic	The Reference during Guided Search	Number of Movements during the Guided Search	Random Variate	Worse Solution Avoidance	Acceptance-rejection Strategy	Source
1	butterfly optimization algorithm (BOA)	the best solution in the current iteration	1	uniform	no	no	[20]
2	hybrid leader-based optimizer (HLBO)	the best solution in the current iteration, a randomly chosen solution, and the corresponding solution	1	uniform	yes	yes	[33]
3	cheetah optimizer (CO)	the global best solution, the best solution in the current iteration, and neighborhood solutions	1	uniform, normal, exponential, sinusoid	no	no	[21]
4	darts game optimizer (DGO)	the best solution in the current iteration and the worst solution in the current iteration.	1	uniform	no	no	[25]
5	election-based optimization algorithm (EBOA)	the best solution in the current iteration, the worst solution in the current iteration, and a randomly chosen solution	1	uniform	yes	yes	[34]
6	golden search optimizer (GSO)	the global best solution and local best solution	1	uniform, sinusoid	no	no	[35]
7	komodo mlipir algorithm (KMA)	the best solution in the current iteration and several best solutions in the current iteration	1	normal	partial	no	[19]
8	marine predator algorithm	local best solution	1	Brownian motion, levy flight	no	no	[17]
9	snow leopard optimization algorithm (SLOA)	the best solution in the current iteration and a randomly chosen solution	2	uniform	no	yes	[14]
10	mixed leader-based optimizer (MLBO)	the best solution in the current iteration and a randomly chosen solution	1	uniform	yes	yes	[36]
11	multi-leader optimizer (MLO)	one of several best solutions in the current iteration	1	uniform	no	yes	[37]
12	modified honey badger algorithm (MHBA)	the global best solution, adjacent solution	1	uniform, sinusoid	no	no	[22]
13	northern goshawk optimizer (NGO)	a randomly chosen solution	1	uniform	yes	yes	[18]
14	average and subtraction-based optimizer (ASBO)	the best solution in the current iteration and worst solution in the current iteration	3	uniform	partial	yes	[38]
15	puzzle optimization algorithm (POA)	a randomly chosen solution	1	uniform	yes	yes	[26]
16	random chosen leader-based optimizer (RSLBO)	a randomly chosen solution	1	uniform	yes	yes	[39]
17	ring toss game-based optimizer (RTGBO)	a randomly chosen solution among the best ten percent of the population	1	uniform	yes	yes	[27]
18	stochastic paint optimizer (SPO)	a randomly chosen solution from the best group, a randomly chosen solution from the mediocre group, a randomly chosen solution from the worst group, and two adjacent solutions	4	uniform	no	yes	[40]
19	pelican optimization algorithm (POA)	a random solution within the solution space	1	uniform	yes	yes	[13]
20	slime mold algorithm (SMA)	the best solution in the current iteration, the worst solution in the current iteration, and two randomly chosen solutions	1	uniform	no	no	[41]
21	this work	the global best solution and several randomly chosen solutions	multiple	uniform	yes	yes	-

Nowadays, many recent metaheuristics adopt swarm intelligence. In swarm intelligence, each solution in the population moves toward a reference autonomously through certain rules. This mechanism can be called as guided search. Particle swarm optimization is an example of an early metaheuristic that adopts swarm intelligence. Through many studies, swarm-based metaheuristic becomes more popular rather than single solution-based metaheuristics or evolution-based metaheuristics.

The popularity of swarm intelligence as a baseline for metaheuristics also comes from many methods that can be chosen during the guided search. Two parts can be explored in the guided search. The first part is the reference selection while the second part is the movement relative to this reference. Several alternatives that can be chosen as the reference are the global best solution, the best solution in the current iteration, the local best solution, several best solutions in the current iteration, and a randomly chosen solution. Meanwhile, several random strategies can be chosen, like a uniform random, normal distribution, levy flight, Brownian movement, sinusoid, and so on. Every strategy gives advantages and disadvantages.

Some metaheuristics apply a strict acceptance-rejection strategy while others do not. The strict acceptance-rejection strategy ensures the agent moves to the new solution only if it is better than the existing solution. This strategy gives the advantage that movement toward a worse solution is avoided. Meanwhile, the disadvantage is that a better solution after a worse solution may not be traced. The opposite condition can be found in metaheuristics that do not implement the strict acceptance-rejection strategy.

The list of shortcoming metaheuristics and their mechanics is presented in Table 1. The mechanics consist of the reference used during the guided search, several interactions during the guided search, random variate, the mechanism of avoiding the worse solution, and the acceptance-rejection strategy. The strategy and characteristics of MIDLO are presented in the last row to provide a clear position regarding this work.

