

# Three on Three Optimizer: A New Metaheuristic with Three Guided Searches and Three Random Searches

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

**Abstract**—This paper presents a new swarm intelligence-based metaheuristic called a three-on-three optimizer (TOTO). This name is chosen based on its novel mechanism in adopting multiple searches into a single metaheuristic. These multiple searches consist of three guided searches and three random searches. These three guided searches are searching toward the global best solution, searching for the global best solution to avoid the corresponding agent, and searching based on the interaction between the corresponding agent and a randomly selected agent. The three random searches are the local search of the corresponding agent, the local search of the global best solution, and the global search within the entire search space. TOTO is challenged to solve the classic 23 functions as a theoretical optimization problem and the portfolio optimization problem as a real-world optimization problem. There are 13 bank stocks from Kompas 100 index that should be optimized. The result indicates that TOTO performs well in solving the classic 23 functions. TOTO can find the global optimal solution of eleven functions. TOTO is superior to five new metaheuristics in solving 17 functions. These metaheuristics are grey wolf optimizer (GWO), marine predator algorithm (MPA), mixed leader-based optimizer (MLBO), golden search optimizer (GSO), and guided pelican algorithm (GPA). TOTO is better than GWO, MPA, MLBO, GSO, and GPA in solving 22, 21, 21, 19, and 15 functions, respectively. It means TOTO is powerful to solve high-dimension unimodal, multimodal, and fixed-dimension multimodal problems. TOTO performs as the second-best metaheuristic in solving a portfolio optimization problem.

**Keywords**—*Optimization; metaheuristic; swarm intelligence; portfolio optimization; Kompas 100; bank*

## I. INTRODUCTION

Many real-world problems can be seen as optimization problems. This circumstance comes from the nature of human behavior or activity in achieving their objective most efficiently. Ironically, people always find certain limitations or constraints. This consideration is the same as the optimization work. In general, optimization is constructed by two elements: objective and constraint. In the optimization problem, many solutions can be chosen in the solution space. However, some solutions are better than others. One solution that is the best one is called the optimal global solution.

The objective of optimization can be minimization or maximization. In the minimization, the optimal global solution is the solution with the lowest value. Some experimental parameters in the minimization, such as delay [1], total order completion time [2], idle time [2], tardiness cost and maintenance [3], project duration [4], energy consumption [5], transmission losses [6], and so on. On the other hand, in

maximization, the optimal global solution is the solution with the highest value. Some parameters in the maximization are profit [7], voltage stability [6], revenue [8], service level [9], and so on.

Two ways can be chosen to solve the optimization problem. These methods are a mathematical method and a metaheuristic [10]. The mathematical method is robust in solving a simple optimization problem. It guarantees finding the optimal global solution. However, the mathematical or deterministic method often fails to solve a complex optimization problem, such as a non-convex or multimodal problem. Moreover, the mathematical method is less flexible in facing various real-world optimization problems [11].

On the other hand, metaheuristic is widely used in many optimizations. Metaheuristics have several advantages. First, it is flexible enough to be implemented in various problems because it focuses on the objectives and constraints [10]. Second, it can be implemented in an environment with limited computational resources because of its approximate approach so that not all possible solutions are traced [12]. This approximate approach comes with the consequence that metaheuristic does not guarantee finding the optimal global solution.

There are hundreds of metaheuristics developed in recent decades. Moreover, many metaheuristics have been developed in recent years. Many of them used metaphors for their name. Many of these shortcoming metaheuristics were inspired by animal behavior, such as the butterfly optimization algorithm (BOA) [13], chameleon swam algorithm (CSA) [14], coati optimization algorithm (COA) [15], Komodo mlipir algorithm (KMA) [16], northern goshawk optimizer (NGO) [17], raccoon optimization algorithm (ROA) [18], marine predator algorithm (MPA) [19], Tasmanian devil optimizer (TDO) [11], snake optimizer (SO) [10], white shark optimizer (WSO) [20], guided pelican algorithm (GPA) [21], and so on. Some metaheuristics were inspired by the mechanics of plants, such as the tunicate swarm algorithm (TSA) [22], flower pollination algorithm (FPA) [23], and so on. Some metaheuristics are named based on their references in their guided search, such as three influential member-based optimizers (TIMBO) [24], mixed leader-based optimizers (MLBO) [25], multi-leader optimizers (MLO) [26], random selected leader-based optimizer (RLSBO) [27], and so on. Some metaheuristics were inspired by human activities, such as stochastic paint optimizer (SPO) [28], modified social forces algorithm (MSFA) [29], driving training-based optimizer (DTBO) [30], and so on. Some metaheuristics were free from metaphor and named based on

their central concept, such as average subtraction-based optimizer (ASBO) [31], golden search optimizer (GSO) [32], total interaction algorithm (TIA) [33], and so on.

Many early metaheuristics depend on a single strategy to find the optimal solution to problems. Unfortunately, as mentioned in the no-free-lunch theory, no metaheuristics can solve all optimization problems [34]. A search may be excellent for solving some optimization problems [34]. Meanwhile, the performance of this search may need to improve in solving other optimization problems. This circumstance gives strong motivation to develop multi search-based metaheuristic. Many shortcomings in metaheuristics were built by accommodating multiple searches in every iteration. Moreover, this multiple search strategy is proven better in solving an optimization problem. However, adding more searches in a single metaheuristic can increase the algorithm complexity and, in the end, the computational resources.

Many existing swarm-based metaheuristics depend more on the guided search than the random one. In some metaheuristics like KMA [16] or COA [15], segregation of roles is implemented in the population. It means that some agents implement segregation of roles while some other agents do not implement segregation of roles. A more guided search is deployed in many swarm-based metaheuristics with multiple searches. Some metaheuristic deploys neighborhood search only or does not implement any random search. Meanwhile, random search is essential in tackling the local optimal entrapment often found in multimodal problems.

Based on this problem and motivated by the no-free-lunch theory, this paper is aimed to propose a new swarm-based metaheuristic with multiple searches where there is a balance between the guided search and the random search. This proposed metaheuristic is named a three-on-three optimizer (TOTO). This name represents the six searches adopted in this metaheuristic. Three searches are guided searches, while three other searches are random.

The scientific contributions of this paper are listed as follows.

