Received: April 29, 2023. Revised: May 21, 2023. 

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

# Swarm Magnetic Optimizer: A New Optimizer that Adopts Magnetic Behaviour

**Purba Daru Kusuma[1]*** **Faisal Candrasyah Hasibuan[1]**

[1]*Computer Engineering, Telkom University, Indonesia*
* Corresponding author's Email: purbodaru@telkomuniversity.ac.id

**Abstract:** This paper introduces a novel swarm-based metaheuristic called swarm magnetic optimizer (SMO). SMO imitates the behaviour of two magnets close to each other: pushing toward or pulling away from each other. This push-pull mechanism is then adopted to become a novel search in SMO. SMO is set up as a swarm of magnets that move autonomously. In the early iteration, the pull strategy is dominant. Meanwhile, it declines during the iteration and is replaced by a push strategy. The determination between these two strategies is calculated stochastically. SMO consists of three sequential phases in every iteration where the corresponding magnet and its reference run a push or pull search in each phase. The reference used in each phase is the global best magnet, a randomly selected magnet, and a randomly generated magnet. This paper evaluates SMO through simulation to find the optimal solution for 23 functions. The performance of SMO is compared with five latest metaheuristics: mixed leader-based optimizer (MLBO), guided pelican algorithm (GPA), zebra optimization algorithm (ZOA), coati optimization algorithm (COA), and clouded leopard optimizer (CLO). The result shows that SMO is better than MLBO, GPA, ZOA, COA, and CLO in consecutively 22, 15, 11, 13, and 14 functions. The superiority of SMO, especially, is in solving high dimension functions. Meanwhile, in the fixed-dimension multimodal functions, SMO is still superior to MLBO but less superior to GPA, ZOA, COA, and CLO.

**Keywords:** Optimization, Metaheuristic, Magnet, Push and pull.

## 1. Introduction

Metaheuristics is a well-known stochastic method for solving various optimization issues. Its popularity is a result of its advantage in locating viable solutions to various optimization problems by utilizing possible computation resources due to its stochastic approach but at the expense of a guarantee of locating the genuine optimality [1]. In every optimization issue, multiple potential solutions must be found [2]; hence, tracing all these potential solutions requires an exorbitant number of computational resources, making the problem nearly challenging to solve. In addition, metaheuristic is versatile enough to be implemented to handle different types of optimization problems without excessive effort on modification since it abstracts the problem detail [3] by focusing on the objectives, decision variables, and constraints.

Various new metaheuristics have been proposed in recent years with their distinct mechanism or strategy. Most of them are swarm-based metaheuristics. Many of these swarm-based metaheuristics imitate animal behavior during hunting or foraging, such as the African vultures optimization algorithm (AVOA) [4], golden jackal optimizer (GJO) [5], Komodo mlipir algorithm (KMA) [6], pelican optimization algorithm (POA) [7], guided pelican algorithm (GPA) [8], zebra optimization algorithm (ZOA) [9], coati optimization algorithm (COA) [10], clouded leopard optimizer (CLO) [11], northern goshawk optimizer (NGO) [12], modified honey badger algorithm (MHBA) [13], fennec fox optimizer (FFA) [14], sparrow search algorithm (SSA) [15], osprey optimization algorithm (OOA) [2], reptile search algorithm (RSA) [16], marine predator algorithm (MPA) [17], Siberian tiger optimizer (STO) [3], and many others. Some metaheuristics, such as coronavirus herd immunity optimizer (CHIO) [18] and coronavirus optimization algorithm (COVIDOA) [19], are inspired by the mechanism of coronavirus. Some metaheuristics imitate the mechanism of leaders or influential

members within a population and their followers. For example, multi-leader optimizer (MLO) [20], mixed leader-based optimizer (MLBO) [21], three influential member-based optimizers (TIMBO) [22], and randomly selected leader-based optimizer (RSLBO) [23], among others. Some metaheuristics, such as the total interaction algorithm (TIA) [24], the multiple interaction optimizer (MIO) [25], the golden search optimizer (GSO) [26], and the average and subtraction-based optimizer (ASBO) [27], do not employ metaphor in their names.

Almost all metaheuristics perform the movement only by the corresponding solution. Meanwhile, the reference remains static. In many metaheuristics, the direction of this movement is unconditional because the corresponding solution always moves toward the reference. Meanwhile, in many latest metaheuristics, the direction of this guided search is conditional. The corresponding solution moves toward the reference only if the reference is better than the corresponding solution. Otherwise, the corresponding solution moves away from this reference.

Ironically, moving toward a better solution or avoiding a worse solution never guarantees improvement. Although the probability of improvement is higher than before, this movement may push the corresponding solution toward the optimal local solution. Based on this circumstance, it is wiser that both directions are traced. Unfortunately, the metaheuristic that searches in both directions is hard to find. Moreover, utilizing the reference to perform a search may improve the searching performance.

Based on this gap and opportunity, this work aims to introduce a new swarm-based metaheuristic called swarm-magnetic optimizer (SMO). Due to its name, SMO adopts the interaction of two magnets that are close to each other, which is the push-pull mechanism. This mechanism is then transformed into a different search in SMO where two types of interaction exist between the corresponding solution and its reference. These two magnets may get closer or move away from each other.

The novelty of this work is mainly on adapting the magnet's behavior, which is pushing and pulling movement, to become a new swarm-based metaheuristic. Meanwhile, the contribution of this work is as follows.

- This work presents the central concept and the formalization of the swarm magnetic optimizer (SMO).
- The 23 functions have been employed as a problem during the performance evaluation of SMO.
- The comparative analysis regarding the performance of SMO is conducted by competing for SMO with five latest metaheuristics.
- The investigation of the performance of SMO based on the evaluation result is conducted to find the strong and weak points of this proposed metaheuristic.

The remainder of this paper is formulated as follows. In section two, the mechanism of swarm-based metaheuristics is reviewed, including some latest metaheuristics, in the context of the interaction between the corresponding member and its reference. The references and the number of strategies in the guided search implemented in these metaheuristics are also reviewed. A detailed description of SMO is presented in section three. The description includes the magnet's behavior, its integration into the SMO concept, the pseudocode presentation of the algorithm, and its mathematical model of processes. In section four, we present the evaluation of SMO's performance in solving the 23 functions, its competition with other optimization methods, and the sensitivity analysis. The in-depth analysis regarding the result and findings, the drawback to the theory, limitations, and the computational complexity of SMO are discussed in section five. After all, the conclusion and proposal for future development are summarized in section six.

