



A New Metaheuristic Algorithm Called Treble Opposite Algorithm and Its Application to Solve Portfolio Selection

Purba Daru Kusuma^{*}, Astri Novianty^{*}

Computer Engineering, Telkom University, Bandung 40258, Indonesia

Corresponding Author Email: purbodaru@telkomuniversity.ac.id

Copyright: ©2024 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/mmep.110326>

ABSTRACT

Received: 15 October 2023

Revised: 3 December 2023

Accepted: 20 December 2023

Available online: 28 March 2024

Keywords:

metaheuristic optimization, swarm intelligence, portfolio optimization, IDX30, banking sector

This work presents a new metaheuristic called treble opposite algorithm (TOA). It consists of three phases. There are two searches that are opposite to each other performed in each phase. In the first phase, the search toward and away from the best solution is carried out. In the second phase, the search toward and away from the middle between two randomly picked solutions is carried out. In the third phase, a neighborhood search around the narrow and large space is carried out. A candidate is selected among the two searches in every phase. TOA is challenged to solve theoretical and practical problems. The 23 functions represent theoretical problems, while the portfolio optimization of stocks in the banking sector listed in IDX30 represents the practical problem. TOA is compared with five metaheuristics: grey wolf optimization (GWO), golden search optimization (GSO), average subtraction-based optimization (ASBO), zebra optimization algorithm (ZOA), and coati optimization algorithm (COA). The result indicates that TOA is superior to its competitors as it is better than GWO, GSO, ZOA, ASBO, and COA in 22, 23, 19, 20, and 19 functions respectively, in handling 23 functions and produces the highest total capital gain in handling portfolio optimization problem. In the future, TOA can be utilized to handle many other real-world optimization problems. Moreover, TOA can be hybridized with other metaheuristics to improve its performance.

1. INTRODUCTION

Optimization becomes an integral part of the wide range of practical works, especially in engineering. The examples are as follows. In the multi-energy system, optimization is needed to minimize the operational expenditures consisting of the energy purchasing cost reduced by energy sales and added with setup cost [1]. In the energy sector, optimization is implemented into wind-based energy systems to minimize the total electricity shortage [2]. Optimization is also used in the battery energy storage system to maximize revenue and minimize long-term capacity fading [3]. In the telecommunication sector, optimization is used in the vehicular ad hoc network (VANET) to increase throughput and reduce delay, packet loss rate, and cost of service [4]. In the manufacturing system, optimization is implemented into production scheduling with objectives such as tardiness, earliness, make-span, number of tardy and early jobs [5].

Optimization plays an important role in engineering because many practical engineering problems need to be solved based on certain objectives. It works by finding the best solution within the solution space or possible solutions [6] and working within the constraints or solution boundaries. Metaheuristics is a long decade tool for optimization. Its popularity comes from its superiority in producing acceptable or high-quality solutions with a limited computational

resource. Moreover, metaheuristic is flexible in handling various optimization problems because it abstracts the problem and focus on its objectives and constraints [7]. As a trial-and-error approach, its stochastic-based method does not guarantee finding the real optimal solution [8].

There are a lot of metaheuristics that already exist in the recent years. Most of them are swarm-based metaheuristics. Many of them use metaphors as the inspiration of their strategy. Many swarm-based metaheuristics imitate animal behavior, such as grey wolf optimization (GWO) [9], marine predator algorithm (MPA) [10], slime mold algorithm (SMA) [11], pelican optimization algorithm (POA) [12], zebra optimization algorithm (ZOA) [13], coati optimization algorithm (COA) [14], butterfly optimization algorithm (BOA) [15], northern goshawk optimization (NGO) [16], guided pelican algorithm (GPA) [17], clouded leopard optimization (CLO) [18], Komodo mlipir algorithm (KMA) [19], cheetah optimizer (CO) [20], cat and mouse based optimizer (CMBO) [21], remora optimization algorithm (ROA) [22], and so on. Certain number of metaheuristics imitate human or social activities, such as modified social force (MSFA) [23], cheb-based optimization algorithm (CBOA) [24], driving training-based optimizer (DTBO) [25], election-based optimization algorithm (EBOA) [26], and so on. Some metaheuristics do not use metaphor but directly use their main strategy for their name, such as average and subtraction-based optimization

(ASBO) [27], golden search optimization (GSO) [28], total interaction algorithm (TIA) [29], and so on.

In almost all swarm-based metaheuristics, guided search becomes the primary strategy, while random or neighborhood search becomes the complementary one. In general, the solution will get closer to the better solution. This better solution can be the global, local, and some best solutions, or a mixture among them. In some metaheuristics, a solution interacts with other solution that is selected randomly. Then, the quality of this randomly selected solution is compared with the corresponding solution. The corresponding solution moves closer to the selected solution if the selected solution is better than the corresponding solution. Otherwise, the corresponding solution chooses to move away. Several metaheuristics also perform the guided search by moving away from the worst solution. This strategy is rational because improvement becomes more probable by moving closer to the better solution or moving away from the worse one.

Unfortunately, in the trial-and-error approach implemented in metaheuristic, getting closer to the better solution never guarantees improvement. Besides, performing neither narrow nor large local search space guarantees improvement too. Getting closer to a better solution may lead to the local optimal entrapment. Neighborhood search with a narrow local search space may lead to stagnation when the current solution is not in the area where the global optimal solution exists. On the other hand, neighborhood search with large local search space may throw away the current solution to the worse area.

The primary objective of this work to introduce a new swarm-based metaheuristic called the treble opposite algorithm (TOA). As its name suggests, TOA consists of three phases that are performed sequentially. The guided searches are performed in the first and second phases, while neighborhood search is performed in the third phase. There are two searches or walks that are opposites to each other. Meanwhile, there are two secondary objectives in this work. The first secondary objective is to evaluate the performance of TOA to solve both theoretical and practical problems. The second secondary objective is to promote the portfolio optimization problem representing the discrete problem as a use case rather than the common mechanical design problem.

The scientific contributions of this work can be described as follows:

- (1) This paper introduces a new swarm-based metaheuristic that performs both getting closer to and away from the reference.
- (2) This new metaheuristic also performs neighborhood searches with both narrow and wide local search space.
- (3) This work evaluates the proposed metaheuristic through simulation to solve the set of 23 classic functions.
- (4) The assessment is also taken by competing the proposed metaheuristic with five new swarm-based metaheuristics: GWO, GSO, ZOA, ASBO, and COA.

