Received: December 8, 2022. Revised: January 12, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Quad Tournament Optimizer: A Novel Metaheuristic Based on Tournament Among Four Strategies

Purba Daru Kusuma[1]* **Meta Kallista[1]**

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

Abstract: This paper introduced a novel metaheuristic that is developed based on the tournament mechanism, namely quad tournament optimizer (QTO). As its name suggests, QTO proposes a new approach of metaheuristic in which there are four searches conducted by each agent in every iteration. These searches are: (1) searching toward the global best solution, (2) searching toward the middle between the global best solution and a randomly selected solution, (3) searching relative to a randomly selected solution, and (4) neighbourhood search around the corresponding solution and the global best solution. A solution candidate is generated by each search. Then, a tournament is carried out to find the best candidate. This strategy is novel because most of metaheuristic deploys only single search or multiple searches where each search is conducted sequentially. QTO is challenged to find the optimal solution of 23 classic functions. In this challenge, QTO is benchmarked against five shortcoming metaheuristics: marine predator algorithm (MPA), slime mould algorithm (SMA), golden search optimizer (GSO), hybrid pelican Komodo algorithm (HPKA), and guided pelican algorithm (GPA). The result indicates that QTO outperforms all these benchmark metaheuristics. QTO is better than MPA, SMA, GSO, HPKA, and GPA in solving 22, 19, 21, 20, and 13 functions consecutively. The result also indicates that QTO has successfully found the global optimal solution for nine functions: Sphere, Schwefel 2.22, Schwefel 1.2, Schwefel 2.21, Rastrigin, Griewank, Six Hump Camel, Branin, and Goldstein-Price.

Keywords: Optimization, Swarm intelligence, Metaheuristic, Tournament.

1. Introduction

Metaheuristics have become a popular method for solving various optimization problems in recent decades, particularly in the engineering field. In the power system, metaheuristic has been used to optimize the solar cell and photovoltaic system [1], multiple energy storage system [2], capacitor bank [3], optimal power flow [4], and so on. In the biomedical works, metaheuristic has been used to optimize the detection of COVID-19 severity [5], lung cancer [6], brain tumor [7], anterior cruciate ligament deficiency [8], melanoma [9], and so on. In the manufacturing and logistics, metaheuristic has been used to optimize the parallel machine scheduling [10], vehicle distribution logistics [11], pickup and delivery problem [12], inventory management and assortment planning [13], logistic distribution center [14], and so on. In transportation sector, metaheuristic has been used to optimize the maintenance scheduling of highway networks [15], forecast the traffic flow [16], and so on.

There are a lot of new metaheuristics proposed in the recent years. This massive development of new metaheuristics also becomes the reason of the popularity of metaheuristic used to solve various optimization problems. Most of these shortcoming metaheuristics used metaphors for its name. Some metaheuristics adopt animal as metaphors, such as cheetah optimizer (CO) [17], marine predator algorithm (MPA) [18], butterfly optimization algorithm (BOA) [19], racoon optimization algorithm (ROA) [20], northern goshawk optimizer (NGO) [21], pelican optimization algorithm (POA) [22], Komodo mlipir algorithm (KMA) [23], guided pelican Komodo algorithm (HPKA) [24], guided pelican algorithm (GPA) [25], tunicate swarm algorithm (TSA) [26], and so on. Some metaheuristics used plant as metaphor, such as slime mold algorithm (SMA) [27], flower pollination algorithm (FPA) [28], and so on. Several metaheuristics use term leader that refers the reference used in their guided search, such as multi leader optimizer (MLO) [29], mixed leader-based optimizer (MLBO) [30], hybrid leader-based optimizer (HLBO) [31], and so on. Several metaheuristics do not use metaphor; they use their core mechanics for their names, such as average subtraction-based optimizer (ASBO) [32], golden search optimizer (GSO) [33], total interaction algorithm (TIA) [34], and so on.

There are several notes regarding the massive development of metaheuristic. First, many metaheuristics used metaphors, mainly terms from nature, as a pseudo novelty. In these metaphor-inspired metaheuristics, the mechanics of metaphors are usually presented as the novelty. In the beginning, the authors presented the mechanism of metaphor used in their proposed metaheuristic and the necessity of the related mechanics should be adopted in the metaheuristic. Ironically, the connection between the mechanism in the metaphor used in the corresponding metaheuristic and the formalization through algorithm and mathematical model is unclear. But through investigation, some metaheuristics make slight modifications to previous metaheuristics. Second, the opportunity of proposing new metaheuristic is still open regarding the no-free-lunch theory. There is no metaheuristic that performs well in solving all optimization problems as a stochastic method. Besides, the optimization problems are also growing and becoming more complex.

Based on the number of strategies implemented in the algorithm, the swarm-based metaheuristics can be classified as follows. In some metaheuristics, each agent uses single common strategy only. In some other metaheuristics, each agent implements multiple strategies carried out sequentially in every iteration. In some other metaheuristics, there are multiple strategies in the entire population. But each agent implements only one strategy. Meanwhile, a metaheuristic where each agent implements multiple strategies and selects the best one for the chosen strategy is rare to find.