## 2. Related works

Swarm intelligence has emerged as a fundamental platform for developing new metaheuristics in recent years. Swarm-based metaheuristics are typically population-based, with a set of solutions constituting the population. Each solution operates as an autonomous agent, working to enhance its quality. Swarm-based metaheuristics, which rely on a population of solutions acting as autonomous agents, exhibit more intense interactions among agents than evolutionary-based metaheuristics like genetic algorithms (GA), resulting in better performance. Many studies that proposed new swarm-based metaheuristic used GA as their benchmark to prove that their proposed metaheuristic is better than GA, such as in KMA [6], MLBO [21], and ZOA [9].

The guided search is the central component of swarm-based metaheuristics. In this search, a member's movement is determined by its interaction with a reference solution. The primary goal of this search is to improve the current solution. The guided search has three common movements: first, the corresponding solution moves towards the reference; second, the corresponding solution avoids the reference; third, the reference avoids the

Table 1. Comparison among latest metaheuristics

| No | Metaheuristic | Guided Search |
|---|---|---|
| 1 | MLBO [21] | Based on the quality comparison, the corresponding member moves toward or avoids the mixture of the best and randomized members. |
| 2 | GPA [8] | The corresponding member moves toward the global best member if the latter is better than the corresponding member. Otherwise, the corresponding member avoids the global best member. |
| 3 | ZOA [9] | First, the corresponding member moves toward the best member. Second, the corresponding member moves toward a randomly selected member. |
| 4 | COA [10] | First, the corresponding member moves toward the best member. Second, the corresponding member moves toward or avoids a randomized member within the search space based on the quality comparison. |
| 5 | CLO [11] | Based on the quality comparison, the corresponding member moves toward or avoids a randomly selected member. As distinction with RSLBO [23] and NGO [12], CLO performs neighbourhood search with non-linear space reduction during the iteration and it begins with wide space. |
| 6 | RSLBO [23] | Based on the quality comparison, the corresponding member moves toward or avoids a randomly selected member. As distinction with CLO [11] and NGO [12], RSLBO does not deploys any neighbourhood search during the iteration. |
| 7 | TIA [24] | Based on the quality comparison, the corresponding member moves toward or avoids all other members based on the quality comparison between the corresponding member and its partner. |
| 8 | MIO [25] | The corresponding member moves toward or avoids some randomly selected members based on the quality comparison. |
| 9 | GSO [26] | The corresponding member moves toward the mixture of the global best member and the local best member. |
| 10 | NGO [12] | Based on the quality comparison, the corresponding member moves toward or avoids a randomly selected member. As distinction with CLO [11] and RSLBO [12], NGO performs neighbourhood search with linear space reduction during the iteration and it begins with narrow space. |
| 11 | OOA [2] | The corresponding member moves toward a randomly selected better member or global best member. |
| 12 | MPA [17] | The corresponding member moves towards the local best member, while the local best member avoids it. Additionally, the corresponding member can move toward the gap between two randomly selected members. The decision is determined based on the phase during the iteration. |
| 13 | MHBA [13] | The global best member avoids the corresponding member. |
| 14 | FFA [14] | Based on the quality comparison, the corresponding member moves toward or avoids a randomly selected member. |
| 15 | this work | Corresponding members and references move toward each other, or corresponding members and references avoid each other. |

corresponding solution. The first movement is employed when the reference solution is either the best or a randomly selected solution, but its quality is superior to that of the corresponding solution. The second movement is performed when the reference is a random solution, and its quality is worse than the corresponding solution. The third movement is performed when the reference is the best solution. The third movement is designed to improve the quality of the best solution by avoiding a worse solution. These three movements are illustrated in Fig. 1. Fig. 1 (a), Fig. 1 (b), and Fig. 1 (c) represent the first, second, and third movements consecutively.

Many swarm-based metaheuristics have adopted these three movements in various ways. Table 1 presents the mechanism in the guided search performed in several latest metaheuristics. The neighborhood search during iteration is reviewed for metaheuristics that performs same guided search as in CLO [11], RSLBO [23], and NGO [12].

Table 1 shows that the corresponding member performs the movement in almost all metaheuristics while the reference remains static. Meanwhile, in a few metaheuristics, the reference performs movement too. Table 1 also indicates several options in choosing the reference, whether it is a global best member, a local best member, a randomly selected member, a randomized member, a randomly selected better member, or a combination.

In some metaheuristics, the quality comparison is implemented to increase the probability of improvement by moving toward a better solution or avoiding a worse one. Meanwhile, as stated

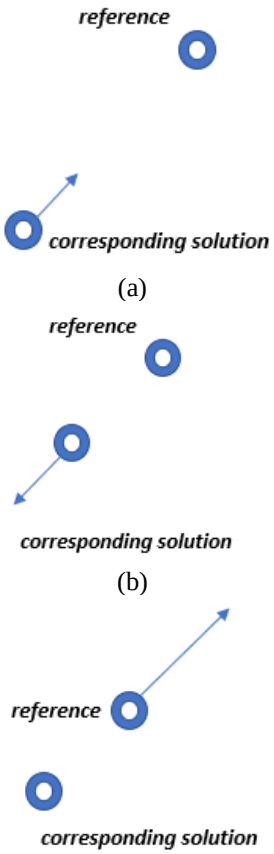


Figure. 1 Common guided search in swarm-based metaheuristic

previously, this movement does not guarantee improvement; in particular, it leads to local optimal entrapment.

Based on this circumstance, several spaces exist for proposing a new swarm-based metaheuristic, which motivates this work. First, a new approach can be chosen by activating the reference so that both the corresponding member and its reference perform searching. Second, in a particular condition, moving toward a worse solution or avoiding a better solution is needed to avoid the local optimal entrapment.

## 3. Model

The central concept of the proposed SMO is based on the behavior of two magnets, especially the push and pull behavior. When two magnets with different poles are close, these two magnets will push toward each other. On the other side, when two magnets with the same pole are close, these two magnets will pull away from each other. This behavior is adopted in the proposed SMO.