- 1) This paper proposes a new swarm-based metaheuristic with multiple searches within this metaheuristic.
- 2) This paper proposes a balanced proportion between the guided search and random search in every iteration.
- 3) The evaluation is performed by benchmarking the performance of the proposed TOTO with five shortcoming metaheuristics: GWO, MPA, MLBO, GSO, and GPA.
- 4) The benchmark test to evaluate the proposed metaheuristic in solving the classic 23 functions and the comparison with other shortcoming metaheuristics is carried out as a proof of concept regarding the significant performance of TOTO.
- 5) The stock optimization problem is chosen as evaluation of the proposed TOTO in solving real-world optimization problem in financial sector and comparison with other metaheuristics.

The remainder of this paper is arranged as follows. The investigation of several shortcomings of metaheuristics is carried out in section two. This investigation includes the mechanics adopted in these metaheuristics and the position of this work to make the novelty and contribution of this work clear. A detailed presentation of the proposed metaheuristic can be found in section three. This presentation includes the concept, algorithm, mathematical model, and algorithm complexity. The test carried out in this work regarding the evaluation of the proposed metaheuristic is presented in section four. The investigation of the result and findings are explored in section five. Finally, the conclusion and the opportunity for future works are summarized in section six.

## II. RELATED WORKS

Metaheuristics have been evolving for decades. Early metaheuristics are simple with a single strategy. In general, they deploy neighborhood search with some strategy to avoid the local optimal. Simulated annealing (SA) and tabu search (TS) are examples of metaheuristics that adopt a simple neighborhood search with a distinct approach to tackling the optimal local problem. In simulated annealing, some probabilistic calculations can still accept a worse solution, where the acceptance becomes more difficult as the iteration goes [35]. Meanwhile, tabu search uses some list to avoid old solutions will be revisited for a certain period [36].

The evolution goes with the development of population-based metaheuristics. Genetic algorithm (GA) and invasive weed optimizer (IWO) are clear examples of these metaheuristics. Population-based metaheuristics give two main advantages. First, they improve the solution faster than the single solution-based metaheuristics. Second, they give broader tracing within the search space. GA has a simple solution in improving the solution by implementing a crossover strategy between two solutions and mutation for exploration [37]. In IWO, multiple new solutions are generated around every existing solution based on normal distribution [38]. Then, all solutions are ranked, and the worse ones are eliminated [38].

The evolution continues with the introduction of a swarm-based metaheuristic. A particle swarm optimizer (PSO) is an example of an early swarm-based metaheuristic. In the swarm-based metaheuristic, each solution can be seen as an autonomous agent that does not have centralized coordination. However, specific interaction and collective intelligence are conducted to help the improvement. As a member of the swarm, each agent conducts a guided search. It means each agent moves in a specific direction, called a guided search toward some references. As an early swarm-based metaheuristic, PSO deploys a simple strategy. In PSO, each agent moves toward the reference at a certain speed [39]. Its references are the combination between the local best solution and the global best solution [39].

In recent decades, the development of swarm-based metaheuristics is more extensive. Various aspects can be used as a baseline for developing new swarm-based metaheuristics. These aspects include reference, stochastic movement, acceptance rejection, segregation of roles, etc. Moreover, many shortcomings of metaheuristics deploy multiple strategies rather than a single strategy to improve its quality in achieving

a better final solution or achieving the objective faster. This multiple strategy/search approach is carried out in a single or multiple phases. Some metaheuristics are also equipped with random search to avoid the convergence achieved faster and

trapped in the local optimal. A detailed review of shortcoming swarm-based metaheuristics is presented in Table I. The mechanics of the proposed metaheuristic is presented in the last row to make a clear position and contribution of this work.

TABLE I. DETAIL SUMMARY OF SEVERAL SHORTCOMING METAHEURISTIC

No	Metaheuristic	Number of Phases	Number of Searches	Guided Searches	Random Searches	Segregation of Roles
1	COA [15]	2	3	searching toward the best solution in the current iteration; searching toward a randomly solution within the search space	searching around the local search space	yes
2	MSFA [29]	1	2	searching toward a target where the target is a random solution within the search space	searching around the local search	no
3	CSA [14]	3	4	searching toward the gap between the local best solution and global best solution; searching toward the mixture between the global best solution and local best solution	searching within the search space; searching within the local search space;	yes
4	KMA [16]	1	4	searching toward the resultant of several best solutions; searching toward the resultant of several better best solutions and avoiding the resultant of several worse best solutions; and searching toward the best solution in the current iteration.	searching within the search space	yes
5	SO [10]	2	4	searching of the best solution to avoid the corresponding solution; searching toward the best solution in different group; searching toward a solution outside the group with same index	searching within the search space	yes
6	ASBO [31]	3	3	searching relative to the average between the best and worst solutions in the current iteration; search toward the gap between the best and worst solutions in the current iteration; searching toward the gap between the corresponding solution and the best solution	-	no
7	TIA [33]	1	1	searching relative to all other solutions in the population	-	no
8	WSO [20]	2	3	searching toward the mixture between the global best solution and local best solution; searching of the best solution to avoid the corresponding solution.	searching between the previous solution and its new guided search.	yes
9	GSO [32]	1	1	searching toward the mixture between the local best solution and the global best solution	-	no
10	NGO [17]	2	2	searching toward a randomly selected solution	search within the local search space.	no
11	GPA [21]	1	2	searching toward the global best solution	searching within the local search space	no
12	MLBO [25]	1	1	searching toward the mixture between the best solution in the current iteration and a randomly selected solution	-	no
13	TIMBO [24]	3	3	searching toward the best solution in the current iteration; searching to avoid the worst solution in the current iteration; and searching relative to the mean member in the current iteration	-	no
14	SPO [28]	1	4	searching toward adjacent solutions; searching toward the gap between a solution from the primary group and a solution from the tertiary group; searching toward the average among a solution from each group; and searching toward the average of a solution from each group plus a randomly selected solution	-	no
15	RSLBO [27]	1	1	searching toward a randomly selected solution	-	no
16	this work	1	6	searching toward the global best solution; searching of the global best solution to avoid the corresponding solution; searching relative to a randomly selected solution;	searching within the local search space of the corresponding solution; searching within the local search space of the global best solution; searching within the search space	no

Table I indicates the limitations of the existing swarm-based metaheuristics. First, many swarm based metaheuristics prioritize guided search rather than random search. Second, some swarm-based metaheuristics deploys only one random search while some others do not deploy any random search. Third, searching methods implemented in every metaheuristic are still less than five methods. Fourth, these multiple searches are performed in the multiple phase process. Based on this circumstance, there is a room in developing swarm-based metaheuristic that deploys multiple searches and puts the balance between guided search and random search as in this work.