Table 1 indicates that there are various methods proposed in every shortcoming metaheuristic. Meanwhile, most metaheuristics implement a single movement only during the guided search in every iteration. Some metaheuristics implement multiple movements but the number of movements during the guided search is static. Based on this circumstance, a room to explore a new metaheuristic where a solution carries out multiple movements during the guided search in every iteration and the number of movements can be adjusted manually is still available.

III. Proposed Model

The basic concept of MIDLO comes from two novel approaches: multiple interactions and dual leadership. In the multiple interaction approach, each solution interacts with multiple solutions in every iteration. In the dual leader approach, there are two references used in the guided search. The first reference is the best solution. The second reference is the randomly chosen solution. In MIDLO, there are two searches in every iteration. The first one is the guided search. The second search is the random search. The guided search is mandatory while the random search is optional. The number of lives is also introduced in MIDLO. The number of lives represents the opportunity for a solution to failing to improve in the guided search without carrying out the random search. In the beginning, the number of lives of each solution is set to maximum. Each time a solution fails to improve, then its number of lives decrements. When its number of lives reaches zero, the solution carries out the random search by generating a new solution randomly within the space. Then, its number of lives is reset to the maximum.

MIDLO consists of two phases: initialization and iteration. In the initialization phase, all solutions are randomly generated within the solution space. In the iteration phase, all solutions carried out multiple interactions. In every interaction, each solution chooses two guided search candidates. The first candidate is generated based on the guided search of the corresponding solution toward the best solution. The second candidate is generated based on the guided search of the corresponding solution relative to the randomly chosen solution. If the chosen solution is better than the corresponding solution, then the second candidate is generated based on the guided search of the corresponding solution toward the chosen solution. Otherwise, the second candidate is generated based on the guided search of the chosen solution toward the corresponding solution. The best candidate between the first candidate and the second candidate is chosen as the final candidate. If the final candidate is better than the corresponding solution use the final candidate as its new solution. Otherwise, the number of lives of the corresponding solution decrements. When the number of lives of a solution becomes zero, then this solution carries out the random search and its number of lives is reset to the maximum level. In the random search, a candidate is randomly generated within the solution space. If this candidate is better than the current solution, then the corresponding solution uses this candidate as its new solution. Each time a solution updates its value, this new solution is used to update the best solution. If the proposed solution is better than the best solution, then the best solution replaces its current solution with this new one.

The concept of MIDLO is then translated into the algorithm and mathematical model. The annotations used in this work are presented below. The algorithm of MIDLO is presented in algorithm 1 using pseudocode. Equation (1) to (9) represents the mathematical model.

b_l, b_u	lower boundary, the upper boundary
n_l	number of lives
n_{lmax}	maximum number of lives
n_{int}	number of interactions
r_1	real random number between 0 and 1
r_2	integer random number between 1 and 2
s	solution
S	set of solutions (population)
s_{sel}	chosen solution
s_{c1}	first candidate
s_{c2}	second candidate
s_c	final candidate
s_{sel}	chosen solution
t	iteration
t_{max}	maximum iteration
U	uniform random

Below is the explanation of (1) to (9). Equation (1) indicates that in the beginning, the solution is randomly chosen within the solution space. Equation (2) indicates that the corresponding solution replaces the best solution only if this corresponding solution is better than the best solution. Equation (3) indicates that a solution is randomly chosen among the population to become the second leader. Equation (4) indicates that the first candidate is generated based on the guided search toward the best solution. Equation (5) indicates that there are two options related to the second guided search. The first option is the movement of the corresponding solution toward the best solution. The second option is the movement of the best solution toward the corresponding solution. Equation (6) indicates that the candidate whose fitness score is better to become the final candidate. Equation (7) indicates that the final candidate replaces the current solution only if this candidate is better than the current solution. Equation (8) indicates that the number of lives belonging to the corresponding solution decrements if the guided search fails to improve the current solution. Equation (9) indicates that in the random search, a new candidate is randomly generated within the problem space.

