https://doi.org/10.31436/iiumej.v24i2.2700

TREBLE SEARCH OPTIMIZER: A STOCHASTIC OPTIMIZATION TO OVERCOME BOTH UNIMODAL AND MULTIMODAL PROBLEMS

PURBA DARU KUSUMA*AND ASHRI DINIMAHARAWATI

Computer Engineering, Telkom University, Bandung, Indonesia

**Corresponding author: purbodaru@telkomuniversity.ac.id*

(Received: 28 December 2022; Accepted: 18 March 2023; Published on-line: 4 July 2023)

ABSTRACT: Today, many metaheuristics have used metaphors as their inspiration and baseline for novelty. It makes the novel strategy of these metaheuristics difficult to investigate. Moreover, many metaheuristics use high iteration or swarm size in their first introduction. Based on this consideration, this work proposes a new metaheuristic free from metaphor. This metaheuristic is called treble search optimizer (TSO), representing its main concept in performing three searches performed by each member in each iteration. These three searches consist of two directed searches and one random search. Several seeds are generated from each search. Then, these searches are compared with each other to find the best seed that might substitute the current corresponding member. TSO is also designed to overcome the optimization problem in the low iteration or swarm size circumstance. In this paper, TSO is challenged to overcome the 23 classic optimization functions. In this experiment, TSO is compared with five shortcoming metaheuristics: slime mould algorithm (SMA), hybrid pelican komodo algorithm (HPKA), mixed leader-based optimizer (MLBO), golden search optimizer (GSO), and total interaction algorithm (TIA). The result shows that TSO performs effectively and outperforms these five metaheuristics by making better fitness scores than SMA, HPKA, MLBO, GSO, and TIA in overcoming 21, 21, 23, 23, and 17 functions, consecutively. The result also indicates that TSO performs effectively in overcoming unimodal and multimodal problems in the low iteration and swarm size.

ABSTRAK: Dewasa ini, terdapat ramai metaheuristik menggunakan metafora sebagai inspirasi dan garis dasar pembaharuan. Ini menyebabkan strategi baharu metaheuristik ini susah untuk dikaji. Tambahan, ramai metaheuristik menggunakan ulangan berulang atau saiz kerumunan dalam pengenalan mereka. Berdasarkan penilaian ini, kajian ini mencadangkan metaheuristk baharu bebas metafora. Metaheuristik ini dipanggil pengoptimum pencarian ganda tiga (TSO), mewakilkan konsep utama dalam pemilihan tiga pencarian yang dilakukan oleh setiap ahli dalam setiap ulangan. Ketiga-tiga carian ini terdiri daripada dua pencarian terarah dan satu pencarian rawak. Beberapa benih dihasilkan dalam setiap carian. Kemudian, carian ini dibandingkan antara satu sama lain bagi mencari benih terbaik yang mungkin berpotensi menggantikan ahli yang sedang digunakan. TSO juga direka bagi mengatasi masalah pengoptimuman dalam ulangan rendah atau lingkungan saiz kerumunan. Kajian ini TSO dicabar bagi mengatasi 23 fungsi pengoptimuman klasik. Eksperimen ini TSO dibandingkan dengan lima kekurangan metaheuristik: algoritma acuan lendir (SMA), algorithma hibrid komodo burung undan (HPKA), Pengoptimum Campuran berdasarkan-Ketua (MLBO), Pengoptimuman Carian Emas (GSO), dan algoritma jumlah interaksi (TIA). Dapatan kajian menunjukkan TSO berkesan menghasilkan dan lebih baik daripada kelima-lima metaheuristik dengan menghasilkan pemarkahan padanan terbaik berbanding SMA, HPKA, MLBO, GSO, dan TIA dalam mengatasi fungsi 21, 21, 23, 23, dan 17, secara

https://doi.org/10.31436/iiumej.v24i2.2700

berurutan. Dapatan kajian juga menunjukkan TSO turut berperanan efektif dalam mengatasi masalah modal tunggal dan modal ganda dalam iterasi rendah dan saiz kerumunan.

KEYWORDS: *optimization; metaheuristic; swarm intelligence; unimodal; multimodal*

1. INTRODUCTION

Metaheuristics is a popular tool extensively used in various optimization problems. Many optimization studies from a wide range of subjects use metaheuristics, such as in smart farming [1], path planning for autonomous robots [2], traffic forecasting [3], power systems [4], electric vehicle charge scheduling [5], and so on. Today, hundreds of metaheuristics exist and are ready to be used in any optimization problem. This circumstance becomes one of several reasons why metaheuristic is so popular. Moreover, there are optimization studies that hybridize a metaheuristic with other methods, whether they are metaheuristics or exact methods. Because metaheuristics are flexible in overcoming various optimization problems and easy to modify, there are many studies on hybridizing metaheuristics.

