Double direction optimization: a new metaheuristic that performs exploitation and exploration simultaneously

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

Article Info

Article history:

Received Feb 10, 2025
Revised Feb 11, 2026
Accepted Mar 5, 2026

Keywords:

Economic emission dispatch
Engineering design
Metaheuristic
Optimization
Power system

ABSTRACT

This research constructs a novel method called double direction optimization (DDO). DDO is constructed based on swarm intelligence (SI) approach and it does not use any metaphor. As its name suggests, it employs a novel algorithm by performing exploitation and exploration simultaneously which is transformed into two sequential searches. In the 1st search, the motion toward the highest quality agent is combined with the motion toward a randomly taken higher quality agent. In the 2nd search, the motion toward the finest entity is combined with the motion relative to a randomly taken agent. In this work, the efficacy of the DDO is assessed using three use cases: 23 functions, four engineering problems, and an economic emission dispatch (EED) problem. In this assessment, there are five metaheuristics that become the benchmark: crayfish optimization algorithm (COA), hiking optimization (HO), osprey optimization algorithm (OOA), carpet weaver optimization (CWO), and dollmaker optimization algorithm (DOA). The result indicates the supremacy of DDO in high dimension functions and competitiveness of DDO in fixed dimension multimodal functions, four engineering problems, and the EED problem.

Corresponding Author:

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

1. INTRODUCTION

Optimization is a popular or well-known study that can be found in many fields, especially engineering. The basic nature of optimization is finding the optimal solution, whether it is the maximum one or minimum one, among the set of available solutions or can be called solution space. This solution space can be continuous or discrete. Many optimization problems in power systems, such as economic dispatch [1], optimal power flow [2], or unit commitment [3] problems are the example of optimization problems with continuous solution space. Many optimization problems in operation research, such as flow-shop scheduling [4], vehicle routing [5], and order allocation [6] problems are considered as optimization problems with discrete solution space. Metaheuristics has become one popular tool that is extensively used in many optimization studies whether their solution space is continuous or discrete.

Many new algorithms are already recently. Some of them are metaphor-imitated algorithms while others are metaphor-free algorithms. The example of the 1st ones are deep sleep optimization (DSO) [7], hiking optimization (HO) [8], prairie dog optimization (PDO) [9], crayfish optimization algorithm (COA) [10], osprey optimization algorithm (OOA) [11], lyrebird optimization algorithm (LOA) [12], carpet weaver optimization (CWO) [13], dollmaker optimization algorithm (DOA) [14], sales training-based optimization (STBO) [15], addax optimization algorithm (AOA) [16], clouded leopard optimization (CLO) [17], golden jackal optimization (GJO) [18], white shark optimization (WSO) [19], elk herd optimization (EHO) [20], Komodo mlipir algorithm (KMA) [21], giant armadillo optimization (GAO) [22], and pufferfish optimization algorithm (POA) [23]. The examples of the 2nd ones are golden search optimization (GSO) [24], average subtraction-based optimization (ASBO) [25], fully informed search algorithm (FISA) [26], and subtraction average-based optimization (SABO) [27].

Unfortunately, there are two concerns or critiques regarding the extensive development of metaheuristics. The 1st concern is the usage of metaphors. The 2nd concern is the trend on outperforming the existing methods. Many metaheuristic optimizations use metaphors, and the most popular metaphor is animal behavior, especially during hunting, searching for food, or mating. Ironically, the existence of the metaphor is often used or promoted as the novel approach [28]. Ironically, by investigating their formal method, especially based on the algorithm through pseudocode and the mathematical model, overall, this metaphor is the mechanics on the searching methods, whether they are directed search toward or away from certain target or random search. Besides, some conditions are the mechanism to implement the multiple search mechanism whether these multiple searches are carried out sequentially, conditionally, or both. The trend of competition between the proposed metaheuristic optimization and the existing ones becomes a concern too. Many new metaheuristics are developed to beat the existing ones, and the measurement of the success is on the outperformance of the proposed one compared to the existing ones.

These two concerns may push the development of metaheuristic in a wrong way. Rather than finding and constructing new ways to find the optimal solution, many recent studies tend to promote new metaphors and outperform the existing ones. Many old school metaheuristics, such as genetic algorithm (GA), simulated annealing (SA), and particle swarm optimization (PSO) are still utilized in many works, such as to solve electronic vehicle optimization [29], path planning [30], water allocation problem [31], task offloading in cloud computing [32], and credit rating model [33]. The relative new metaheuristics such as grey wolf optimization (GWO) [34] or marine predator algorithm (MPA) [35] are also still popular, such as to solve task scheduling in cloud system [36], COVID-19 forecasting [37], and power flow [38]. Meanwhile, all these existing metaheuristics have been taken as benchmarks for new metaheuristics and have been beaten many times. This contradiction means that the unmatched performance is not everything. Besides, the simplicity and novel approach of these existing algorithms play a significant role in becoming popular.

This work is aimed at proposing a new method called double direction optimization (DDO). It is constructed using SI framework so that it comprises a set of autonomous entities. DDO is designed as a metaphor-free algorithm so that its novel approach in performing both exploration and exploitation in a simultaneous manner can be clearly investigated and recognized. DDO is then implemented to solve three use cases. Based on this explanation, the contributions of this work are listed: i) it promotes a novel metaphor-free algorithm called as DDO, ii) DDO has a novel searching method by performing exploitation and exploration simultaneously in equal proportion, iii) the efficacy of DDO is assessed by challenging it to solve 23 functions, 4 engineering problems, and economic emission dispatch (EED) problem in Indonesia, and iv) the quality of DDO is benchmarked using five new metaheuristics.