As a swarm-based metaheuristic, the system can be perceived as a collection or swarm of magnets. Each magnet represents each solution. The interaction

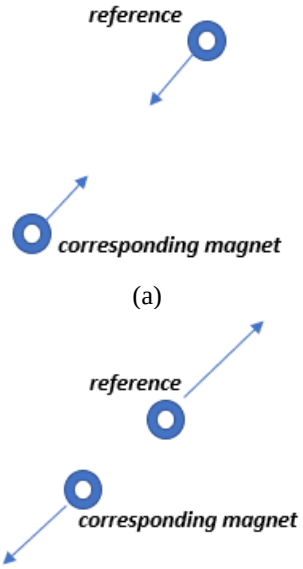


Figure. 2 Push and pull behaviour

between a magnet and its reference guides the former's movement within the search space. During each interaction, two movements occur, one for the corresponding magnet and another for the reference. This movement can be pushed toward or pulled away from each other. This process is visualized using Fig. 2, where the push behavior is presented in Fig. 2 (a) while the pulling behavior is presented in Fig. 2 (b). Each movement generates a candidate. It means there are two candidates in every interaction. Then, the best candidate between these two candidates is selected to compare with the corresponding magnet. This final candidate replaces the current corresponding magnet only if it can improve the corresponding magnet. After a magnet moves to a new location, the global best magnet will be updated. This corresponding magnet replaces the current value of the global best magnet only if the improvement occurs.

There are three phases performed sequentially by each magnet in every iteration. Interaction is performed in every phase. The corresponding magnet interacts with the global best magnet in the first phase. In the second phase, the corresponding magnet interacts with a randomly selected magnet among the swarm of magnets. In the third phase, the corresponding magnet interacts with a randomized magnet within the search space.

The iteration stochastically determines and controls the decision between push and pull. During early iterations, the magnet prefers pulling away rather than pushing towards, but this preference decreases as the iteration increases. In the late iteration, the corresponding magnet prefers pushing

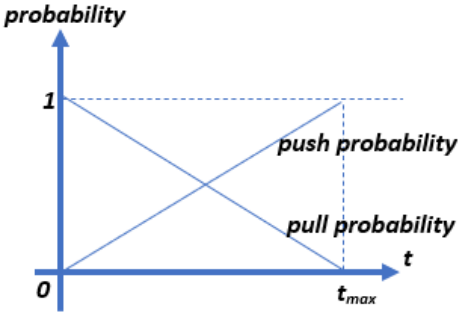


Figure. 3 Probability of push and pull during iteration

| | |
|---|---|
| $c_{11}$ | first candidate of the first phase |
| $c_{12}$ | second candidate of the first phase |
| $c_{21}$ | first candidate of the second phase |
| $c_{22}$ | second candidate of the second phase |
| $c_{31}$ | first candidate of the third phase |
| $c_{32}$ | second candidate of the third phase |
| $c_b$ | best candidate |
| $f$ | objective function |
| $m$ | magnet |
| $M$ | collection of magnets |
| $m_b$ | global best magnet |
| $m_t$ | targeted magnet |
| $m_s$ | selected magnet |
| $m_l$ | the lower boundary of the magnet |
| $m_u$ | the upper boundary of the magnet |
| $r_1$, $r_3$ | floating point random number between 0 and 1 |
| $r_2$ | integer random number between 1 and 2 |
| $t$ | iteration |
| $t_m$ | maximum iteration |
| $th$ | threshold |

towards rather than pulling away. The probability of push and pull during the iteration is visualized using Fig. 3. The pushing behavior represents intensification, while the pulling behavior represents diversification. Below is the annotation used in the formalization of the model. The formalization of the proposed SMO is presented in Algorithm 1. Meanwhile, Eq. (1) to Eq. (12) are used to explain each process.

As a metaheuristic, SMO consists of two steps. The first step is initialization, presented in lines 2 to 6 in algorithm 1. The second step is iteration which is presented in lines 7 to 25 in algorithm 25. In the initialization, each magnet is distributed uniformly within the search space as presented in Eq. (1). Then, the global best magnet is updated by using Eq. (2), where the objective score between the corresponding magnet and the global best magnet is considered during this evaluation for the update.

$$m = U(m_l, m_u) \tag{1}$$

| | algorithm 1: swarm magnetic optimizer |
|---|---|
| 1 | **begin** |
| 2 | **for all** $m$ in $M$ |
| 3 | distribute $m$ uniformly using Eq. (1) |
| 5 | update $m_b$ using Eq. (2) |
| 6 | **end for** |
| 7 | **for** $t$=1: $t_m$ |
| 8 | define $th$ using Eq. (3) |
| 9 | define $m_t$ using Eq. (1) |
| 10 | **for all** $m$ **in** $M$ |
| 11 | calculate $c_{11}$, $c_{12}$ using Eq. (4), Eq. (5) |
| 12 | find $c_b$ using Eq. (6) |
| 13 | update $m$ using Eq. (7) |
| 14 | update $m_b$ using Eq. (2) |
| 15 | define $m_s$ using Eq. (8) |
| 16 | calculate $c_{21}$, $c_{22}$ using Eq. (9), Eq. (10) |
| 17 | find $c_b$ using Eq. (6) |
| 18 | update $m$ using Eq. (7) |
| 19 | update $m_b$ using Eq. (2) |
| 20 | calculate $c_{31}$, $c_{32}$ using Eq. (11), Eq. (12) |
| 21 | find $c_b$ using Eq. (6) |
| 22 | update $m$ using Eq. (7) |
| 23 | update $m_b$ using Eq. (2) |
| 24 | **end for** |
| 25 | **end for** |
| 26 | **end** |
| 27 | **output:** $m_b$ |

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

There are two processes determined at the beginning of every iteration. The first process calculates the threshold, which is formalized using Eq. (3). The second process determines the targeted magnet used for the reference in the third phase.

$$th = \frac{t}{t_m} \tag{3}$$

The first phase is presented in lines 11 to 14 in Algorithm 1. It is shown that the global best magnet is used for reference in this first phase. In the first phase, two candidates are generated using Eq. (4) and (5), respectively. Eq. (6) selects the best candidate, and Eq. (7) determines whether the best candidate should replace the corresponding magnet's current value. Eq. (7) is also used in the second and third phases to decide whether the best candidate should replace the current value of the corresponding magnet.