algorithm 1: MIDLO

output: s_{best}
begin
  for all S
    initialize s_i using (1)
    $n_{l,i} = n_{lmax}$
    update s_{best} using (2)
  end for
  t =1
  while $t <= t_{max}$
    for all S
      for k=1 **to** n_{int}
        select $s_{sel,i}$ using (3)
        calculate $s_{c1,i}$ using (4)
        calculate $s_{c2,i}$ using (5)
        select $s_{c,i}$ using (6)
        update s_i using (7)
        update s_{best} using (2)
        update $n_{l,i}$ using (8)
        if $n_{l,i}$ = 0 **then**
          $n_{l,i} = n_{lmax}$
          calculate $s_{c,i}$ using (9)
          update s_i using (7)
        end if
      end for
    end for
    $t = t + 1$
  end while
end

$$s = U(b_l, b_u) \tag{1}$$

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

$$s_{sel} = U(S) \tag{3}$$

$$s_{c1} = s + r_1.(s_{best} - r_2.s) \tag{4}$$

$$s_{c2} = \begin{cases} s + r_1.(s_{sel} - r_2.s), f(s_{sel}) < f(s) \\ s_{sel} + r_1.(s - r_2.s_{sel}), else \end{cases} \tag{5}$$

$$s_c = \begin{cases} s_{c1}, f(s_{c1}) < f(s_{c2}) \\ s_{c2}, else \end{cases} \tag{6}$$

$$s' = \begin{cases} s_c, f(s_c) < f(s) \\ s, else \end{cases} \tag{7}$$

$$n_l{}' = \begin{cases} n_l - 1, s' \geq s \\ n_l, else \end{cases} \tag{8}$$

$$s_c = U(b_l, b_u) \tag{9}$$

Based on algorithm 1, the complexity of MIDLO can be presented as $O(t_{max}.n(S).n_{int})$. It means that the complexity of MIDLO is proportional to the maximum iteration, population size, or the number of interactions.

IV. Simulation

MIDLO is then challenged to solve the optimization problem through simulation. In this work, the 23 classic functions represent the optimization problems. These functions cover all kinds of problems. They consist of seven high-dimension unimodal functions (functions 1 to 7), six high-dimension multimodal functions (functions 8 to 13), and ten fixed-size multimodal functions (functions 14 to 23). These functions cover problems with narrow to large problem spaces. These functions also cover problems that are simple and easy to solve to problems that have many locally optimal solutions.

In this work, MIDLO is competed with five other metaheuristics: PSO, stochastic marine predator algorithm with multiple candidates (SMPA-MC), GSO, ASBO, and guided pelican algorithm (GPA). All these metaheuristics are based on swarm intelligence. PSO represents the early metaheuristic, and it has been implemented in many optimization studies. The other metaheuristics represent the shortcoming metaheuristics that are proven better than PSO but still less popular.

Several parameters are set as follows. The population size is 10 and the maximum iteration is 50 which represents the optimization process with a low population and low iteration. This maximum iteration is in contrast with many works proposing a new metaheuristic where the maximum iteration was set to 1000, such as in the first introduction of NGO [18] or POA [13]. The dimension is set to 40. In MIDLO, the maximum number of lives and number of iterations is set to 5. In PSO, all weights are set as 0.3. In SMPA-MC, the fishing aggregate devices are set to 0.2 and the number of candidates is 5. The simulation result is presented in Table 2. Meanwhile, the comparison between MIDLO and the sparing metaheuristics based on the functions group is presented in Table 3.