The structure of the rest of this paper is as provided below. Section 2 investigates and summarizes the recent development or studies that proposed a new method. Section 3 describes the model of DDO including the concept and formalization. Section 4 provides the result and discussion of DDO. Section 5 provides the conclusion and proposal for studies in the future.

2. LITERATURE REVIEW

Metaheuristics have been known as an effective optimization tool since long decades ago. Its popularity comes from its simplicity and effectiveness in solving problems with various circumstances. It started with the individual search era, where algorithms such as tabu search, SA, and variable neighborhood search gained their popularity. Even today, these algorithms are still widely used whether using their basic form or their modification ones. Then, the population-based algorithms rose to provide improvement since they achieve faster performance and give better chance to avoid local optimal by spreading the solution within space. In this era, several algorithms like GA or evolutionary programming (EP) rose their popularity. The evolution continued to the emerging of swarm intelligence (SI) era. Still constructed using population-based algorithm, SI provided breakthrough by activating all members of the population to become autonomous agents that search both individually and collectively in every iteration. PSO was known as the early version of SI-based algorithm which is also still popular until today. Since then, many metaheuristics were developed based on SI framework.

Then, it comes nature inspired optimization (NIO) era. This era, many metaheuristics adopt or imitate the behavior from the nature. Animal behavior becomes the most popular inspiration although plantation, material, or even human behavior are also adopted as metaphors. Many nature-inspired or metaphor-based algorithms were constructed based on SI framework, specifically PSO imitates the natural behavior of birds. Meanwhile, there are some nature-inspired algorithms that were developed based on evolutionary systems or the combination of SI and evolutionary systems. Despite the explosion of the metaphor-based metaheuristics, there is critique that's state that many of them hide their mere or trivial novelty behind the metaphor [28]. By abstracting the metaphors, many searching methods in these metaphor-based metaheuristics in general are the same or similar. In general, the highest quality agent or solution becomes the most popular entity taken to become a target for the directed search. This highest quality agent can be the global version or the local one. This entity has various name in many metaphor-based metaheuristics. Besides the highest quality agent, there are also many common entities like higher quality agent, and other agent, that also have many names. Besides the entity, the types of searches also have various names, for example the neighborhood search. Below are the examples of several metaphor-based metaheuristics including the explanation of searching methods plus the metaphor that is used for the searching methods.

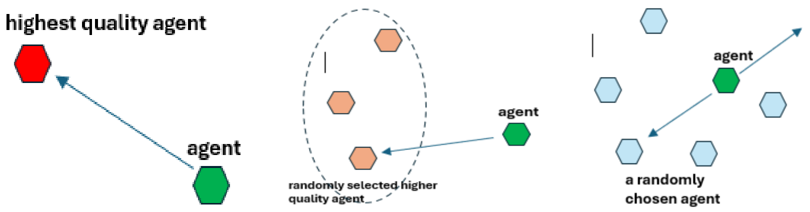
COA is a metaheuristic that imitates the behavior of colony of crayfish [10]. It uses several metaphor-based terms such as temperature, food, and shade. Temperature is a randomized variable that is used to determine the strategy or searching method that is taken in every iteration. Shade is the entity that is in the middle of the highest quality solution in the recent iteration and the highest quality solution until the recent iteration. The food is the highest quality solution. Meanwhile, there are three stages: competition, foraging, and summer resort. By abstracting these metaphors, they are the condition based on the randomized number and the threshold. OOA is a metaheuristic that imitates the behavior of the osprey hunting the prey [11]. It employs two sequential phases. The 1st phase employs exploration that uses position identification and fish hunting as metaphors. Meanwhile, the 2nd phase employs exploitation that uses carrying the caught prey to certain position as metaphors. In the 1st phase, a set that consists of all higher quality agents plus the highest quality agent is called fish. The target in this phase is a randomly selected fish. In the 2nd phase, carrying the fish is a metaphor for neighborhood search.

KMA is a metaheuristic that imitates the behavior of Komodo dragon during hunting for prey and mating [21]. It employs population split into three groups based on the quality of the agents: big males, females, and small males. The big males are high quality agents; female are the moderate quality agents; and small males are low quality agents. Females perform two types of searches: motion toward the highest quality agent which metaphor is sexual reproduction and random search which metaphor is asexual reproduction. The motion of the high-quality agents tends to be exploitation. The motion of the low-quality agents tends to be exploration. Meanwhile, the motion of the females is balance between exploitation which is motion toward the highest quality agent and exploration which is the random search, but one action can be done at one time. PDO is a method that imitates the behavior of prairie dog [9]. There are four options that represent searching methods: foraging, burrowing, food alarm, and antipredation. Foraging and burrowing represent exploration while food alarm and antipredation represent exploitation. The strategy selection is based on the iteration where there are four equal size periods from the 1st iteration to the maximum iteration. There are two entities that are used for reference: the highest quality solution so far and the highest quality solution in the recent iteration. EHO is a metaheuristic that imitates the breeding process of elk herd [20]. There are three phases: rutting, calving, and selection. Rutting is the process of identifying each agent based on the normalized quality. Calving is the process of creating new offsprings based on the motion toward a randomly taken higher quality agent. Selection is a mechanism to determine which entities can be continued to the next iteration.