### III. PROPOSED MODEL

TOTO is a swarm-based metaheuristic with multiple strategies in every iteration. The multiple-strategy approach aims to cover each strategy's disadvantages because each has its strengths and weaknesses. The multiple strategy approach is interpreted by conducting six searches by each agent in every iteration. It means that these searches are mandatory for each agent. In other words, TOTO does not implement segregation of roles. It differs from other metaheuristics, such as RDA, KMA, or SKA, where the segregation of roles is implemented among the population. These searches are three guided searches and three random searches. Each search generates a candidate. Then, the best candidate among these six candidates is chosen as the selected candidate to be compared with the corresponding agent. If this candidate is better than the current solution of the corresponding agent, this candidate becomes the replacement as a new solution for the corresponding agent. This mechanism is also different from some metaheuristics, such as DTBO, GPA, NGO, and so on, where the searches are conducted sequentially. This new solution is then compared with the current global best solution. If this new solution is better than the global best solution, then the global best solution is updated.

The global best solution and a randomly selected agent become the reference in these three guided searches. In the first guided search, a candidate is generated based on the movement of the corresponding agent toward the best global solution. In

the second guided search, a candidate is generated based on the movement of the global best solution avoiding the corresponding solution. In the third guided search, a candidate is generated based on the movement of the corresponding agent relative to a randomly selected agent. If the randomly selected agent is better than the corresponding agent, then the third candidate is generated based on the movement of the corresponding agent toward the randomly selected agent. Otherwise, the third candidate is generated based on the movement of the corresponding agent, avoiding the randomly selected agent.

The local search space is used for the first and second random searches. On the other hand, local search space is not needed in the third random search. The local search space width declines linearly due to the increase in the iteration. In the first random search, a candidate is generated within the local search space of the corresponding agent. In the second search space, a candidate is generated within the local search space of the global best solution. A candidate is generated within the search space in the third random search. The illustration of these six searches is presented in Fig. 1.

Based on the previous explanation, the rationale of the proposed strategy is highlighted and summarized as follows.

- The multiple searches are proven better than single search as the multiple search approach is adopted in various shortcoming metaheuristics.
- The balance between the guided searches and random searches is designed to give balance between exploration capability and exploitation capability.
- Multiple references are adopted to expand the searching capability because searching process cannot depend on only single reference.
- The strict acceptance-rejection approach is adopted to avoid the searching process moves to the worse solution or area.

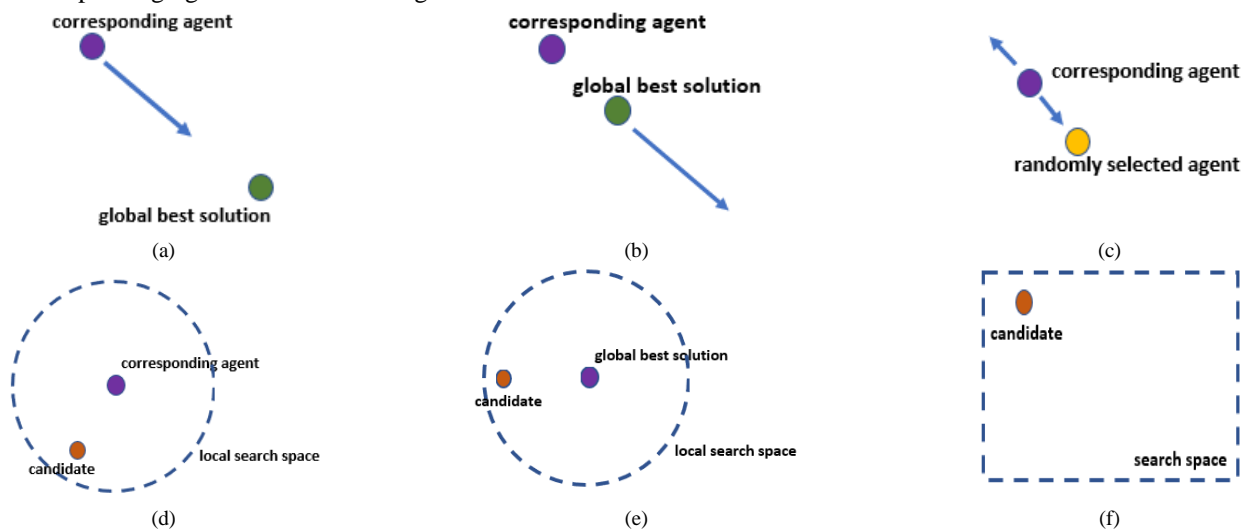


Fig. 1. Illustration of six searches in TOTO: (a) First guided search, (b) Second guided search, (c) Third guided search, (d) First random search, (e) Second random search, and (f) Third random search.

The algorithm of TOTO is then constructed based on its main concept as the formalization of this metaheuristic. This algorithm is presented in algorithm 1. Equation (1) to (11) is used as formalization of the related process. The annotations used in this paper are as follows.

$a$	agent
$A$	set of agents
$a_{best}$	global best agent
$a_{sel}$	randomly selected agent
$a_l$	lower boundary
$a_u$	upper boundary
$c_1$	first candidate
$c_2$	second candidate
$c_3$	third candidate
$c_4$	fourth candidate
$c_5$	fifth candidate
$c_6$	sixth candidate
$c_{sel}$	selected candidate
$f$	fitness function
$r_1$	real uniform random number between 0 and 1
$r_2$	integer uniform random number between 1 and 2
$r_3$	real uniform random number between -1 and 1
$t$	iteration
$t_{max}$	maximum iteration
$u$	uniform random

**algorithm 1:** main algorithm of TOTO

```

1  Begin
2  for all  $a$  in  $A$ 
3    generate  $a$  using (1)
4    update  $a_{best}$  using (2)
5  end for
6  for  $t = 1$  to  $t_{max}$ 
7    for all  $a$  in  $A$ 
8      first guided search using (3)
9      second guided search using (4)
10     third guided search using (5) and (6)
11     first random search using (7)
12     second random search using (8)
13     third random search using (9)
14     choose  $c_{sel}$  using (10)
15     update  $a$  using (11)
16     update  $a_{best}$  using (2)
17   end for
18 end for
19 end