This work is carried out based on certain methodology. First, certain number of new swarm-based metaheuristics is reviewed, especially their fundamental strategy. Second, these existing metaheuristics are mapped and then the position of the proposed metaheuristic is investigated and introduced to make clear position to fill the gap in the recent development of metaheuristics. Third, the formal model includes the algorithm and mathematical representation of TOA constructed. Fourth, the assessment of the performance of TOA is carried out by challenging TOA to solve both theoretical and practical problems and its comparison with several latest metaheuristics.

Fifth, the in-depth investigation regarding the assessment result and the drawback to the theoretical aspect is carried out to develop a comprehensive assessment of the proposed TOA, including its strength, weakness, limitation, complexity, and proposal for future studies.

The arrangement of the rest of this paper is as follows. The review regarding new swarm-based metaheuristics, especially related to their metaphor, mechanics, and the use case for assessment, is presented and summarized in section 2. A detailed description of the proposed metaheuristic, which consists of the fundamentals, algorithm, and formalization, is presented in section 3. The assessment of the proposed metaheuristic and its result is presented in section 4. The in-depth analysis regarding the result, findings, linkage to the theoretical basis, and the limitation of this work are discussed in section 5. In the end, the concluding remark and the potential for future works or developments are summarized in section 6.

2. RELATED WORKS

The no-free-lunch (NFL) theory has been becoming one important reason for the massive development of metaheuristics. As stated in NFL theory, there is not any metaheuristic that powerful and superior to solve all kinds of optimization problems [16]. A metaheuristic can perform well in handling some problems but end with mediocre results in handling other ones [16]. On the other hand, there are various problems, especially in engineering, needing to be optimized.

Through decades, there is an evolution in the development of metaheuristics. In the beginning, a lot of metaheuristics were single agent-based ones performing neighborhood searches, such as tabu search, simulated annealing, and so on. The single agent-based metaheuristic can be defined as a metaheuristic that consists of one active agent searching for the optimal solution. Then, population-based metaheuristics were introduced to boost the convergence and improve the exploration effort as in genetic algorithm (GA), invasive weed optimization (IWO), artificial bee colony (ABC), and so on. In the population-based metaheuristic, there are certain number of solutions that are tried to improve in every iteration. Later, metaheuristics developed based on swarm intelligence became more popular because of its nature in improving solutions through interaction among solutions within the population.

In general, swarm intelligence performs the searching process through information sharing among autonomous agents within the population. Regarding their autonomy, each agent searches for improvement independently without any control from a central command. Every agent moves toward the reference with a certain degree of randomness. PSO, as the early version of the swarm-based metaheuristic, utilizes the global and local best solutions as the reference of the guided search performed by each agent [30]. Then, the development of a later swarm-based metaheuristic was performed in various ways, such as in determining the reference, the walk relative to the reference, and the random search enriching the guided search. Some metaheuristics perform single-phase search, such as in BOA [15] or KMA [19], while some others perform multiple-phase searches, such as in NGO [16] or EBOA [26]. Some metaheuristics perform a single strategy while others deploy multiple strategies performed in a single phase or multiple phases. Some metaheuristics perform segregation of roles while others do not. In the metaheuristic that deploys

segregation of roles, the mechanism of one agent or some agents may be different from the others. Some metaheuristics implement a rigid acceptance approach, while others do not. In the rigid acceptance approach, a new solution replaces the current solution only if this new solution is better than the current one. The list of new swarm-based metaheuristics, including the strategy implemented by them, is presented in Table 1. Meanwhile, the strategy implemented in the proposed metaheuristic is presented in the last row of Table 1.

Due to the large number of new metaheuristics already exists and were developed in the recent years, it is impossible to accommodate all of them in a single table. Therefore, metaheuristics selected in Table 1 represents the variety of them. The summarized strategy of each metaheuristic presented in Table 1 is also designed to show that despite the use of any metaphors, in general, there are several common searches, references, and directions implemented in these metaheuristics.

Table 1. Summary of several metaheuristics

No.	Metaheuristic	Metaphor	Strategy	Rigid Acceptance
1	GWO [9]	grey wolf	guided search toward the three best solutions in every iteration	no
2	KMA [19]	komodo	guided search toward the resultant of some better solutions, guided search to avoid the resultant of some better solutions but worse than the corresponding solution, crossover with the best solution, and full random search	no
3	CLO [18]	clouded leopard	guided search relative to randomly selected other solution, neighbourhood search within large but decreased local space	yes
4	BOA [15]	butterfly	guided search toward the best solution among population, neighbourhood search	no
5	COA [14]	coati	guided search toward the best solution among population, guided search relative to a randomized solution within the space, neighbourhood search within the large but decreased space	yes
6	ZOA [13]	zebra	guided search toward the global best solution, guided search toward a randomly selected solution, and neighbourhood search within narrow and decreased local space.	yes
7	NGO [16]	northern goshawk	guided search relative to a randomly selected solution, neighbourhood search within narrow and decreased local space.	yes
8	EBOA [26]	voting	guided search relative to the best solution or a randomly selected solution among some best solutions, neighbourhood search within narrow and decreased local space	yes
9	TIA [29]	-	guided search relative to all other solutions	yes
10	ASBO [27]	-	guided search relative to the average of best and worst solutions, guided search toward the subtraction of the best solution with the worst solution, guided search to avoid the best solution	yes
11	GSO [28]	-	guided search toward the global best solution and local best solution	no
12	this work	-	guided search toward and avoid the global best solution, guided search toward and avoid the middle between two randomly selected solutions among population, neighbourhood search within the large and narrow but decreased local space	yes

Based on Table 1, it is shown that almost all swarm-based metaheuristics move closer to the better reference and move away from the worse reference. In many cases, the corresponding solutions will move closer to the global best solution, the local best solution, the best solution among the population in the current iteration, the resultant of some best solutions, and so on. In these cases, the references are assumed to be better than the corresponding solution. Meanwhile, when the quality of the reference is unknown, such as a randomized solution within the search space or a randomly selected solution among the population, quality checking is performed first. Then, this reference is compared with the corresponding solution based on its quality. The corresponding solution moves closer to the reference if it is better than the corresponding solution. Otherwise, the corresponding solution chooses to move away.

On the other hand, a metaheuristic may choose to perform a neighborhood search or a full random search. Each metaheuristic chooses its own strategy. But a metaheuristic that performs multiple random searches is rare to find.

These circumstances become the reasoning and background to develop a new metaheuristic in this work. The distinct positioning of the proposed metaheuristic among the others is that it performs both getting closer and getting away without considering the quality of the reference. Moreover, it performs two opposite random searches. The first uses a

narrow local search space, while the second uses a large local search space.