Based on this explanation, proposing new metaheuristic that is free from metaphor is challenging. Moreover, it is better that the proposed algorithm is kept simple. By do not using metaphor and keeping the algorithm simple, the proposed algorithm will be easier to learn. This consideration becomes the motivation of developing new metaheuristic that simple and free from metaphor. Moreover, it is also challenging to develop a metaheuristic that performs both exploration and exploitation simultaneously as in general, many metaheuristics split the exploration and exploitation.

3. METHOD

The proposed DDO is developed based on the concept of balance between exploitation and exploration. Both activities, which in nature are different, are then merged into single search. There are three motions that are carried out in DDO as shown in Figure 1. The 1st motion is the motion toward the highest quality agent as in Figure 1(a). The 2nd motion is the motion toward a randomly taken higher quality agent as in Figure 1(b). The third motion is the motion based toward or away from a randomly taken agent as in Figure 1(c).



(a) (b) (c)

Figure 1. Three motions in DDO of (a) 1st motion, (b) 2nd motion, and (c) third motion

These three motions are then transformed into two sequential searches. The 1st search is the blending of the motion toward the highest quality agent and the motion toward a randomly taken higher quality agent. The 2nd search is the blending of the motion toward the highest quality agent and the motion toward or away from a randomly taken agent. The motion toward the highest quality agent represents the exploitation-oriented motion. Meanwhile, the motion toward a randomly taken higher quality agent and the motion toward or away from a randomly taken agent represents the exploration-oriented motion. In both searches, the portion of the exploitation-oriented motion is 0.5 so that the portion of the exploration-oriented motion is also 0.5. Each search generates a solution candidate. Stringent acceptance is employed in DDO so that this candidate replaces the recent value of the agent only if this candidate provides improvement. The formal explanation of DDO is provided in Algorithm 1. The formulation of each process is presented in (1) to (10). Before that, the notations are provided in Table 1.

Algorithm 1. Double direction optimizer
```
start
  for all agents
    initiate a_i
    update a_highest
  end for
  for t =1 to t_max
    for all agents
      perform 1st search and update a_highest
      perform 2nd search and update a_highest
    stop
  stop
  return a_highest
stop
```

Table 1. Notation list

Notation	Description
a	agent
A	set of agents
A_{pool}	set of higher quality agents
a_{sel1}, a_{sel2}	1st and 2nd selected agent
c_1, c_2	1st and 2nd candidate
of	objective function
b_{upper}, b_{lower}	upper boundary, lower boundary
u_1	random [0, 1]
u_2	random among a set
u_3	random {1, 2}
t	iteration
t_{max}	maximum iteration
i	index of agent
j	index of dimension
k	index of another agent
d	dimension size

DDO consists of a set of autonomous agents as presented in (1). Then, there are two processes that are carried out during the initialization. The 1st process is generating the initial agent randomly within space as presented in (2). Then, the 2nd process is updating the highest quality agent as presented in (3).

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

$$a_{i,j} = b_{lower,j} + u_1(b_{upper,j} - b_{lower,j}) \tag{2}$$

$$a_{highest}' = \begin{cases} a_i, of(a_i) < of(a_{highest}) \\ a_{highest}, else \end{cases} \tag{3}$$

The process of the 1st search is presented in (4) to (7). In the beginning, a pool that consists of higher quality agents plus the highest quality agent is set using (4). Then, an agent is randomly taken from this pool using (5). The search that combines the 1st and the 2nd search in a balance manner is presented in (6) to generate the 1st solution candidate. Finally, the agent is updated based on the 1st solution candidate and this process is presented in (7).

$$A_{pool,i} = \{\forall a_k \in A \wedge of(a_k) < of(a_i)\} \cup a_{highest} \tag{4}$$

$$a_{sel1,i} = u_2(A_{pool,i}) \tag{5}$$

$$c_{1,i,j} = a_{i,j} + \frac{u_1(a_{highest,j} - u_3 a_{i,j})}{2} + \frac{u_1(a_{sel1,i,j} - u_3 a_{i,j})}{2} \tag{6}$$

$$a_i' = \begin{cases} c_{1,i}, of(c_{1,i}) < of(a_i) \\ a_i, else \end{cases} \tag{7}$$

The process of the 1st search is presented in (8) to (10). In the beginning, an agent is randomly taken from the swarm using (8). The search that combines the 1st and the 3rd search in a balance manner is presented in (9) to generate the 2nd solution candidate. The motion is toward the randomly taken agent if this randomly taken agent is better than the agent. Otherwise, the direction is to move away from the target. Finally, the agent is updated based on the 1st solution candidate and this process is presented in (10).

$$a_{sel2,i} = u_2(A) \tag{8}$$

$$c_{2,i,j} = \begin{cases} a_{i,j} + \frac{u_1(a_{highest,j} - u_3 a_{i,j})}{2} + \frac{u_1(a_{sel2,i,j} - u_3 a_{i,j})}{2}, of(a_{sel2,i}) < of(a_i) \\ a_{i,j} + \frac{u_1(a_{highest,j} - u_3 a_{i,j})}{2} + \frac{u_1(a_{i,j} - u_3 a_{sel,i,j})}{2}, else \end{cases} \tag{9}$$

$$a_i' = \begin{cases} c_{2,i}, of(c_{2,i}) < of(a_i) \\ a_i, else \end{cases} \tag{10}$$

The complexity of DDO can be presented as $O(n(A).d)$ during the initialization and $O(n(A)^2 d.t_{max})$. During initialization, the complexity is linear to the swarm size or the dimension size. Meanwhile, during iteration, the complexity is linear to the dimension size or the maximum iteration, and quadratic proportional to the swarm size.