$$c_{11} = \begin{cases} m + r_2.(m_b - r_3.m), r_1 < th \\ m + r_2.(m - r_3.m_b), else \end{cases} \tag{4}$$

$$c_{12} = \begin{cases} m_b + r_2.(m - r_3.m_b), r_1 < th \\ m_b + r_2.(m_b - r_3.m), else \end{cases} \quad (5)$$

$$c_b = \begin{cases} c_{*1}, f(c_{*1}) < f(c_{*2}) \\ c_{*2}, else \end{cases} \quad (6)$$

$$m' = \begin{cases} c_b, f(c_b) < f(m) \\ m, else \end{cases} \quad (7)$$

The second phase is presented in lines 15 to 19 in Algorithm 1. In the beginning, a reference is randomly selected among the swarm of magnets, and this process is formalized using Eq. (8). In this second phase, the first candidate is determined by using Eq. (9). In contrast, the second candidate is determined by using Eq. (10).

$$m_s = U(M) \quad (8)$$

$$c_{21} = \begin{cases} m + r_2.(m_s - r_3.m), r_1 < th \\ m + r_2.(m - r_3.m_s), else \end{cases} \quad (9)$$

$$c_{22} = \begin{cases} m_s + r_2.(m - r_3.m_s), r_1 < th \\ m_s + r_2.(m_s - r_3.m), else \end{cases} \quad (10)$$

The third phase is presented in lines 20 to 23 in Algorithm 1. In this third phase, the first candidate is generated using Eq. (11), while the second is generated using Eq. (12).

$$c_{31} = \begin{cases} m + r_2.(m_t - r_3.m), r_1 < th \\ m + r_2.(m - r_3.m_t), else \end{cases} \quad (11)$$

$$c_{32} = \begin{cases} m_t + r_2.(m - r_3.m_t), r_1 < th \\ m_t + r_2.(m_t - r_3.m), else \end{cases} \quad (12)$$

## 4. Simulation and result

In this section, the proposed SMO is evaluated through a performance competition with five other metaheuristics in the first part of the evaluation. The second part of the evaluation will be described separately. The second part is the evaluation of the hyperparameter. Due to their extensive coverage, the 23 classic functions are chosen as the problems. The list of these 23 functions is presented in Table 2.

In the first evaluation, there are five competitors: MLBO, GPA, ZOA, COA, and CLO. These five competitors are the latest ones. All these competitors deploy a strict acceptance approach because, in many previous works, strict acceptance has been proven better than non-strict acceptance. GPA is the only metaheuristic that generates multiple candidates in a single search. Meanwhile, the other metaheuristics, including SMO, generates a single candidate only. MLBO is the only competitor that performs single-phase search only.

Table 2 shows that these 23 functions can be split into three groups: high-dimension unimodal, high-dimension multimodal, and fixed-dimension multimodal functions. The result is presented in Table 3, Table 4, and Table 5, which represent the first, second, and third groups consecutively. In this test, the population size is five, and the maximum iteration is 20.

Table 3 shows the superior performance of SMO among its competitors. SMO becomes the best performer in solving almost all high-dimension unimodal functions. All metaheuristics achieve the same result, the optimal global solution for solving Schwefel 2.22. Meanwhile, MLBO is the worst performer, and GPA is the second worst performer in solving five functions. Both SMO and ZOA perform equally in solving Sphere. The result achieved by SMO and ZOA is far better than the other metaheuristics. The vast performance gap between SMO and MLBO and GPA is also found in Schwefel 1.2, Schwefel 2.21, Rosenbrock, Step, and Quartic.

Table 4 shows that competition among these metaheuristics becomes tighter in solving the high-dimension multimodal functions. SMO performs in the first rank in solving three functions (Rastrigin, Griewank, and Penalized), second rank in solving one function (Penalized 2), and fourth rank in solving two functions (Schwefel and Ackley). Table 4 also shows that the performance among metaheuristics in solving Schwefel is narrow, although the performance gap between the best and worst performers in other functions is wide.

Table 5 shows that the competition among metaheuristics is tough in solving fixed-dimension multimodal functions. SMO performs best in only two functions (Six hump camel and Hartman 3). Meanwhile, SMO is in the second rank in two functions (Shekel Foxholes and Kowalik), third rank in one function (Goldstein-Price), fourth rank in two functions (Branin and Shekel 5), and fifth rank in three functions (Hartman 6, Shekel 7, and Shekel 10).

Table 5 also shows that the performance gap among metaheuristics is narrow compared to the circumstance in high dimension function. The performance gap is tight in eight functions (Kowalik, Six Hump Camel, Branin, Hartman 3, Hartman 6, Shekel 5, Shekel 7, and Shekel 10). It means that SMO is still competitive in solving fixed-dimension functions, although it is not superior to high-dimension functions.

The result in Table 3 to Table 5 is summarized in Table 6. Table 6 presents SMO's superiority over its competitors based on the group of functions. The data

Table 2. List of 23 functions

| No | Function | Type | Dim | Space | Target |
|---|---|---|---|---|---|
| 1 | Sphere | high dimension unimodal function | 50 | [-100, 100] | 0 |
| 2 | Schwefel 2.22 | high dimension unimodal function | 50 | [-100, 100] | 0 |
| 3 | Schwefel 1.2 | high dimension unimodal function | 50 | [-100, 100] | 0 |
| 4 | Schwefel 2.21 | high dimension unimodal function | 50 | [-100, 100] | 0 |
| 5 | Rosenbrock | high dimension unimodal function | 50 | [-30, 30] | 0 |
| 6 | Step | high dimension unimodal function | 50 | [-100, 100] | 0 |
| 7 | Quartic | high dimension unimodal function | 50 | [-1.28, 1.28] | 0 |
| 8 | Schwefel | high dimension multimodal function | 50 | [-500, 500] | -20,945 |
| 9 | Ratsrigin | high dimension multimodal function | 50 | [-5.12, 5.12] | 0 |
| 10 | Ackley | high dimension multimodal function | 50 | [-32, 32] | 0 |
| 11 | Griewank | high dimension multimodal function | 50 | [-600, 600] | 0 |
| 12 | Penalized | high dimension multimodal function | 50 | [-50, 50] | 0 |
| 13 | Penalized 2 | high dimension multimodal function | 50 | [-50, 50] | 0 |
| 14 | Shekel Foxholes | fixed dimension multimodal function | 2 | [-65, 65] | 1 |
| 15 | Kowalik | fixed dimension multimodal function | 4 | [-5, 5] | 0.0003 |
| 16 | Six Hump Camel | fixed dimension multimodal function | 2 | [-5, 5] | -1.0316 |
| 17 | Branin | fixed dimension multimodal function | 2 | [-5, 5] | 0.398 |
| 18 | Goldstein-Price | fixed dimension multimodal function | 2 | [-2, 2] | 3 |
| 19 | Hartman 3 | fixed dimension multimodal function | 3 | [1, 3] | -3.86 |
| 20 | Hartman 6 | fixed dimension multimodal function | 6 | [0, 1] | -3.32 |
| 21 | Shekel 5 | fixed dimension multimodal function | 4 | [0, 10] | -10.153 |
| 22 | Shekel 7 | fixed dimension multimodal function | 4 | [0, 10] | -10.402 |
| 23 | Shekel 10 | fixed dimension multimodal function | 4 | [0, 10] | -10.536 |