In general, this massive development of metaheuristics comes from two reasons. The first reason is that various things can be used as inspiration for searching mechanisms, especially nature. Many metaheuristics use nature, especially animal behavior, as their inspiration and transform it into an optimization or searching strategy. Several shortcoming metaheuristics that use animal behavior as their inspiration, such as the Komodo mlipir algorithm (KMA) [6], northern goshawk optimizer (NGO) [7], marine predator algorithm (MPA) [8], hybrid pelican Komodo algorithm (HPKA) [9], coati optimization algorithm (COA) [10], cheetah optimizer (CO) [11], chameleon swarm algorithm (CSA) [12], and so on. Several metaheuristics used the term leader to represent the reference during the directed search, such as mixed leader-based optimizer (MLBO) [13], random selected leader-based optimizer (RSLBO) [14], hybrid leader-based optimizer (HLBO) [15], and so on. Meanwhile, several metaheuristics declared their main concept or strategy for their name rather than using metaphors, such as total interaction algorithm (TIA) [16], golden search optimizer (GSO) [17], average and subtraction-based optimizer (ASBO) [18], and so on. The second reason is that no metaheuristic is suitable or superior in overcoming any optimization problem, as stated in the no-free-lunch theory. Each strategy has its strengths and weaknesses. In other words, no metaheuristic can accommodate all strategies.

There are several critiques following the massive development of new metaheuristics. First, many metaphor-based metaheuristics use their metaphor as a novelty or contribution. However, through the investigation of the algorithm and mathematical model, their method is slightly different from the previous ones [19]. Second, many studies proposing new metaheuristics exploited their ability to outperform the previous metaheuristics rather than highlighted their distinct mechanics in a clear explanation [19]. Besides, the performance of many metaheuristics is investigated in the high iteration or swarm in their first appearance. Moreover, these circumstances need to be clarified in several studies proposing a new metaheuristic. For example, NGO uses the behavior of the northern goshawk as metaphor and the maximum iteration is set to 1,000 during the evaluation [7]. The maximum iteration is also set to 1,000 in the first introduction of ASBO [18]. In the first introduction of GSO, the maximum iteration is set to 1,000 while the swarm size is set to 30 [17]. Unfortunately, the performance of these metaheuristics has not been investigated in the low swarm and low iteration circumstance.

https://doi.org/10.31436/iiumej.v24i2.2700

The objective of this work is to promote a new simple and metaphor-free swarm-based metaheuristic that works effectively with a low iteration number and swarm size. This metaheuristic is called a treble search optimizer (TSO) which comes from the three searches performed in the algorithm. These searches include two directed searches and one random search. The global optimal member and one randomly selected member become the references in these two directed searches. Meanwhile, the random search focuses on finding a better member near or around the corresponding member. In TSO, each member performs all these three searches in every iteration, which means TSO does not implement segregation of roles. One best seed is selected in every search, so three seeds from three searches are generated by a corresponding member in every search. Then, the best seed becomes the final seed as a substitute for the current corresponding member.

Based on this explanation, below are the novelties and contributions of this work.

1) This work promotes a novel swarm-based metaheuristic that is free from using metaphors, named treble search optimizer (TSO).
2) TSO was designed to overcome the optimization problem in the low iteration and swarm circumstance.
3) TSO performs three searches (two directed searches and one random search) where several seeds are generated from each search.
4) The performance of TSO is investigated using 23 classic functions to overcome.

2. RELATED WORKS

Investigating the existing metaheuristics is the first and main critical step in proposing a new metaheuristic. Within this investigation, it is very important to highlight the distinction, novelty, or uniqueness of the metaheuristic. This step is also important because there are hundreds of metaheuristics already in existence. Proposing a new metaheuristic without investigating the existing ones, especially the shortcoming ones, may end with proposing a metaheuristic like the existing ones. This investigation is also important because a new metaheuristic can be developed by modifying or hybridizing several existing metaheuristics.

Investigating a metaheuristic can be performed by classifying the metaheuristic based on several parameters. First, a metaheuristic should be classified as whether it uses metaphors. As mentioned, a metaphor-based metaheuristic should be rigorously investigated to address its distinct approach by abstracting the metaphor. Second, the algorithm and mathematical model following the algorithm should also be reviewed. After that, several parameters can be used to classify the metaheuristic, such as the number of searches, segregation of roles, and so on.

In many shortcoming metaheuristics, implementing multiple search strategies has become more popular rather than performing a single search strategy, as shown in Table 1. The main reason is that there is not any single search that can guarantee finding the optimal member. Using the global best member or the best member among the swarm becomes the most popular option so that this member is used in many swarm-based metaheuristics, such as KMA [6], HPKA [9], ASBO [18], and so on. Some other metaheuristics use local best member for their reference, such as in MPA [8], GSO [17], and so on. Meanwhile, there are also metaheuristics using the other members within the swarm as their reference, such as in TIA [16], NGO [7], and so on. Even moving toward

https://doi.org/10.31436/iiumej.v24i2.2700

the best member may guide the entire swarm toward the local optimal entrapment because the global best member lies somewhere else in the search space.