4. RESULTS AND DISCUSSION

4.1. Assessment result

This subsection provides an assessment to investigate the efficacy of DDO to solve optimization problems. Three cases are taken for the assessment. The 1st case is set of 23 functions. The 2nd case is 4 engineering problems. The 3rd case is EED problem. The 1st case represents unconstrained and standard problems. The 2nd case represents constrained and standard problems. The third use case represents constrained and practical problems. In this work, DDO is benchmarked with five existing algorithms: COA [10], HO [8], OOA [11], CWO [13], and DOA [14]. All these five algorithms are new. COA and HO represent algorithms that employ loose acceptance while OOA, CWO, and DOA represent algorithms that employ stringent acceptance. The swarm size is 5. The maximum iteration is 20.

The 1st use case is the set of 23 functions. Its seven high dimension unimodal functions are used to observe the exploitation capability as each of these functions has only one optimal solution [39]. Its six high dimension multimodal functions are used to observe exploration capability as each of these functions has multiple optimal solutions but only one global optimal solution [39]. It means the capability to escape from the local optimal solutions matters. Its ten fixed dimension multimodal functions are used to observe the balancing capability between exploitation and exploration [39]. The detailed information of these functions can be found in [10]. The dimension for high dimension functions is 30. The decimal point less than 10-4 is rounded to 0.

Table 2 displays the supremacy of DDO in high dimension unimodal functions. It attains the best result in all seven functions. It also attains the global optimal for three functions (F1, F2, and F4). Meanwhile, CWO becomes the worst one as it attains the worst result in six functions (F1 to F6). Then, HO becomes the 2nd worst one as it attains the 2nd worst result in five functions (F1, F2, and F4 to F6). Besides DDO, COA, OOA, and DOA also attains the global optimal in F2. The result also indicates a widespread between the best and worst algorithms in these high dimension unimodal functions. This circumstance occurs in all seven functions. Table 3 displays the supremacy still occurs for DDO in high dimension multimodal functions. It attains the best result in four functions (F9 to F11, and F13). DDO also attains the global optimal in two functions (F9 and F10). Meanwhile, DDO becomes the 2nd best in F12 and fourth best in F8. COA becomes the 2nd best algorithm as it attains the 2nd best result in three functions (F9 and F11). CWO becomes the worst algorithm as it attains the worst result in four functions (F10 to F13). The spread between the best result and worst result is mixed in these functions. Widespread occurs in five functions (F9 to F13). Meanwhile, narrow spread occurs in F8.

Table 2. Result on high dimension unimodal functions

F	Parameter	COA [10]	HO [8]	OOA [11]	CWO [13]	DOA [14]	DDO
1	mean	0.0009	6.9097×10^{1}	0.1992	1.7616×10^{4}	0.4930	0.0000
	range	0.0139	1.4068×10^{2}	0.3310	1.4309×10^{4}	2.2460	0.0000
	position	2	5	3	6	4	1
2	mean	0.0000	9.9050×10^{3}	0.0000	3.2116×10^{31}	0.0000	0.0000
	range	0.0000	1.0417×10^{5}	0.0000	7.0656×10^{32}	0.0000	0.0000
	position	1	5	1	6	1	1
3	mean	3.7437	1.0828×10^{3}	1.1349×10^{3}	3.5890×10^{4}	9.2550×10^{2}	0.0016
	range	4.0904×10^{1}	5.2197×10^{3}	7.4532×10^{3}	3.6250×10^{4}	3.2686×10^{3}	0.0116
	position	2	4	5	6	3	1
4	mean	0.1439	3.4356	0.8963	5.1600×10^{1}	1.3129	0.0000
	range	3.1167	3.7783	1.6887	1.4211×10^{1}	2.9405	0.0000
	position	2	5	3	6	4	1
5	mean	3.8643×10^{1}	4.6163×10^{4}	3.2262×10^{1}	2.4873×10^{7}	3.6289×10^{1}	2.8942×10^{1}
	range	1.2843×10^{2}	2.3427×10^{5}	1.4598×10^{1}	3.2954×10^{7}	2.9888×10^{1}	0.1753
	position	4	5	2	6	3	1
6	mean	7.0849	7.9495×10^{1}	6.1723	1.8815×10^{4}	6.5198	6.1137
	range	0.9700	1.7381×10^{2}	2.4558	1.1582×10^{4}	3.1813	2.0996
	position	4	5	2	6	3	1
7	mean	0.0397	2.0501×10^{2}	0.0354	1.2094×10^{1}	0.0454	0.0112
	range	0.1730	3.5240×10^{2}	0.0632	1.3509×10^{1}	0.1845	0.0268
	position	3	6	2	5	4	1