TABLE II
BENCHMARK SIMULATION RESULT

F	Fitness Score						Better Than
	PSO [15]	SMPA-MC [42]	GSO [35]	ASBO [38]	GPA [43]	MIDLO	
1	$2.452x10^{4}$	$5.643x10^{3}$	$1.518x10^{4}$	$2.153x10^{-8}$	$6.624x10^{1}$	$1.294x10^{-85}$	PSO, SMPA-MC, GSO, ASBO, GPA
2	$9.156x10^{44}$	0.000	$8.911x10^{50}$	0.000	0.000	0.000	PSO, GSO
3	$6.768x10^{4}$	$9.501x10^{3}$	$3.333x10^{4}$	$6.144x10^{1}$	$4.089x10^{3}$	$1.324x10^{-37}$	PSO, SMPA-MC, GSO, ASBO, GPA
4	$5.019x10^{1}$	$2.459x10^{1}$	$3.969x10^{1}$	$1.311x10^{-3}$	$1.756x10^{1}$	$1.098x10^{-32}$	PSO, SMPA-MC, GSO, ASBO, GPA
5	$2.673x10^{7}$	$1.711x10^{6}$	$1.219x10^{7}$	$3.873x10^{1}$	$3.891x10^{3}$	$3.891x10^{1}$	PSO, SMPA-MC, GSO, GPA
6	$2.560x10^{4}$	$5.158x10^{3}$	$1.478x10^{4}$	4.726	$6.532x10^{1}$	7.908	PSO, SMPA-MC, GSO, GPA
7	$1.564x10^{1}$	1.396	8.152	$2.518x10^{-2}$	$2.040x10^{-1}$	$1.667x10^{-3}$	PSO, SMPA-MC, GSO, ASBO, GPA
8	$-3.427x10^{3}$	$-4.748x10^{3}$	$-3.944x10^{3}$	$-4.197x10^{3}$	$-9.295x10^{3}$	$-2.068x10^{3}$	-
9	$3.845x10^{2}$	$2.662x10^{2}$	$2.586x10^{2}$	$1.270x10^{1}$	$1.229x10^{2}$	0.000	PSO, SMPA-MC, GSO, ASBO, GPA
10	$1.741x10^{1}$	$1.228x10^{1}$	$1.834x10^{1}$	2.220	3.849	$3.997x10^{-15}$	PSO, SMPA-MC, GSO, ASBO, GPA
11	$2.274x10^{2}$	$4.864x10^{1}$	$1.426x10^{2}$	$1.563x10^{-1}$	1.6586	0.000	PSO, SMPA-MC, GSO, ASBO, GPA
12	$2.402x10^{7}$	$9.604x10^{3}$	$5.346x10^{6}$	$1.048x10^{-2}$	8.045	$9.871x10^{-1}$	PSO, SMPA-MC, GSO, GPA
13	$7.506x10^{7}$	$5.972x10^{5}$	$3.351x10^{7}$	7.433	$2.551x10^{1}$	3.026	PSO, SMPA-MC, GSO, ASBO, GPA
14	$1.431x10^{1}$	2.015	9.100	1.901	1.178	6.291	PSO, GSO
15	$4.838x10^{-2}$	$1.352x10^{-3}$	$1.248x10^{-2}$	$9.440x10^{-2}$	$5.253x10^{-3}$	$1.309x10^{-3}$	PSO, SMPA-MC, GSO, ASBO, GPA
16	$-9.901x10^{-1}$	-1.031	-1.032	$-2.238x10^{-2}$	-1.032	-1.027	PSO, GPA
17	$5.932x10^{-1}$	$3.981x10^{-1}$	$3.981x10^{-1}$	$6.394x10^{-1}$	$3.981x10^{-1}$	1.031	-
18	$1.237x10^{1}$	3.001	3.092	3.000	3.000	$1.371x10^{1}$	-
19	$-2.410x10^{-1}$	$-4.954x10^{-2}$	$-4.210x10^{-2}$	$-4.954x10^{-2}$	$-4.954x10^{-2}$	$-4.954x10^{-2}$	PSO, GSO
20	-2.387	-3.231	-2.993	$-8.811x10^{-1}$	-3.278	-2.691	PSO, ASBO
21	-3.344	-7.486	-5.438	-6.393	-7.290	-3.924	PSO
22	-2.517	-8.760	-7.067	-6.734	-7.849	-3.837	PSO
23	-3.020	-9.013	-5.705	-5.856	-9.727	-4.033	PSO

Table 2 indicates that MIDLO performs as a competitive metaheuristic by achieving the acceptable solution in low population and low iteration circumstances. MIDLO is also superior compared to other sparing metaheuristics, especially in solving high-dimension problems. It can find the best result in solving nine functions. Meanwhile, its performance is not superior in solving the fixed dimension problem. MIDLO also performs inferior in solving three functions by producing the worst solution in solving these functions.

TABLE III
GROUP BASED COMPARISON

Cluster	PSO [15]	SMPA-MC [42]	GSO [35]	ASBO [38]	GPA [43]
1	7	6	7	4	6
2	5	5	5	4	5
3	8	1	3	1	2
Total	20	12	15	9	13

Table 3 strengthens the competitiveness of MIDLO compared with the five metaheuristics. MIDLO is better than PSO, SMPA-MC, GSO, ASBO, and GPA in solving 20, 12, 15, 9, and 13 functions respectively. PSO becomes the metaheuristic that is easiest to outperform while ASBO becomes the most difficult one. Table 2 also indicates the superiority of MIDLO is mostly in solving the high-dimension functions, whether they are in the first or second groups. On the other hand, MIDLO is less superior to SMPA-MC, GSO, ASBO, and GPA in solving the fixed-dimension multimodal functions. MIDLO is superior only to PSO in solving fixed-dimension multimodal functions.

The second test is to evaluate the performance of MIDLO in solving the 23 functions in the high iteration scenario. In this test, the maximum iteration is set to 100 and 150. Theoretically, higher maximum iteration may improve the performance of any metaheuristic. The result is presented in Table 4.