```

$$a = U(a_l, a_u) \quad (1)$$

$$a_{best}' = \begin{cases} a, & f(a) < f(a_{best}) \\ a_{best}, & \text{else} \end{cases} \quad (2)$$

$$c_1 = a + r_1(a_{best} - r_2a) \quad (3)$$

$$c_2 = a_{best} + r_1(a_{best} - r_2a) \quad (4)$$

$$a_{sel} = U(A) \quad (5)$$

$$c_3 = \begin{cases} a + r_1(a_{sel} - r_2a), & f(a_{sel}) < f(a) \\ a + r_1(a - r_2a_{sel}), & \text{else} \end{cases} \quad (6)$$

$$c_4 = a + r_3 \left(1 - \frac{t}{t_m}\right) (a_u - a_l) \quad (7)$$

$$c_5 = a_{best} + r_3 \left(1 - \frac{t}{t_m}\right) (a_u - a_l) \quad (8)$$

$$c_6 = a_l + r_1(a_u - a_l) \quad (9)$$

$$c_{sel} = c \in C, \min(f(c)) \quad (10)$$

$$a' = \begin{cases} c_{sel}, & f(c_{sel}) < f(a) \\ a, & \text{else} \end{cases} \quad (11)$$

The explanation of (1) to (11) is as follows. Equation (1) states that the initial solution is randomized within the search space. Equation (2) states that the global best solution is updated by comparing the current value of the global best solution and the new value of the corresponding agent. If the corresponding agent is better than the global best solution, then the corresponding agent replaces the current global best solution. Equation (3) states that the candidate of the first guided search is generated based on the movement of the corresponding agent toward the global best solution. Equation (4) states that the candidate of the second guided search is generated based on the movement of the global best solution avoiding the corresponding agent. Equation (5) states that an agent is randomly selected among the population. Equation (6) states that the candidate of the third guided search is generated based on the movement of the corresponding agent relative to a randomly selected agent. Equation (7) states that the candidate of the first random search is generated based on the neighborhood search around the corresponding agent. Equation (8) states that the candidate of the second random search is generated based on the neighborhood search around the global best solution. Equation (9) states that the candidate of the third random search is generated based on the constant random search within the search space. Equation (10) states that the best candidate among these six candidates is selected to be compared with the corresponding solution. Equation (11) states that this candidate replaces the corresponding solution if this candidate is better than the corresponding solution.

The algorithm complexity of TOTO is presented as  $O(6t_{max}.n(A))$ . Based on this presentation, the complexity of TOTO is affected by two parameters: maximum iteration and population size. Each parameter has a linear proportion to the complexity.

#### IV. SIMULATION AND RESULT

This section discusses the performance analysis regarding the proposed metaheuristic. Some optimization tests are carried out due to provide the performance data. The first test is carried out to evaluate the performance of TOTO in solving classic 23 functions. These functions are standard as benchmark tests in many works proposing new metaheuristics. These functions consist of seven high dimension unimodal functions (Sphere, Schwefel 2.22, Schwefel 1.2, Schwefel 2.21, Rosenbrock, Step, and Quartic), six high-dimension multimodal functions (Schwefel, Rastrigin, Ackley, Griewank, Penalized, and Penalized 2), and ten fixed dimension multimodal functions (Shekel Foxholes, Kowalik, Six Hump Camel, Branin, Goldstein-Price, Hartman 3, Hartman 6, Shekel 5, Shekel 7, and Shekel 10). The second test is carried out to analyze the

hyperparameters of TOTO. The third test is carried out to evaluate the performance of TOTO in solving the practical optimization problem. The third test is carried out to evaluate the performance of TOTO in solving a real-world optimization problem. The portfolio optimization problem is chosen as the use case in this work.

The first test compares TOTO with five shortcoming swarm-based metaheuristics: GWO, MPA, MLBO, GSO, and GPA. GWO and MPA are older than the others but widely used in various optimization works. On the other hand, MLBO, GSO, and GPA are newer, but the use of these metaheuristics is still rare.

Several parameters regarding the first test are set as follows. The population size is 10. The maximum iteration is 50. The problem dimension is 50. The fishing aggregate devices for MPA are 0.5. The number of candidates for GPA is 5. The result is presented in Table II. The average fitness score, less than  $10^{-4}$ , is rounded to the nearest  $10^{-4}$  value. The best score in every function is presented in bold font. The cluster-based comparison is presented in Table III.

Table II indicates the excellent performance of TOTO in solving the 23 classic functions and competing with five other metaheuristics. The result presents that TOTO can find the

optimal global solution of eight functions (Sphere, Schwefel 2.22, Schwefel 2.21, Rastrigin, Ackley, Six Hump Camel, and Goldstein-Price). Moreover, TOTO also performs as the best metaheuristics in solving 17 functions. Seven high-dimension unimodal functions (Sphere, Schwefel 2.22, Schwefel 1.2, Schwefel 2.21, Rosenbrock, Step, and Quartic), five high-dimension multimodal functions (Rastrigin, Ackley, Griewank, Penalized, and Penalized 2), and five fixed dimension multimodal functions (Kowalik, Six Hump Camel, Branin, Goldstein-Price, and Shekel 7). But other metaheuristics also find the optimal global solutions in solving three functions (Schwefel 2.22, Six Hump Camel, and Branin).

Table III strengthens the superiority of TOTO. TOTO is better than GWO, MPA, MLBO, GSO, and GPA in solving 22, 21, 21, 19, and 15 functions, respectively. GWO becomes the easiest metaheuristic to beat. On the other hand, GPA becomes the most challenging metaheuristic to beat. Compared to GPA, TOTO is superior in solving high-dimension functions. On the other hand, GPA is superior in solving fixed-dimension functions.

In the second test, the performance of TOTO is evaluated due to the increase in the maximum iteration. This test has three maximum iterations: 100, 150, and 200. The result is presented in Table IV.