Table 1: List of shortcoming metaheuristics

No	Metaheuristic	Metaphor	Segregation of Roles	Number of Searches	Maximum Iteration	Swarm Size
1	KMA [6]	komodo dragon	yes	4	5,000 evaluations	5, 20-200
2	NGO [7]	northern goshawk	no	2	1,000	20-80
3	HPKA [9]	pelican and komodo dragon	yes	4	200	20
4	MPA [8]	marine predator	yes	5	500	50
5	CO [11]	cheetah	yes	3	$12x10^5$ evaluations	6
6	COA [10]	coati	no	3	200, 1,000	n/a
7	HLBO [15]	leader	no	2	1,000	n/a
8	MLBO [13]	leader	no	1	n/a	n/a
9	RSLBO [14]	leader	no	1	n/a	n/a
10	ASBO [18]	-	no	3	1,000	20-80
11	GSO [17]	-	no	1	1,000	30
12	TIA [16]	-	no	1	50	10
13	this work	-	no	3	40	5

The list of swarm-based shortcoming metaheuristics is displayed in Table 1. Table 1 consists of five pieces of information related to the corresponding metaheuristics. The first information is the metaphor used in the metaheuristic. The second information is whether the corresponding metaheuristic implements the segregation of roles. The third information is the number of searches implemented in the corresponding metaheuristic. The fourth information is the maximum iteration set in the first appearance of the corresponding metaheuristic. The fifth information is the swarm size set in the first appearance of the corresponding metaheuristic. There are 14 things that could be improved in metaheuristics in Table 1. The last row of Table 1 presents the attributes of TSO to give a clear view regarding the novelty and position of this work.

Table 1 presents many areas for improvement in the metaheuristic's use of metaphors, especially animals. Some metaheuristics use the term leader while others do not use a metaphor. Most metaheuristics do not perform segregation of roles so that in these metaheuristics, all members perform all searches adopted in the corresponding metaheuristic. Besides, most shortcoming metaheuristics perform multiple searches rather than a single search. In their first appearance, many metaheuristics are challenged to overcome optimization problems in the high maximum iteration or swarm size. Based on this explanation, the opportunity to propose a new swarm-based metaheuristic that is metaphor-free implements multiple searches and is challenged to overcome problems in the low maximum iteration and low swarm size is still open.

3. MODEL

TSO is built based on two approaches. First, TSO performs three searches where each output will be compared to find the best output. Second, there are multiple seeds generated in every search so that the best seed among these seeds will be chosen to compete with other selected seeds from other searches mentioned in the first approach.

https://doi.org/10.31436/iiumej.v24i2.2700

algorithm 1: treble search optimizer (TSO)

output: s_b
for all s **in** S
  initialize s using Eq. (1)
  update s_b using Eq. (2)
end for
for t=1 **to** t_{max}
  for all s **in** S
    select s_s using Eq. (3)
    for j=1 **to** n_c
      generate c_1, c_2, c_3 using Eq. (4), Eq. (5), Eq. (6)
    end for
    select c_{s1}, c_{s2}, c_{s3} using Eq. (7), Eq. (8), Eq. (9)
    select c_f using Eq. (10)
    update s using Eq. (11)
    update s_b using Eq. (2)
  end for
end for

c_1	first search seed
c_2	second search seed
c_3	third search seed
C_1	set of first search seed
C_2	set of second search seed
C_3	set of third search seed
c_{s1}	selected seed among first search seeds
c_{s1}	selected seed among second search seeds
c_{s3}	selected seed among third search seeds
c_f	final seed
f	objective function
n_c	number of seeds
s	member
S	set of members
s_b	global best member
s_s	selected member
s_u	upper boundary
s_l	lower boundary
t	iteration
t_{max}	maximum iteration
U	uniform random
U_r	real uniform random number
U_i	integer random number

In TSO, each corresponding member performs three searches which are two directed searches and one random search. The first directed search generates several seeds along the way, the corresponding member toward the global best member. The second directed search generates seeds relative to a selected random member within the swarm. In the second directed search, these seeds may be in the direction of the corresponding member toward the selected member or away from the selected member. This choice depends on the quality of the corresponding member and the randomly selected member. The first direction occurs if this randomly selected member is better than the corresponding member. Otherwise, the second direction takes place. In the third search, several seeds are generated around the corresponding member.

As a metaheuristic, TSO consists of two phases: initialization and iteration. In the initialization, all members are uniformly randomized within the search space. Meanwhile, the iteration phase represents the improvement where three searches are performed. At the end of the process, the global best member becomes the final member, i.e., the algorithm output.

The best seed is then chosen in every search. It means that there are now three selected seeds from three searches. Then, these seeds will compete among each other so that the best seed among these three seeds becomes the final seed. This final seed is then compared with the corresponding member. This final seed substitutes the corresponding member only if this final seed is better than the current corresponding member. Otherwise, the corresponding member remains static at the end of this iteration. This concept is then transformed into an algorithm and mathematical model. The mathematical model is displayed in Eq. (1) to Eq. (11). The formalization of TSO is displayed in algorithm 1. Below are annotations used in algorithm 1 and the following mathematical model. The visualization of TSO is presented in Fig. 1.

