Received: September 27, 2022. Revised: November 7, 2022.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Total Interaction Algorithm: A Metaheuristic in which Each Agent Interacts with All Other Agents

Purba Daru Kusuma[1]* **Astri Novianty[1]**

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

Abstract: This research introduced a novel simple metaheuristic called as total interaction algorithm (TIA). TIA is a swarm intelligence which relies on the interaction among solutions in the population. The core and distinct concept of TIA is that each solution interacts with all other solutions in every iteration to find the best possible solution. Then, this new alternative replaces the current solution if it is better than the current solution. In this research, TIA is tested to solve 23 functions that represent both unimodal and multimodal problems. TIA is benchmarked with five metaheuristics: particle swarm optimization (PSO), marine predator algorithm (MPA), golden search optimizer (GSO), guided pelican algorithm (GPA), and driving training-based optimizer (DTBO). The result indicates that TIA is superior to other benchmark metaheuristics, especially in solving the high dimension functions. TIA is better than PSO, MPA, GSO, GPA, and DTBO in 22, 21, 16, 11, 13 functions. The result also indicates that the increase of the maximum iteration improves the performance of TIA mostly in solving high dimension unimodal functions. Meanwhile, the increase of the population size is less significant to improve its performance. Overall, this research resumes that the interaction with as many as possible individuals is proven better than with only selected individuals as implemented in many other metaheuristics.

Keywords: Computational intelligence, Swarm intelligence, Metaheuristic, Optimization.

1. Introduction

Swarm intelligence-based metaheuristic is popular in many optimization studies, especially in the numerical problem. Its popularity comes from the adoption of a stochastic approach so that it is flexible in the system with limited computational resource because it does not trace all possible solutions. It is also flexible to solve any optimization problem because it implements trial and error approach by focusing on the objectives and constraints.

Today there are a lot of swam intelligence-based metaheuristics. Some metaheuristics, such as particle swarm optimization (PSO), grey wolf optimizer (GWO), and marine predator algorithm (MPA), have been utilized extensively in many optimization studies. The examples are as follows. PSO was combined with firefly optimization to solve power flow system with various objectives, i.e., minimizing the total generation cost, improving voltage profile, reducing power loss in the transmission line, and enhancing voltage stability [1]. PSO was also used in the path planning for autonomous robot [2]. MPA was utilized to optimize the power flow in the multi-region system [3].

In recent years, there are many studies proposing new swarm intelligence-based metaheuristics. Many metaheuristics were built based on the animal's behavior, such as emperor penguin colony (EPC) [4], red deer algorithm (RDA) [5], butterfly optimization algorithm (BOA) [6], Komodo mlipir algorithm (KMA) [7], stochastic Komodo algorithm (SKA) [8], squirrel search optimizer (SSO) [9], modified honey badger algorithm (MHBA) [10], northern goshawk optimizer (NGO) [11], guided pelican algorithm (GPA) [12], and so on. Some metaheuristics were built based on the human social activities, such as teaching learning-based optimizer (TLBO) [13], election-based optimization algorithm (EBOA) [14], driving training-based optimizer (DTBO) [15], modified social forces algorithm (MSF) [16], and so

Received: September 27, 2022. Revised: November 7, 2022.

on. Some metaheuristics were built based on the game mechanics, such ring toss game-based optimizer (RTGBO) [17], darts game optimizer (DGO) [18], and puzzle optimization algorithm (POA) [19]. Some metaheuristics used term leader as the inspiration, such as random selected leader-based optimizer (RSLBO) [20], hybrid leader-based optimizer (HLBO) [21], mixed leader-based optimizer (MLBO) [22], multi leader optimizer (MLO) [23], and so on.

In general, the collective intelligence becomes the central consideration in any swarm intelligence-based metaheuristics. It mostly becomes the guidance for every agent to improve its quality. The guidance is usually chosen from the agent in the system. In some metaheuristics, such as BOA [6] and golden search algorithm (GSO) [24], each agent interacts only with the highest quality agent. In some metaheuristics, such as GWO [25], PSO [26], MLO [23], SSO [9], and KMA [7], each agent interacts with several best agents. In some metaheuristics, such as DTBO [15] and NGO [11], each agent interacts with a randomly selected agent. In some metaheuristics, such as HLBO [21] and MLBO [22], each agent interacts with the combination between the highest quality agent and randomly selected agent. Meanwhile, metaheuristic in which each agent interacts with all other agents is hard to find. The main problem of using only one or several agents is that the system may lose bigger opportunity to improve the current solution.

Therefore, this research proposes a new swarm intelligence-based metaheuristic called as total interaction algorithm (TIA). As its name suggests, each agent interacts with all other agents in the system to determine the target for its guided movement. TIA is built as a single-phase metaheuristic which carries out a random search only in the initialization phase. TIA does not deploy local search as the additional strategy. Moreover, the calculation in TIA is simple. TIA also implements strict acceptance-rejection approach to ensure that the accepted solution is always better than the previous solution.

The main contribution of this work is proposing a new metaheuristic in which each agent interacts with all other agents in the system. Commonly in swarm intelligence-based metaheuristics, each agent interacts only with single agent or several agents. TIA uses single-phase approach and simple calculation while most of swarm intelligence-based metaheuristics consist of two or three phases. The local search or random search which is usually carried out in the last phase is usually deployed to counter the local optimal issue.