3. PROPOSED MODEL

The proposed treble opposite algorithm (TOA) is developed as a metaphor-free swarm-based metaheuristic. As a metaphor-free metaheuristic, TOA uses its main strategy for its name. Meanwhile, as a swarm-based metaheuristic, TOA uses guided search as its core search and random search as complementary. TOA consists of three phases performed sequentially. A guided search relative to the global best solution is performed in the first phase. A guided search relative to a target between two randomly selected solutions is performed in the second phase. A random search is performed in the third phase. Each phase consists of two searches where each search generates a candidate. Then, the better candidate between these two candidates will be assessed to replace the current solution. The rigid acceptance approach is implemented in TOA.

There are several reasons behind this concept. First, there is not any guarantee that moving toward a better solution guarantees improvement, especially in handling multimodal problems. Besides, the walk toward the best or better solution may end with the circumstance where the optimization process

is entrapped within the region of the local optimal. Meanwhile, moving toward any direction without any certain guidance is far from wise. Second, the multiple search approach becomes more important as each has its advantages and disadvantages. Meanwhile, there is not any metaheuristic can accommodate all kinds of searches. Third, the walk should not rely only on the global best solution or the best solution among the population. Fourth, swarm-based metaheuristic should be enriched with random search. A rigid acceptance approach is needed to prevent the agent from moving toward a worse solution.

As a swarm-based metaheuristic, TOA is also a population-based metaheuristic. It means that TOA consists of a certain number of solutions. In general, as with other metaheuristics, TOA consists of two steps. The first step is initialization while the second step is iteration. In the initialization, all solutions are generated uniformly within the search space. It means all solutions within the search space have an equal opportunity to become the initial solution.

The first phase consists of a guided search relative to the global best solution. There are two walks in this phase. The first walk is the walk toward the global best solution. The second walk is the walk to avoid the global best solution. The first walk is common in many swarm-based metaheuristics because the improvement is more probable by getting closer to the global best solution. But moving toward the global best solution does not guarantee improvement, especially in handling multimodal problems. On the other hand, avoiding the global best solution can be seen as the exploration to anticipate if moving toward the global best solution may lead to stagnation or local optimal entrapment. This process is illustrated in Figure 1.

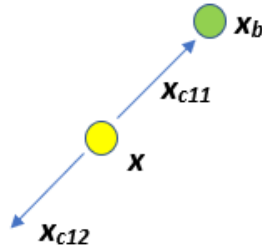


Figure 1. Guided search toward and opposite the global best solution

The second guided search consists of the guided search relative to the solution in the middle between two randomly selected solutions. This process can be viewed as a guided exploration. In many metaheuristics where there is an interaction between the corresponding solution and other solutions within the population, the reference is the randomly selected solution. Meanwhile, in TOA, there are two solutions that are selected randomly. Then, the target is not one of the selected solutions but the average of two random solutions. The reasoning is that the walk toward the existing solution is unnecessary because the quality of these solutions is already known. Meanwhile, it is important to search for someplace between two found solutions. There are two walks performed in this search. The first walk is a walk toward the target while the second walk is a walk to avoid the target. This strategy is different from many shortcoming metaheuristics regarding the interaction among solutions where the corresponding solution moves toward a randomly selected solution if this randomly selected solution is better than the corresponding solution and

moves in the opposite way when the opposite circumstance takes place. This process is illustrated in Figure 2.

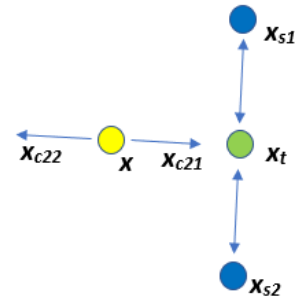


Figure 2. Guided search toward and opposite the middle between two randomly selected solutions

The third phase consists of two random searches, i.e., neighborhood searches. As neighborhood searches, these searches try to find better solutions around the corresponding solution. Meanwhile, there are differences between these two neighborhood searches. The first search is performed within a narrow local search space, while the second search is performed within the large local search space. The similarity among them is that the local search space declines linearly through iteration. This process is illustrated in Figure 3.

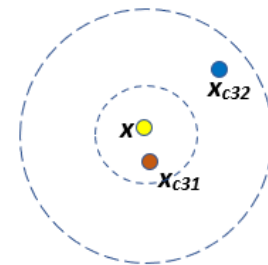


Figure 3. Neighbourhood search within the large and narrow local space

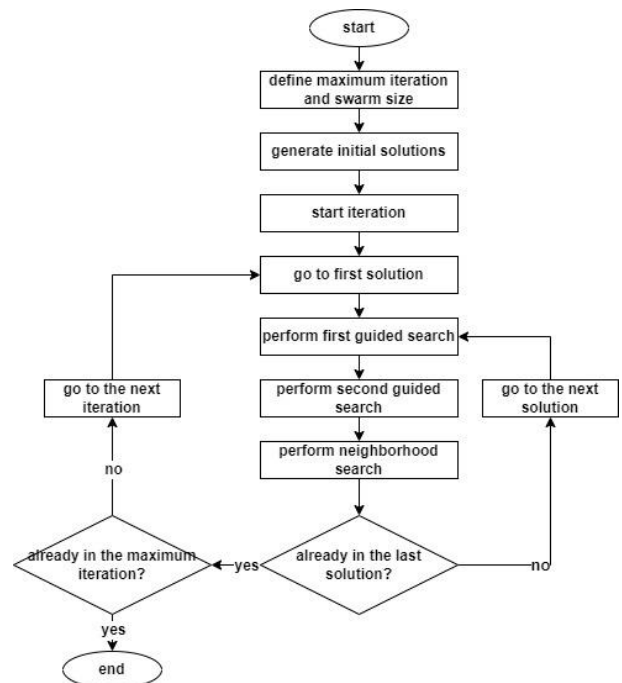


Figure 4. Flowchart of TSO

The rigid acceptance approach is performed for all solutions and the global best solution. The new solution or candidate replaces the previous solution only if the improvement happens. Otherwise, this candidate will be rejected. Moreover, each time there is a new solution or replacement, the global best solution will be updated. The rigid acceptance approach is also performed to update the global best solution.

This concept is then formalized in pseudocode presented in algorithm 1. x_b becomes the final solution. The initialization step is presented from line 2 to line 5. Meanwhile, the iteration step is presented from line 6 to line 15. Moreover, the illustration of the overall process of TOA is presented in the flowchart in Figure 4. The more detailed procedure in each process is mathematically presented using Eq. (1) to Eq. (15).