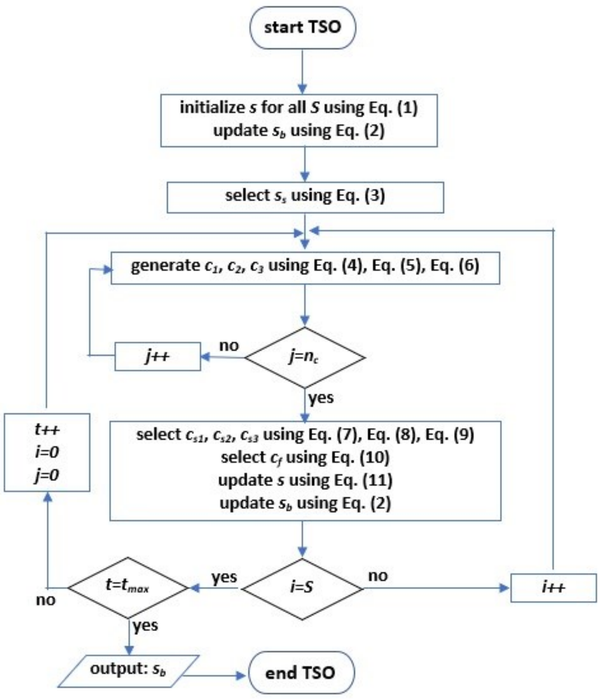

Fig. 1: Flowchart of treble search optimizer.

$$s = U_r(s_l, s_u) \tag{1}$$

$$s_b' = \begin{cases} s, f(s) < f(s_b) \\ s_b, otherwise \end{cases} \tag{2}$$

$$s_s = U(S) \tag{3}$$

$$c_1 = s + U_r(0,1)(s_b - U_i(1,2)s) \tag{4}$$

$$c_2 = \begin{cases} s + U_r(0,1)(s_s - U_i(1,2)s) \\ s + U_r(0,1)(s - U_i(1,2)s_s) \end{cases} \tag{5}$$

$$c_3 = s + 0.1U_r(-1,1)(s_u - s_l) \tag{6}$$

$$c_{s1} = c_1 \in C_1 \wedge min\big(f(c_1)\big) \tag{7}$$

$$c_{s2} = c_2 \in C_2 \wedge min\big(f(c_2)\big) \tag{8}$$

$$c_{s3} = c_3 \in C_3 \wedge min\big(f(c_3)\big) \tag{9}$$

$$c_f = c, min(c_1, c_2, c_3) \tag{10}$$

$$s' = \begin{cases} c_f, f\left(c_f\right) < f(s) \\ s, otherwise \end{cases} \tag{11}$$

Below is the detailed explanation of Eq. (1) to Eq. (11). The global best member becomes the final solution. Lines 2 to 5 represent the initialization phase. Lines 6 to 17 represent the iteration phase. Equation (1) describes that the initial member is uniformly randomized between the lower and upper boundary, i.e., search space. Equation (2) describes that the corresponding member substitutes the current global best member if this corresponding member is better than the current global best member. Equation (3) describes randomly selecting a member among the set of members. Equation (4) describes that the seed of the first search is generated along the way from the corresponding member toward the global best member. Equation (5) describes that the seed of the second search is generated based on the relation between the corresponding member and the randomly selected member. Equation (6) describes that the seed of the third search is generated near the corresponding member. Equations (7) to Eq. (9) describes that the best seed is selected from among seeds in every search. Equation (10) describes that the best seed among these three selected seeds becomes the final seed. Equation (11) describes that the final seed substitutes the current corresponding member if this final seed is better than the current corresponding member.

4. RESULTS

This section presents the experiment performed to evaluate the performance of TSO and its result. There are two experiments regarding this work. The first experiment is performed to evaluate the performance of TSO in overcoming a set of benchmark functions and the performance comparison between TSO and the sparing metaheuristics. The second experiment is performed to evaluate the hyperparameter of TSO. The 23 classic functions are chosen as the benchmark functions.

These 23 classic functions are chosen based on several reasons. The first reason is that these functions represent various problems with specific circumstances and challenges. The second reason is that these functions are very popular, so they are chosen in many studies proposing a new metaheuristic. These functions can be categorized into three groups: seven high-dimension unimodal functions, six high-dimension multimodal functions, and ten fixed-dimension multimodal functions. These functions also represent problems with various search spaces, from narrow to large ones.

In the first experiment, TSO is compared with five shortcoming metaheuristics: SMA, HPKA, MLBO, GSO, and TIA. These metaheuristics are chosen mainly because they are new. In their first appearance, these metaheuristics outperformed many previous metaheuristics. SMA outperformed many metaheuristics, such as the whale optimization algorithm (WOA), moth-flame optimizer (MFO), grey wolf optimizer (GWO), bat algorithm (BA), sine cosine algorithm (SCA), particle swarm optimization (PSO), firefly

https://doi.org/10.31436/iiumej.v24i2.2700

algorithm (FA), multi-verse optimizer (MVO), salp swarm algorithm (SSA), ant lion optimizer (ALO), and differential evolution (DE) [20]. HPKA outperformed four metaheuristics: GWO, MPA, KMA, and POA [9]. MLBO outperforms several metaheuristics, such as PSO, genetic algorithm (GA), teaching-learning based optimizer (TLBO), GWO, emperor penguin optimizer (EPOA), and so on [13]. GSO outperformed four metaheuristics: gravitational search algorithm (GSA), SCA, tunicate swarm algorithm (TSA), and GWO [17]. TIA outperformed five sparing metaheuristics: PSO, marine predator algorithm (MPA), GSO, directed pelican algorithm (GPA), and driving training-based optimizer (DTBO) [16].