The remaining content of the paper is arranged as follows. The shortcoming studies regarding the swarm intelligence-based metaheuristics are discussed in the second section. The detailed formalization of TIA is presented in the third section. The test carried out to evaluate the performance of TIA is presented in the fourth section. The in-depth investigation regarding the test result, its relationship with the theory, and limitation of this work are discussed in the fifth section. The conclusion and future research potential are summarized in the sixth section.

2. Related works

There are several parts to be concerned in proposing a new swarm intelligence-based metaheuristic or modifying the existing swarm intelligence-based metaheuristic. The first part is the partner which an agent interacts with to improve its current solution. In some metaheuristics, an agent interacts only with one partner. Meanwhile, in some others, an agent interacts with several partners. The second part is the phase carried out in every iteration. Some metaheuristics deploy single phase while some others deploy multiple phases. The third part is the random variation. Some metaheuristics use uniform distribution while some others use normal, or other distribution. The fourth part is the calculation regarding the movement. Some metaheuristics deploy simple calculation while some others deploy complicated calculation. The fifth part is the acceptance-rejection approach regarding the possibility of a new solution replaces the current solution. In the strict acceptance-rejection approach, new solution replaces the current solution only if it is better than the current solution. In the soft acceptance-rejection approach, a worse solution may replace the current solution based on certain condition. In non-acceptance-rejection approach, the new solution replaces the current solution without considering the quality of the new solution. The sixth part is the segregation of roles among agents. The detail strategy carried out in the recent swarm intelligence-based metaheuristics are explained below.

In BOA, an agent may interact with the best agent or with two randomly selected agents in every iteration [6]. This choice depends on the random number generated for the corresponding agent in every iteration. If the random number is less than the predetermined threshold, then the agent takes a step toward the best agent. It means that BOA implements the segregation of roles stochastically. The best agent is selected among the population in the beginning of every iteration. Otherwise, the agent carries out a random search between the two selected agents. BOA carries out only a single phase in every iteration. The calculation of the agent movement is also simple where the guided search or random search can be written in single process. BOA deploys uniform random number only. BOA does not implement either strict acceptance-rejection approach.

In HLBO, an agent interacts with two partners in every iteration [21]. The first partner is the best agent in the current iteration. The second partner is a randomly selected agent. HLBO carries out two phases in every iteration. The first phase is the movement toward the hybrid leader. The hybrid leader is constructed from three elements, i.e., the corresponding agent, the best agent, and the randomly selected agent. The proportion of each element depends on the normalized quality of the element relative to all three elements. The agent moves toward the hybrid leader if the hybrid leader is better than the corresponding agent. Otherwise, the corresponding agent avoids the hybrid leader. The second phase is the local search carried out by the corresponding agent. In this phase, the local search space width declines linearly due to the increase of iteration. HLBO also implements strict acceptance-rejection approach, but it does not deploy segregation of roles.

In DGO, an agent interacts with one partner [18]. This partner is the best agent among population in every iteration. In DGO, each agent carries out only the guided search. Several steps are needed to determine the step size in the guided search. First, all agents are sorted based on their fitness score in the beginning of every iteration. Second, the agent's fitness score is normalized. Third, this normalization is used to build the roulette wheel. Fourth, the corresponding agent takes three trials stochastically to get the dart score. Fifth, this dart score is used to determine the step size in the guided search.

In EBOA, an agent interacts with the best agent or a randomly selected agent as a partner [14]. This selection is based on the normalized quality of the corresponding agent. Agent whose quality is better tends to choose the best agent while agent whose quality is worse tends to choose the randomly selected agent. The corresponding agent moves toward its partner if the partner is better than the corresponding agent. Otherwise, the corresponding agent avoids its partner. This mechanism is carried out in the first phase. In the second phase, the agent carries out local search in which the local search space width declines linearly due to the increase of iteration. The calculation in EBOA is simple. EBOA uses uniform random number and implements strict acceptance-rejection approach. It does not deploy segregation of roles.

In KMA, there is segregation of roles in the population. The population is split into three groups: high quality agents (big male), middle quality agents (female), and low-quality agents (small male) [7]. A high-quality agent interacts with all other high-quality agents in two ways. The high-quality agent moves toward the resultant of better high-quality agents and avoids the resultant of worse high-quality agents. There are two options for the middle-quality agents carried out stochastically. The first option is moving toward the highest quality agents. The second option is carrying out a random search. A low-quality agent moves toward the resultant of all high-quality agents. KMA deploys both uniform and normal random numbers. Meanwhile, it does not deploy acceptance-rejection approach. It consists of single phase only.

SKA becomes the improved version of KMA. As in KMA, SKA deploys segregation of roles. But SKA eliminates the sorting mechanism carried out in the beginning of every iteration [8]. The role for every agent is determined stochastically so that the role does not depend on the quality of the corresponding agent. Like KMA, SKA does not deploy acceptance-rejection approach. Different from KMA, SKA uses the uniform random number only and consists of single phase.

In SSO, there is segregation of roles. The population is divided into three groups. The first group consists of the highest quality agents [9]. The second group consists of three following best agents. The third group consists of the rest of the agents. SSO consists of two phases. In the first phase, every agent moves toward the best agent or carries out a random search. In the second phase, some agents also carry out a random search. SSO carries out a complicated calculation in determining the movement. Moreover, SSO carries out both uniform random and levy random.