This paper is aimed at proposing a new metaheuristic where each agent carries out several searches in every iteration. Rather than carried out sequentially as in many existing metaheuristics, these searches are then competed against one another to find the best candidate. Then, this candidate will be proposed as replacement for the current solution.

Based on the previous explanation, this paper contributes to several aspects, as follows.

1) This paper proposes a novel mechanic in metaheuristic studies where several candidates generated by several searches are competed in the tournament mechanism to find the best candidate for replacement.
2) This paper proposes a metaheuristic that is free from the use of metaphor.
3) This paper proposes a metaheuristic where sorting is not carried out at the beginning of every iteration to minimize the computational process.
4) The proposed metaheuristic consists only population size and maximum iteration as the adjusted parameters to avoid making the wrong adjustment that may lead to poor performance.
5) This paper carries out the sensitivity analysis regarding the adjusted parameters and the performance of the proposed metaheuristic.

The structure of the rest of this paper is as follows. Section 2 reviews the concept and strategy of some shortcoming metaheuristics to make clearer perspective and the contribution of this work. Section 3 presents a detailed description of the proposed metaheuristic, which includes the concept, algorithm, and mathematical presentation. Section 4 presents the evaluation of the proposed metaheuristic. There are two evaluations in this work. The first evaluation is the performance test and comparison of QTO with other metaheuristics in solving 23 classic functions. The second evaluation is test regarding the hyper parameters. Section 5 discusses the in-depth analysis regarding the test result, algorithm complexity, and the limitation of this work. Section 6 summarizes the concluding remark and the future work potential based on the advantages of QTO and its limitations in the current work.

2. Related works

Any swarm-based metaheuristic will perform two types of searches. The first search is the guided search. The guided search can be defined as searching process of the corresponding solution relative the reference. This reference can be the global best solution, some best solutions, a randomly selected solution, and so on. This search is mandatory in all swarm -based metaheuristics. The second search is the random search. In general, random search within the search space is carried out by all agents during the initialization step. Meanwhile, random search is implemented in some metaheuristics and is not implemented in some other ones. This random search can be random search within the entire search space or random search within the local search space. The second type is also named as neighborhood search.

Table 1. Comparison among metaheuristics regarding number of searches and phases

No	Metaheuristic	Reference of Guided Search	Random or Local Search	Number of Phases	Number of Strategies	Each Agent Carries Out All Strategies	Sorting in the Beginning of Every Iteration
1	ASBO [32]	the average between the best and worst solutions; the gap between the best and worst solutions; the gap between corresponding and the best solutions.	no	3	3	yes	yes
2	GSO [33]	the mixture of the global best solution and local best solution	no	1	1	yes	yes
3	TIA [34]	movement relative to all other solutions	no	1	1	yes	no
4	MLO [29]	a randomly selected solution among several best solutions	yes	2	2	yes	yes
5	MLBO [30]	the mixture between the best solution and a randomly selected solution	no	1	1	yes	no
6	HLBO [31]	the mixture among the best solution, a randomly selected solution, and corresponding solution	yes	2	2	yes	no
7	SMA [27]	the global best solution, two randomly selected solutions	yes	1	3	no	no
8	POA [22]	search space	yes	2	2	yes	no
9	KMA [23]	high quality solutions; the highest quality solution	yes	1	4	no	no
10	GPA [24]	global best solution	yes	2	2	yes	yes
11	HPKA [25]	global best solution; a randomly selected solution	yes	1	5	no	no
12	TSA [26]	best solution	no	1	1	yes	no
13	CO [17]	global best solution; mixture between global best solution and neighborhood solution	yes	2	4	no	no
14	NGO [21]	a randomly selected solution	yes	2	2	yes	no
15	MPA [18]	local best solution and two randomly selected solutions	yes	2	5	yes	no
16	this work	global best solution; a randomly selected solution	yes	1	4	yes	no

Some swarm intelligence-based metaheuristics implement single strategy while the others implement multiple strategies. In the single strategy metaheuristic, there is only one guided search implemented for each agent and there is not any random search. In the multiple strategy metaheuristics, these strategies can be carried out in single phase or multiple phases. When these multiple strategies are carried out in one phase, these strategies are distributed among agents, whether the distribution is deterministic or stochastic. This mechanism can be called as segregation of roles. On the other hand, in the multiple-phase multiple-strategy metaheuristics, these strategies are carried out by each agent during the iteration. These strategies are carried out sequentially. The number of phases usually represents the number of strategies. The detail mapping regarding this strategy in some shortcoming metaheuristics is presented in Table 1. The mechanism of the proposed metaheuristic in this work is written in the last row to make clear perspective regarding the novelty and position of the proposed metaheuristic.

Table 1 indicates that metaheuristic that implements multiple strategies in single phase and each agent implements all strategies during the iteration is hard to find. Meanwhile, the multiple strategy approach is very important to improve the performance of metaheuristic. Every strategy always has advantages and disadvantages. It means that multiple strategy approach can cover the disadvantage of one strategy with the others. Moreover, it is also important that all agents in the population carry out all installed strategies. Based on this situation, there is still room for developing a new metaheuristic that implements multiple strategies in a single phase and has each agent carry out all strategies.