Table 3. Result on high dimension multimodal functions

F	Parameter	COA [10]	HO [8]	OOA [11]	CWO [13]	DOA [14]	DDO
8	mean	-1.6495×10^{3}	-8.9384×10^{1}	-3.2233×10^{3}	-3.2210×10^{3}	-2.8575×10^{3}	-2.2803×10^{3}
	range	4.0904×10^{3}	5.8782×10^{1}	3.0789×10^{3}	2.3603×10^{3}	1.9878×10^{3}	2.5584×10^{3}
	position	5	6	1	2	3	4
9	mean	0.7676	3.3282×10^{2}	3.5354	2.9411×10^{2}	4.1539	0.0000
	range	7.8096	2.0907×10^{2}	3.4619×10^{1}	5.0041×10^{1}	2.1662×10^{1}	0.0000
	position	2	6	3	5	4	1
10	mean	0.0006	6.3110	0.1260	1.7797×10^{1}	0.2239	0.0000
	range	0.0133	6.4759	0.2671	2.3461	1.4136	0.0000
	position	2	5	3	6	4	1
11	mean	0.1035	0.6609	0.1608	1.7426×10^{2}	0.2981	0.0047
	range	1.4294	0.8064	0.5087	1.0161×10^{2}	0.7454	0.0566
	position	2	5	3	6	4	1
12	mean	1.7670	7.1148	1.0737	1.7621×10^{7}	1.1105	1.0931
	range	3.4031	1.0516×10^{1}	0.8075	3.9357×10^{7}	0.8421	0.6449
	position	4	5	1	6	3	2
13	mean	3.6052	8.3395	3.3655	6.0101×10^{7}	3.3689	3.1343
	range	6.1277	3.3368×10^{1}	0.7923	1.2195×10^{8}	1.2225	0.3208
	position	4	5	2	6	3	1

Table 4 displays the competitiveness of DDO in the fixed dimension multimodal functions. DDO becomes the fourth best algorithm in eight functions (F14, F15, F17, F18, and F20 to F23), the 2nd best in

F19 and fifth best in F16. COA becomes the 2nd worst algorithm while HO becomes the worst algorithm. Although DDO loses its supremacy, the spread among metaheuristics in these ten functions is narrow. This circumstance takes place in all ten functions. This fact also displays fierce confrontation among metaheuristics. Despite poor performance in almost ten functions, COA becomes the best algorithm in F19. OOA becomes the best algorithm by achieving the best result in five functions (F14 to F18).

Table 4. Result on fixed dimension multimodal functions

F	Parameter	COA [10]	HO [8]	OOA [11]	CWO [13]	DOA [14]	DDO
14	mean	1.1641×10^{1}	2.1802×10^{1}	8.8796	9.0067	9.4877	1.1247×10^{1}
	range	1.0677×10^{1}	9.5106×10^{1}	1.5445×10^{1}	1.6243×10^{1}	1.5627×10^{1}	1.2385×10^{1}
	position	5	6	1	2	3	4
15	mean	0.0785	0.2624	0.0036	0.0134	0.0048	0.0246
	range	0.1410	2.5534	0.0324	0.0357	0.0233	0.1045
	position	5	6	1	3	2	4
16	mean	-0.3765	1.0774	-1.0275	-0.9797	-1.0241	-0.9148
	range	1.0162	4.4438	0.0267	0.2288	0.0432	0.7763
	position	5	6	1	3	2	4
17	mean	4.0192	3.2455	0.3985	0.4175	0.4040	3.0204
	range	1.0965×10^{1}	8.4598	0.0019	0.0920	0.0334	1.8305×10^{1}
	position	6	5	1	3	2	4
18	mean	3.8223×10^{1}	7.3453×10^{2}	3.0098	3.6395	3.1257	3.7510×10^{1}
	range	1.0072×10^{2}	2.8771×10^{3}	0.0480	3.0187	0.5721	1.2630×10^{2}
	position	5	6	1	3	2	4
19	mean	-2.1864	-0.0328	-0.0495	-0.0495	-0.0495	-0.0495
	range	0.5081	0.0495	0.0000	0.0000	0.0000	0.0000
	position	1	6	2	2	2	2
20	mean	-1.2498	-0.4572	-3.0024	-3.0115	-3.0969	-1.8282
	range	2.3583	1.8893	0.5844	0.6046	0.5030	2.4238
	position	5	6	3	2	1	4
21	mean	-0.6381	-0.7144	-2.5681	-3.5367	-4.4096	-2.2534
	range	1.7716	2.4038	3.7301	4.4547	7.5920	3.3735
	position	6	5	3	2	1	4
22	mean	-0.8288	-0.9226	-2.8291	-4.7293	-4.3212	-2.0946
	range	1.6121	3.4908	4.2509	5.0288	6.6768	3.4280
	position	6	5	3	1	2	4
23	mean	-1.0853	-1.1567	-2.8011	-4.3247	-4.3164	-2.0985
	range	2.9944	1.9820	3.3525	5.7714	5.1837	2.5867
	position	6	5	3	1	2	4

Table 5 strengthens the competitiveness of DDO compared with its benchmark. Overall, DDO is superior to COA and HO. Moreover, DDO is absolute superior to HO. Meanwhile, DDO has proven competitive compared to OOA, CWO, and DOA. In general, DDO is dominant in the high dimension functions. Unfortunately, although DDO is superior to COA and HO, it is inferior to OOA, CWO, and DOA in fixed dimension functions. Fortunately, as shown in Table 5, the narrow spread in fixed dimension functions proofs that DDO is still competitive to OOA, CWO, and DOA.

Table 5. Summary of supremacy of DDO

Cluster	COA [10]	HO [8]	OOA [11]	CWO [13]	DOA [14]
1	6	7	6	7	6
2	6	6	4	5	5
3	9	10	0	0	0
Total	21	23	10	12	11

The 2nd use case is four engineering problems. They include pressure vessels, speed reducers, welded beam, and spring design problems. These problems are constrained problems where in each design problem, there are multiple constraints. It makes the solution can be put anywhere within space as the value in certain dimension makes limitations for solution in other dimensions. Detailed information of them can be found in [20]. The result displays that DDO is also competitive although not superior in engineering problems. It becomes the third best in the welded beam design problem, fourth best in both pressure vessel and spring design problems, and fifth best in the speed reducer design problem. CWO becomes the best algorithm as it attains the best result in three problems (speed reducer, welded beam, and spring) while it becomes the 2nd best in pressure vessel design problem. HO becomes the worst algorithm as it attains the worst result in three design problems (pressure vessel, welded beam, and spring).