Table 3. Fitness score comparison in solving high dimension unimodal functions

| F | Parameter | MLBO [21] | GPA [8] | ZOA [9] | COA [10] | CLO [11] | SMO |
|---|---|---|---|---|---|---|---|
| 1 | mean | $2.8645\text{x}10^{4}$ | $1.1244\text{x}10^{4}$ | 0.0008 | 3.2444 | $2.0701\text{x}10^{1}$ | 0.0008 |
| | standard dev | $5.3100\text{x}10^{3}$ | $2.6471\text{x}10^{3}$ | 0.0012 | 2.4053 | $1.2466\text{x}10^{1}$ | 0.0013 |
| | mean rank | 6 | 5 | 1 | 3 | 4 | 1 |
| 2 | mean | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | standard dev | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | mean rank | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | mean | $6.8084\text{x}10^{4}$ | $3.9778\text{x}10^{4}$ | $1.6941\text{x}10^{2}$ | $7.5127\text{x}10^{3}$ | $1.9333\text{x}10^{4}$ | $5.6140\text{x}10^{1}$ |
| | standard dev | $2.8906\text{x}10^{4}$ | $1.0676\text{x}10^{4}$ | $3.2175\text{x}10^{2}$ | $7.3670\text{x}10^{3}$ | $1.0256\text{x}10^{4}$ | $7.6023\text{x}10^{1}$ |
| | mean rank | 6 | 5 | 2 | 3 | 4 | 1 |
| 4 | mean | $5.0506\text{x}10^{1}$ | $4.2358\text{x}10^{1}$ | 0.0709 | 4.2816 | 9.7554 | 0.0838 |
| | standard dev | 5.0704 | $1.4808\text{x}10^{1}$ | 0.0509 | 1.9818 | 7.5739 | 0.0372 |
| | mean rank | 6 | 5 | 1 | 3 | 4 | 2 |
| 5 | mean | $2.4788\text{x}10^{7}$ | $4.2962\text{x}10^{6}$ | $4.8955\text{x}10^{1}$ | $1.5018\text{x}10^{2}$ | $7.3821\text{x}10^{2}$ | $4.8920\text{x}10^{1}$ |
| | standard dev | $8.4343\text{x}10^{6}$ | $1.7169\text{x}10^{6}$ | 0.0325 | $1.5383\text{x}10^{2}$ | $6.7116\text{x}10^{2}$ | 0.0439 |
| | mean rank | 6 | 5 | 2 | 3 | 4 | 1 |
| 6 | mean | $2.7414\text{x}10^{4}$ | $1.0514\text{x}10^{4}$ | $1.0708\text{x}10^{1}$ | $1.5564\text{x}10^{1}$ | $3.4657\text{x}10^{1}$ | $1.0438\text{x}10^{1}$ |
| | standard dev | $4.0526\text{x}10^{3}$ | $1.5871\text{x}10^{3}$ | 0.5725 | 2.3770 | $1.8700\text{x}10^{1}$ | 0.3540 |
| | mean rank | 6 | 5 | 2 | 3 | 4 | 1 |
| 7 | mean | $2.4160\text{x}10^{1}$ | 4.1759 | 0.0221 | 0.0879 | 0.0835 | 0.0133 |
| | standard dev | $1.4549\text{x}10^{1}$ | 1.1952 | 0.0122 | 0.0400 | 0.0494 | 0.0093 |
| | mean rank | 6 | 5 | 2 | 4 | 3 | 1 |

Table 4. Fitness score comparison in solving high-dimension multimodal functions

| F | Parameter | MLBO [21] | GPA [8] | ZOA [9] | COA [10] | CLO [11] | SMO |
|---|---|---|---|---|---|---|---|
| 8 | mean | $-3.5269\text{x}10^{3}$ | $-5.1767\text{x}10^{3}$ | $-2.8683\text{x}10^{3}$ | $-4.7466\text{x}10^{3}$ | $-4.4451\text{x}10^{3}$ | $-4.4367\text{x}10^{3}$ |
| | standard dev | $7.0070\text{x}10^{2}$ | $7.9327\text{x}10^{2}$ | $6.1864\text{x}10^{2}$ | $5.6517\text{x}10^{2}$ | $6.1889\text{x}10^{2}$ | $4.8114\text{x}10^{2}$ |
| | mean rank | 5 | 1 | 6 | 2 | 3 | 4 |
| 9 | mean | $4.5401\text{x}10^{2}$ | $4.1164\text{x}10^{2}$ | 0.0079 | 4.1679 | $1.2851\text{x}10^{2}$ | 0.0067 |
| | standard dev | $4.8060\text{x}10^{1}$ | $3.6427\text{x}10^{1}$ | 0.0211 | 4.2408 | $6.4398\text{x}10^{1}$ | 0.0111 |
| | mean rank | 6 | 5 | 2 | 3 | 4 | 1 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 10 | mean | $1.7181 \times 10^{1}$ | $1.3832 \times 10^{1}$ | 0.0052 | 0.4640 | 1.8293 | $1.0999 \times 10^{1}$ |
| | standard dev | 0.5836 | 0.6989 | 0.0024 | 0.1892 | 0.6243 | $1.0264 \times 10^{1}$ |
| | mean rank | 6 | 5 | 1 | 2 | 3 | 4 |
| 11 | mean | $2.5520 \times 10^{2}$ | $9.0778 \times 10^{1}$ | 0.0090 | 0.5662 | 1.0674 | 0.0086 |
| | standard dev | $4.5564 \times 10^{1}$ | $1.5608 \times 10^{1}$ | 0.0294 | 0.3236 | 0.3863 | 0.0319 |
| | mean rank | 6 | 5 | 2 | 3 | 4 | 1 |
| 12 | mean | $2.0627 \times 10^{7}$ | $2.3164 \times 10^{5}$ | 1.0598 | 1.2221 | 1.4696 | 0.9404 |
| | standard dev | $1.5016 \times 10^{7}$ | $1.8363 \times 10^{5}$ | 0.1468 | 0.2260 | 0.3814 | 0.1213 |
| | mean rank | 6 | 5 | 2 | 3 | 4 | 1 |
| 13 | mean | $6.8830 \times 10^{7}$ | $6.3041 \times 10^{6}$ | 3.1206 | 4.0046 | 6.1079 | 3.1340 |
| | standard dev | $4.0582 \times 10^{7}$ | $3.2578 \times 10^{6}$ | 0.0552 | 0.4259 | 2.8029 | 0.0603 |
| | mean rank | 6 | 5 | 1 | 3 | 4 | 2 |