Algorithm 1: Treble opposite algorithm

```

1  begin
2  for all  $x$  in  $X$ 
3    initialize  $x_i$  using Eq. (2)
4    update  $x_b$  using Eq. (4)
5  end for
6  for  $t=1:t_m$ 
7    for all  $x$  in  $X$ 
8      perform 1st guided search using Eq. (5) to Eq. (7)
9      update  $x_i$  using Eq. (3) and  $x_b$  using Eq. (4)
10     perform 2nd guided search using Eq. (8) to Eq. (12)
11     update  $x_i$  using Eq. (3) and  $x_b$  using Eq. (4)
12     perform random search using Eq. (13) to Eq. (15)
13     update  $x_i$  using Eq. (3) and  $x_b$  using Eq. (4)
14   end for
15 end for
16 end
17 output  $x_b$ 

```

This population is formalized by using Eq. (1), where this population is presented with a set of solutions. x denotes the solution, X denotes the set of solutions, and n denotes the population size.

$$X = \{x_1, x_2, x_3, \dots, x_n\} \quad (1)$$

The initialization is formalized by using Eq. (1). In Eq. (2), $x_{i,j}$ denotes the solution i at dimension j , $x_{l,j}$ denotes the lower boundary of dimension j and $x_{u,j}$ denotes the upper boundary of dimension j . U represents the uniform random generator.

$$x_{i,j} = U(x_{l,j}, x_{u,j}) \quad (2)$$

The updating process of the current solution and the global best solution processes is formalized using Eq. (3) and Eq. (4). The improvement is measured based on the objective function f . In this work, the context of optimization is minimization. It means the lower objective score means this solution is better. x_b denotes the global best solution representing the best solution so far.

$$x'_i = \begin{cases} x_c, & f(x_c) < f(x_i) \\ x_i, & \text{else} \end{cases} \quad (3)$$

$$x'_b = \begin{cases} x_i, & f(x_i) < f(x_b) \\ x_b, & \text{else} \end{cases} \quad (4)$$

The first guided search procedure is formalized by using Eq. (5) to Eq. (7), where x_{c11} and x_{c12} denote the first and second

walks in the first phase. Meanwhile, x_{c1} denotes the candidate of the first phase.

$$x_{c11,j} = x_{i,j} + U(0,1) \cdot (x_{b,j} - 2x_{i,j}) \quad (5)$$

$$x_{c12,j} = x_{i,j} + U(0,1) \cdot (x_{i,j} - 2x_{b,j}) \quad (6)$$

$$x_{c1} = \begin{cases} x_{c11}, & f(x_{c11}) < f(x_{c12}) \\ x_{c12}, & \text{else} \end{cases} \quad (7)$$

Processes in the second phase are formalized by using Eq. (8) to Eq. (12). Eq. (8) states that two randomly selected solutions are picked from the population uniformly where x_s denotes the selected solution. Eq. (9) states that the target (x_t) is determined in the middle between these two solutions. Eq. (10) represents the walk toward the target, while Eq. (11) represents the walk to avoid the target. $x_{c21,j}$ and $x_{c22,j}$ denotes the first and second candidates generated in the second phase at dimension j . x_{c2} denotes the final candidate of the second phase.

$$x_{s1}, x_{s2} = U(X) \quad (8)$$

$$x_{t,j} = \frac{x_{s1,j} + x_{s2,j}}{2} \quad (9)$$

$$x_{c21,j} = x_{i,j} + U(0,1) \cdot (x_{t,j} - 2x_{i,j}) \quad (10)$$

$$x_{c22,j} = x_{i,j} + U(0,1) \cdot (x_{i,j} - 2x_{t,j}) \quad (11)$$

$$x_{c2} = \begin{cases} x_{c21}, & f(x_{c21}) < f(x_{c22}) \\ x_{c22}, & \text{else} \end{cases} \quad (12)$$

The processes in the third phase are formalized using Eq. (13) to Eq. (15). t denotes the iteration while t_m denotes the maximum iteration. $x_{c31,j}$ and $x_{c32,j}$ denotes the first and second candidates generated in the third phase at dimension j . x_{c3} denotes the final candidate of the second phase.

$$x_{c31,j} = x_{i,j} + 0.1U(-1,1) \cdot \left(1 - \frac{t}{t_m}\right) \cdot (x_{u,j} - x_{l,j}) \quad (13)$$

$$x_{c32,j} = x_{i,j} + U(-1,1) \cdot \left(1 - \frac{t}{t_m}\right) \cdot (x_{u,j} - x_{l,j}) \quad (14)$$

$$x_{c3} = \begin{cases} x_{c31}, & f(x_{c31}) < f(x_{c32}) \\ x_{c32}, & \text{else} \end{cases} \quad (15)$$

4. RESULT

In this section, the performance of TOA is assessed through simulation. There are two assessments regarding the proposed metaheuristic. In the first assessment, TOA is challenged to solve a set of 23 classic functions that can be split into three groups: seven high dimension unimodal functions, six high dimension multimodal functions, and ten fixed dimension multimodal functions. A detailed description of these 23 functions is presented in Table 2. In the second assessment, TOA is challenged to solve the portfolio optimization problem. The set of 23 functions represents the theoretical problem while the portfolio optimization problem represents the practical problem.

Table 2. List of 23 functions

No.	Function	Dim	Space	Target
1	Sphere	50	[-100, 100]	0
2	Schwefel 2.22	50	[-100, 100]	0
3	Schwefel 1.2	50	[-100, 100]	0
4	Schwefel 2.21	50	[-100, 100]	0
5	Rosenbrock	50	[-30, 30]	0
6	Step	50	[-100, 100]	0
7	Quartic	50	[-1.28, 1.28]	0
8	Schwefel	50	[-500, 500]	-20,948
9	Rastrigin	50	[-5.12, 5.12]	0
10	Ackley	50	[-32, 32]	0
11	Griewank	50	[-600, 600]	0
12	Penalized	50	[-50, 50]	0
13	Penalized 2	50	[-50, 50]	0
14	Shekel Foxholes	2	[-65, 65]	1
15	Kowalik	4	[-5, 5]	0.0003
16	Six Hump Camel	2	[-5, 5]	-1.0316
17	Branin	2	[-5, 5]	0.398
18	Goldstein-Price	2	[-2, 2]	3
19	Hartman 3	3	[1, 3]	-3.86
20	Hartman 6	6	[0, 1]	-3.32
21	Shekel 5	4	[0, 10]	-10.153
22	Shekel 7	4	[0, 10]	-10.402
23	Shekel 10	4	[0, 10]	-10.536

There are several reasons for choosing the 23 functions and the portfolio optimization problem as the use cases for the assessment. First, these functions cover the variety of the optimization problems so that it can be used to investigate the performance of any metaheuristics. This set of functions contains both unimodal functions and the multimodal ones. This set also contains functions with various problem spaces, from the very narrow ones, moderate, to the very large ones. In several functions, the global optimal solution is right in the middle of the space while some others are distributed anywhere in the problem space. Some functions have smooth shapes so that a better solution can be identified easily. Meanwhile, some functions have ambiguous shape, such as very curly or dominant with the flat shape with vary narrow slope. Second, this set of functions is popular enough and it is used in many studies proposing metaheuristics, such as KMA,

GSO, TIA, and so on.