Table 6 also reveals the variety of spread between the algorithms, especially between the best and worst algorithms. The spread is very wide in the pressure vessel design problems. There are four clusters in this problem. The 1st cluster contains CWO and DOA. The 2nd cluster contains OOA. The third cluster contains DDO and COA. Meanwhile the fourth cluster contains HO. Table 7 reveals the very narrow spread among algorithms in speed reducer design problems. It is indicated based on the spread between CWO as the best algorithm and COA as the worst algorithm. There is only a single cluster in this problem. Meanwhile, based on the range, there are two clusters of algorithms in this problem. The 1st cluster contains CWO and DOA. The 2nd cluster contains COA, HO, OOA, and DDO.

Table 6. Result in pressure vessel design problems

Metaheuristic	Mean	Range	Rank
COA [10]	7.3674×10^{5}	3.3916×10^{6}	5
HO [8]	8.2946×10^{12}	1.9054×10^{11}	6
OOA [11]	2.5899×10^{2}	4.3729×10^{3}	3
CWO [13]	2.6005×10^{1}	7.6470×10^{1}	2
DOA [14]	1.0107×10^{1}	1.5683×10^{1}	1
DDO	9.4723×10^{3}	1.2680×10^{5}	4

Table 7. Result in speed reducer design problems

Metaheuristic	Mean	Range	Rank
COA [10]	3.6284×10^{3}	3.6826×10^{2}	6
HO [8]	3.6011×10^{3}	1.8726×10^{2}	3
OOA [11]	3.6336×10^{3}	2.9471×10^{2}	4
CWO [13]	3.5379×10^{3}	3.0308×10^{1}	1
DOA [14]	3.5383×10^{3}	4.0981×10^{1}	2
DDO	3.6163×10^{3}	3.0173×10^{2}	5

Table 8 reveals the widespread among algorithms in welded beam design problems. It is indicated based on the spread between CWO and HO. These algorithms can be grouped into four clusters. The 1st cluster contains CWO and DOA. The 2nd cluster contains OOA. The 3rd cluster contains COA and DDO. The 4th cluster contains HO. Table 9 reveals the widespread among algorithms in spring design problems. The spread between CWO and HO is wide. These methods can be grouped into three clusters. The 1st cluster contains OOA, CWO, and DOA. The 2nd cluster contains DDO. Then, the third cluster contains COA and HO. The 3rd case is the EED problem in Java-Bali grid system. Java-Bali grid system is the biggest grid system in Indonesia since Java is the most populous and industrialized island in Indonesia. It makes the power consumption in Java Island very high. Meanwhile, the location of Bali is beside Java. This grid system contains eight power plants where six of them are thermal power plants while the two others are hydroelectric power plants [40]. The specification of this system includes the power range of each power plant, the cost constants, and the power demand can be found in [40]. In this paper, the power demand is set to 12,228 MW/hour while cost weight is set to 0.5. The result is shown in Table 10.

Table 8. Result in welded beam design problems

Metaheuristic	Mean	Range	Rank
COA [10]	1.5361×10^{10}	2.6609×10^{11}	5
HO [8]	1.0290×10^{12}	1.6107×10^{13}	6
OOA [11]	1.6531×10^{9}	1.3215×10^{10}	4
CWO [13]	3.0772×10^{8}	8.9356×10^{8}	1
DOA [14]	1.5348×10^{8}	9.9828×10^{8}	2
DDO	1.7408×10^{10}	1.0914×10^{10}	3

Table 9. Result in spring design problems

Metaheuristic	Mean	Range	Rank
COA [10]	2.7200×10^{2}	4.3065×10^{3}	5
HO [8]	8.7078×10^{2}	5.3383×10^{3}	6
OOA [11]	4.2879	5.1650	2
CWO [13]	4.2316	2.5533	1
DOA [14]	5.2796	2.3052×10^{1}	3
DDO	3.0303×10^{1}	8.8224×10^{1}	4

Table 10. Result on EED problem of Java-Bali grid system

Metaheuristic	Mean	Range	Rank
COA [10]	2.3079×10^{10}	4.1731×10^{9}	6
HO [8]	2.3692×10^{10}	3.7716×10^{9}	5
OOA [11]	2.1266×10^{10}	8.1013×10^{8}	4
CWO [13]	2.1242×10^{10}	7.1741×10^{8}	3
DOA [14]	2.0938×10^{10}	5.5360×10^{8}	1
DDO	2.1273×10^{10}	6.3325×10^{8}	2

4.2. Discussion

In general, this result displays that DDO is competitive in the optimization problems. DDO is dominant in high dimension unimodal functions and high dimension multimodal functions. It means that DDO has dominant exploitation and exploration capabilities. Then, DDO is competitive in fixed dimension multimodal functions which means that DDO has balancing capability between exploitation and exploration.

The competitiveness of DDO in the constrained problems whether the engineering problems or EED problems displays that DDO can find optimal solutions with more limited space.