Table 5. Fitness score comparison in solving fixed dimension multimodal functions

| F | Parameter | MLBO [21] | GPA [8] | ZOA [9] | COA [10] | CLO [11] | SMO |
|---|---|---|---|---|---|---|---|
| 14 | mean | $1.3264 \times 10^{1}$ | 5.6492 | 3.1506 | 5.2939 | 6.4448 | 4.3548 |
| | standard dev | 6.5580 | 3.9200 | 2.1696 | 3.2633 | 3.7604 | 2.7929 |
| | mean rank | 6 | 4 | 1 | 3 | 5 | 2 |
| 15 | mean | 0.0366 | 0.0123 | 0.0034 | 0.0023 | 0.0048 | 0.0030 |
| | standard dev | 0.0299 | 0.0102 | 0.0052 | 0.0021 | 0.0047 | 0.0053 |
| | mean rank | 6 | 5 | 3 | 1 | 4 | 2 |
| 16 | mean | -0.9025 | -1.0203 | -1.0291 | -1.0303 | -1.0305 | -1.0306 |
| | standard dev | 0.1889 | 0.0116 | 0.0051 | 0.0019 | 0.0016 | 0.0020 |
| | mean rank | 6 | 5 | 4 | 3 | 2 | 1 |
| 17 | mean | 0.5050 | 0.4043 | 0.4010 | 0.4020 | 0.4003 | 0.4026 |
| | standard dev | 0.1689 | 0.0050 | 0.0067 | 0.0012 | 0.0053 | 0.0090 |
| | mean rank | 6 | 5 | 2 | 3 | 1 | 4 |
| 18 | mean | $2.5947 \times 10^{1}$ | 3.0670 | 3.1012 | 8.9158 | $1.1096 \times 10^{1}$ | 3.1460 |
| | standard dev | $3.7037 \times 10^{1}$ | 0.0572 | 0.2389 | $1.6070 \times 10^{1}$ | $1.9451 \times 10^{1}$ | 0.6249 |
| | mean rank | 6 | 1 | 2 | 4 | 5 | 3 |
| 19 | mean | -0.0353 | -0.0495 | -0.0495 | -0.0495 | -0.0495 | -0.0495 |
| | standard dev | 0.0164 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | mean rank | 6 | 1 | 1 | 1 | 1 | 1 |
| 20 | mean | -2.6473 | -3.2120 | -2.9871 | -3.0724 | -2.9959 | -2.9425 |
| | standard dev | 0.4478 | 0.0928 | 0.1647 | 0.1664 | 0.1948 | 0.1616 |
| | mean rank | 6 | 1 | 4 | 2 | 3 | 5 |
| 21 | mean | -2.0629 | -5.4980 | -2.9400 | -4.3721 | -4.5464 | -3.3548 |
| | standard dev | 2.0741 | 2.8134 | 1.1973 | 1.8207 | 2.0702 | 1.9131 |
| | mean rank | 6 | 1 | 5 | 3 | 2 | 4 |
| 22 | mean | -1.8598 | -4.3249 | -3.2309 | -5.3018 | -4.9378 | -2.5638 |
| | standard dev | 0.7794 | 2.4282 | 1.4528 | 2.4630 | 2.0911 | 0.8129 |
| | mean rank | 6 | 3 | 4 | 1 | 2 | 5 |
| 23 | mean | -2.2247 | -5.7304 | -3.9953 | -3.9771 | -4.2944 | -3.3806 |
| | standard dev | 1.3760 | 2.6453 | 2.0663 | 1.8925 | 1.4361 | 1.7756 |
| | mean rank | 6 | 1 | 3 | 4 | 2 | 5 |

Table 6. Group-based superiority of SMO

| Group | Number of Functions Where SMO is Better | | | | |
|---|---|---|---|---|---|
| | MLBO [21] | GPA [8] | ZOA [9] | COA [10] | CLO [11] |
| 1 | 6 | 6 | 4 | 6 | 6 |
| 2 | 6 | 5 | 4 | 4 | 4 |
| 3 | 10 | 4 | 3 | 3 | 4 |
| Total | 22 | 15 | 11 | 13 | 14 |

represents the number of functions where SMO is better than its competitor. Table 6 strengthens the circumstance that SMO is superior in solving high-dimension functions, whether unimodal or multimodal functions. Meanwhile, SMO is less superior in fixed dimension functions, except compared to MLBO. SMO is superior to MLBO by achieving better performance in ten out of ten functions.

Table 7. Relation between maximum iteration and the average fitness score

| F | Average Fitness Score | | Significantly Improved? |
|---|---|---|---|
| | $t_m$ = 40 | $t_m$ = 80 | |
| 1 | 0.0000 | 0.0000 | no |
| 2 | 0.0000 | 0.0000 | no |
| 3 | 0.0017 | 0.0000 | yes |
| 4 | 0.0000 | 0.0000 | no |
| 5 | $4.8897 \times 10^{1}$ | $4.8928 \times 10^{1}$ | no |
| 6 | $1.0317 \times 10^{1}$ | $1.0202 \times 10^{1}$ | no |
| 7 | 0.0071 | 0.0019 | yes |
| 8 | $-4.7562 \times 10^{3}$ | $-5.1448 \times 10^{3}$ | no |
| 9 | 0.0000 | 0.0000 | yes |
| 10 | $1.6281 \times 10^{1}$ | 8.6025 | no |
| 11 | 0.0040 | 0.0000 | yes |
| 12 | 0.9682 | 1.0151 | no |
| 13 | 3.0189 | 3.0151 | no |
| 14 | 2.1839 | 1.9064 | no |
| 15 | 0.0043 | 0.0017 | yes |
| 16 | -1.0314 | -1.0314 | no |
| 17 | 0.3998 | 0.3981 | no |
| 18 | 3.0284 | 3.0000 | no |
| 19 | -0.0495 | -0.0495 | no |
| 20 | -3.0953 | -3.1052 | no |
| 21 | -4.5351 | -4.4114 | no |
| 22 | -3.9071 | -5.2746 | no |
| 23 | -4.5654 | -5.0012 | no |