The portfolio optimization problem is chosen due to its rarity in studies proposing a new metaheuristic. This problem is far less popular than the mechanical design problem or the optimal power flow problem. Meanwhile, the portfolio optimization problem gives another kind of challenge where the solution space is discrete which is different from the floating point-based optimization problem in the mechanical design or the theoretical problems, such as the set of 23 functions.

In this assessment, TOA is compared with five other metaheuristics: GWO, GSO, ZOA, ASBO, and COA. The several reasons of choosing these metaheuristics as competitors is as follows. First, all these metaheuristics do not have any adjusted parameters except the common ones which are the population size and maximum iteration. These metaheuristics are new although GWO is the oldest among them. On the other hand, COA is the newest one. GWO and GSO represent the metaheuristics that do not deploy rigid acceptance approach. ZOA, ASBO, and COA represent metaheuristics that deploy rigid acceptance approach. GWO and GSO perform single phase search while ZOA, ASBO, and COA perform multiple phase search.

In the first assessment, the population size is set 5 while the maximum iteration is set 20. On the other hand, the dimension for the high dimension functions is set to 50. It means TOA and all these competitors are challenged to solve high dimension problems with low population size and low maximum iteration. The result of the first, second, and third result is presented in Table 3, Table 4 and Table 5 respectively. The information regarding each function consists of the mean score, standard deviation, and mean rank. The result with decimal point less than 10^{-4} is rounded to 0.0000.

Table 3 indicates that TOA performs superiorly in handling the high dimension unimodal functions. In this group, TOA becomes the best performer in handling five functions and the second best in handling one function (Step). Fortunately, the gap between TOA and ZOA as the best performer in handling Step function is very narrow. Moreover, TOA can find the global optimal solution in handling four functions (Sphere, Schwefel 2.22, Schwefel 1.2, and Schwefel 2.21).

Table 3. Evaluation result in solving high-dimension unimodal functions

F	Parameter	GWO	GSO	ZOA	ASBO	COA	TOA
1	mean	1.2804×10^4	5.8594×10^4	0.0005	1.7053	3.8420	0.0000
	std dev	7.6652×10^3	1.3382×10^4	0.0004	0.9249	2.7363	0.0000
	mean rank	5	6	2	3	4	1
2	mean	0.0000	1.4018×10^{67}	0.0000	0.0000	0.0000	0.0000
	std dev	0.0000	5.6933×10^{67}	0.0000	0.0000	0.0000	0.0000
	mean rank	1	6	1	1	1	1
3	mean	3.4079×10^5	2.0564×10^5	9.0720×10^1	6.9451×10^3	8.4605×10^3	0.0000
	std dev	4.4531×10^5	1.0905×10^5	1.2367×10^2	4.0960×10^3	8.4154×10^3	0.0000
	mean rank	6	5	2	3	4	1
4	mean	7.2948×10^1	6.2390×10^1	0.0604	1.7270	3.7789	0.0000
	std dev	2.1715×10^1	6.7105	0.0304	0.6480	1.5287	0.0000
	mean rank	6	5	2	3	4	1
5	mean	5.1738×10^7	1.4096×10^8	4.8951×10^1	6.8074×10^1	1.1414×10^2	4.8951×10^1
	std dev	4.9234×10^7	6.2870×10^7	0.0345	1.8070×10^1	6.4230×10^1	0.0187
	mean rank	5	6	1	3	4	1
6	mean	1.1610×10^4	6.0576×10^4	1.0730×10^1	1.1222×10^1	1.4394×10^1	1.1071×10^1
	std dev	7.3676×10^3	1.1716×10^4	0.5848	1.5504	2.4375	0.3448
	mean rank	5	6	1	3	4	2
7	mean	4.3895×10^1	1.0465×10^2	0.0270	0.0681	0.0754	0.0059
	std dev	6.2312×10^1	3.9674×10^1	0.0154	0.0354	0.0452	0.0037
	mean rank	5	6	2	3	4	1

Table 4 indicates that TOA also performs superiorly in handling high dimension multimodal functions. In this group, TOA becomes the best performer in handling five functions and the second best in handling one function (Penalized). Moreover, TOA can find the global optimal solution in handling three functions (Rastrigin, Ackley, and Griewank).

This result also splits these six metaheuristics into two clusters based on their performance. The first cluster consists of GWO and GSO. On the other hand, the second cluster consists of ZOA, ASBO, COA, and TOA. The performance gap between these two clusters is wide.

Table 4. Evaluation result in solving high-dimension multimodal functions

F	Parameter	GWO	GSO	ZOA	ASBO	COA	TOA
8	mean	-7.1601×10^1	-3.2981×10^3	-2.8693×10^3	-3.9086×10^3	-4.5522×10^3	-4.7660×10^3
	std dev	3.9837×10^2	9.3244×10^2	6.6108×10^2	5.7547×10^2	5.0252×10^2	5.1712×10^2
	mean rank	6	4	5	3	2	1
9	mean	1.5507×10^2	5.2199×10^2	0.0019	2.1274×10^1	8.2210	0.0000
	std dev	4.7037×10^1	5.3209×10^1	0.0018	3.7314	7.9341	0.0000
	mean rank	5	6	2	4	3	1
10	mean	1.2446×10^1	1.9139×10^1	0.0050	2.8705	0.4925	0.0000
	std dev	2.6049	0.5420	0.0025	0.4190	0.2094	0.0000
	mean rank	5	6	2	4	3	1
11	mean	1.2748×10^2	5.2380×10^2	0.0087	0.6766	0.5505	0.0000
	std dev	7.6051×10^1	1.1741×10^2	0.0283	0.2142	0.3137	0.0000
	mean rank	5	6	2	4	3	1
12	mean	2.3014×10^8	2.0093×10^8	1.0471	0.1638	1.1589	0.9533
	std dev	2.1350×10^8	1.0902×10^7	0.1561	0.0827	0.2134	0.1490
	mean rank	6	5	3	1	4	2
13	mean	2.5432×10^8	5.5124×10^8	3.1386	1.1350×10^1	4.0352	3.1279
	std dev	2.4953×10^8	3.9614×10^8	0.0389	1.7262	0.3895	0.0158
	mean rank	5	6	2	4	3	1