3. Model

The core concept of QTO is creating the tournament-based competition between four common strategies in many existing metaheuristics. Three of the strategies are guided searches while one strategy is random search. The first strategy is moving toward or surpass the global best solution. The second strategy is moving toward the middle solution between the global best solution and a randomly selected solution. The third strategy is the movement relative to a randomly selected solution. The fourth strategy is combining a solution around the corresponding solution and a solution around the global best solution. The reasoning between these four strategies is presented in the next paragraphs.

The movement toward or surpassing the global best has two objectives. The first objective is tracing the better solution along the way between the corresponding solution and the global best solution. The second objective is tracing the better solution in some areas after the global best solution. If the corresponding solution is not the global best solution, then it becomes more probable to move toward the global best solution to improve the current corresponding solution rather than searching a solution somewhere else. Meanwhile, the second objective is constructed based on the assumption that the global best solution needs to improve. It is more probable to improve the global best solution by avoiding the solution worse than the global best solution.

The movement toward the middle between the global best solution and a randomly selected solution is designed to diversify the guided search. This movement will trace the possibility of the global optimal solution between the global best solution and a randomly selected solution.

The movement relative to a randomly selected solution has two objectives. The first objective is to improve the current solution somewhere besides the global best solution if this selected area is better than the corresponding solution. The second objective is to avoid the corresponding solution moves to the worse solution. In this context, a randomly selected solution is chosen. Then, the corresponding solution moves toward this randomly selected solution if this reference is better than the corresponding solution. Otherwise, the corresponding solution moves away from this reference.

The random search in the fourth strategy is a compromised neighborhood search between the corresponding solution and the global best solution. This random search has several characteristics. First, it is a combination between neighborhood searches around the corresponding solution and neighborhood searches around the global best solution. Second, the local search space declines due to the increase of the iteration. Third, in the early iteration, the corresponding solution has dominant proportion. Then, its dominance decreases as the iteration goes. On the other side, the proportion of the global best solution increases as the iteration goes. At the end of the iteration, the global best solution has a dominant proportion.

These four strategies are then compared based on their quality. The strategy whose quality is the best becomes the selected candidate. This selected candidate is then compared with the corresponding solution. The corresponding solution updates its solution by replacing its current solution with the selected candidate only if this candidate is better than the corresponding solution. Then, the global best solution is updated too.

The formalization of this concept is presented in algorithm 1 and Eq. (1) to Eq. (11). Algorithm 1 represents the sequence of process in the proposed algorithm while Eq. (1) to Eq. (11) becomes represents the detail mechanics in every process. Below are the annotations used in this work.

b_l	lower boundary
b_u	upper boundary
c_1	first candidate
c_2	second candidate
c_3	third candidate
c_4	fourth candidate
C	set of candidates
c_s	selected candidate
f	fitness function
r_1	real random number between -1 and 1
r_2	real random number between 0 and 1
r_3	integer random number between 1 and 2
t	iteration
t_m	maximum iteration
x	solution
x_s	selected solution
x_b	global best solution
X	set of solutions

algorithm 1: quad tournament optimizer

Begin
  for all x **in** X **do**
    set initial x using Eq. (1)
    update x_b using Eq. (2)
  end for
  for t=1 **to** t_{max} **do**
    for all x **in** X **do**
      find x_s using Eq. (3)
      generate c_1 using Eq. (4)
      generate c_2 using Eq. (5)
      generate c_3 using Eq. (6)
      generate c_4 using Eq. (7) to Eq. (9)
      find c_s using Eq. (10)
      update x using Eq. (11)
      update x_b using Eq. (2)
  end for
End
return x_b

$$x = U(b_l, b_u) \quad (1)$$

$$x_b' = \begin{cases} x, f(x) < f(x_b) \\ x_b, else \end{cases} \quad (2)$$

$$x_s = U(X) \quad (3)$$

$$c_1 = x + r_2(x_b - r_3 x) \quad (4)$$

$$c_2 = x + r_2\left(\frac{x_b + x_s}{2} - r_3 x\right) \quad (5)$$

$$c_3 = \begin{cases} x + r_2(x_s - r_3 x), f(x_s) < f(x) \\ x + r_2(x - r_3 x_s), else \end{cases} \quad (6)$$

$$w_1 = \left(1 - \frac{t}{t_m}\right)^2 r_1(b_u - b_l) \quad (7)$$

$$w_2 = \left(\frac{t}{t_m}\right)\left(1 - \frac{t}{t_m}\right) r_1(b_u - b_l) \quad (8)$$