Table 8. Relation between swam size and the average fitness score

| F | Average Fitness Score | | Significantly Improved? |
|---|---|---|---|
| | $n(M)$ = 10 | $n(M)$ = 20 | |
| 1 | 0.0005 | 0.0011 | no |
| 2 | 0.0000 | 0.0000 | no |
| 3 | $1.1868 \times 10^{2}$ | $9.8309 \times 10^{1}$ | no |
| 4 | 0.0735 | 0.0621 | no |
| 5 | $4.9231 \times 10^{1}$ | $4.8869 \times 10^{1}$ | no |
| 6 | 9.3452 | 8.7973 | no |
| 7 | 0.0101 | 0.0064 | no |
| 8 | $-4.6261 \times 10^{3}$ | $-5.0398 \times 10^{3}$ | no |
| 9 | 0.0091 | 0.0030 | yes |
| 10 | $1.4368 \times 10^{1}$ | $1.3369 \times 10^{1}$ | no |
| 11 | 0.0115 | 0.0027 | yes |
| 12 | 0.8240 | 0.6979 | no |
| 13 | 3.1349 | 3.0632 | no |
| 14 | 1.9687 | 1.1863 | no |
| 15 | 0.0030 | 0.0038 | no |
| 16 | -1.0314 | -1.0316 | no |
| 17 | 0.3981 | 0.3981 | no |
| 18 | 3.0000 | 3.0000 | no |
| 19 | -0.0495 | -0.0495 | no |
| 20 | -3.0651 | -3.1408 | no |
| 21 | -4.2685 | -5.8953 | no |
| 22 | -4.1789 | -5.9434 | no |
| 23 | -4.2616 | -5.6032 | no |

Meanwhile, in the third group, SMO is better than GPA and CLO in four functions and better than ZOA and COA in three functions. Overall, SMO is better than MLBO, GPA, ZOA, COA, and CLO in consecutively, 22, 15, 11, 13, and 14 functions. Although SMO is better than ZOA in 11 functions, it is draw in 3. It means that ZOA is better than SMO in 9 functions.

The second evaluation is regarding the hyperparameter test. As the SMO has no adjusted parameters, the hyperparameter test is performed to evaluate the influence of the maximum iteration and swarm size. In this evaluation, the 23 functions are used as problems. In the test to evaluate the relation between maximum iteration and the performance of SMO, there are two values of maximum iteration: 40 and 80. The result is presented in Table 7. Meanwhile, in the test to evaluate the relation between swarm size and the performance of SMO, there are two values of swarm size: 10 and 20. The result is presented in Table 8.

Table 7 indicates that the increase in maximum iteration improves the performance of SMO in five functions. Two functions are high-dimension unimodal, two are high-dimension multimodal, and one is fixed-dimension multimodal. Meanwhile, in high maximum iteration, SMO can find the global optimal of eight functions (Sphere, Schwefel 2.22, Schwefel 1.2, Schwefel 2.21, Rastrigin, Griewank, Branin, and Goldstein Price).

Table 8 shows that the increase in swarm size improves the performance of SMO significantly only in two functions. These two functions are Rastrigin and Griewank. Meanwhile, in the high swarm size circumstance, there are four functions with their global optimal: Schwefel 2.22, Six Hump Camel, Branin, and Goldstein Price.

## 5. Discussion

The in-depth analysis of SMO is conducted based on the intensification and diversification capabilities of SMO. The superior performance of SMO in solving high-dimensional functions with a single peak attest to its total intensification capacity. On the other hand, the superiority of SMO in solving high-dimension multimodal functions proves that SMO also has superior diversification capabilities. Meanwhile, the fierce competition in solving fixed-dimension multimodal functions shows that the competition in proposing a new metaheuristic becomes more difficult.

The superiority of SMO to these five competitors shows that implementing both getting closer or moving away from the better reference can be an

alternative for improvement rather than just getting closer to the better reference or avoiding worse reference as implemented in these all competitors. Pulling away gives two advantages. First, it allows tracing other search spaces in case getting closer to the reference means getting closer to the local optimal entrapment. Second, it also forces the optimal global solution to move on and not just sit as a reference. The corresponding solution performs the movement or search in these five competitors while the reference remains static.

The competitiveness of ZOA can be traced to the neighbourhood search implemented in ZOA. In the second phase, the zebra's escaping strategy inspires the search when a predator attack [9]. In this neighbourhood search, the local search space is narrow and reduced linearly during the iteration. Meanwhile, the SMO does not perform any neighbourhood search because all movements performed in SMO are guided searches. This circumstance can be seen as an opportunity to improve the current form of SMO by implementing neighbourhood search. In ZOA, this neighbourhood search is not mandatory but determined stochastically [9]. In the second phase of ZOA, there are two choices in the second phase: neighbourhood search and guided search toward a randomized solution [9].

The neighbourhood search is also performed in COA [10], CLO [11], and GPA [8]. The local search space is also reduced during the iteration of these three metaheuristics. It starts with a vast space. This circumstance indicates that the short jump is still needed besides the long jump. The long jump can make the agent traces the search space faster. However, the short jump seems necessary in tackling the problem where the optimal global solution lies in the narrow area. The long jump may throw the agent away from this narrow area.

The almost absolute superiority of SMO to MLBO indicates that multiple searches performed explicitly in multiple phases in the iteration stage become the critical factor in creating a significant performance gap between SMO and MLBO. Although MLBO uses not only a single reference, MLBO performs only one search because it mixes the best solution and randomized solution to become a single reference [21].

The complexity analysis of SMO can be divided into two stages. The complexity can be presented as $O(n(X)d)$ in the initialization stage. This representation means that the computational complexity of SMO in the initialization stage is proportional to the swarm size and the dimension. Meanwhile, in the iteration stage, the complexity can be presented as $O(6n(X)d.t_m)$. It means that the complexity is proportional to the maximum iteration besides the swarm size and the dimension. Meanwhile, six movements are distributed into three phases, and each phase performs two movements.