Table 5. Evaluation result in solving fixed-dimension multimodal functions

F	Parameter	GWO	GSO	ZOA	ASBO	COA	TOA
14	mean	1.4891×10^2	1.1792×10^1	8.1578	4.5056	5.7096	4.4929
	std dev	2.0891×10^2	5.4872	3.3760	3.5496	3.1420	2.8173
	mean rank	6	5	4	2	3	1
15	mean	0.8900	0.0333	0.0079	0.1256	0.0067	0.0013
	std dev	2.1766	0.0225	0.0122	0.0327	0.0085	0.0007
	mean rank	6	4	3	5	2	1
16	mean	1.1195×10^2	-0.8298	-0.8800	-0.0049	-1.0298	-1.0302
	std dev	5.1121×10^2	0.4097	0.2193	0.0205	0.0026	0.0016
	mean rank	6	4	3	5	2	1
17	mean	5.5672×10^1	1.5696	4.8955	2.0446	0.4095	0.4031
	std dev	1.1630×10^1	3.4805	6.1640	4.6518	0.0223	0.0090
	mean rank	6	3	5	4	2	1
18	mean	1.7159×10^3	1.9956×10^1	4.0241×10^1	1.5500×10^1	1.2322×10^1	3.0993
	std dev	4.8681×10^3	2.7394×10^1	5.2763×10^1	5.8630×10^1	2.1838×10^1	0.1895
	mean rank	4	5	6	3	2	1
19	mean	-0.0026	-0.0151	-0.0495	-0.0495	-0.0495	-0.0495
	std dev	0.0070	0.0168	0.0000	0.0000	0.0000	0.0000
	mean rank	6	5	1	1	1	1
20	mean	-0.0053	-1.9786	-2.0295	-0.7399	-3.0412	-2.8111
	std dev	0.0010	0.7019	0.5996	0.5286	0.1877	0.6573
	mean rank	6	4	3	5	1	2
21	mean	-0.2731	-1.4029	-2.0139	-3.3757	-4.3245	-4.3152
	std dev	0.0000	1.2351	1.2025	3.3023	1.7608	0.4108
	mean rank	6	5	4	3	1	2
22	mean	-0.2936	-2.4186	-2.5916	-3.4876	-4.3249	-4.8074
	std dev	0.0000	1.9652	1.5957	3.1984	1.7673	1.0910
	mean rank	6	5	4	3	2	1
23	mean	-0.3217	-2.1062	-2.0959	-4.6137	-3.6712	-4.6240
	std dev	0.0000	1.4188	1.0323	3.6199	1.5581	0.5674
	mean rank	6	4	5	2	3	1

Table 5 indicates that TOA is still superior in handling the fixed-dimension multimodal functions. Among the ten functions in this group, TOA becomes the best performer in handling eight functions and the second best one in handling two functions (Hartman 6 and Shekel 5). Different from the circumstance in the first and second groups, fierce competition

takes place in handling functions in the third group. The performance gap among metaheuristics is narrow. On the other hand, there is no discrimination between GSO and GWO on one side and TOA, ASBO, ZOA, and COA on the other side. It means that GWO and GSO are competitive enough in handling fixed-dimension multimodal functions.

The result in Tables 3-5 is then summarized in Table 6 which presents the superiority of TOA to other metaheuristics in handling functions based on the groups. Table 6 strengthens the fact that TOA is superior to these five competitors in all groups of functions. Moreover, TOA is superior to GSO in handling all 23 functions. Overall, TOA is better than GWO, GSO, ZOA, ASBO, and COA in handling 22, 23, 19, 20, and 19 functions, respectively.

Table 6. Number of functions in every group where TOA is better

Cluster	Number of Functions				
	GWO	GSO	ZOA	ASBO	COA
1	6	7	4	6	6
2	6	6	6	5	6
3	10	10	9	9	7
Total	22	23	19	20	19

The second assessment is the portfolio optimization problem. In this assessment, TOA is challenged to solve the stock arrangement that should be purchased in the investment action. The stocks of the banking company listed IDX30 represent the decision variables. IDX30 index is the index or list of Indonesian stock exchange that consists of stocks provided by companies that have high liquidity, high capitalization, and strong corporate fundamentals. There are four stocks from the banking sector listed in IDX30. The list contains the detailed information regarding these four stocks which is presented in Table 7. There are three pieces of information regarding the list: stock index, current stock price, and year-on-year capital gain. The data is obtained from Google on 12 February 2023. The stock price and capital gain are presented in rupiah per share.

Table 7. List of stocks in IDX30

No.	Stock	Price (Rp)	Capital Gain (Rp)
1	BBCA	8,450	375
2	BBNI	9,025	925
3	BBRI	4,820	300
4	BMRI	10,375	2,625

Table 8. Result in solving a portfolio optimization problem

No.	Metaheuristic	Total Capital Gain (Rp)
1	TOA	354,561,190
2	COA	348,783,068
3	ASBO	349,709,880
4	ZOA	323,643,854
5	GSO	331,220,113
6	GWO	173,173,409

The objective of this portfolio optimization is to maximize the total capital gain through the arrangement of stock. As there are four stocks that should be arranged and all stocks provide positive capital gain, this problem can be seen as low dimension unimodal functions. The stock quantity should be purchased for each stock ranging from 200 to 1,000 lots. One lot consists of 100 shares. Meanwhile, the total investment should not surpass two billion rupiahs. In this assessment, the maximum iteration is set to 20 while the population size is set to 5. The result is presented in Table 8.

Table 8 indicates that the proposed TOA is superior compared to all these five competitors in handling the portfolio

optimization problem. Overall, the gap in total capital gain among TOA, COA, ASBO, ZOA, and GSO is narrow. On the other hand, the gap between these metaheuristics and GWO is wide.

5. DISCUSSION

In general, the performance of any metaheuristics can be drawn back to their exploration and exploitation capabilities. Exploration represents the ability of the metaheuristic in tracing solution within the search space [26]. Exploration is needed to find the area where the global optimal solution exists and to avoid the local optimal entrapment. In this context, exploration capability is usually linked to multimodal problems where these problems consist of multiple optimal solutions. In many cases, there are a lot of optimal solutions in one multimodal problem which means a metaheuristic will be easily trapped in the local optimal so that the area of global optimal solution will never be found. Exploitation represents the ability to improve the current solution by tracing possible solution near the current solution [26]. Exploitation can also be viewed as the ability to conduct the local search. The other issue is related to the condition when the agent is thrown away from the area where the global optimal solution. Based on these issues, in this section, the capability of TOA in handling exploration, exploitation, and avoiding being thrown away from the area of the global optimal solution will be discussed.