In general, COA and HO perform poorly in whether the unconstrained or constrained problems. The difference between COA and HO in one group and OOA, CWO, DOA, and DDO in another group is the acceptance mechanism where both metaheuristics perform loose acceptance while the others perform stringent acceptance. It means that in general, the stringent acceptance is better than the loose one. In other words, it tends to be better to be stuck in the local optimal rather than being pushed to a worsening solution. Compared between COA and HO, HO tends to be worse than COA. By investigating the searching mechanism, it is shown that COA has more searching algorithms than HO. This circumstance also occurs in OOA, CWO, DOA, and DDO that also have multiple searches. This circumstance displays that algorithm that employ multiple searches tend to be better than algorithms that employ only single searches.

The assessment result also proves the no-free-lunch (NFL) theory. Despite being dominant in high dimension functions, the performance of DDO is competitive or moderate in other cases. CWO is superior in engineering problems although its performance in the 23 functions is moderate. Although the performance of COA is poor in general, its performance in the high dimension unimodal functions is very competitive. The existence of NFL theory is also proven by investigating the performance of spread between the best and worst algorithms. Overall, this spread is wide in the high dimension unimodal functions, some high dimension multimodal functions, and some engineering problems. This spread is narrow in the fixed dimension multimodal functions, some engineering problems, and the EED problems.

Despite its competitiveness, DDO especially and this work generally still have limitations. There are a lot of algorithms and rules that can be adopted to develop a new method. Besides, there are also a lot of use cases that can be taken to investigate the performance DDO or its modification in a more comprehensive manner. These two limitations can be explored to create new metaheuristic or modified DDO in future studies. For example, DDO can be modified to become more adaptive or exploring other entities like lower quality agents or the lowest quality agent to become additional reference. Another interesting algorithm is the marking mechanism so that the agents do not search in the area that has been visited, and the quality of the solution is not sophisticated.

5. CONCLUSION

This paper has presented a new metaphor-free metaheuristic that is called as DDO. This presentation includes the concept of combining exploration and exploitation within single search, formalization through pseudocode and mathematical formulation, and assessment. The result displays that DDO has proven acceptable in finding the quasi-optimal solution. DDO has proven competitive in three cases: 23 functions, four engineering problems, and EED problems. The result also displays the dominance of DDO in high dimension functions. DDO also can find the global optimal of five functions out of 23 functions. This result displays that COA and HO tend to perform poorly rather than the others. In the future, it is challenging to implement DDO to solve various practical optimization problems from broader sectors. Moreover, the modification of DDO by inserting a new search or adaptive algorithm is also important in future studies.

FUNDING INFORMATION

The publication of this paper is funded by Telkom University, Indonesia, without any specific research grant.

AUTHOR CONTRIBUTIONS STATEMENT

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

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

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

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

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

REFERENCES

[1] M. F. Tabassum, M. Saeed, N. A. Chaudhry, J. Ali, M. Farman, and S. Akram, "Evolutionary simplex adaptive Hooke-Jeeves algorithm for economic load dispatch problem considering valve point loading effects," *Ain Shams Engineering Journal*, vol. 12, no. 1, pp. 1001–1015, 2021, doi: 10.1016/j.asej.2020.04.006.

[2] H. M. Hasanien, I. Alsaleh, A. Alassaf, and A. Alateeq, "Enhanced coati optimization algorithm-based optimal power flow including renewable energy uncertainties and electric vehicles," *Energy*, vol. 283, 2023, doi: 10.1016/j.energy.2023.129069.

[3] V. K. Kamboj and O. P. Malik, "Optimal unit commitment and generation scheduling of integrated power system with plug-in electric vehicles and renewable energy sources," *Energies*, vol. 17, no. 1, 2024, doi: 10.3390/en17010123.

[4] S. Yang, J. Wang, and Z. Xu, "Real-time scheduling for distributed permutation flowshops with dynamic job arrivals using deep reinforcement learning," *Advanced Engineering Informatics*, vol. 54, 2022, doi: 10.1016/j.aei.2022.101776.

[5] Y. Hou, C. Wang, C. Zhang, L. Dang, and C. Xiao, "A hybrid max-min ant system algorithm for electric capacitated vehicle routing problem," *IAENG International Journal of Computer Science*, vol. 51, no. 3, pp. 195–203, 2024.

[6] Q. Dong and Y. Yuan, "Data-driven distributionally robust supplier selection and order allocation problems considering carbon emissions," *International Transactions in Operational Research*, vol. 32, no. 2, pp. 1119–1145, 2025, doi: 10.1111/itor.13328.

[7] S. O. Oladejo, S. O. Ekwe, L. A. Akinyemi, and S. A. Mirjalili, "The deep sleep optimizer: a human-based metaheuristic approach," *IEEE Access*, vol. 11, pp. 83639–83665, 2023, doi: 10.1109/ACCESS.2023.3298105.

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

[9] A. E. Ezugwu, J. O. Agushaka, L. Abualigah, S. Mirjalili, and A. H. Gandomi, "Prairie dog optimization algorithm," *Neural Computing and Applications*, vol. 34, no. 22, pp. 20017–20065, 2022, doi: 10.1007/s00521-022-07530-9.

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

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

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

[13] S. Alomari *et al.*, "Carpet weaver optimization: a novel simple and effective human-inspired metaheuristic algorithm," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 4, pp. 230–242, 2024, doi: 10.22266/IJIES2024.0831.18.

[14] S. A. Omari *et al.*, "Dollmaker optimization algorithm: a novel human-inspired optimizer for solving optimization problems," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 3, pp. 816–828, 2024, doi: 10.22266/ijies2024.0630.63.

[15] T. Hamadneh *et al.*, "Sales training based optimization: a new human-inspired metaheuristic approach for supply chain management," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 6, pp. 1325–1334, 2024, doi: 10.22266/ijies2024.1231.96.