$$c_4 = w_1 x + w_2 x_b \quad (9)$$

$$c_s = c \in C, min\big(f(c)\big) \quad (10)$$

$$x' = \begin{cases} c_s, f(c_s) < f(x) \\ x, else \end{cases} \quad (11)$$

Below is the explanation of Eq. (1) to Eq. (11). Eq. (1) represents the random search within the solution space to generate the initial solution. Eq. (2) states that the corresponding solution replaces the current global best solution if this corresponding solution is better than the current global best solution. Eq. (3) states that a reference is randomly selected among the population. Eq. (4) is used to determine the first candidate. Eq. (4) states that the first guided search is the movement toward or surpassing the global best solution. Eq. (5) is used to determine the second candidate. Eq. (5) states that the second guided search is the movement toward a solution in the middle between the randomly selected solution and the global best solution. Eq. (6) is used to determine the third candidate. Eq. (6) states the third guided search as a movement relative to the randomly selected solution. The corresponding solution moves toward this reference if this reference is better than the corresponding solution. Otherwise, the corresponding solution moves away from the reference. Eq. (7) to Eq. (9) are used to determine the fourth candidate. Eq. (7) and Eq. (8) are used to determine the weight of the random search where Eq. (7) is used for the corresponding solution while Eq. (8) is used for the global best solution. Eq. (9) states that the fourth candidate is the combination between the neighborhood search of the corresponding solution and the neighborhood search of the global best solution. Eq. (10) states that the selected candidate is the candidate whose quality is the highest. Eq. (11) states that the selected candidate replaces the current solution if this candidate is better than the current solution.

4. Simulation and result

This section presents the ability of QTO in solving the optimization problem. The set of 23 classic functions is chosen as the problem. These functions are popular and widely used in many studies proposing new metaheuristics. One important reason is these functions cover various types of optimization problems. These functions are clustered into three groups. There are seven high dimensional unimodal functions in the first group (Sphere, Schwefel 2.22, Schwefel 1.2, Schwefel 2.21, Rosenbrock, Step, and Quartic). There are six high dimensional multimodal functions in the second group (Schwefel, Rastrigin, Ackley, Griewank, Penalized, and Penalized 2). There are ten fixed dimension multimodal functions in the third group (Shekel Foxholes, Kowalik, Six Hump Camel, Branin, Goldstein-Price, Hartman 3, Hartman 6, Shekel 5, Shekel 7, and Shekel 10). These functions also cover problems with narrow, moderate, and large problem spaces.

There are three tests carried out in this work. The first test is related to the performance evaluation of QTO and the comparison with other metaheuristics. This first test is performed to evaluate the performance of the proposed QTO and compare its performance with other existing metaheuristics. The

Table 2. Simulation result

Function	Average Fitness Score					
	MPA [18]	**SMA [27]**	**GSO [33]**	**HPKA [24]**	**GPA [25]**	**QTO**
1	$1.766x10^{2}$	$4.017x10^{3}$	$3.937x10^{4}$	$5.138x10^{4}$	$2.293x10^{2}$	**0.000**
2	$4.362x10^{-66}$	**0.000**	$3.968x10^{57}$	$3.317x10^{46}$	$4.202x10^{30}$	**0.000**
3	$2.774x10^{3}$	$4.115x10^{4}$	$1.099x10^{5}$	$1.365x10^{5}$	$7.534x10^{3}$	**0.000**
4	6.916	$3.186x10^{1}$	$4.885x10^{1}$	$7.989x10^{1}$	$2.459x10^{1}$	**0.000**
5	$1.092x10^{4}$	$8.482x10^{7}$	$9.016x10^{7}$	$1.502x10^{8}$	$1.076x10^{4}$	$\mathbf{3.889x10^{1}}$
6	$1.213x10^{2}$	$5.455x10^{3}$	$3.643x10^{4}$	$4.743x10^{4}$	$2.414x10^{2}$	**7.265**
7	$1.109x10^{-1}$	$2.773x10^{2}$	$5.789x10^{1}$	$1.120x10^{2}$	$3.778x10^{-1}$	$\mathbf{1.956x10^{-3}}$
8	$-2.894x10^{3}$	$-6.588x10^{3}$	$-4.080x10^{3}$	$-4.758x10^{3}$	$\mathbf{-8.410x10^{3}}$	$-3.995x10^{3}$
9	$4.524x10^{1}$	$3.941x10^{1}$	$3.960x10^{2}$	$4.562x10^{2}$	$1.833x10^{2}$	**0.000**
10	2.984	8.004	$1.982x10^{1}$	$1.975x10^{1}$	5.225	$\mathbf{4.441x10^{-16}}$
11	2.795	$4.452x10^{1}$	$4.057x10^{2}$	$6.234x10^{2}$	3.159	**0.000**
12	3.024	$2.773x10^{6}$	$1.185x10^{8}$	$2.212x10^{8}$	$2.184x10^{1}$	$\mathbf{8.346x10^{-1}}$
13	$8.665x10^{2}$	$1.639x10^{7}$	$3.152x10^{8}$	$6.879x10^{8}$	$1.527x10^{2}$	**2.949**
14	$1.174x10^{1}$	4.922	9.546	$1.331x10^{1}$	**1.494**	6.662
15	$2.066x10^{-2}$	$1.252x10^{-1}$	$3.028x10^{-2}$	$1.033x10^{-2}$	$3.714x10^{-3}$	$\mathbf{2.106x10^{-3}}$
16	$-9.417x10^{-1}$	$-3.940x10^{-2}$	$-9.621x10^{-1}$	$-9.650x10^{-1}$	**-1.031**	**-1.031**
17	4.066	$6.541x10^{-1}$	$4.019x10^{-1}$	2.033	$\mathbf{3.981x10^{-1}}$	$4.242x10^{-1}$
18	$2.143x10^{1}$	$6.357x10^{1}$	$1.757x10^{1}$	$2.873x10^{1}$	**3.000**	3.021
19	**-3.253**	$-4.954x10^{-2}$	$-2.890x10^{-2}$	$-4.802x10^{-2}$	$-4.954x10^{-2}$	$-4.954x10^{-2}$
20	-1.431	$5.314x10^{-1}$	-2.566	-2.712	**-3.281**	-2.588
21	$-8.404x10^{-1}$	-3.898	-3.207	-2.333	**-5.594**	-4.497
22	$-8.848x10^{-1}$	-3.609	-3.510	-4.207	**-7.541**	-3.827
23	$-9.701x10^{-1}$	-2.955	-3.633	-3.525	**-9.388**	-4.929