The main difference of TOA from other swarm-based metaheuristics is that TOA performs not just moving closer but also moving away. It means that without considering the quality of its reference, whether it is the global best solution or the randomly selected solution. It is different from GSO [28] and GWO [9] where these two metaheuristics performs moving closer only. This approach is also different from ASBO [27], COA [14], and ZOA [13] where the direction of the walk relative to other solutions is performed conditionally. The getting closer walk is performed only if the reference is better than the corresponding solution and choose to get away when the opposite circumstance takes place. As mentioned previously there is not any guarantee that getting closer to the better solution will end with improvement. Tracing both directions as implemented in TOA is proven better than implementing getting closer only or the optional direction.

The exploration capability of TOA is also enriched with the walk relative to the middle between two randomly selected solutions. It provides additional exploration because this location may not be traced yet. This concept is like GWO where the target is the resultant of the three best solutions in the current iteration or ASBO where one of the targets is the middle between the best and worst solutions in the current iteration.

Neighborhood searches performed in the third phase represent both exploration and exploitation capabilities. The neighborhood search with wide local search space represents the exploration while the neighborhood search with narrow local search space represents the exploitation. Although the local search space width of both neighborhood searches decreases linearly, the width of the second neighborhood search is still ten times than the first one.

The effort to avoid the agent moving to a worse solution or being thrown away from the current area is facilitated mainly by the rigid acceptance approach. This approach is important due to the result in handling the high-dimensional functions. It is shown that metaheuristics that implement rigid-acceptance

approaches are far better than those that do not. The second protection comes from the local search space that is reduced linearly through iteration. This second approach gives more flexibility in the early iteration, and this flexibility becomes harder as the iteration goes. Simulated annealing is the early metaheuristic that introduces iteration-controlled flexibility. Meanwhile, MPA is the famous swarm-based metaheuristic that uses this approach by reducing the local search space through iteration. This approach is then adopted and modified by the later metaheuristics, such as POA, ZOA, COA, etc.

The superiority of TOA does not mean it has become the perfect metaheuristic. TOA is a metaheuristic that performs better than GWO, GSO, ZOA, COA, and ASBO in handling almost all functions in the set of 23 functions. Meanwhile, ZOA is better than TOA in handling Step, ASBO is better than TOA in handling Penalized, and COA is better than TOA in handling Hartman 6 and Shekel 5. Moreover, the final solution of TOA is still far from the global optimal in handling Rosenbrock, Step, Schwefel, Hartman 3, Shekel 5, Shekel 7, and Shekel 10.

The superiority of TOA in handling the portfolio optimization problem also does not guarantee that TOA is superior in handling all practical optimization problems. This portfolio optimization problem is generally an integer-based optimization problem because the decision variables are presented in integer format. The objective function is also simple accumulating the capital gain of all shares. Moreover, there is no stock in which the capital gain is negative. Meanwhile, this problem is more unique than the set of 23 functions because the quantity of stocks that should be purchased is limited by not only the maximum range but also the total investment. Meanwhile, there are various practical optimization problems in engineering where the decision variables are presented in floating point, and the objective function is more complex. Besides, there are various combinatorial optimization problems that can be explored to evaluate the effectiveness of TOA rather than numerical optimization problems as in this work.

This result motivates the modification and improvement of TOA in future work. Besides, there are a huge number of optimization problems spanning the wide area of engineering. These problems have their own characteristics, such as the objective and constraints. The objective can be a single objective or multiple objectives. Meanwhile, in many problems, some constraints may depend on other constraints, making these problems more difficult to solve. Many problems are combinatorial rather than numerical.

Moreover, future studies can be carried out based on the advantages and limitations of this current work. In regard that the searching in both opposite directions is better than searching in a single direction, this approach can be maintained in future development. Meanwhile, this TOA uses only two references: the best solution, and two randomly picked solutions. On the other hand, there are existing metaheuristics that use a better solution in the swarm or a collection of better solutions in the swarm as reference. This approach can be hybridized into the current TOA as improvement. Besides, the investigation of hybridizing TOA as a swarm-based metaheuristic with any evolution-based metaheuristic is also challenging. In the end, the modification or improvement can also be performed by implementing other stochastic distributions, such as exponential, Poisson, normal, Brownian, and so on.

6. CONCLUSIONS

This paper has presented the new swarm-based metaheuristic called treble opposite algorithm (TOA). The main concept of this metaheuristic is implementing a multiple phase strategy where there are two searches that is opposite to each other in every phase. This strategy is designed to improve the exploration capability while maintaining the exploitation capability. Getting closer to the reference is designed to maintain the exploitation while getting away from the reference improves the exploration. This approach is different from many existing swarm-based metaheuristics where the swarm members move toward their reference. The result indicates that TOA is superior to five other metaheuristics. In handling the set of 23 functions, TOA is better than GWO, GSO, ZOA, ASBO, and COA in 22, 23, 19, 20, and 19 functions, respectively. Moreover, TOA produces the highest total capital gain in handling portfolio optimization problems. This result proves that searching in two opposite directions is better than searching only in a single direction.

Future studies can be performed through many tracks. First, there is not any guarantee that TOA is superior in handling all kinds of problems. It means that more tests can be performed to evaluate the performance of TOA more comprehensively. The use cases may come from the wide range of engineering problems, whether they are numerical or combinatorial ones. Implementing TOA to solve large scale problems that consist of hundreds of decision variables is also challenging. Second, it is challenging to hybridize the proposed TOA with other metaheuristics, especially the evolution-based ones.

FUNDING

This work was funded by Telkom University, Indonesia.

REFERENCES

- [1] Kamper, A., Delorme, R., Leenders, L., Bardow, A. (2023). Boosting operational optimization of multi-energy systems by artificial neural nets. *Computers and Chemical Engineering*, 173: 108208. <http://doi.org/10.1016/j.compchemeng.2023.108208>
- [2] Ala, A., Mahmoudi, A., Mirjalili, S., Simic, V., Pamucar, D.E. (2023). Evaluating the performance of various algorithms for wind energy optimization: A hybrid decision-making model. *Expert Systems with Applications*, 221: 119731. <http://doi.org/10.1016/j.eswa.2023.119731>
- [3] Yao, J., Hedengren, J.D., Gao, T., Powell, K.M. (2023). A two-level optimization framework for battery energy storage systems to enhance economics and minimize long-term capacity fading. *Journal of Energy Storage*, 63: 106943. <http://doi.org/10.1016/j.est.2023.106943>
- [4] Yang, H., Pu, C., Wu, J., Wu, Y., Xia, Y. (2023). Enhancing OLSR protocol in VANETs with multi-objective particle swarm optimization. *Physica A: Statistical Mechanics and its Applications*, 614: 128570. <http://doi.org/10.1016/j.physa.2023.128570>
- [5] Azevedo, B.F., Montanno-Vega, R., Leonilde, M., Varela, R., Pereira, A.I. (2023). Bio-inspired multi-objective algorithms applied on production scheduling problems. *International Journal of Industrial Engineering*