Table 2: The simulation result of the first experiment

Function	Average Fitness Score					
	SMA	HPKA	MLBO	GSO	TIA	TSO
1	6.6663×10^4	6.2570×10^4	2.2305×10^4	6.0029×10^4	**0.0000**	**0.0000**
2	**0.0000**	**0.0000**	1.1222×10^{51}	7.5089×10^{69}	1.7930×10^{55}	**0.0000**
3	2.1289×10^5	2.0975×10^5	5.7327×10^4	1.7531×10^5	0.0316	**0.0000**
4	8.2362×10^1	7.9015×10^1	4.7480×10^1	5.6846×10^1	**0.0000**	**0.0000**
5	1.7908×10^8	1.9984×10^8	1.9085×10^7	1.2710×10^8	4.8879×10^1	**4.8803×10^1**
6	6.9099×10^4	6.9444×10^4	2.1638×10^4	5.9789×10^4	9.4253	**8.2081**
7	1.6824×10^2	1.3522×10^2	1.6547×10^1	9.2918×10^1	0.0122	**0.0021**
8	**-6.2418×10^3**	-5.9791×10^3	-3.9934×10^3	-4.3938×10^3	-2.3210×10^3	-4.6397×10^3
9	5.4539×10^2	5.6253×10^2	4.4459×10^2	5.1443×10^2	**0.0000**	**0.0000**
10	1.9750×10^1	1.9947×10^1	1.6483×10^1	1.9463×10^1	**0.0000**	**0.0000**
11	6.2192×10^2	7.1348×10^2	1.9505×10^2	5.3193×10^2	0.0087	**0.0000**
12	4.6092×10^8	3.3887×10^8	8.1216×10^6	2.2263×10^8	0.8217	**0.6182**
13	1.1149×10^9	8.5180×10^8	5.0042×10^7	4.7534×10^8	3.0902	**2.8085**
14	1.1963×10^1	1.3565×10^1	7.6193	1.1118×10^1	7.5131	**3.7733**
15	0.0106	0.0111	0.0141	0.0277	0.0046	**0.0006**
16	-0.8973	-0.8722	-0.9689	-0.9862	-1.0219	**-1.0314**
17	0.9389	0.7064	0.4966	0.8170	1.7409	**0.3986**
18	8.0109×10^1	2.2714×10^1	9.8071	2.6717×10^1	1.7821×10^1	**3.0137**
19	-0.0455	-0.0387	-0.0486	-0.0193	**-0.0495**	**-0.0495**
20	-2.7549	-2.7309	-2.8287	-2.5539	-2.3823	**-3.2980**
21	-2.9231	-4.0803	-2.8597	-2.7178	-4.1103	**-8.2464**
22	-3.7697	-3.0703	-2.8335	-3.3707	-3.4117	**-7.6623**
23	-3.5599	-2.8837	-2.3017	-3.1063	-2.5551	**-8.8215**

In the first experiment, several parameters are set based on a certain value. The swarm size is set to 5, which represents the low swarm. The maximum iteration is set to 40, which represents low iteration. The dimension is set to 50, which represents a high-dimension problem. In HPKA, all searches have equal opportunity. The result is displayed in Table 2, while the superiority of the TSO compared with the other metaheuristics based on the group of functions is displayed in Table 3. In Table 2, the best score is written in bold font. Meanwhile, the floating-point accuracy is set to 10^{-4} so that a score less than 10^{-4} is rounded to 0.

Table 2 indicates the excellent performance of TSO in terms of finding the optimal global member and producing the best scores among the metaheuristics. TSO could find the optimal global member of seven functions: Sphere, Schwefel 2.22, Schwefel 1.2, Schwefel 2.21, Rastrigin, Ackley, and Griewank. Meanwhile, TSO could also find the

https://doi.org/10.31436/iiumej.v24i2.2700

member near the global optimal member of three functions: Kowalik, Six Hump Camel, and Branin. TSO also performed the best of 22 functions out of 23 functions. However, several metaheuristics also performed the same value in six functions. These six functions are Sphere, Schwefel 2.22, Schwefel 2.21, Rastrigin, Ackley, and Hartman 3. TIA performed the same value in five functions (Sphere, Schwefel 2.21, Rastrigin, Ackley, and Hartman 3), while SMA and HPKA performed the same value in Schwefel 2.22.

Table 3 indicates the superiority of TSO among other competing metaheuristics in all groups of functions. TSO was better than SMA, HPKA, MLBO, GSO, and TIA in overcoming 21, 21, 23, 23, and 17 functions, respectively. It means that TSO was superior to MLBO and GSO. TSO was superior to SMA and HPKA. Meanwhile, TSO was still superior to TIA, although TIA is the most challenging metaheuristic to beat. This result indicates the superiority of TSO in overcoming all three kinds of problems: high-dimension unimodal problems, high-dimension multimodal problems, and fixed-dimension multimodal problems.