TABLE II. SIMULATION RESULT (AVERAGE FITNESS SCORE)

Function	GWO	MPA	MLBO	GSO	GPA	TOTO	Better than
Sphere	7.4328x10 <sup>2</sup>	2.8092x10 <sup>3</sup>	1.5375x10 <sup>4</sup>	2.2247x10 <sup>4</sup>	1.5742x10 <sup>2</sup>	<b>0.0000</b>	GWO, MPA, MLBO, GSO, GPA
Schwefel 2.22	0.0000	0.0000	0.0000	9.9113x10 <sup>63</sup>	0.0000	<b>0.0000</b>	GSO
Schwefel 1.2	2.6197x10 <sup>4</sup>	1.9080x10 <sup>4</sup>	3.8128x10 <sup>4</sup>	4.5054x10 <sup>4</sup>	8.5787x10 <sup>3</sup>	<b>0.0003</b>	GWO, MPA, MLBO, GSO, GPA
Schwefel 2.21	1.6220x10 <sup>1</sup>	2.2948x10 <sup>1</sup>	4.1616x10 <sup>1</sup>	4.0027x10 <sup>1</sup>	2.2449x10 <sup>1</sup>	<b>0.0000</b>	GWO, MPA, MLBO, GSO, GPA
Rosenbrock	2.2470x10 <sup>6</sup>	9.1381x10 <sup>5</sup>	1.0097x10 <sup>7</sup>	1.9217x10 <sup>7</sup>	7.9361x10 <sup>3</sup>	<b>4.8833x10<sup>1</sup></b>	GWO, MPA, MLBO, GSO, GPA
Step	9.2781x10 <sup>2</sup>	3.0696x10 <sup>3</sup>	1.5544x10 <sup>4</sup>	2.0205x10 <sup>4</sup>	1.4096x10 <sup>2</sup>	<b>9.0134</b>	GWO, MPA, MLBO, GSO, GPA
Quartic	0.8404	0.7544	8.0704	1.6288x10 <sup>1</sup>	0.3552	<b>0.0037</b>	GWO, MPA, MLBO, GSO, GPA
Schwefel	-1.2504x10 <sup>1</sup>	-3.6788x10 <sup>3</sup>	-5.125x10 <sup>3</sup>	-5.1120x10 <sup>3</sup>	<b>-1.1075x10<sup>4</sup></b>	-5.648x10 <sup>3</sup>	GWO, MPA, MLBO, GSO
Rastrigin	1.3626x10 <sup>1</sup>	3.1846x10 <sup>2</sup>	3.8775x10 <sup>2</sup>	3.7542x10 <sup>2</sup>	1.8105x10 <sup>2</sup>	<b>0.0000</b>	GWO, MPA, MLBO, GSO, GPA
Ackley	2.1327	9.1212	1.5494x10 <sup>1</sup>	1.9245x10 <sup>1</sup>	4.5612	<b>0.0000</b>	GWO, MPA, MLBO, GSO, GPA
Griewank	1.6133	2.4440x10 <sup>1</sup>	1.4181x10 <sup>2</sup>	2.0615x10 <sup>2</sup>	2.2037	<b>0.0021</b>	GWO, MPA, MLBO, GSO, GPA
Penalized	2.0545x10 <sup>5</sup>	9.0031x10 <sup>2</sup>	1.5497x10 <sup>6</sup>	1.1357x10 <sup>7</sup>	1.6725x10 <sup>1</sup>	<b>0.6670</b>	GWO, MPA, MLBO, GSO, GPA
Penalized 2	3.0459x10 <sup>7</sup>	3.4758x10 <sup>5</sup>	1.9640x10 <sup>7</sup>	4.4838x10 <sup>7</sup>	4.4657x10 <sup>1</sup>	<b>2.9620</b>	GWO, MPA, MLBO, GSO, GPA
Shekel Foxholes	2.3602x10 <sup>1</sup>	8.1399	3.9583	9.3023	<b>1.0927</b>	1.1388	GWO, MPA, MLBO, GSO
Kowalik	0.1970	0.0121	0.0046	0.0150	0.0053	<b>0.0015</b>	GWO, MPA, MLBO, GSO, GPA
Six Hump Camel	0.2901	-1.0037	-1.0235	-1.0316	-1.0316	<b>-1.0316</b>	GWO, MPA, MLBO
Branin	5.5155x10 <sup>1</sup>	0.9437	0.4016	0.3981	0.3981	<b>0.3981</b>	GWO, MPA, MLBO
Goldstein-Price	6.0122x10 <sup>2</sup>	8.5646	3.8505	4.2277	3.0001	<b>3.0000</b>	GWO, MPA, MLBO, GSO, GPA
Hartman 3	-0.0010	<b>-3.5541</b>	-0.0495	-0.0426	-0.0495	-0.0495	GWO, GSO
Hartman 6	-0.0051	-1.5787	-3.1239	-2.9718	<b>-3.2998</b>	-3.2103	GWO, MPA, MLBO, GSO
Shekel 5	-0.2731	-1.0622	-3.5332	<b>-6.6056</b>	-5.6785	-6.5205	GWO, MPA, MLBO, GPA
Shekel 7	-0.2936	-1.3635	-5.5027	-5.3298	-8.1483	<b>-7.6367</b>	GWO, MPA, MLBO, GSO, GPA
Shekel 10	-0.3217	-1.4165	-4.3085	-6.0164	<b>-9.5547</b>	-5.7171	GWO, MPA, MLBO

TABLE III. CLUSTER BASED COMPARISON

Cluster	GWO	MPA	MLBO	GSO	GPA
1	6	6	6	7	6
2	6	6	6	6	5
3	10	9	9	6	4
Total	22	21	21	19	15

TABLE IV. RELATION BETWEEN MAXIMUM ITERATION AND AVERAGE FITNESS SCORE

Function	Average Fitness Score		
	$t_{max} = 100$	$t_{max} = 150$	$t_{max} = 200$
Sphere	0.0000	0.0000	0.0000
Schwefel 2.22	0.0000	0.0000	0.0000
Schwefel 1.2	0.0000	0.0000	0.0000
Schwefel 2.21	0.0000	0.0000	0.0000
Rosenbrock	$4.8843 \times 10^1$	$4.8848 \times 10^1$	$4.8845 \times 10^1$
Step	8.7200	8.5779	8.9453
Quartic	0.0011	0.0008	0.0003
Schwefel	$-6.4723 \times 10^3$	$-6.7829 \times 10^3$	$-7.3773 \times 10^3$
Rastrigin	0.0000	0.0000	0.0000
Ackley	0.0000	0.0000	0.0000
Griewank	0.0000	0.0000	0.0000
Penalized	0.6270	0.5446	0.5199
Penalized 2	2.9369	2.8683	2.8744
Shekel Foxholes	1.1869	0.9980	0.9980
Kowalik	0.0015	0.0015	0.0006
Six Hump Camel	-1.0316	-1.0316	-1.0316
Branin	0.3981	0.3981	0.3981
Goldstein-Price	3.0000	3.0000	3.0000
Hartman 3	-0.0495	-0.0495	-0.0495
Hartman 6	-3.2681	-3.2221	-3.2169
Shekel 5	-7.3673	-7.7048	-8.2857
Shekel 7	-8.1151	-8.8114	-8.8197
Shekel 10	-8.5191	-8.4954	-9.0716

Table IV indicates that the increase in maximum iteration does not improve the performance of TOTO in most functions. There are two possible reasons for this circumstance. The first reason is the optimal global solution has been found when the maximum iteration is still low. The second reason is TOTO fails to improve, although the optimal global solution has yet to be found. In some functions, the increase of maximum iteration improves the performance of TOTO but is less significant. Fortunately, when the maximum iteration increases, TOTO can find the optimal global solution of Schwefel 1.2, Griewank, and Shekel Foxholes.