[16] T. Hamadneh *et al.*, "Addax optimization algorithm: a novel nature-inspired optimizer for solving engineering applications," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 3, pp. 732–743, 2024, doi: 10.22266/ijies2024.0630.57.

[17] E. Trojovska and M. Dehghani, "Clouded leopard optimization: a new nature-inspired optimization algorithm," *IEEE Access*, vol. 10, pp. 102876–102906, 2022, doi: 10.1109/ACCESS.2022.3208700.

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

[19] 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, 2022, doi: 10.1016/j.knosys.2022.108457.

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

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

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

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

[24] M. Noroozi, H. Mohammadi, E. Efatinasab, A. Lashgari, M. Eslami, and B. Khan, "Golden search optimization algorithm," *IEEE Access*, vol. 10, pp. 37515–37532, 2022, doi: 10.1109/ACCESS.2022.3162853.

[25] M. Dehghani, Š. Hubálovský, and P. Trojovský, "A new optimization algorithm based on average and subtraction of the best and worst members of the population for solving various optimization problems," *PeerJ Computer Science*, vol. 8, 2022, doi: 10.7717/PEERJ-CS.910.

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

[27] P. Trojovský and M. Dehghani, "Subtraction-average-based optimizer: a new swarm-inspired metaheuristic algorithm for solving optimization problems," *Biomimetics*, vol. 8, no. 2, 2023, doi: 10.3390/biomimetics8020149.

[28] J. Swan *et al.*, "Metaheuristics 'in the large,'" *European Journal of Operational Research*, vol. 297, no. 2, pp. 393–406, 2022, doi: 10.1016/j.ejor.2021.05.042.

[29] K. K. Mathew and D. M. Abraham, "Particle swarm optimization based sliding mode controllers for electric vehicle onboard charger," *Computers and Electrical Engineering*, vol. 96, 2021, doi: 10.1016/j.compeleceng.2021.107502.
[30] I. Thammachantuek and M. Ketcham, "Path planning for autonomous mobile robots using multi-objective evolutionary particle swarm optimization," *PLoS ONE*, vol. 17, 2022, doi: 10.1371/journal.pone.0271924.
[31] D. Cheng, "Water allocation optimization and environmental planning with simulated annealing algorithms," *Mathematical Problems in Engineering*, vol. 2022, 2022, doi: 10.1155/2022/2281856.
[32] T. Gao, Q. Tang, J. Li, Y. Zhang, Y. Li, and J. Zhang, "A particle swarm optimization with lévy flight for service caching and task offloading in edge-cloud computing," *IEEE Access*, vol. 10, pp. 76636–76647, 2022, doi: 10.1109/ACCESS.2022.3192846.
[33] R. Estran, A. Souchaud, and D. Abitbol, "Using a genetic algorithm to optimize an expert credit rating model," *Expert Systems with Applications*, vol. 203, 2022, doi: 10.1016/j.eswa.2022.117506.
[34] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.
[35] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine predators algorithm: a nature-inspired metaheuristic," *Expert Systems with Applications*, vol. 152, 2020, doi: 10.1016/j.eswa.2020.113377.
[36] M. Nanjappan, P. Krishnadoss, J. Ali, G. Natesan, and B. Ananthakrishnan, "Task scheduling based on cost and execution time using ameliorate grey wolf optimizer algorithm in cloud computing," *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 3, pp. 417–427, 2023, doi: 10.22266/ijies2023.0630.33.
[37] M. A. A. Al-Qaness, A. A. Ewees, H. Fan, L. Abualigah, and M. A. Elaziz, "Marine predators algorithm for forecasting confirmed cases of COVID-19 in Italy, USA, Iran, and Korea," *International Journal of Environmental Research and Public Health*, vol. 17, no. 10, 2020, doi: 10.3390/ijerph17103520.
[38] R. A. Swief, N. M. Hassan, H. M. Hasanien, A. Y. Abdelaziz, and M. Z. Kamh, "Multi-regional optimal power flow using marine predators algorithm considering load and generation variability," *IEEE Access*, vol. 9, pp. 74600–74613, 2021, doi: 10.1109/ACCESS.2021.3081374.
[39] 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, doi: 10.1109/ACCESS.2022.3172789.
[40] K. M. D. Puspitasari, J. Raharjo, A. S. Sastrosubroto, and B. Rahmat, "Generator scheduling optimization involving emission to determine emission reduction costs," *International Journal of Engineering*, vol. 35, no. 8, Aug. 2022, pp. 1468-1478, doi: 10.5829/IJE.2022.35.08B.02.

BIOGRAPHIES OF AUTHORS

Purba Daru Kusuma received his bachelor's and master's degrees in Electrical Engineering from Institut Teknologi Bandung, Indonesia, in 2002 and 2009, respectively. He received his doctoral degree in Computer Science from Universitas Gadjah Mada, Indonesia, in 2017. Recently, he is an associate professor at Telkom University, Indonesia. His research area is artificial intelligence, machine learning, and operational research. He can be contacted at email: purbodaru@telkomuniversity.ac.id.

Helmy Widyantara recently, he is an associate professor in Telkom University, Indonesia. He received his bachelor's degree from STMIK Surabaya in 202. He received master's degree from Universitas Gadjah Mada in 2010 and doctoral degree from Institut Teknologi Sepuluh Nopember in 2020. The field of research interests are but are not limited to embedded systems, the internet of things, and smart farming. He can be contacted at email: helmywidyantara@telkomuniversity.ac.id.