Table 4 indicates that there are two possible responses due to the increase in maximum iteration. First, the increase in maximum iteration decreases the fitness score significantly. Second, the increase of maximum iteration does not affect the fitness score, or the effect is less significant. The first circumstance can be found in five functions. Meanwhile, the second circumstance can be found in eighteen other functions. Among these five functions, four functions are high dimension unimodal functions, and one function is fixed dimension multimodal function. The second circumstance can be seen as the convergence has been achieved in the low maximum iteration. It means that the higher maximum iteration does not change the result.

TABLE IV
CONVERGENCE RESULT

Function	Average Fitness Score	
	$t_{max} = 100$	$t_{max} = 150$
1	$6.333x10^{-177}$	$7.340x10^{-266}$
2	0.000	0.000
3	$2.665x10^{-86}$	$1.133x10^{-130}$
4	$3.506x10^{-67}$	$2.958x10^{-101}$
5	$3.723x10^{1}$	$3.890x10^{1}$
6	8.061	7.961
7	$1.138x10^{-3}$	$3.482x10^{-4}$
8	$-2.171x10^{3}$	$-2.293x10^{3}$
9	0.000	0.000
10	$3.997x10^{-15}$	$3.997x10^{-15}$
11	0.000	0.000
12	$9.036x10^{-1}$	$9.420x10^{-1}$
13	3.020	2.998
14	7.197	7.358
15	$1.423x10^{-3}$	$4.573x10^{-4}$
16	-1.018	-1.015
17	$8.719x10^{-1}$	$6.532x10^{-1}$
18	$1.039x10^{1}$	8.984
19	$-4.954x10^{-2}$	$-4.954x10^{-2}$
20	-2.709	-2.673
21	-3.979	-3.866
22	-4.194	-4.691
23	-4.068	-4.1939

The third test is carried out to evaluate the relation between the number of interactions and the fitness score. In this test, the number of interactions is set to 2 and 8. The first value represents the limited interaction carried out by each solution in every iteration. Meanwhile, the second value represents

intensive interaction carried out by each solution in every iteration. Theoretically, more interaction may produce better performance, i.e., a lower fitness score. The result can be seen in Table 5.

TABLE V
NUMBER OF INTERACTION SIMULATION RESULT

Function	Average Fitness Score	
	$n_{int} = 2$	$n_{int} = 8$
1	2.636×10^{-31}	2.429×10^{-144}
2	0.000	0.000
3	5.031×10^{-8}	5.598×10^{-70}
4	4.260×10^{-13}	1.043×10^{-52}
5	3.890×10^{1}	3.892×10^{1}
6	7.694	8.115
7	5.088×10^{-3}	7.963×10^{-4}
8	-1.799×10^{3}	-2.243×10^{3}
9	0.000	0.000
10	3.997×10^{-15}	3.658×10^{-15}
11	0.000	0.000
12	8.907×10^{-1}	9.289×10^{-1}
13	3.065	3.026
14	9.199	6.305
15	9.732×10^{-4}	8.859×10^{-4}
16	-1.024	-1.024
17	9.996×10^{-1}	4.403×10^{-1}
18	1.221×10^{1}	7.012
19	-4.954×10^{-2}	-4.954×10^{-2}
20	-2.477	-2.710
21	-3.387	-3.753
22	-3.462	-4.580
23	-3.726	-3.838

Table 5 indicates that the increase in the number of interactions is less significant to the improvement of the algorithm's performance. Performance improvement due to the increase in the number of interactions occurs only in four functions (f_1, f_3, f_4, and f_7). In these four functions, performance improvement is very significant. These four functions are unimodal. Meanwhile, in three functions (f_2, f_9, and f_{11}), stagnation occurs because the global optimal solution has been achieved in the low number of iterations circumstance. This result indicates that the increase in interaction is less significant in the multimodal functions.

The fourth test is carried out to evaluate the relation between the maximum number of lives and the algorithm's performance. Theoretically, a higher maximum number of lives gives more time for the solution still in the current solution although it fails to improve before moving randomly within the solution space. In this work, the maximum number of lives is set to 2 and 8. The first value means that the solution is forced to leave the current solution quickly when it fails to improve. Meanwhile, the second value means that the solution can stay on the current solution longer although it fails to improve. The result is presented in Table 6.

Table 6 indicates that there are three circumstances due to the relation between the maximum number of lives and the algorithm's performance. In most functions, the increase in the maximum number of lives does not affect the algorithm's performance. In three functions (f_2, f_9, and f_{11}), this circumstance occurs because the global optimal solution has been found. In five functions (f_3, f_4, f_7, f_{15}, and f_{17}), the increase in the maximum number of lives improves the algorithm's performance. In two functions (f_1 and f_{18}), the increase in the maximum number of lives worsens the algorithm's performance.