The summary of strategy and mechanism implemented in these shortcoming metaheuristics are presented in Table 1. The seventh column indicates whether the metaheuristic implements the acceptance-rejection strategy. Meanwhile, the strategy and mechanism of the proposed metaheuristic is presented in the last row so that the novelty and contribution are clearly investigated.

Hence, there are many varieties of recent swarm intelligence-based metaheuristics. Some of them may be similar in certain parts but different in other parts. The proposing of a new metaheuristic can be conducted based on one or more existing metaheuristics with modifications in some parts.

As a resume, the existence of swarm intelligence-based metaheuristic in which the agent interacts with all other agents is rare to find. Among different kind of metaheuristics, an agent may interact with only one agent, two agents, or some agents. Some

Received: September 27, 2022. Revised: November 7, 2022.

Table 1. Strategy mapping of the shortcoming swarm intelligence-based metaheuristics

No	Metaheuristic	Partner	Number of Phases	Random	Segregation of Roles	A/R	Random Search within the Search Space
1	BOA [6]	the best solution in the current iteration or two randomly selected solutions	1	uniform	yes, stochastic	no	initialization
2	HLBO [21]	the best solution in the current iteration	2	uniform	no	yes	initialization, second phase
3	DGO [18]	the best solution in the current iteration	1	uniform	no	no	initialization
4	EBOA [14]	the best solution and a randomly selected solution	2	uniform	yes, stochastic	yes	initialization, second phase
5	KMA [7]	the highest quality solution, and several best solutions	1	uniform, normal	yes, deterministic	no	initialization, optional for the moderate quality agent
6	SKA [8]	the highest quality solution, and several best solutions	1	uniform	yes, stochastic	no	initialization, optional for the moderate quality agent
7	SSO [9]	the highest quality solution and several best solutions	2	uniform and levy	yes, deterministic	no	initialization, optional in first and second phases
8	RTGBO [17]	a randomly selected solution among several best solutions in the current iteration	1	uniform	no	yes	initialization
9	POA [19]	two randomly selected solutions	2	uniform	no	yes	initialization
10	GPA [12]	global best solution	2	uniform	no	yes	initialization, second phase
11	MHBA [10]	global best solution	2	uniform, sinusoid	no	no	initialization
12	DTBO [15]	a randomly selected solution	3	uniform	no	yes	initialization, third phase
13	NGO [11]	a randomly selected solution	2	uniform	no	yes	initialization, second phase
14	GSO [24]	global best solution, local best solution	1	uniform, sinusoid	no	no	initialization
15	MLBO [22]	best solution in the current iteration and a randomly selected solution	1	uniform	no	yes	initialization
16	this work	all other solutions	1	uniform	no	yes	initialization

metaheuristics carries out following the target. Meanwhile, some other metaheuristics also deploy the target avoidance. Therefore, the research opportunity to develop a new swarm intelligence-based metaheuristic in which every agent interacts with all other agents is still available.

3. Model

The main concept of TIA is simple. The system consists of certain number of solutions or agents. In the initialization phase, each solution is uniformly distributed within the solution space. The iteration phase is used to improve the solution quality. Each solution interacts with all other solutions in every iteration. There is a candidate generated for the corresponding solution in every interaction. If the partner is better than the corresponding solution, then the candidate is obtained based on the movement of the corresponding solution relative to the partner. On the other hand, if the partner is not better than the corresponding solution, then the candidate is obtained based on the movement of the partner relative to the corresponding solution. Then, the candidate with best quality becomes the final candidate for the corresponding solution. The final candidate is then compared with its corresponding solution. If the final candidate is better than the corresponding solution, then the final candidate replaces the current solution. Otherwise, the corresponding solution remains unchanged. This strategy represents the strict acceptance-rejection approach.

A variable called as the best solution becomes the collective intelligence used in TIA. This variable stores the solution with the highest quality. The variable is updated every time a solution is updated. A new solution becomes the best solution only if its value is better than the best solution.

There are several variables used in this research as described below. The formalization of TIA is presented in algorithm 1 and supported with Eq. (1) to Eq. (5).

b_l	lower boundary
b_u	upper boundary
r_1	real random number between 0 and 1
r_2	integer random number between 0 or 1
s	Solution
S	set of solutions
s_b	best solution
s_c	solution candidate
s_{cc}	chosen candidate
t	Iteration
t_{max}	maximum iteration

$$s = U(b_l, b_u) \tag{1}$$

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

$$s_{c,i,j} = \begin{cases} s_i + r_1.(s_j - r_2.s_i), f(s_j) < f(s_i) \\ s_j + r_1.(s_i - r_2.s_j), else \end{cases} \tag{3}$$

$$s_{cc,i}' = \begin{cases} s_{c,i,j}, f(s_{c,i,j}) < f(s_{cc,i}) \\ s_{cc,i}, else \end{cases} \tag{4}$$

$$s' = \begin{cases} s_{cc}, f(s_{cc}) < f(s) \\ s, else \end{cases} \tag{5}$$

Eq. (1) states that the initial solution is distributed uniformly within the solution space. Eq. (2) states that the corresponding solution will replace the current best solution only if the corresponding solution is better than the current best solution. Eq. (3) states that there are two options regarding the movement candidate. The first option is the movement closer to the other solution if the other solution is better than the corresponding solution. The second option is the movement to avoid the other solution if the other solution is not better than the corresponding solution. Eq. (4) states that the chosen candidate for the corresponding solution is the candidate with the highest quality. Eq. (5) states that the final candidate will replace the current solution if the final candidate is better than the current solution.