In the third test, the performance of TOTO is evaluated due to the increase of population size. There are three values of population size: 20, 30, and 40. The result is presented in Table V.

TABLE V. RELATION BETWEEN POPULATION SIZE AND AVERAGE FITNESS SCORE

Function	Average Fitness Score		
	$n(S) = 20$	$n(S) = 30$	$n(S) = 40$
Sphere	0.0000	0.0000	0.0000
Schwefel 2.22	0.0000	0.0000	0.0000
Schwefel 1.2	0.0003	0.0000	0.0000
Schwefel 2.21	0.0000	0.0000	0.0000
Rosenbrock	$4.8806 \times 10^1$	$4.8727 \times 10^1$	$4.8723 \times 10^1$
Step	8.0915	7.7506	7.2435
Quartic	0.0027	0.0012	0.0011
Schwefel	$-6.0975 \times 10^3$	$-6.746 \times 10^1$	$-6.8492 \times 10^3$
Rastrigin	0.0000	0.0000	0.0000
Ackley	0.0000	0.0000	0.0000
Griewank	0.0006	0.0000	0.0000
Penalized	0.5811	0.5286	0.4791
Penalized 2	2.8680	2.7882	2.8189
Shekel Foxholes	0.9980	0.9980	0.9980
Kowalik	0.0015	0.0006	0.0009
Six Hump Camel	-1.0316	-1.0316	-1.0316
Branin	0.3981	0.3981	0.3981
Goldstein-Price	3.0000	3.0000	3.0000
Hartman 3	-0.0495	-0.0495	-0.0495
Hartman 6	-3.2551	-3.2150	-3.2796
Shekel 5	-7.8440	-8.1561	-8.0214
Shekel 7	-8.1235	-8.7475	-9.3842
Shekel 10	-7.4649	-9.1384	-9.4117

Table V indicates there are also two circumstances regarding the increase in the population size. First, the performance of TOTO needs to improve the increase in population size. Like in the second test, this circumstance happens because the optimal global solution has been found or TOTO fails to improve, although the optimal global solution has yet to be found. Like in the first test, the population size increase also makes TOTO find the optimal global solution of three more functions: Schwefel 1.2, Griewank, and Shekel Foxholes. Meanwhile, in some other functions, TOTO can improve its performance.

In the fourth test, TOTO is challenged to solve the portfolio optimization problem as a real-world problem. In this work, the optimization determines the stocks the investor should buy. The stocks selected in this work are stocks from the banking sector in Indonesia, which are listed in Kompas 100 index. Kompas 100 is a list that consists of preferred stocks in Indonesia. 13 stocks in the banking sector are listed in Kompas 100. Detailed information regarding these stocks is presented in Table VI. In Table VI, the second column represents the stock index, the third column represents the stock price taken on 11 November 2022, and the third column represents the six-month capital gain of these stocks. The price and capital gain are presented in rupiah per share.

TABLE VI. STOCK INFORMATION

No	Stock Index	Price	Capital Gain
1	AGRO	525	-410
2	AMAR	366	42
3	ARTO	5,700	-2,100
4	BACA	136	-23
5	BBCA	8,725	1,350
6	BBHI	2,380	-2,450
7	BBNI	9,175	325
8	BBRI	4,590	260
9	BTPN	2,620	90
10	BMRI	10,100	2,225
11	BNBA	1,530	-310
12	BRIS	1,360	-55
13	BTPS	3,060	110

The objective of this portfolio optimization is to maximize the total capital gain. The total capital gain is calculated by accumulating the capital gain from the stocks. Two constraints limit this optimization. The first constraint is that the investor can purchase stock from 100 to 1,000 lots for each stock. One lot means 100 shares. The second constraint is that the maximum investment is 4,000,000,000 rupiah. This work benchmarked TOTO with GPA, GSO, MLBO, MPA, and GWO. The population size is ten, and the maximum iteration is 50. The result is presented in Table VII.

TABLE VII. PORTFOLIO OPTIMIZATION RESULT

No	Metaheuristic	Total Capital Gain
1	TOTO	382,157,681
2	GPA	384,955,295
3	GSO	289,744,968
4	MLBO	361,041,190
5	MPA	188,953,890
6	GWO	-16,222,768

Table VII indicates the good performance of TOTO. It is shown that TOTO outperforms GSO, MLBO, MPA, and GWO. Meanwhile, GPA is better than TOTO with very narrow gap. GWO becomes the metaheuristic with the poorest performance because it produces negative total capital gain which means loss.

## V. DISCUSSION

This fifth section will discuss the more profound analysis of the result and findings. This discussion consists of four parts. The first part is the discussion regarding the performance of TOTO in solving the 23 classic functions. The second part is the discussion regarding hyperparameters. The third part is the discussion regarding the performance of TOTO in solving the portfolio optimization problem. The fourth part is the discussion regarding the limitation of this work, especially the proposed metaheuristic.

The first discussion is about the performance of TOTO. TOTO performs well in solving the classic 23 functions and benchmarking with five other metaheuristics. As mentioned previously, TOTO performs as the best metaheuristic in solving 17 functions in the low population size and low maximum iteration circumstances. Meanwhile, TOTO can find the optimal global solution of eight functions in these circumstances. Moreover, TOTO can find the optimal global solution of three additional functions in the high maximum iteration or population size circumstances. Comparing metaheuristics, TOTO is better than GWO, MPA, MLBO, and GSO in almost all functions. Meanwhile, TOTO is better than GPA in 15 functions.