## 6. Conclusion

The swarm magnetic optimizer (SMO), a new swarm-based metaheuristic that adopts the magnet's behavior, has been presented in this work. SMO's performance has also been evaluated through simulation that challenges SMO to find the optimal solution for the 23 functions. Moreover, SMO has competed with five latest metaheuristics. The result shows that SMO performs well by outperforming MLBO, GPA, ZOA, COA, and CLO consecutively in the 22, 15, 11, 13, and 14 functions. SMO is superior to these five metaheuristics in solving high-dimension functions. Meanwhile, although the performance gap is narrow, SMO is less superior to GPA, ZOA, COA, and CLO in solving fixed-dimension multimodal functions.

There are several opportunities for future work based on the development of SMO. SMO can be modified to solve multi-objective problems or combinatorial problems. More studies can also be conducted to implement SMO in solving various real-world optimization problems in various sectors, whether engineering or non-engineering problems. The hybridization of SMO with other metaheuristics is also challenging work. Specifically, the hybridization of SMO and ZOA as its most difficult competitor to beat is interesting.

## Conflicts of interest

The authors declare no conflict of interest.

## Author contributions

Conceptualization, Kusuma; methodology, Kusuma; software, Kusuma; formal analysis, Kusuma and Hasibuan; investigation, Kusuma and Hasibuan; data curation, Kusuma; writing-original paper draft, Kusuma; writing-review and editing: Hasibuan; supervision: Hasibuan; funding acquisition, Kusuma.

## Acknowledgments

This work is funded and supported by Telkom University, Indonesia.

## References

[1] D. Acharya and D. K. Das, “A Novel Human Conception Optimizer for Solving Optimization

Problems", *Scientific Reports*, Vol. 12, ID. 21631, pp. 1-20, 2022.

[2] M. Dehghani and P. Trojovsky, "Osprey Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Solving Engineering Optimization Problems", *Frontiers in Mechanical Engineering*, Vol. 8, ID. 1126450, pp. 1-43, 2023.

[3] P. Trojovsky, M. Dehghani, and P. Hanus, "Siberian Tiger Optimization: A New Bio-Inspired Metaheuristic Algorithm for Solving Engineering Optimization Problems", *IEEE Access*, Vol. 10, pp. 132396-132431, 2022.

[4] B. Abdollahzadeh, F. S. Gharehchopogh, and S. Mirjalili, "African Vultures Optimization Algorithm: A New Nature-Inspired Metaheuristic Algorithm for Global Optimization Problems", *Computers & Industrial Engineering*, Vol. 158, ID. 107408, 2021.

[5] N. Chopra and M. M. Ansari, "Golden Jackal Optimization: A Novel Nature-Inspired Optimizer for Engineering Applications", *Expert Systems with Applications*, Vol. 198, ID. 116924, pp. 1-15, 2022.

[6] S. Suyanto, A. A. Ariyanto, and A. F. Ariyanto, "Komodo Mlipir Algorithm", *Applied Soft Computing*, Vol. 114, pp. 1–17, 2022.

[7] 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.

[8] 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, doi: 10.22266/ijies2022.1231.18.

[9] 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.

[10] M. Dehghani, Z. Montazeri, E. Trojovska, and P. Trojovsky, "Coati Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems", *Knowledge-Based Systems*, Vol. 259, ID. 110011, pp. 1-43, 2023.

[11] E. Trojovska and M. Dehghani, "Clouded Leopard Optimization: A New Nature-Inspired Optimization Algorithm", *IEEE Access*, Vol. 10, pp. 102876-102906, 2022.

[12] M. Dehghani, S. Hubalovsky, and P. Trojovsky, "Northern Goshawk Optimization: A New Swarm-Based Algorithm for Solving Optimization Problems", *IEEE Access*, Vol. 9, pp. 162059–162080, 2021.

[13] 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, doi: 10.22266/ijies2022.0831.14.

[14] E. Trojovska, M. Dehghani, and P. Trojovsky, "Fennec Fox Optimization: A New Nature-Inspired Optimization Algorithm", *IEEE Access*, Vol. 10, pp. 84417-84443, 2022.

[15] J. Xue and B. Shen, "A Novel Swarm Intelligence Optimization Approach: Sparrow Search Algorithm", *Systems Science & Control Engineering*, Vol. 8, No. 1, pp. 22-34, 2020.

[16] L. Abualigah, M. A. Elaziz, P. Sumari, Z. W. Geem, and A. H. Gandomi, "Reptile Search Algorithm (RSA): A Nature-Inspired Meta-Heuristic Optimizer", *Expert Systems with Applications*, Vol. 191, ID. 116158, pp. 1-33, 2022.

[17] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine Predators Algorithm: A Nature-inspired Metaheuristic", *Expert System with Applications*, Vol. 152, ID: 113377, 2020.

[18] M. A. A. Betar, Z. A. A. Alyasseri, M. A. Awadallah, and I. A. Doush, "Coronavirus Herd Immunity Optimizer (CHIO)", *Neural Computing and Applications*, Vol. 33, pp. 5011–5042, 2021.

[19] A. M. Khalid, K. M. Hosny, and S. Mirjalili, "COVIDOA: A Novel Evolutionary Optimization Algorithm Based on Coronavirus Disease Replication Lifecycle", *Neural Computing and Applications*, Vol. 34, pp. 22465-22492, 2022.

[20] M. Dehghani, Z. Montazeri, A. Dehghani, R. A. R. 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, doi: 10.22266/ijies2020.1231.32.

[21] F. Zeidabadi, S. 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. 472–479, 2021, doi: 10.22266/ijies2021.0831.41.

[22] F. A. Zeidabadi, M. Dehghani, and O. P. Malik, "TIMBO: Three Influential Members Based Optimizer", *International Journal of Intelligent*

*Engineering and Systems*, Vol. 14, No. 5, pp. 121-128, 2021, doi: 10.22266/ijies2021.1031.12.

[23] F. A. Zeidabadi, M. Dehghani, and O. P. Malik, "RSLBO: Random Selected Leader Based Optimizer", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 5, pp. 529–538, 2021, doi: 10.22266/ijies2021.1031.46.

[24] P. D. Kusuma and A. Novianty, "Total Interaction Algorithm: A Metaheuristic in Which Each Agent Interacts with All Other Agents", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 1, pp. 224-234, 2023, doi: 10.22266/ijies2023.0228.20.

[25] P. D. Kusuma and A. Novianty, "Multiple Interaction Optimizer: A Novel Metaheuristic and Its Application to Solve Order Allocation Problem", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 2, pp. 440-453, 2023, doi: 10.22266/ijies2023.0430.35.

[26] 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.

[27] 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.