Algorithm 1: Total interaction algorithm

for all s **in** S
  generate initial s using Eq. (1)
  update s_{best} using Eq. (2)
end
for t = 1 **to** t_{max}
  for i = 1 **to** $n(S)$
    for j = 1 **to** $n(S)$
      if $i \neq j$ **then**
        generate $s_{c,i,j}$ using Eq. (3)
        update $s_{cc,i}$ using Eq. (4)
      end if
    end for
    update s_i using Eq. (5)
    update s_b using Eq. (2)
  End
End
return s_{best}

4. Simulation and result

In this study, TIA is tested to find the optimal solution of 23 functions. These functions are commonly used in a lot of works that propose new metaheuristic, such as in the first introduction of RTGBO [17]. These functions can be divided into three clusters. The first cluster consists of seven high dimension unimodal functions. The second cluster consists of six high dimension multimodal functions. The third cluster consists of ten fixed dimension multimodal functions. The detail description of the functions can be seen in [12] and [24]. In this research, the dimension for the high dimension functions is set to 20.

In this test, TIA is benchmarked with five other metaheuristics: PSO, MPA, GSO, GPA, and DTBO. These five metaheuristics are built based on swarm intelligence. PSO represents the early generation of swarm intelligence-based metaheuristic while GSO, GPA, and DTBO represent the latest versions of swarm intelligence-based metaheuristic. Meanwhile, MPA represents the metaheuristic which each agent consists of a pair of a predator and its prey [27]. In this test, the population size is 10 and the maximum

Table 2. Simulation results

F	Average Fitness Score					
	PSO	MPA	GSO	GPA	DTBO	TIA
1	7.092×10^{3}	7.929×10^{2}	3.681×10^{3}	4.260	1.384×10^{-8}	**4.032×10^{-22}**
2	1.686×10^{21}	1.478×10^{6}	2.538×10^{24}	**0.000**	5.515×10^{-100}	4.869×10^{-219}
3	1.504×10^{4}	2.267×10^{3}	7.739×10^{3}	6.909×10^{1}	9.366	**2.497×10^{-11}**
4	3.664×10^{1}	6.351	2.803×10^{1}	2.365	1.210×10^{-3}	**1.467×10^{-9}**
5	5.020×10^{6}	2.763×10^{3}	1.897×10^{6}	2.073×10^{2}	1.890×10^{1}	**1.875×10^{1}**
6	6.163×10^{3}	8.567×10^{2}	4.364×10^{3}	7.266	2.745	**1.832**
7	1.805	1.829×10^{-1}	7.400×10^{-1}	3.021×10^{-2}	8.578×10^{-3}	**4.112×10^{-3}**
8	-2.209×10^{3}	-2.336×10^{3}	-2.873×10^{3}	**-5.583×10^{3}**	-3.424×10^{3}	-1.826×10^{3}
9	1.608×10^{2}	1.081×10^{2}	8.164×10^{1}	1.826×10^{1}	3.175×10^{1}	**0.000**
10	1.551×10^{1}	8.106	1.615×10^{1}	2.342	3.498×10^{-5}	**5.306×10^{-12}**
11	6.268×10^{1}	8.197	4.392×10^{1}	8.701×10^{-1}	3.726×10^{-3}	**0.000**
12	1.515×10^{6}	1.944×10^{1}	2.356×10^{5}	1.427	4.310×10^{-1}	**2.601×10^{-1}**
13	1.181×10^{7}	3.248×10^{4}	2.677×10^{6}	**4.362×10^{-1}**	2.267	1.805
14	1.012×10^{1}	8.276	9.736	**9.980×10^{-1}**	2.082	5.455
15	3.759×10^{-2}	1.483×10^{-2}	1.198×10^{-2}	1.668×10^{-3}	6.372×10^{-4}	**3.974×10^{-4}**
16	-1.016	-1.009	**-1.032**	**-1.032**	**-1.032**	-1.030
17	1.751	1.256	**3.981×10^{-1}**	**3.981×10^{-1}**	**3.981×10^{-1}**	4.085×10^{-1}
18	1.932×10^{1}	9.401	**3.000**	**3.000**	4.227	4.967
19	-6.550×10^{-2}	**-3.574**	-4.097×10^{-2}	-4.954×10^{-2}	-4.954×10^{-2}	-4.954×10^{-2}
20	-1.971	-1.726	-3.013	**-3.300**	-3.282	-3.016
21	-3.361	-1.602	-5.831	-7.734	**-1.015×10^{1}**	-5.142
22	-2.732	-1.117	-6.527	-8.409	**-9.889**	-6.050
23	-2.865	-1.239	-6.127	-9.435	**-9.601**	-4.471

Table 3. Cluster based comparison

Cluster	Number of Functions where TIA is Better				
	PSO	MPA	GSO	GPA	DTBO
1	7	7	7	6	7
2	5	5	5	4	5
3	10	9	4	1	1
Total	22	21	16	11	13

iteration is 50. This setting represents the low population and low iteration system. In PSO, all weights are set 0.2. In MPA, the fishing aggregate devices is set 0.5. In GPA, the number of candidates is 10. The result is presented in Table 2. The best result in every function is presented in bold font. The cluster-based comparison between TIA and five other metaheuristics is presented in Table 3. Data in Table 3 indicates the number of functions which TIA is better than the corresponding metaheuristics.