second and third tests are tests related to hyperparameter analysis. The second test is conducted based on several values of maximum iteration. This test is performed to evaluate the speed of the proposed QTO in reaching the convergence or the acceptable solution. The third test is conducted based on several values of population size. This test is performed to evaluate the improvement of the proposed QTO during the increase of population size.

In this work, QTO is benchmarked with five other shortcoming metaheuristics: MPA [18], SMA [27], GSO [33], HPKA [24], and GPA [25]. The reason of choosing these five metaheuristics is as follow. All these metaheuristics are swarm intelligence-based metaheuristics. MPA and SMA are older than GSO, HPKA, and GPA, but more popular and widely used. On the other hand, GSO, HPKA, and GPA are newer metaheuristics introduced in 2022. Not all shortcoming metaheuristics in Table 1 are used as compared technique in this simulation because it is impossible to compare a new metaheuristic with too many compared techniques. Meanwhile, these five metaheuristics are chosen because of their distinct strategy. MPA is chosen because of its approach in changing strategy for the guided search as the iteration increases [18]. SMA is chosen because it represents metaheuristic that implements segregation of roles [27]. HPKA is chosen because it represents a new metaheuristic that hybridizes two existing metaheuristics [24]. GSO is chosen because it represents a new version of metaheuristic that combines the global best solution and local best solution for its reference in the guided search [33], which is firstly introduced in PSO. GPA represents a metaheuristic that uses global best solution as the sole reference during the guided search and generates multiple candidates in every guided search [25].

This test is carried out based on certain circumstances. The population size is 5 that represents low population. The maximum iteration is 50 that also represents low iteration. On the other hand, the dimension of the high dimension functions (F1 to F13) is 40 that represents problems with high dimension. In MPA, the fishing aggregate devices is 0.1. In HPKA, the threshold is set at 0.2, 0.4, and 0.6. In GPA, the number of candidates is 5. The result is shown in Table 2. The best result is presented in bold font. Moreover, the information representing the number of functions where QTO is better than the corresponding benchmark metaheuristic in every group is presented in Table 3.

Table 2 indicates that QTO performs well based on two perspectives. The first perspective is that in general, QTO can find the acceptable solution for the 23 functions. Second, QTO can find the best result in solving 14 functions (Sphere, Schwefel 2.22, Schwefel 1.2, Schwefel 2.21, Rosenbrock, Step,

Table 3. Cluster based comparison

Cluster	Number of Functions where QTO is Better				
	MPA [18]	SMA [27]	GSO [33]	HPKA [24]	GPA [25]
1	7	6	7	7	7
2	6	5	5	5	5
3	9	8	9	8	1
Total	22	19	21	20	13

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

F	Average Fitness Score		
	t_{max} = 100	t_{max} = 150	t_{max} = 200
1	0.000	0.000	0.000
2	0.000	0.000	0.000
3	0.000	0.000	0.000
4	0.000	0.000	0.000
5	$3.888\text{x}10^{1}$	$3.883\text{x}10^{1}$	$3.887\text{x}10^{1}$
6	7.256	7.304	7.176
7	$7.849\text{x}10^{-4}$	$1.680\text{x}10^{-4}$	$3.618\text{x}10^{-4}$
8	$-4.092\text{x}10^{3}$	$-4.427\text{x}10^{3}$	$-4.719\text{x}10^{3}$
9	0.000	0.000	0.000
10	$4.441\text{x}10^{-16}$	$4.441\text{x}10^{-16}$	$4.441\text{x}10^{-16}$
11	0.000	0.000	0.000
12	$8.312\text{x}10^{-1}$	$7.653\text{x}10^{-1}$	$8.249\text{x}10^{-1}$
13	2.884	2.873	2.857
14	7.720	5.963	5.265
15	$6.972\text{x}10^{-4}$	$8.350\text{x}10^{-4}$	$6.239\text{x}10^{-4}$
16	-1.031	-1.031	-1.031
17	$4.269\text{x}10^{-1}$	$4.417\text{x}10^{-1}$	$4.268\text{x}10^{-1}$
18	3.003	3.000	3.002
19	$-4.954\text{x}10^{-2}$	$-4.954\text{x}10^{-2}$	$-4.954\text{x}10^{-2}$
20	-2.545	-2.699	-2.689
21	-4.669	-4.296	-4.240
22	-4.664	-4.924	-4.278
23	-4.011	-4.182	-5.183