TABLE VI
NUMBER OF LIFES SIMULATION RESULTS

Function	Average Fitness Score	
	$n_{life} = 2$	$n_{life} = 8$
1	4.895×10^{-87}	4.005×10^{-85}
2	0.000	0.000
3	2.002×10^{-40}	1.547×10^{-41}
4	1.260×10^{-32}	7.276×10^{-33}
5	3.892×10^{1}	3.890×10^{1}
6	7.904	7.836
7	1.644×10^{-3}	1.194×10^{-3}
8	-2.190×10^{3}	-2.097×10^{3}
9	0.000	0.000
10	3.997×10^{-15}	3.997×10^{-15}
11	0.000	0.000
12	9.616×10^{-1}	9.035×10^{-1}
13	3.028	3.019
14	7.195	7.720
15	7.499×10^{-4}	3.921×10^{-4}
16	-1.024	-1.027
17	1.084	7.267×10^{-1}
18	6.623	9.666
19	-4.954×10^{-2}	-4.954×10^{-2}
20	-2.574	-2.640
21	-3.565	-3.351
22	-4.667	-4.405
23	-4.211	-4.551

V. Discussion

In this section, the in-depth and more profound analysis of the test result will be discussed. This discussion consists of three parts. The first part is the discussion regarding the performance of MIDLO and the comparison with the sparing metaheuristics. The second part is the discussion regarding hyperparameters. The third part is the limitation of this work and metaheuristic so that it will be useful as a baseline for future development.

Overall, MIDLO is a competitive metaheuristic, and it has met the main objective of solving high-dimension problems. Table 2 indicates that MIDLO is superior to the four sparing metaheuristics (PSO, SMPA-MC, GSO, and GPA) but still inferior to ASBO. Table 2 also indicates that multiple interaction and dual leader approaches adopted in MIDLO makes this metaheuristic superior to all sparing metaheuristics, especially in solving high-dimension problems, both unimodal and multimodal.

The competitiveness of MIDLO can be drawn back to its distinct mechanism compared to the mechanism adopted by the sparing metaheuristics. In PSO, each solution carries out one-time interaction with the local best solution and the global best solution [15]. In SMPA-MC, each solution carries out two sequential interactions [42]. The first interaction is with the local best solution and the second interaction is with two randomly chosen solutions [42]. In GSO, each solution carries out one-time interaction with the local best solution and the global best solution [35]. In ASBO, each solution carries out two-time interaction with the best solution and worst solution chosen in every iteration [38]. In GPA, the guided search referred only to the global best but with multiple candidates along the way toward the reference [43]. Choosing the global best solution and the randomly chosen solution as references for the guided search is useful, especially in solving the high dimension problems, both unimodal and multimodal.

The test result indicates that MIDLO performs well with limited resources or computation. The result has shown that

the acceptable solution, whether quasi-optimal or global optimal can be achieved in low population and low iteration circumstances. Moreover, this result can be achieved in a low number of interactions which means the computational process can be reduced. The test result has shown that a higher maximum iteration or the number of interactions improves the algorithm's performance significantly, especially in some high-dimension unimodal functions. But there is a question of whether the superior result regarding these functions is necessary. Besides, Table 1 has indicated that its superiority has been achieved in low population size and low maximum iteration. After that, increasing the population size or maximum iteration may not be necessary for the result but increases the computational process.

Based on the sensitivity analysis, the maximum iteration and number of interactions are more sensitive than the maximum number of lives. The increase in maximum iteration and number of interactions improves the algorithm's performance. But after passing a certain value, the increase of these two parameters does not improve the result significantly. Meanwhile, the response of the algorithm due to the increase in the maximum number of lives is various. It depends on the problems it tries to solve. It means that the increase in the maximum number of lives does not guarantee performance improvement.

There are several limitations regarding this work and the proposed metaheuristic. First, MIDLO is still less competitive in solving the fixed dimension multimodal functions, except compared with PSO. Although the dimension is low, the area where the global optimal solution exists is difficult to find. This circumstance can be used as a baseline to improve the MIDLO. MIDLO adopts a strict acceptance-rejection approach in both guided search and random search. The improvisation can be carried out in many ways, for example by modifying the acceptance-rejection strategy or implementing another random search. Second, MIDLO has not been implemented to solve practical optimization problems, whether these problems are classics, such as the mechanical problem or optimal flow problem, or any other problems. Future research can be carried out by implementing MIDLO to solve any practical problems, whether numerical or combinatorial.