In general, the superiority of TOTO occurs in all three groups of these 23 functions. It means that TOTO can tackle various problems indicated in these functions. Due to the consideration that multimodal functions are used to test the exploration capability while unimodal functions are used to test the exploitation capability [40], the proposed TOTO is proven in having superior exploration and exploitation capabilities. Meanwhile, the inferiority of TOTO in solving the fixed dimension multimodal functions compared to GPA is that GPA performs very well in solving the fixed dimension multimodal problems. But the superiority of GPA in solving these functions comes with the consequence that the complexity of GPA is higher than TOTO. GPA generates several candidates in its searches, whether it's guided search toward the global best solution or its local search [21]. On the other hand, the tournament of six searches adopted in TOTO is proven better than the multiple candidate strategy adopted in GPA [21]. The result also proves that the sorting mechanism at the beginning of the iteration, as adopted in GWO [41], is unimportant.

The second discussion is related to hyperparameter analysis. Theoretically, the population size and maximum iteration positively affect the performance of metaheuristics. Meanwhile, the result in Table III and Table IV indicates that after some level, the increase of maximum iteration or population size remains the same. In the low maximum iteration or population size, the increase of one of these two parameters may improve the result. But, increasing one of these parameters does not produce a better result in the high maximum iteration and population size. In almost all functions, the acceptable solution, whether globally optimal or sub-optimal, has been found in the low maximum iteration and population size.

The third discussion is related to the performance of TOTO in solving the portfolio optimization problem. The result indicates that TOTO performs well in solving this problem by producing the second-best total capital gain. On the other hand, GPA becomes the best one. Fortunately, the performance gap between GPA and TOTO is very narrow. This test also indicates that real-world optimization problems should be used to test all metaheuristics. This result shows that the performance gap in the portfolio optimization problem is narrow compared with the classic 23 functions. It is also indicated that gaining significant improvement in the real-world optimization problem is much more complicated than in the theoretical optimization problem, primarily when the solution is based on an integer number.



The fourth discussion is regarding the limitation of this work and the proposed metaheuristic. This paper has presented that metaheuristics with multiple strategies generally perform better than other metaheuristics with one or two strategies. Meanwhile, TOTO adopts only six strategies (three guided and three random searches). On the other hand, there are a lot of other guided searches, and random searches can be chosen. It means TOTO can be improved by embedding more searches into this metaheuristic. But there are also limitations in embedding new searches. The first limitation is that it can only accommodate some searches into a single metaheuristic. The second limitation is that accommodating various searches into a single metaheuristic is also not wise because some may be less effective than other searches. The question is which searches are better than others. In other words, in which condition is some searches better than others.

The limitation that has been previously discussed makes the development of a new metaheuristic still possible. Although the references used in the guided search converge to the best or randomly selected solution, the selection of these references is still various. For example, many metaheuristics choose the best solution so far like in GPA [21], the best solution in the current iteration like in COA [15], or some best solutions as the leader. In many metaheuristics, for example in RLSBO [27], a randomly selected solution or some randomly selected solutions are uniformly selected among the population. On the other hand, like in SO [10], some solutions from some fixed-size groups are randomly chosen to reduce dependency on the best solution that may come with convergence too early.

The second opportunity comes from the motive to minimize the maximum iteration or population size. This work demonstrated that TOTO could perform well in low population sizes and maximum iteration. Theoretically, the performance of any metaheuristic can be improved by increasing the population size or maximum iteration to a very high number. Meanwhile, this work has demonstrated that choosing an appropriate strategy can be a better option than just scaling up the maximum iteration or population size, which is closer to the greedy approach.

The third opportunity comes from the scalability aspect. For example, large-scale problems with very high dimensions can be found easily in many real-world optimization problems, such as optimizing the purchase order of a supermarket with hundreds of stock-keeping units or optimizing the investor's portfolio with hundreds of stocks can be chosen. Large-scale optimization problems consume more computational resources, whether from increasing the maximum iteration or population size so that a sub-optimal solution can be found. It is still challenging to develop a new metaheuristic that does not need excessive computational resources in handling the large-scale problems.

## VI. CONCLUSION

A new swarm-based metaheuristic, namely TOTO, with its superior performance, has been presented in this paper. This paper also has presented a novel mechanism for adopting three guided and three random searches in a competition to find a better solution. The test result indicates that TOTO and precisely the strategy adopted in this metaheuristic perform

better not just in beating some previous metaheuristics but also finding acceptable solutions in the low maximum iteration and low population size circumstances. This work presents that TOTO can find the optimal global solution of eleven functions. Moreover, TOTO is better than GWO, MPA, MLBO, GSO, and GPA in solving 22, 21, 21, 19, and 15 functions, respectively. It means TOTO can tackle problems faced in the three groups of functions. In the test related to a portfolio optimization problem, TOTO is better than GWO, MPA, MLBO, and GSO by producing better total capital gain.

Due to the limitations of this work, this work and especially the proposed metaheuristic can become the baseline for future studies. These future studies can be carried out by improving TOTO or implementing TOTO to solve various real-world optimization problems.

## ACKNOWLEDGMENT

This work was financially supported by Telkom University, Indonesia.

## REFERENCES

- [1] E. Haitam, R. Najat, and J. Abouchabaka, "A vehicle routing problem for the collection of medical samples at home: case study of Morocco", *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 4, pp. 345-351, 2021.
- [2] M. Fekri, M. Heydari, and M. M. Mazdeh, "Two-objective optimization of preventive maintenance orders scheduling as a multi-skilled resource-constrained flow shop problem", *Decision Science Letters*, vol. 12, no. 1, pp. 41-54, 2023.
- [3] T. J. Kumar and M. Thangaraj, "An ordered precedence constrained flow shop scheduling problem with machine specific preventive maintenance", *Journal of Project Management*, vol. 8, no. 1, pp. 45-46, 2023.
- [4] A. Golab, E. S. Gooya, A. A. Falou, and M. Cabon, "A multilayer feed-forward neural network (MLFNN) for the resource-constrained project scheduling problem (RCPPSP)", *Decision Science Letters*, vol. 11, no. 4, pp. 407-418, 2022.
- [5] M. Subramanian, M. Narayanan, B. Bhasker, S. Gnanavel, M. H. Rahman, and C. H. P. Reddy, "Hybrid electro search with ant colony optimization algorithm for task scheduling in a sensor cloud environment for agriculture irrigation control system", *Complexity*, vol. 2022, ID. 4525220, pp. 1-15, 2022.
- [6] B. -G. Risi, F. Riganti-Fulginei, and A. Laudani, "Modern techniques for the optimal power flow problem: state of the art", *Energies*, vol. 15, ID. 6387, pp. 1-20, 2022.
- [7] S. R. Singh and R. Chaudhary, "Effect of inflation on EOQ model with multivariate demand and partial backlogging and carbon tax policy", *Journal of Future Sustainability*, vol. 3, no. 1, pp. 35-58, 2023.
- [8] J. Yang, "Big data perspective on financial operations revenue management approach", *Mathematical Problems in Engineering*, vol. 2022, ID. 8901666, pp. 1-9, 2022.
- [9] M. Wang, B. Mao, Y. Yang, R. Shi, and J. Huang, "Determining the level of service scale of public transport system considering the distribution of service quality", *Journal of Advanced Transportation*, vol. 2022, ID. 5120401, pp. 1-14, 2022.
- [10] F. A. Hashim and A. G. Hussein, "Snake optimizer: a novel metaheuristic optimization algorithm", *Knowledge-Based Systems*, vol. 242, ID. 108320, 2022.
- [11] M. Dehghani, S. Hubalovsky, and P. Trojovsky, "Tasmanian devil optimization: a new bio-inspired optimization algorithm for solving optimization problem", *IEEE Access*, vol. 10, pp. 19599-19620, 2022.
- [12] M. Dehghani, Z. Montazeri, G. Dhiman, O. P. Malik, R. Morales-Menendez, R. A. Ramirez-Mendoza, A. Dehghani, J. M. Guerrero, and L. Parra-Arroyo, "A spring search algorithm applied to engineering