Quartic, Rastrigin, Ackley, Griewank, Penalized, Penalized 2, Kowalik, and Six Hump Camel). The second perspective is QTO can find the global optimal solution in solving six functions (Sphere, Schwefel 2.22, Schwefel 1.2, Schwefel 2.21, Rastrigin, and Ackley) and the near global optimal solution in solving two functions (Six Hump Camel, Branin, and Goldstein-Price). It should be noted that SMA is also successful in finding the global optimal solution of Schwefel 2.22 while GPA is also successful in finding the same quality result as QTO of Six Hump Camel.

Table 3 indicates that in general, QTO outperforms the five benchmark metaheuristics. QTO is better than MPA, SMA, GSO, HPKA, and GPA in solving 22, 19, 21, 20, and 13 functions consecutively. MPA becomes the easiest metaheuristic to defeat. Meanwhile, GPA emerges as the most difficult metaheuristic to outperform.

The second test is conducted to evaluate the relationship between the increase of maximum iteration and the performance of QTO. In this test, the maximum iteration is set at 100, 150, and 200. The result is presented in Table 4.

Table 4 indicates there are two behaviours regarding the increase of maximum iteration. The first behaviour is that the increase of maximum iteration makes the average fitness score decrease. This behaviour can be seen in Schwefel and Shekel Foxholes. Meanwhile, this improvement is not significant. The second behaviour is that the increase of maximum iteration does not affect the average fitness score. There are two reasons regarding this behaviour. The first reason is that the global optimal solution or near global optimal solution has been found. This reason can be seen in Sphere, Schwefel 2.22, Schwefel 1.2, Schwefel 2.21, Rastrigin, Six Hump Camel, and Goldstein-Price. The second reason is that the final solution is not near the global optimal solution but fails to improve. This reason can be seen in Rosenbrock, Step, Quartic, Ackley, Penalized, Penalized 2, Kowalik, Branin, Hartman 3, Hartman 6, Shekel 5, Shekel 7, and Shekel 10.

The third test is carried out to evaluate the relation between population size and the average fitness score. In this test, the population size is set at 10, 20, and 30. The result is presented in Table 5.

Table 5 indicates there are two responses regarding the increase of population size. The first response is the average fitness score decreases due to the increase of population size. This response can be seen in Step, Quartic, Schwefel, Penalized, Penalized 2, Shekel Foxholes, Kowalik, Hartman 6, Shekel 5, Shekel 7, and Shekel 10. The second response is that the average fitness core is not affected by the increase of population size. In some functions, this response is caused by the finding of the global optimal solution as in Sphere, Schwefel 2.22, Schwefel 1.2, Schwefel 2.21, Rastrigin, Griewank, Six Hump Camel, Branin, and Goldstein-Price. On the other hand, this response happens although the global optimal solution has not yet been found, such as in Rosenbrock, Ackley, and Hartman 3.

5. Discussion

The test result indicates the outstanding performance of QTO. QTO is successful in finding the global optimal solution of six functions in the low population size and low maximum iteration circumstance. Four functions are high dimensional unimodal functions while the two others are high dimensional multimodal functions. Moreover, there

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

F	**Average Fitness Score**		
	***n*(*X*) = 10**	***n*(*X*) = 20**	***n*(*X*) = 30**
1	0.000	0.000	0.000
2	0.000	0.000	0.000
3	0.000	0.000	0.000
4	0.000	0.000	0.000
5	$3.886 x 10^{1}$	$3.880 x 10^{1}$	$3.876 x 10^{1}$
6	6.435	5.831	5.327
7	$8.978 x 10^{-4}$	$1.920 x 10^{-4}$	$1.358 x 10^{-4}$
8	$-4.418 x 10^{3}$	$-4.734 x 10^{3}$	$-4.748 x 10^{3}$
9	0.000	0.000	0.000
10	$4.441 x 10^{-16}$	$4.441 x 10^{-16}$	$4.441 x 10^{-16}$
11	0.000	0.000	0.000
12	$6.288 x 10^{-1}$	$4.833 x 10^{-1}$	$4.590 x 10^{-1}$
13	2.879	2.614	2.532
14	4.108	2.926	1.716
15	$6.511 x 10^{-4}$	$4.866 x 10^{-4}$	$4.294 x 10^{-4}$
16	-1.032	-1.032	-1.032
17	$4.057 x 10^{-1}$	$3.981 x 10^{-1}$	$3.981 x 10^{-1}$
18	3.001	3.000	3.000
19	$-4.954 x 10^{-2}$	$-4.954 x 10^{-2}$	$-4.954 x 10^{-2}$
20	-2.968	-3.084	-3.185
21	-4.969	-6.070	-7.045
22	-5.289	-7.653	-8.027
23	-4.855	-7.768	-6.575

are more than three functions whose global optimal solution can be found in the high population size and low maximum iteration. These three additional functions are fixed dimension multimodal functions.