- Computations, 14(2): 415-436. <http://doi.org/10.5267/j.ijiec.2022.12.001>
- [6] Trojovska, E., Dehghani, M., Trojovsky, P. (2022). Fennec fox optimization: A new nature-inspired optimization algorithm. *IEEE Access*, 10: 84417-84443. <http://doi.org/10.1109/ACCESS.2022.3197745>
- [7] Braik, M.S. (2021). Chameleon swarm algorithm: A bio-inspired optimizer for solving engineering design problems. *Expert Systems with Applications*, 174: 114685. <https://doi.org/10.1016/j.eswa.2021.114685>
- [8] Dehghani, M., Hubalovsky, S., Trojovsky, P. (2022). Tasmanian devil optimization: A new bio-inspired optimization algorithm for solving optimization algorithm. *IEEE Access*, 10: 19599-19620. <http://doi.org/10.1109/ACCESS.2022.3151641>
- [9] Mirjalili, S., Mirjalili, S.M., Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, 69: 46-61. <http://doi.org/10.1016/j.advengsoft.2013.12.007>
- [10] Faramarzi, A., Heidarinejad, M., Mirjalili, S., Gandomi, A.H. (2020). Marine predators algorithm: A nature-inspired metaheuristic. *Expert Systems with Applications*, 152: 113377. <http://doi.org/10.1016/j.eswa.2020.113377>
- [11] Li, S., Chen, H., Wang, M., Heidari, A.A., Mirjalili, S. (2020). Slime mould algorithm: A new method for stochastic optimization. *Future Generation Computer Systems*, 111: 300-323. <http://doi.org/10.1016/j.future.2020.03.055>
- [12] Trojovsky, P., Dehghani, M. (2022). Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications. *Sensors*, 22: 855. <http://doi.org/10.3390/s22030855>
- [13] Trojovska, E., Dehghani, M., Trojovsky, P. (2022). Zebra optimization algorithm: A new bio-inspired optimization algorithm for solving optimization algorithm. *IEEE Access*, 10: 49445-49473. <http://doi.org/10.1109/ACCESS.2022.3172789>
- [14] Dehghani, M., Montazeri, Z., Trojovska, E., Trojovsky, P. (2023). Coati optimization algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems. *Knowledge-Based Systems*, 259: 110011. <https://doi.org/10.1016/j.knsys.2022.110011>
- [15] Arora, S., Singh, S. (2019). Butterfly optimization algorithm: a novel approach for global optimization. *Soft Computing*, 23(3): 715-734. <http://doi.org/10.1007/s00500-018-3102-4>
- [16] Dehghani, M., Hubalovsky, S., Trojovsky, P. (2021). Northern goshawk optimization: A new swarm-based algorithm for solving optimization problems. *IEEE Access*, 9: 162059-162080. <http://doi.org/10.1109/ACCESS.2021.3133286>
- [17] Kusuma, P.D., Prasasti, A.L. (2022). Guided pelican algorithm. *International Journal of Intelligent Engineering and Systems*, 15(6): 179-190. <http://doi.org/10.22266/ijies2022.1231.18>
- [18] Trojovska, E., Dehghani, M. (2022). Clouded leopard optimization: A new nature-inspired optimization algorithm. *IEEE Access*, 10: 102876-102906. <http://doi.org/10.1109/ACCESS.2022.3208700>
- [19] Suyanto, S., Ariyanto, A.A., Ariyanto, A.F. (2022). Komodo mlipir algorithm. *Applied Soft Computing*, 114: 108043. <http://doi.org/10.1016/j.asoc.2021.108043>
- [20] Akbari, M.A., Zare, M., Azizipanah-abarghooee, R., Mirjalili, S., Deriche, M. (2022). The cheetah optimizer: A nature-inspired metaheuristic algorithm for large-scale optimization problems. *Scientific Reports*, 12: 10953. <http://doi.org/10.1038/s41598-022-14338-z>
- [21] Dehghani, M., Hubalovsky, S., Trojovsky, P. (2021). Cat and mouse based optimizer: A new nature-inspired optimization algorithm. *Sensors*, 21: 5214. <http://doi.org/10.3390/s21155214>
- [22] Jia, H., Peng, X., Lang, C. (2021). Remora optimization algorithm. *Expert Systems with Applications*, 185: 115665. <http://doi.org/10.1016/j.eswa.2021.115665>
- [23] Kusuma, P.D., Adiputra, D. (2022). Modified social forces algorithm: From pedestrian dynamic to metaheuristic optimization. *International Journal of Intelligent Engineering and Systems*, 15(3): 294-303. <http://doi.org/10.22266/ijies2022.0630.25>
- [24] Trojovska, E., Dehghani, M. (2022). A new human-based metaheuristic optimization method based on mimicking cooking training. *Scientific Reports*, 12: 14861. <http://doi.org/10.1038/s41598-022-19313-2>
- [25] Dehghani, M., Trojovska, E., Trojovsky, P. (2022). A new human-based metaheuristic algorithm for solving optimization problems on the base of simulation of driving training process. *Scientific Reports*, 12: 9924. <http://doi.org/10.1038/s41598-022-14225-7>
- [26] Trojovsky, P., Dehghani, M. (2022). A new optimization algorithm based on mimicking the voting process for leader selection. *PeerJ Computer Science*, 8: e976. <http://doi.org/10.7717/peerj-cs.976>
- [27] Dehghani, M., Hubalovsky, S., Trojovsky, P. (2022). 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*, 8: e910. <http://doi.org/10.7717/peerj-cs.910>
- [28] Noroozi, M., Mohammadi, H., Efatinasab, E., Lashgari, A., Eslami, M., Khan, B. (2022). Golden search optimization algorithm. *IEEE Access*, 10: 37515-37532. <http://doi.org/10.1109/ACCESS.2022.3162853>
- [29] Kusuma, P. D., Novianty, A. (2023). Total interaction algorithm: A metaheuristic in which each agent interacts with all other agents. *International Journal of Intelligent Engineering and Systems*, 16(1): 224-234. <http://doi.org/10.22266/ijies2023.0228.20>
- [30] Gad, A.G. (2023). Particle swarm optimization algorithm and its applications: A systematic review. *Archives of Computational Methods in Engineering*, 30(5): 3471. <http://doi.org/10.1007/s11831-022-09762-3>