Table 2 indicates the good performance of TIA. Overall, TIA can find the acceptable solution for all functions. Its performance is the best among metaheuristics tested in 11 functions. Moreover, it can find the global optimal solution of Rastrigin and Griewank. Unfortunately, TIA is entrapped in solving Hartman 3, as also happens with all benchmark metaheuristics except MPA.

Table 2 indicates that TIA is superior compared to other metaheuristics. TIA is very superior to PSO, MPA, and GSO. Meanwhile, TIA is competitive enough compared to GPA and DTBO. TIA is superior in solving functions in the second cluster. TIA is almost superior in solving functions in the first cluster. TIA is less superior compared to GPA in solving Schwefel 2.22 where GPA achieves the global optimal solution. In the third cluster, TIA is superior compared to PSO and MPA. On the other hand, TIA is inferior compared to GSO, GPA, and DTBO.

The second test is carried out to evaluate the performance of TIA regarding the various maximum iteration. In this test, the maximum iteration is set to be 75, 100, and 125 which are higher than the default value that is set in the first test. The result is presented in Table 4.

Table 4 indicates that there are two circumstances happens due to the increase of the maximum iteration. In three functions, i.e., Sphere, Schwefel 1.2, and Schwefel 2.21, the increase of maximum iteration improves the algorithm's performance significantly. These three functions are fixed dimension unimodal functions. In four functions, i.e., Schwefel 2.22, Rastrigin, Griewank, and Six Hump Camel, the increase of maximum iteration does not affect the algorithm's performance because the global optimal solution has been already found or almost found. Three of them are high dimension functions, and the

Received: September 27, 2022. Revised: November 7, 2022.

Table 4. Relation between maximum iteration and TIA's performance

F	Average Fitness Score		
	t_{max} = 75	t_{max} = 100	t_{max} = 125
1	6.087×10^{-35}	2.846×10^{-48}	2.846×10^{-61}
2	0.000	0.000	0.000
3	5.348×10^{-19}	4.076×10^{-26}	1.867×10^{-34}
4	4.991×10^{-15}	1.806×10^{-20}	1.125×10^{-25}
5	1.875×10^{1}	1.875×10^{1}	1.876×10^{1}
6	1.703	1.682	1.602
7	1.617×10^{-3}	8.186×10^{-4}	8.791×10^{-4}
8	-1.824×10^{3}	-1.886×10^{3}	-1.913×10^{3}
9	0.000	0.000	0.000
10	3.997×10^{-15}	3.997×10^{-15}	3.997×10^{-15}
11	0.000	0.000	0.000
12	2.442×10^{-1}	2.582×10^{-1}	2.563×10^{-1}
13	1.833	1.787	1.781
14	5.848	4.731	4.843
15	4.572×10^{-4}	4.000×10^{-4}	5.011×10^{-4}
16	-1.030	-1.031	-1.030
17	4.049×10^{-1}	6.978×10^{-1}	4.179×10^{-1}
18	6.601	6.290	6.723
19	-4.954×10^{-2}	-4.954×10^{-2}	-4.954×10^{-2}
20	-3.045	-2.893	-3.095
21	-6.103	-6.246	-7.455
22	-6.017	-5.239	-5.497
23	-5.604	-4.461	-4.972

Table 5. Relation between population size and TIA's performance

F	Average Fitness Score		
	$n(S)$ = 20	$n(S)$ = 30	$n(S)$ = 40
1	3.269×10^{-26}	1.729×10^{-28}	5.690×10^{-30}
2	0.000	0.000	0.000
3	4.302×10^{-14}	3.311×10^{-15}	1.586×10^{-16}
4	3.014×10^{-11}	3.476×10^{-12}	9.264×10^{-13}
5	1.868×10^{1}	1.863×10^{1}	1.861×10^{1}
6	1.120	8.333×10^{-1}	7.792×10^{-1}
7	9.150×10^{-4}	8.286×10^{-4}	5.452×10^{-4}
8	-1.904×10^{3}	-1.979×10^{3}	-2.130×10^{3}
9	0.000	0.000	0.000
10	5.567×10^{-14}	4.174×10^{-15}	3.997×10^{-15}
11	0.000	0.000	0.000
12	1.424×10^{-1}	9.788×10^{-2}	8.680×10^{-2}
13	1.218	1.021	8.903×10^{-1}
14	2.296	1.544	1.055
15	3.819×10^{-4}	3.898×10^{-4}	3.831×10^{-4}
16	-1.032	-1.032	-1.032
17	3.982×10^{-1}	3.981×10^{-1}	3.981×10^{-1}
18	3.005	3.000	3.000
19	-4.954×10^{-2}	-4.954×10^{-2}	-4.954×10^{-2}
20	-3.246	-3.241	-3.296
21	-8.058	-7.515	-8.400
22	-8.459	-9.381	-9.523
23	-7.320	-7.759	-7.304

remain is fixed dimension function. In other functions, the increase of maximum iteration does not affect the algorithm's performance although the final solution is not very close to the global optimal solution.

The third test is carried out to evaluate the algorithm's performance regarding the increase of population size. The population size is 20, 30, and 40. The result is presented in Table 5.