Table 3: TSO superiority among other metaheuristics based on group of functions

Group	Number of Functions				
	SMA	HPKA	MLBO	GSO	TIA
1	6	6	7	7	4
2	5	5	6	6	4
3	10	10	10	10	9
Total	21	21	23	23	17

The second experiment was performed to evaluate the sensitivity or hyperparameter. There were three parameters evaluated in this experiment: maximum iteration, swarm size, and the number of seeds. This experiment was performed by implementing TSO to overcome 23 classic functions with several values of these parameters. The maximum iteration in the first sub-experiment was set at 10, 20, and 30. The result is displayed in Table 4. In the second sub-experiment, the swarm size was set at 10, 15, and 20. The result is displayed in Table 5. In the third experiment, the number of seeds was set to 3, 6, and 9. The result is displayed in Table 6.

Table 4 indicates that all functions have achieved an acceptable member in the low iteration. In almost all functions, the result produced by TSO in the low iteration circumstance was still competitive compared with the result produced by other metaheuristics, as seen in Table 2. Moreover, convergence was achieved in the early iteration in nine functions (Schwefel 2.22, Schwefel, Penalized, Penalized 2, Shekel Foxholes, Six Hump Camel, Branin, Hartman 3, and Hatman 6). Among these nine functions, one function was a high dimension unimodal function, two functions were high-dimension multimodal functions, and six functions were fixed-dimension multimodal functions.

Table 5 indicates that the increase in swarm size after five members is insignificant in improving the member quality in almost all functions. Stagnancy occurred in nine functions because the optimal global member was achieved. Stagnancy also occurred in five functions, although a globally optimal member had yet to be achieved. Less significant improvement occurred in nine functions.

Table 4: Performance of TSO with several values of maximum iteration

Function	Average Fitness Score		
	t_{max} = 10	t_{max} = 20	t_{max} = 30
1	2.6467	0.0001	0.0000
2	0.0000	0.0000	0.0000
3	7.7478x10^{2}	1.3350	0.0034
4	1.8470	0.0259	0.0003
5	1.3821x10^{2}	4.8819x10^{1}	4.8819x10^{1}
6	1.1822x10^{1}	8.0214	8.1741
7	0.0193	0.0043	0.0019
8	-4.0668x10^{3}	-4.2352x10^{3}	-4.6166x10^{3}
9	3.5869x10^{1}	0.0024	0.0000
10	0.7767	0.0020	0.0000
11	0.5366	0.0060	0.0035
12	0.7894	0.6502	0.6806
13	3.2482	2.9358	2.8653
14	4.4341	4.9429	3.6936
15	0.0017	0.0006	0.0005
16	-1.0304	-1.0309	-1.0308
17	0.3989	0.3984	0.3984
18	6.6955	3.0042	3.0079
19	-0.0495	-0.0495	-0.0495
20	-3.2050	-3.239	-3.2961
21	-5.7886	-7.2599	-7.9212
22	-5.8045	-7.1234	-7.5447
23	-6.1753	-6.9030	-7.8198

Table 5: Performance of TSO with several values of swarm size

Function	Average Fitness Score		
	N(X) = 10	N(X) = 15	N(X) = 20
1	0.0000	0.0000	0.0000
2	0.0000	0.0000	0.0000
3	0.0000	0.0000	0.0000
4	0.0000	0.0000	0.0000
5	4.8717x10^{1}	4.8717x10^{1}	4.8713x10^{1}
6	6.8744	6.4879	6.2007
7	0.0007	0.0005	0.0004
8	-5.7240x10^{3}	-5.2949x10^{3}	-5.3398x10^{3}
9	0.0000	0.0000	0.0000
10	0.0000	0.0000	0.0000
11	0.0000	0.0000	0.0000
12	0.4685	0.4531	0.3879
13	2.6890	2.5259	2.4414
14	2.4780	1.1478	1.0927
15	0.0014	0.0004	0.0004
16	-1.0316	-1.0316	-1.0316
17	0.3981	0.3981	0.3981
18	3.0015	3.0000	3.0000
19	-0.0495	-0.0495	-0.0495
20	-3.3071	-3.3038	-3.3137
21	-9.2349	-9.4629	-9.6812
22	-9.6167	-1.0084x10^{1}	-1.0193x10^{1}
23	-9.1412	-1.0034x10^{1}	-1.0293x10^{1}

Table 6: Performance of TSO with several values of number of seeds

Function	Average Fitness Score		
	n(C) = 3	n(C) = 6	n(C) = 9
1	0.0000	0.0000	0.0000
2	0.0000	0.0000	0.0000
3	0.0005	0.0000	0.0000
4	0.0000	0.0000	0.0000
5	$4.8850\text{x}10^{1}$	$4.8775\text{x}10^{1}$	$4.8775\text{x}10^{1}$
6	8.5787	7.7692	7.4645
7	0.0033	0.0012	0.0005
8	$-4.4692\text{x}10^{3}$	$-4.6142\text{x}10^{3}$	$-4.8921\text{x}10^{3}$
9	0.0000	0.0000	0.0000
10	0.0000	0.0000	0.0000
11	0.0000	0.0000	0.0000
12	0.7405	0.6131	0.5368
13	2.9124	2.7398	2.7218
14	4.7629	5.9951	3.5355
15	0.0023	0.0015	0.0013
16	-1.0310	-1.0310	-1.0316
17	0.3987	0.3986	0.3982
18	3.0053	3.0007	3.0009
19	-0.0495	-0.0495	-0.0495
20	-3.2625	-3.2822	-3.2983
21	-8.0977	-6.3341	-8.3292
22	-7.0566	-8.3640	-7.9018
23	-6.5042	-8.6724	-8.5935