This outstanding performance is also strengthened by the fact that QTO is superior to other benchmark metaheuristics in solving 14 functions. These functions are seven high dimension unimodal functions, five high dimension multimodal functions, and two fixed dimension multimodal functions. It indicates that QTO is very superior at solving the high dimensional problems.

Table 3 indicates that GPA becomes the most difficult metaheuristic to beat. Meanwhile, the other four metaheuristics are easy to beat. The main difference between GPA and these four metaheuristics is that multiple candidates are generated in every phase. On the other hand, there is only one candidate generated in every phase of the four metaheuristics. This circumstance triggers new question whether generating multiple candidates can improve the performance of the metaheuristic although the consequence is increasing the complexity linear to the number of candidates.

Table 5 indicates that the weakness in solving the fixed dimension multimodal functions can be tackled by increasing the population size. The high population size can improve many functions in the third group. Moreover, the global optimal solution of three functions in the third group is achieved in the high population size circumstance. The reason is that high population size improves the diversity of solutions.

The complexity of QTO can be presented as $O(4n(X).t_m)$. This presentation indicates that the computational resource of QTO increases linearly due to the increase of one parameter, whether the population size or maximum iteration. Besides, number 4 represents the four methods included in the tournament conducted by every solution in every iteration. The complexity of QTO is far less than many other metaheuristics that deploy sorting processes in every iteration, such as in DTBO, DGO, GSO, GWO, and so on. The sorting process needs quadratic computational resources regarding the population size.

There are several limitations in this current study. First, QTO implements only four methods to be placed in the tournament: moving toward the global best solution, moving toward the middle point between the global best solution and a randomly selected solution, moving relative to a randomly selected solution, and the weighted neighbourhood search between the corresponding solution and the global best solution. Meanwhile, QTO can be enriched by including more methods into the tournament. The use of local best solution can be the alternative, whether this local best solution is combined with the global best solution such as in PSO, BOA, or SSO, or it becomes the exclusive reference as in MPA. The elimination of a solution also becomes the other alternative. On the other hand, adopting sorting mechanism at the beginning of iteration is not recommended because it consumes additional computation in the quadratic manner based on the population size. Second, tests carried out in this work are limited to solving the theoretical optimization problems and evaluating the hyperparameters. QTO has not been tested to solve the practical optimization problem although the result in Table 2 shows that QTO can tackle optimization problems under various circumstances. Third, there are hundreds of metaheuristics in the present days. Meanwhile, QTO is benchmarked with only five metaheuristics.

6. Conclusion

This study has demonstrated the outstanding performance of the proposed metaheuristic, namely quad tournament optimizer (QTO). This outstanding performance can be seen in two perspectives. The first perspective is that QTO has successfully found the global optimal solution of nine functions: Sphere, Schwefel 2.22, Schwefel 1.2, Schwefel 2.21, Rastrigin, Griewank, Six Hump Camel, Branin, and Goldstein-Price. The global optimal solution of six functions has been found in the low population size and low maximum iteration circumstances. Meanwhile, the global optimal solution of the other three functions has been found in the high population size and low maximum iteration circumstance. The second perspective is that QTO outperforms all five benchmark metaheuristics in solving the 23 functions. QTO is better than MPA, SMA, GSO, HPKA, and GPA in solving 22, 19, 21, 20, and 13 functions consecutively. This outstanding performance proves that the tournament-based strategy adopted in QTO is better than implementing only one or several methods for every agent in every iteration.

This work can be improved in two ways in the future based on the limitations in the current study. First, QTO can be improved by including more methods in the tournament rather than only four methods in the current work. Second, QTO can be adopted to solve practical optimization problems, from the common problems such as welded beam design or optimal electric flow to combinatorial problems such as scheduling, assignment, and so on.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

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

Acknowledgments

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

References

[1] M. A. E. Dabah, R. A. E. Sehiemy, M. A. Ebrahim, Z. Alaas, and M. M. Ramadan, “Identification Study of Solar Cell/Module Using Recent Optimization Techniques”, *International Journal of Electrical and Computer Engineering*, Vol. 12, No. 2, pp. 1189-1198, 2022.

[2] L. Ooha, R. K. Radha, and N. C. Kotainah, “Self-adaptive Dragonfly Algorithm for Optimal Allocation of Multiple Energy Storage Systems with Var Support of Islanded Microgrid Operation”, *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 6, pp. 142-151, 2022, doi: 10.22266/ijies2022.1231.15.

[3] R. Puppala and C. Sekhar, “Optimal Allocation of Capacitor Banks and DSTATCOMs in Radial Distribution System Considering Electric Vehicle Load Growth”, *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 6, pp. 45-53, 2022, doi: 10.22266/ijies2022.1231.05.

[4] 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.

[5] N. A. M. Aseri, M. A. Ismail, A. S. Fakharudin, A. O. Ibrahim, S. Kasim, N. H. Zakaria, and T. Sutikno, “Comparison of Meta-heuristic Algorithms for Fuzzy Modelling of COVID-19 Illness’ Severity Classification”, *IAES International Journal of Artificial Intelligence*, Vol. 11, No. 1, pp. 50-64, 2022.