Table 5 indicates various response due to the increase of population size. The increase of population size improves the algorithm's performance significantly in three high dimension unimodal functions, namely Sphere, Schwefel 1.2, and Schwefel 2.21. There are six functions which are not affected by the increase of the population size, i.e., Schwefel 2.22, Rastrigin, Griewank, Six Hump Camel, Goldstein Price, and Branin. The increase of population size does not improve the six functions because the global optimal solution has been already found. In other functions, the increase of population size does not improve the algorithm's performance although the final solution is not very close to the global optimal solution.

The fourth test is carried out to evaluate the performance of TIA due to the increase of the dimension of the problem. In this test, the 13 multimodal functions represent the problems. The dimension is set as 30, 40, and 50. The population size is 10 and the maximum iteration is 50. The result is presented in Table 6.

Table 6 indicates that the performance of TIA is stable due to the increase of dimension. The average fitness score tends to stagnant or fluctuates in five functions (Schwefel 2.22, Quartic, Schwefel, Rastrigin, and Griewank). Meanwhile, the average fitness score increases with very low gradient in eight functions (Sphere, Schwefel 1.2, Schwefel 2.21, Rosenbrock, Step, Ackley, Penalized, and Penalized 2).

5. Discussion

In this section, the in-depth analysis regarding the algorithm's performance is discussed. The discussion consists of eight parts. The first part is the algorithm's performance regarding the characteristic of functions. The second part is the comparison between TIA and the benchmark metaheuristics regarding the result and the mechanics. The third part is the performance of TIA regarding the increase of maximum iteration. The fourth part is the performance of TIA regarding the increase of population size. The sixth part is the complexity analysis of TIA. The seventh part is the problem dimension analysis. The eighth part is the limitation of TIA and this work.

Table 6. Relation between dimension and TIA's performance

F	**Average Fitness Score**		
	***D* = 30**	***D* = 40**	***D* = 50**
1	2.114×10^{-19}	7.583×10^{-18}	7.392×10^{-17}
2	1.490×10^{-222}	0.000	0.000
3	5.882×10^{-9}	4.148×10^{-8}	6.132×10^{-7}
4	1.765×10^{-8}	9.041×10^{-8}	2.009×10^{-7}
5	2.878×10^{1}	3.880×10^{1}	4.877×10^{1}
6	3.426	5.232	7.466
7	3.895×10^{-3}	3.033×10^{-3}	4.008×10^{-3}
8	-2.017×10^{3}	-2.504×10^{3}	-2.783×10^{3}
9	0.000	0.000	0.000
10	8.788×10^{-11}	5.182×10^{-10}	1.516×10^{-9}
11	4.037×10^{-17}	0.000	4.657×10^{-12}
12	3.575×10^{-1}	4.964×10^{-1}	5.193×10^{-1}
13	2.377	2.652	2.788

In general, TIA is successful as a simple metaheuristic. Its performance is superior although it consists of only single mechanism. TIA is superior in solving high dimension unimodal functions and high dimension multimodal functions. Meanwhile, TIA is very competitive compared to PSO and MPA but less competitive compared to GSO, GPA, and DTBO in solving fixed dimension multimodal functions. This result indicates that TIA has made significant improvement compared to the early swarm intelligence-based metaheuristics. Meanwhile, the competition between TIA and the shortcoming metaheuristics is still though.

Table 2 indicates that the problem space width does not affect the performance of TIA. TIA can achieve the good result in solving functions with very narrow problem space, such as Quartic and Rastrigin. On the other hand, TIA is superior in solving problems with large problem space, such as Griewank, Sphere, Schwefel 2.22, Schwefel 1.2, Schwefel 2.21, and Step. Unfortunately, TIA is inferior in solving Schwefel.

The result indicates that interaction with all agents is better than interaction with only the best agent. This statement is based on the comparison between TIA and PSO. In PSO, the improvement of an agent relies on the interaction of the corresponding agent with the global best solution and its local best solution [26]. It means that the interaction with other agents in the population is not considered. The result also indicates that interaction with other agents, whether their quality is not the best is still important in finding better possibility to improve the quality of the current solution. This paradigm is important due to the significant gap between TIA and PSO in solving many functions. Meanwhile, TIA is like PSO in its simplicity.

The result also indicates that simple metaheuristic can outperform the complicated one as in MPA. MPA can be seen as a complicated metaheuristic as it adopts various strategy. MPA adopts strict separation between exploration and exploitation by dividing the iteration in rigid shifting from exploration to exploitation [27]. MPA also adopts complicated random mechanism, the Brownian motion and cuckoo movement, rather than simple uniform random [27]. MPA also deploys two phase strategy in every iteration. Unfortunately, the complicated strategy still fails to outperform TIA.

The superiority of TIA to MPA can also be seen as superiority of total interaction to the limited interaction. In MPA, the agent interacts only with its local best solution called the predator in the first phase [27]. It means there is not any interaction with other agents in the population. The limited interaction is conducted in the second phase where two randomly selected agents become the guidance for random movement. But this strategy is not mandatory because this strategy competes with random search based on the fishing aggregate devices during eddy formation [27].

Comparison between TIA and GSO proofs that simple strategy adopted by TIA can outperform complicated strategy adopted by GSO. Like PSO, GSO also uses the local best solution and global best solution as guidance for the movement of every agent [24]. Different from PSO, the sine and cosine functions are embedded in determining the movement [24]. Moreover, GSO deploys additional phase which the worst solution is replaced by randomly selected solution in every iteration [24]. Unfortunately, this strategy is still inferior to TIA, especially in solving high dimension functions. Meanwhile, multiple phase strategy as in GSO is important to tackle the fixed dimension multimodal functions.