Table 6 indicates that the increase in the number of seeds was less sensitive to the improvement of the performance of TSO. The average fitness score tended to fluctuate or remain static in almost all problems. Meanwhile, the less significant improvement occurred in six functions (Penalized, Shekel Foxholes, Branin, Quartic, Shekel 5, Shekel 7, and Shekel 10).

5. DISCUSSION

This section presents an in-depth evaluation of the relation between the result and the findings. This discourse is divided into four parts. The first part is a discourse related to the performance of the TSO and the linkage with the chosen exploitation-exploitation strategy. The second part is a discourse regarding the hyperparameter evaluation. The third part is a discourse related to the algorithm complexity of TSO. The fourth part is a discourse related to the limitation of this work, especially the metaheuristic.

The first discourse is related to the evaluation of the experiment result. TSO performed effectively in overcoming the 23 classic functions. Its performance was superior in all groups of these functions. Based on the superior result in overcoming unimodal functions, TSO performed exploitation effectively. Moreover, TSO was also good at performing exploration based on the superior result in overcoming multimodal functions, whether they were high-dimension multimodal functions or fixed-dimension multimodal functions. The performance gap between TSO and the sparing metaheuristics was also broad, especially in overcoming high-dimension functions. This gap needs to be more comprehensive in overcoming fixed-dimension functions.

https://doi.org/10.31436/iiumej.v24i2.2700

The superiority of TSO proves that the strategy implemented in TSO is better than the strategy implemented in the sparing metaheuristics. First, implementing multiple searches is proven better than a single search because each has its strengths and weaknesses. Second, each member needs to perform multiple searches in every iteration. Third, the tournament-based approach was better than the sequential-based approach.

The second discourse is related to the sensitivity analysis of the hyperparameter. This work evaluated three parameters: maximum iteration, swarm size, and the number of seeds. In general, the increase in maximum iteration improves the quality of the member. In the low maximum iteration, increasing maximum iteration improves the member mostly in overcoming the unimodal functions. On the other hand, the increase in the maximum iteration is less significant in improving the quality of the member. On the contrary, the swarm size does not improve the member quality in overcoming unimodal functions. The increase in swarm size improves the member quality, mostly in overcoming the multimodal functions. However, this improvement is also insignificant because the near-optimal or optimal global member has been found in the low maximum iteration and low swarm size circumstances. Increasing the number of seeds improves the quality of members in overcoming several functions. These functions can be found in both unimodal and multimodal functions. However, the improvement could be more significant.

The third discourse is related to algorithm complexity. The complexity of TSO can be displayed as $O(3t_{max}.n(X).n(C))$. Based on this presentation, the complexity is linear to one of three parameters: the maximum iteration, swarm size, or the number of seeds. Fortunately, the computational process of TSO is still competitive because the acceptable member has reached a low maximum iteration, low swarm size, and a low number of seeds.

The fourth discourse is related to the limitation of the algorithm and this work. There are several limitations regarding this algorithm and this work. First, TSO still needs to find the final member, the optimal global member or near-optimal member in overcoming six functions. These six functions are two high-dimension unimodal functions (Rosenbrock and Step), three high-dimension multimodal functions (Schwefel, Penalized, and Penalized 2), and one fixed-dimension multimodal function (Hartman 3). This fact strengthens the no-free-lunch theory. Although TSO is superior among the sparing metaheuristics, there are still problems where TSO needs to find the optimal global member. Meanwhile, there is an opportunity to find the optimal global member for these functions by setting the system to high maximum iteration and swarm size. However, this scenario was not performed in this work as its focus is on the low maximum iteration and low swarm size. This limitation can be used as a baseline to improve the current form of TSO so that the improved version can overcome these six functions in future studies. This work has a limitation on implementing TSO in overcoming theoretical optimization problems only. In some studies, the proposed metaheuristic has been challenged to overcome the theoretical problems only, while in other studies, the metaheuristic has also been challenged to overcome practical problems.

6. CONCLUSION

The development and evaluation of a new swarm-based metaheuristic, treble search optimizer (TSO) has been introduced in this paper. Referring to its name, the central concept of TSO is performing three searches: two directed searches and one random search. Each member in every iteration performs these three searches. Several seeds are generated from each search. The experiment result presents that TSO performed effectively in overcoming 23 classical functions. TSO outperformed five shortcoming metaheuristics chosen as the sparing metaheuristics in this work. TSO was better than SMA, HPKA, MLBO, GSO, and TIA in overcoming 21, 21, 23, 23, and 17 functions, respectively. TSO could find the optimal global member of seven functions in low maximum iteration and low swarm size circumstances. When the swarm size was set to moderate, there were three more functions where TSO can find their optimal global member. TSO also performed the best member of 22 functions out of 23 functions.