[6] M. Pradhan, A. Bhuiyan, S. Mishra, T. Thieu, and I. L. Coman, “Histopathological Lung Cancer Detection Using Enhanced Grasshopper Optimization Algorithm with Random Forest”, *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 6, pp. 11-20, 2022, doi: 10.22266/ijies2022.1231.02.

[7] P. Srinivasalu and A. Palaniappan, “Brain Tumor Detection by Modified Particle Swarm Optimization Algorithm and Multi-Support Vector Machine Classifier”, *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 6, pp. 91-100, 2022, doi: 10.22266/ijies2022.1231.10.

[8] G. Wang, X. Zeng, G. Lai, G. Zhong, K. Ma, and Y. Zhang, “Efficient Subject-Independent Detection of Anterior Cruciate Ligament Deficiency Based on Marine Predator Algorithm and Support Vector Machine”, *IEEE Journal of Biomedical and Health Informatics*, Vol. 26, No. 10, pp. 4936-4947, 2022.

[9] A. Damarla and D. Sumathi, “An Approach for Optimization of Features using Gorilla Troop Optimizer for Classification of Melanoma”, *International Journal of Advanced Computer Science and Applications*, Vol. 13, No. 10, pp. 275-286, 2022.

[10] Y. Chen, Z. Guan, C. Wang, F. D. Chou, and L. Yue, “Bi-objective Optimization of Identical Parallel Machine Scheduling with Flexible Maintenance and Job Release Times”,

International Journal of Industrial Engineering Computations, Vol. 13, No. 4, pp. 457-472, 2022.

[11] W. Fang, Z. Guan, L. Yue, Z. Zhang, H. Wang, and L. Meng, "Heterogeneous-vehicle Distribution Logistics Planning for Assembly Line Station Materials with Multiple Time Windows and Multiple Visits", *International Journal of Industrial Engineering Computations*, Vol. 13, No. 4, pp. 473-490, 2022.

[12] A. Berahhou, Y. Benadada, and K. Bouanane, "Memetic Algorithm for the Dynamic Vehicle Routing Problem with Simultaneous Delivery and Pickup", *International Journal of Industrial Engineering Computations*, Vol. 13, No. 4, pp. 587-600, 2022.

[13] S. J. Sajadi and A. Ahmadi, "An Integrated Optimization Model and Metaheuristics for Assortment Planning, Shelf Space Allocation, and Inventory Management of Perishable Products: A Real Application", *PLoS One*, Vol. 17, No. 3, ID. e0264186, pp. 1-30, 2022.

[14] T. X. Lin, Z. H. Wu, and W. T. Pan, "Optimal Location of Logistics Distribution Centres with Swarm Intelligent Clustering Algorithms", *PLoS One*, Vol. 17, No. 8, ID. e0271928, pp. 1-16, 2022.

[15] B. Tong, J. Wang, X. Wang, F. Zhou, X. Mao, and Y. Duan, "Modelling Maintenance Scheduling Strategies for Highway Networks", *PLoS One*, Vol. 17, No. 6, ID. e0269656, pp. 1-24, 2022.

[16] R. Sivakumar, S. A. Angayarkanni, Y. V. R. Rao, and A. S. Sadiq, "Traffic Flow Forecasting using Natural Selection based Hybrid Bald Eagle Search-Grey Wolf Optimization Algorithm", *PLoS One*, Vol. 17, No. 9, ID. e0275104, pp. 1-15, 2022.

[17] M. A. Akbari, M. Zare, R. A. Abarghooee, S. Mirjalili, and M. Deriche, "The Cheetah Optimizer: A Nature-inspired Metaheuristic Algorithm for Large-scale Optimization Problems", *Scientific Reports*, Vol. 12, ID. 10953, pp. 1-20, 2022.

[18] 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.

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

[20] S. Z. Koohi, N. A. W. A. Hamid, M. Othman, and G. Ibragimov, "Raccoon Optimization Algorithm", *IEEE Access*, Vol. 7, pp. 5383-5399, 2019.

[21] 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.

[22] P. Trojovsky and M. Dehghani, "Pelican Optimization Algorithm: A Novel Nature-Inspired Algorithm for Engineering Applications", *Sensors*, Vol. 22, ID. 855, pp. 1-34, 2022.

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

[24] P. D. Kusuma and A. Dinimaharawati, "Hybrid Pelican Komodo Algorithm", *International Journal of Advanced Computer Science and Applications*, Vol. 13, No. 6, pp. 46-55, 2022.

[25] 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.

[26] 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.

[27] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime Mould Algorithm: A New Method for Stochastic Optimization", *Future Generation Computer Systems*, Vol. 111, pp. 300-323, 2020.

[28] 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.

[29] M. Dehghani, Z. Montazeri, A. Dehghani, R. A. R. Mendoza, H. Samet, J. M. Guerrero, and G. Dhiman, "MLO: Multi Leader Optimizer", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 6, pp. 364–373, 2020, doi: 10.22266/ijies2020.1231.32.

[30] F. A. Zeidabadi, S. A. 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.

[31] 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.

[32] 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.

[33] 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.

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