Comparison between TIA and DTBO proofs that interaction with all agents is important to tackle the high dimension functions and multiple phase strategy is important to tackle the fixed dimension multimodal functions. As presented in Table 2, TIA is superior to DTBO in solving high dimension unimodal functions and high dimension multimodal functions. On the other hand, DTBO is superior to TIA in solving fixed dimension multimodal functions.

Same result can also be found by comparing TIA and GPA. The random search carried out in the second phase of GPA makes GPA powerful in solving the fixed dimension multimodal functions. On the other hand, limited interaction among agents in GPA makes TIA is superior to GPA in solving the high dimension unimodal functions and high

Received: September 27, 2022. Revised: November 7, 2022.

dimension multimodal functions. This comparison also proofs that interaction between the corresponding agents and all other agents as carried out in TIA is more important than generating multiple candidates along the way between the corresponding agent and the global best solution as in GPA [12].

The result also indicates that TIA is powerful enough in finding the acceptable solutions in the low iteration and low population size. The increase of maximum iteration improves the quality of the final solution for some problems. Meanwhile, the increase of population size also improves the quality of the final solution for some problems. On the other hand, the increase of maximum iteration or population size does not affect to the quality of the solution in many other problems. There are two factors regarding this situation. First, the final solution is the global optimal solution or very close to the global optimal solution. It makes the increase of maximum iteration or population size is not necessary anymore. Second, the additional strategy should be embedded to TIA, for example the neighbourhood search, to make significant improvement.

The complexity of TIA can be presented as $O(t_{max}.n(S)^2)$. This presentation indicates that the maximum iteration is linear to the complexity while the population size is quadratic to the complexity. The comparison between the result in Table 4 and Table 5 shows that increasing the maximum iteration is wiser than increasing the population size. There are two reasons as follows. First, the increase of maximum iteration creates more significant quality improvement than the increase of population size. Second, the increase of maximum iteration consumes less computational resource than the increase of population size. Although the relation of population size to the complexity is quadratic, TIA is still better than other metaheuristics where sorting mechanism is deployed in the beginning of every iteration, such as in DGO [18], RTGBO [17], GWO [25], and so on. This sorting mechanism is also quadratic to the complexity. Then, it is summed with the linear complexity regarding the movement carried out by every agent. On the other hand, TIA is more efficient rather than other metaheuristics that conduct multiple phases carried out by every agent in every iteration.

TIA is proven stable in the circumstance where the dimension of the problem increases as indicated in Table 6. In some functions, the average fitness score tends to fluctuate with low ripple while in other functions, the average fitness score increases with very low gradient. This circumstance means that TIA is very potential to solve many problems with high or very high dimension without facing significant performance drop.

There are several limitations regarding TIA and this research. The limitations can be used as the baseline for future research. First, TIA is still inferior in solving the fixed dimension multimodal problems compared with the shortcoming metaheuristics that carry out multiple phases in every iteration. The following challenge is whether this weakness can be overcome by adding the neighbourhood search in the second phase and what consequence follows this effort. Second, this research has presented the basic form of TIA to solve the numerical optimization problem. On the other hand, this basic form needs modification so that TIA can be implemented to solve the combinatorial optimization problem. Third, the cases used in the test are 23 functions. On the other hand, there are many other sets of functions that can be used, such as complex CEC 2019 [14], or IEEE CEC 2017 [21]. Moreover, TIA has not been tested to solve the practical optimization problem. Fourth, TIA also should be tested to solve problems with very high dimension. This test is important because there are many problems with very high dimension in the real world, such as optimizing the shelf arrangement in a department store that consists of hundreds of stock-keeping units, scheduling hundreds of orders in the manufacturer, or optimizing the financial portfolio that consists of hundreds of stocks.

6. Conclusion

This research has demonstrated that the interaction with all agents in a metaheuristic, as implemented in the proposed algorithm, is better than the interaction with only selected agents, as in many existing metaheuristics. In general, TIA outperforms the five benchmark metaheuristics in finding the optimal solution of 23 functions. TIA is better than PSO, MPA, GSO, GPA, and DTBO in 22, 21, 16, 11, and 13 functions. Its superiority especially comes from the high dimensional problems. On the other hand, TIA is not superior in solving fixed dimension multimodal problems, especially compared to GSO, GPA, and DTBO. TIA can find the global optimal solution of Rastrigin and Griewank in the low population and low iteration setting that means TIA is powerful in solving problems with narrow or large problem space. TIA also can find the global optimal solution of Schwefel 2.22 in the high iteration setting. Meanwhile, TIA can find the global optimal solution of Six Hump Camel, Goldstein Price, and Branin in the high population setting.

Many future studies can be conducted based on this study. There is space to modify TIA or to combine TIA with other metaheuristics, especially to improve its performance in solving the fixed dimension multimodal problems. Moreover, TIA should be tested to optimize many practical problems to evaluate its strength and weakness more comprehensively.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization: Kusuma; methodology: Kusuma, software: Kusuma, formal analysis: Kusuma and Novianty; writing-original paper draft: Kusuma; writing-review and editing: Novianty; supervision: Novianty; project administration: Kusuma; funding acquisition: Kusuma.