VI. Conclusion

This study has demonstrated that the distinct mechanics implemented in MIDLO is effective in solving the high dimension problems. These mechanics are especially on the multiple interactions and choosing both the global best solution and randomly chosen solution as references in the guided search. Through the test result, it is shown that MIDLO is superior to the sparing metaheuristics, especially in solving the high-dimension functions. MIDLO is better than PSO, SMPA-MC, GSO, ASBO, and GPA in solving 20, 12, 15, 9, and 13 functions respectively. Meanwhile, MIDLO is not superior in solving fixed-dimension multimodal functions.

This work has also suggested several areas that can be used in future research. The improvement or modification in the random search is important to make MIDLO superior in solving the fixed-dimension multimodal functions. Moreover, future research can also be conducted by using MIDLO to solve many kinds of practical optimization problems.

References

[1] Y. Hou, B. Liu, L. Dang, W. He, and W. Gu, "A local search-based metaheuristic algorithm framework for the school bus routing problem," *Engineering Letters*, vol. 30, no. 1, pp. 17-28, 2022.

[2] G. Chen, P. Qiu, X. Hu, F. Long, and H. Long, "Research of short-term wind speed forecasting based on the hybrid model of optimized quadratic decomposition and the improved monarch butterfly," *Engineering Letters*, vol. 30, no. 1, pp. 73-90, 2022.

[3] A. Alejo-Reyes, E. Cuevas, A. Rodriguez, A. Mendoza, and E. Olivares-Benitez, "An improved grey wolf optimizer for a supplier selection and order quantity allocation problem," *Mathematics*, vol. 8, ID: 1457, pp. 1-24, 2020.

[4] B. -G. Risi, F. Riganti-Fulginei, and A. Laudani, "Modern techniques for the optimal power fow problem: state of the art," *Energies*, vol. 15, ID: 6387, pp. 1-20, 2022.

[5] S. J. Sajadi and A. Ahmadi, "An integrated optimization model and metaheuristics for assortment planning, shelf space allocation, and inventory management of perishable products: a real application," *PLOS ONE*, vol. 17, no. 3, ID: e0264186, pp. 1-30, 2022.

[6] F. F. Korkmaz, M. Subran, and A. R. Yildiz, "Optimal design of aerospace structures using recent meta-heuristic algorithms," *Materials Testing*, vol. 63, no. 11, pp. 1025-1031, 2021.

[7] P. D. Kusuma and R. A. Nugrahaeni, "Collaborative flow-shop scheduling using simulated annealing and first price sealed bid auction to minimize total cost and make-span," *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 1, pp. 530-539, 2022.

[8] A. K. Mishra and D. Shrivastava, "A discrete Jaya algorithm for permutation flow-shop scheduling problem," *International Journal of Industrial Engineering Computations*, vol. 11, pp. 415-428, 2020.

[9] G. Li, Y. Tian, and M. Xie, "Improved whale optimization algorithm and low-energy consumption design of circuit breaker," *Mathematical Problems in Engineering*, vol. 2022, ID: 8349258, pp. 1-12, 2022.

[10] P. D. Kusuma, R. A. Nugrahaeni, and D. Adiputra, "Coordinated ambulance routing problem for COVID-19 by using cloud-theory-based simulated annealing to minimize number of unserved patients and total travel distance," *Engineering Letters*, vol. 30, no. 3, pp. 955-963, 2022.

[11] A. I. Alhujaylan and M. I. Hosny, "A GRASP-based solution construction approach for the multi-vehicle profitable pickup and delivery problem," *International Journal of Advanced Computer Science and Application*, vol. 10, no. 4, pp. 111-120, 2019.

[12] 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.

[13] 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.

[14] P. Coufal, S. Hubalovsky, M. Hubalovska, and Z. Balogh, "Snow leopard optimization algorithm: a new nature-based optimization algorithm for solving optimization problems," *Mathematics*, vol. 9, ID. 2832, pp. 1-26, 2021.

[15] D. Freitas, L.G. Lopes, and F. Morgado-Dias, "Particle swarm optimization: a historical review up to the current developments," *Entropy*, vol. 22, ID. 362, pp. 1-36, 2020.

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

[17] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine predators algorithm: a nature-inspired metaheuristic," *Experts Systems with Applications*, vol. 152, ID. 113377, pp. 1-48, 2020.

[18] 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.

[19] S. Suyanto, A. A. Ariyanto, and A. F. Ariyanto, "Komodo mlipir algorithm," *Applied Soft Computing*, vol. 114, ID. 108043, pp. 1-17, 2022.