- optimization problems”, Applied Sciences, vol. 10, ID. 6173, pp. 1-21, 2020.
- [13] S. Arora and S. Singh, “Butterfly optimization algorithm: a novel approach for global optimization”, Soft Computing, vol. 23, no. 3, pp. 715–734, 2019.
- [14] M. S. Braik, “Chameleon swarm algorithm: a bio-inspired optimizer for solving engineering design problems”, Expert Systems with Applications, vol. 174, ID. 114685, pp. 1-25, 2021.
- [15] 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.
- [16] S. Suyanto, A. A. Ariyanto, and A. F. Ariyanto, “Komodo mlipir algorithm”, Applied Soft Computing, vol. 114, pp. 1–17, 2022.
- [17] 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.
- [18] S. Z. Koohi, N. A. W. A. Hamid, M. Othman, and G. Ibragimov, “Raccoon optimization algorithm”, IEEE Access, vol. 7, pp. 5383-5399, 2019.
- [19] 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.
- [20] M. Braik, A. Hammouri, J. Atwan, M. A. Al-Betar, M. A. Awadallah, “White shark optimizer: a novel bio-inspired meta-heuristic algorithm for global optimization problems”, Knowledge-Based Systems, vol. 243, ID. 108457, pp. 1-29, 2022.
- [21] P. D. Kusuma and A. L. Prasasti, “Guided pelican algorithm”, International Journal of Intelligent Engineering and Systems, vol. 15, no. 6, pp. 179-190, 2022.
- [22] S. Kaur, L. K. Awasthi, A. L. Sangal, and G. Dhiman, “Tunicate swarm algorithm: a new bio-inspired based metaheuristic paradigm for global optimization”, Engineering Applications of Artificial Intelligence, vol. 90, ID. 103541, 2020.
- [23] M. I. A. Latiffi, M. R. Yaakub, and I. S. Ahmad, “Flower pollination algorithm for feature selection in tweets sentiment analysis”, International Journal of Advanced Computer Science and Applications, vol. 13, no. 5, pp. 429-436, 2022.
- [24] 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.
- [25] F. A. Zeidabadi, S. A. Doumari, M. Dehghani, and O. P. Malik, “MLBO: mixed leader based optimizer for solving optimization problem”, International Journal of Intelligent Engineering and Systems, vol. 14, no. 4, pp. 472–479, 2021.
- [26] M. Dehghani, Z. Montazeri, A. Dehghani, R. A. Ramirez-Mendoza, H. Samet, J. M. Guerrero, and G. Dhiman, “MLO: multi leader optimizer”, International Journal of Intelligent Engineering and Systems, vol. 13, no. 6, pp. 364–373, 2020.
- [27] 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.
- [28] A. Kaveh, S. Talatahari, and N. Khodadadi, “Stochastic paint optimizer: theory and application in civil engineering”, Engineering with Computers, vol. 38, pp. 1921-1952, 2022.
- [29] P. D. Kusuma and D. Adiputra, “Modified social forces algorithm: from pedestrian dynamic to metaheuristic optimization”, International Journal of Intelligent Engineering and Systems, vol. 15, no. 3, pp. 294–303, 2022.
- [30] M. Dehghani, E. Trojovská, and P. Trojovský, “A new human-based metaheuristic algorithm for solving optimization problems on the base of simulation of driving training process”, Scientific Report, vol. 12, No. 1, pp. 1–21, 2022.
- [31] 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.
- [32] 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.
- [33] 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.
- [34] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization”, IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, pp. 67-82, 1997.
- [35] A. Kuznetsov, L. Wieclaw, N. Poluyanenko, L. Hamera, S. Kandy, and Y. Lohachova, “Optimization of simulated annealing algorithm for s-boxes generating”, Sensors, vol. 22, ID. 6073, pp. 1-19, 2022.
- [36] M. A. Noman, M. Alatefi, A. M. Al-Ahmari, and T. Ali, “Tabu search algorithm based on lower bound and exact algorithm solutions for minimizing the makespan in non-identical parallel machines scheduling”, Mathematical Problems in Engineering, vol. 2021, ID. 1856734, pp. 1-9, 2021.
- [37] S. Katoch, S. S. Chauhan, and V. Kumar, “A review on genetic algorithm: past, present, and future”, Multimedia Tools and Applications, vol. 80, pp. 8091-8126, 2021.
- [38] M. Misaghi and M. Yaghoobi, “Improved invasive weed optimization algorithm (IWO) based on chaos theory for optimal design of PID controller”, Journal of Computational Design and Engineering, vol. 6, pp. 284-295, 2019.
- [39] A. G. Gad, “Particle swarm optimization algorithm and its applications: a systematic review”, Archives of Computational Methods in Engineering, vol. 29, pp. 2531-2561, 2022.
- [40] P. Trojovsky and M. Dehghani, “A new optimization algorithm based on mimicking the voting process for leader selection”, PeerJ Computer Science, vol. 8, ID. e976, pp. 1-40, 2022.
- [41] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer”, Advances in Engineering Software, vol. 69, pp. 46-61, 2014.