Acknowledgments

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

References

[1] A. Khan, H. Hizam, N. I. A. Wahab, and M. L. Othman, "Optimal Power Flow Using Hybrid Firefly and Particle Swarm Optimization Algorithm", *PLoS One*, Vol. 15, No. 8, ID: e0235668, pp. 1-21, 2020.

[2] I. Thammachantuek and M. Ketcham, "Path Planning for Autonomous Mobile Robots Using Multi-Objective Evolutionary Particle Swarm Optimization", *PLoS One*, Vol. 17, No. 8, ID: e0271924, pp. 1-48, 2022.

[3] R. A. Swief, N. M. Hassan, H. M. Hasanien, A. Y. Abdelaziz, and M. Z. Kamh, "Multi-Regional Optimal Power Flow Using Marine Predators Algorithm Considering Load and Generation Variability", *IEEE Access*, Vol. 9, pp. 74600–74613, 2021.

[4] S. Harifi, M. Khalilian, J. Mohammadzadeh, and S. Ebrahimnejad, "Emperor Penguins Colony: A New Metaheuristic Algorithm for Optimization", *Evolutionary Intelligence*, Vol. 12, No. 2, pp. 211–226, 2019.

[5] A. M. F. Fard, M. H. Keshteli, and R. T. Moghaddam, "Red Deer Algorithm (RDA): A New Nature-inspired Meta-heuristic", *Soft Computing*, Vol. 24, No. 19, pp. 14637–14665, 2020.

[6] S. Arora and S. Singh, "Butterfly Optimization Algorithm: A Novel Approach for Global Optimization", *Soft Computing*, Vol. 23, No. 3, pp. 715–734, 2019.

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

[8] P. D. Kusuma and M. Kallista, "Stochastic Komodo Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 4, pp. 156-166, 2022, doi: 10.22266/ijies2022.0831.15.

[9] M. Suman, V. P. Sakthivel, and P. D. Sathya, "Squirrel Search Optimizer: Nature Inspired Metaheuristic Strategy for Solving Disparate Economic Dispatch Problems", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 5, pp. 111–121, 2020, doi: 10.22266/ijies2020.1031.11.

[10] S. A. Yasear and H. M. A. Ghanimi, "A Modified Honey Badger Algorithm for Solving Optimal Power Flow Optimization Problem", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 4, pp. 142–155, 2022, doi: 10.22266/ijies2022.0831.14.

[11] 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.

[12] P. D. Kusuma and A. L. Prasasti, "Guided Pelican Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 6, pp. 179-190, 2022, doi: 10.22266/ijies2022.1231.18.

[13] X. Mi, Z. Liao, S. Li, and Q. Gu, "Adaptive Teaching–Learning-based Optimization with Experience Learning to Identify Photovoltaic Cell Parameters", *Energy Reports*, Vol. 7, pp. 4114–4125, 2021.

[14] P. Trojovský 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.

[15] 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.

[16] 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.

[17] S. A. Doumari, H. Givi, M. Dehghani, and O. P. Malik, "Ring Toss Game-Based Optimization Algorithm for Solving Various Optimization Problems", *International Journal of Intelligent*

Engineering and Systems, Vol. 14, No. 3, pp. 545–554, 2021, doi: 10.22266/ijies2021.0630.46.
[18] M. Dehghani, Z. Montazeri, H. Givi, J. M. Guerrero, and G. Dhiman, "Darts Game Optimizer: A New Optimization Technique Based on Darts Game", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 5, pp. 286–294, 2020, doi: 10.22266/ijies2020.1031.26.
[19] F. A. Zeidabadi and M. Dehghani, "POA: Puzzle Optimization Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 1, pp. 273–281, 2022, doi: 10.22266/ijies2022.0228.25.
[20] F. A. Zeidabadi, M. Dehghani, and O. P. Malik, "RSLBO: Random Selected Leader Based Optimizer", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 5, pp. 529–538, 2021, doi: 10.22266/ijies2021.1031.46.
[21] M. Dehghani and P. Trojovský, "Hybrid Leader Based Optimization: A New Stochastic Optimization Algorithm for Solving Optimization Applications", *Scientific Reports*, Vol. 12, No. 1, pp. 1–16, 2022.
[22] F. Zeidabadi, S. Doumari, M. Dehghani, and O. P. Malik, "MLBO: Mixed Leader Based Optimizer for Solving Optimization Problems", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 4, pp. 472–479, 2021, doi: 10.22266/ijies2021.0831.41.
[23] M. Dehghani, Z. Montazeri, A. Dehghani, R. A. R. Mendoza, H. Samet, J. M. Guerrero, and G. Dhiman, "MLO: Multi Leader Optimizer", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 6, pp. 364–373, 2020, do: 10.22266/ijies2020.1231.32.
[24] M. Noroozi, H. Mohammadi, E. Efatinasab, A. Lashgari, M. Eslami, and B. Khan, "Golden Search Optimization Algorithm", *IEEE Access*, Vol. 10, pp. 37515–37532, 2022.
[25] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer", *Advances in Engineering Software*, Vol. 69, pp. 46–61, 2014.
[26] D. Freitas, L. G. Lopes, and F. M. Dias, "Particle Swarm Optimisation: A Historical Review Up to the Current Developments", *Entropy*, Vol. 22, No. 3, ID: 362, pp. 1-36, 2020.
[27] 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.