[20] S. Arora and S. Singh, "Butterfly optimization algorithm: a novel approach for global optimization," *Soft Computing*, vol. 23, pp. 715-734, 2019.

[21] M. A. Akbari, M. Zare, R. Azizipanah-abarghooee, S. Mirjalili, and M. Deriche, "The cheetah optimizer: a nature-inspired metaheuristic algorithm for large-scale optimization problems," *Scientific Reports*, vol. 12, ID. 10953, pp. 1-20, 2022.

[22] 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.
[23] M. Suman, V. P. Sakhtivel, 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.
[24] A. M. Fathollahi-Fard, M. Hajiaghaei-Keshteli, and R. Tavakkoli-Moghaddam, "Red deer algorithm (RDA): a new nature-inspired meta-heuristic," *Soft Computing*, vol. 24, pp. 14637-14665, 2020.
[25] 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.
[26] F. A. Zeidabadi and M. Dehghani, "POA: puzzle optimization algorithm," *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 1, pp. 273-281, 2022.
[27] S. A. Doumari, H. Givi, M. Dehghani, and O. P. Malik, "Ring toss game-based optimization algorithm for solving various optimization problems," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 3, pp. 545-554, 2021.
[28] M. Dehghani, Z. Montazeri, O. P. Malik, H. Givi, and J. M. Guerrero, "Shell game optimization: a novel game-based algorithm," *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 3, pp. 246-255, 2020.
[29] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, pp. 67-82, 1997.
[30] O. Yulmaz, A. A. Altun, and M. Koklu, "Optimizing the learning process of multi-layer perceptrons using a hybrid algorithm based on MVO and SA," *International Journal of Industrial Engineering Computations*, vol. 13, no. 4, pp. 617-640, 2022.
[31] O. O. Dursun and A. Ozger, "Multi-depot heterogeneous fleet vehicle routing problem with time windows: airline and roadway integrated routing," *International Journal of Industrial Engineering Computations*, vol. 13, vol. 3, no. 435-456, 2022.
[32] A. C. Adamuthe and S. M. Kagwade, "Hybrid and adaptive harmony search algorithm for optimizing energy efficiency in VMP problem in cloud environment," *Decision Science Letters*, vol. 11, no. 2, pp. 113-126, 2022.
[33] M. Dehghani and P. Trojovsky, "Hybrid leader based optimization: a new stochastic optimization algorithm for solving optimization applications," *Scientific Reports*, vol. 12, ID. 5549, pp. 1-16, 2022.
[34] P. Trojovsky and M. Dehghani, "A new optimization algorithm based on mimicking the voting process for leader selection," *PeerJ Computer Science*, vol. 8, ID. e976, pp. 1-40, 2022.
[35] 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.
[36] F. A. Zeidabadi, S. A. Doumari, M. Dehghani, and O. P. Malik, "MLBO: mixed leader based optimizer for solving optimization problems," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 4, pp. pp. 472-479, 2021.
[37] M. Dehghani, Z. Montazeri, A. Dehghani, R. A. Ramirez-Mendoza, H. Samet, J. M. Guerrero, and G. Dhiman, "MLO: multi leader optimizer," *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 6, pp. 364-373, 2020.
[38] 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.
[39] F. A. Zeidabadi, M. Dehghani, and O. P. Malik, "RLSBO: random chosen leader based optimizer," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 5, pp. 529-538, 2021.
[40] 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.
[41] 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.
[42] P. D. Kusuma and R. A. Nugrahaeini, "Stochastic marine predator algorithm with multiple candidates," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 4, pp. 241-251, 2022.
[43] 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.

Purba Daru Kusuma is an assistant professor in computer engineering in Telkom University, Indonesia. He received his bachelor 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 currently becomes a member of IAENG.

Anggunmeka Luhur Prasasti received her Engineering degree in Telecommunication Engineering from Telkom Institute of Technology (Telkom University) in 2012. She obtained her master's degree from School of Electrical Engineering and Informatics, Bandung Institute of Technology, Indonesia. Her current research interests are artificial intelligence in signal processing and machine learning implementation to detect any kind of diseases, forecast conditions such as rain possibility, crowd density in an area, and also blockchain technology. She's used to compete in several international IT competitions such as Microsoft Imagine Cup and Asia Pacific ICT Awards, to name a few. Now, she is a Computer Engineering lecturer in Telkom University, Indonesia. She could be reached through anggunmeka@telkomuniversity.ac.id (orcid id: 0000-0001-6197-4157).