Future studies can be conducted in several ways. Improvement is still open for TSO, especially in finding the optimal global member of the six functions that TSO still needs to find in this work. More theoretical experiments should be performed to enrich the investigation of the strengths and weaknesses of TSO. Moreover, future studies can also be performed by implementing TSO to overcome many kinds of practical optimization problems.

ACKNOWLEDGMENT

This work was financially supported by Telkom University, Indonesia.

REFERENCES

[1] Ramezanpour MR, Farajpour M. (2022) Application of artificial neural networks and genetic algorithm to predict and optimize greenhouse banana fruit yield through nitrogen, potassium and magnesium. PLoS ONE, 17(2): e0264040. https://doi.org/10.1371/journal.pone.0264040.

[2] Thammachantuek I, Ketcham M. (2022) Path planning for autonomous mobile robots using multi-objective evolutionary particle swarm optimization. PLoS ONE, 17(8): e0271924. https://doi.org/10.1371/journal.pone.0271924.

[3] Sivakumar R, Angayarkanni SA, Ramana RYV, Sadiq AS. (2022) Traffic flow forecasting using natural selection based hybrid bald eagle search-grey wolf optimization algorithm. PLoS ONE, 17(9): e0275104. https://doi.org/10.1371/journal.pone.0275104.

[4] Hoballah A, Azmy AM. (2023) Constrained economic dispatch following generation outage for hot spinning reserve allocation using hybrid grey wolf optimizer. Alexandria Engineering Journal, 62: 169-180. https://doi.org/10.1016/j.aej.2022.07.033.

[5] Sowmya R, Sankaranarayanan V. (2022) Optimal scheduling of electric vehicle charging at geographically dispersed charging stations with multiple charging piles. International Journal of Intelligent Transportation Systems Research, 20: 672-695. https://doi.org/10.1007/s13177-022-00316-2.

[6] Suyanto, Ariyanto AA, Ariyanto AF. (2022) Komodo mlipir algorithm. Applied Soft Computing, 114: 108043. https://doi.org/10.1016/j.asoc.2021.108043.

[7] Dehghani M, Hubalovsky S, Trojovsky P. (2021) Nortehrn goshawk optimization: a new swarm-based algorithm for solving optimization problems. IEEE Access, 9: 162059-162080. doi: 10.1109/ACCESS.2021.3133286.

[8] Faramarzi A, Heidarinejad M, Mirjalili S, Gandomi AH. (2020) Marine predators algorithm: a nature-inspired metaheuristic. Expert Systems with Applications, 152: 113377. https://doi.org/10.1016/j.eswa.2020.113377.

[9] Kusuma PD, Dinimaharawati A. (2022) Hybrid pelican komodo algorithm. International Journal of Advanced Computer Science and Applications, 13(6): 46-55. https://dx.doi.org/10.14569/IJACSA.2022.0130607.

[10] 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.knosys.2022.110011.

[11] Akbari MA, 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. https://doi.org/10.1038/s41598-022-14338-z.
[12] Braik MS. (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.
[13] Zeidabadi FA, Doumari SA, Dehghani M, Malik OP. (2021) MLBO: mixed leader based optimizer for solving optimization problems. International Journal of Intelligent Engineering and Systems, 14(4): 472-479. doi: 10.22266/ijies2021.0831.41.
[14] Zeidabadi FA, Dehghani M, Malik OP. (2021) RSLBO: random selected leader based optimizer. International Journal of Intelligent Engineering and Systems, 14(5): 529-538. doi: 10.22266/ijies2021.1031.46.
[15] Dehghani M, Trojovsky P. (2022) Hybrid leader based optimization: a new stochastic optimization algorithm for overcoming optimization applications. Scientific Reports, 12: 5549. https://doi.org/10.1038/s41598-022-09514-0.
[16] Kusuma PD, 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. doi: 10.22266/ijies2023.0228.20.
[17] Noroozi M, Mohammadi H, Efatinasab E, Lashgari A, Eslami M, Khan B. (2022) Golden search optimization algorithm. IEEE Access, 10: 37515-37532. https://doi.org/10.1109/ACCESS.2022.3162853.
[18] 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. https://doi.org/10.7717/peerj-cs.910.
[19] Swan J, Adriaensen S, Brownlee AEI, Hammond K, Johnson CG, Kheiri A, Krawiec F, Merelo JJ, Minku LL, Ozcan E, Pappa GL, Garcia-Sanchez P, Sorensen K, Vob S, Wagner M, White DR. (2022) Metaheuristics in the large. European Journal of Operational Research, 297(2): 393-406. https://doi.org/10.1016/j.ejor.2021.05.042.
[20] Li S, Chen H, Wang M, Heidari AA, Mirjalili S. (2020) Slime mould algorithm: A new method for stochastic optimization. Future Generation Computer Systems, 111: 300-323. https://doi.org/10.1016/j.future.2020.03.055.