Tiki-taka algorithm: a novel metaheuristic inspired by football playing style

Mohd Fadzil Faisae Ab. Rashid

Department of Industrial Engineering, College of Engineering, Universiti Malaysia Pahang, Kuantan, Malaysia

Received 6 March 2020
Revised 4 May 2020
26 May 2020
Accepted 27 May 2020

Abstract

Purpose – Metaheuristic algorithms have been commonly used as an optimisation tool in various fields. However, optimisation of real-world problems has become increasingly challenging with to increase in system complexity. This situation has become a pull factor to introduce an efficient metaheuristic. This study aims to propose a novel sport-inspired algorithm based on a football playing style called tiki-taka.

Design/methodology/approach – The tiki-taka football style is characterised by short passing, player positioning and maintaining possession. This style aims to dominate the ball possession and defeat opponents using its tactical superiority. The proposed tiki-taka algorithm (TTA) simulates the short passing and player positioning behaviour for optimisation. The algorithm was tested using 19 benchmark functions and five engineering design problems. The performance of the proposed algorithm was compared with 11 other metaheuristics from sport-based, highly cited and recent algorithms.

Findings – The results showed that the TTA is extremely competitive, ranking first and second on 84% of benchmark problems. The proposed algorithm performs best in two engineering design problems and ranks second in the three remaining problems.

Originality/value – The originality of the proposed algorithm is the short passing strategy that exploits a nearby player to move to a better position.

Keywords Metaheuristic, Tiki-taka, Optimisation algorithm, Football-inspired

Paper type Research paper

1. Introduction

Optimisation is a process of searching for a combination of variables to obtain the best solution for a particular problem subjected to constraints. In most cases, the best solution is presented as a minimum or maximum of the objective function. Optimisation is necessary in all aspects of life for decision-making. However, most real-world problems are complex with numerous nonlinear constraints. This situation makes optimisation to become increasingly challenging.

A popular optimisation approach used to accurately overcome computational time problems is known as a heuristic. This approach provides fast results but cannot ensure optimality. A heuristic is a problem-dependent optimisation approach that is constructed to solve a particular problem type. A heuristic called Greedy Approximation Algorithm was proposed for the Knapsack problem that prioritises the item with the highest value per unit mass (Akçay *et al.*, 2006). Metaheuristics are another class of optimisation approach. Compared with heuristics, which are applicable for a particular problem, metaheuristics are not limited to a specific domain of problems. A metaheuristic is an iterative searching approach that is

The author would like to acknowledge the Ministry of Education, Malaysia and Universiti Malaysia Pahang for funding this research under FRGS grant (FRGS/1/2019/TK03/UMP/02/3).

guided by predefined mechanisms/strategies to explore and find the optimal (or near-optimal) solution in the search space. The popular metaheuristics are genetic algorithm (GA), ant colony optimisation (ACO), simulated annealing (SA) and particle swarm optimisation (PSO) (Abdel-Basset *et al.*, 2018; Kennedy and Eberhart, 1995; Dorigo *et al.*, 1996).

Previous researchers established many metaheuristic classifications. The first metaheuristic classification consists of single-based and population-based solutions. Single-solution metaheuristics such as SA and Tabu Search, start with one solution and improve it through iterations. Metaheuristics are also classified into static and dynamic objective functions. The objective function in most metaheuristics is consistent throughout the optimisation. However, the objective function in metaheuristics such as Guided Local Search changes during iteration to suit the searching condition. Metaheuristics can be categorised on the basis of memory usage and memory-less techniques. Memory usage technique refers to the utilisation of historical information for an algorithm to determine the next solution. Metaheuristics are also classified into nature-inspired and non-nature-inspired. This classification is based on the original concept of the manner an algorithm works. Nature-inspired metaheuristics are the active cluster in metaheuristic research. The nature concept such as from animals, plants and physical behaviour, is mathematically modelled as metaheuristics. By contrast, non-nature-inspired metaheuristics are based on other concepts from daily activities.

Since 2015, many metaheuristic algorithms based on different inspirations have been proposed. The largest metaheuristic category from 2015 to 2020 is the algorithm inspired by animal or insect behaviour. The popular animal-based algorithms in terms of citation are Whale Optimisation Algorithm (WOA), Ant Lion Optimiser (ALO) and Moth Flame Optimisation (MFO) (Mirjalili and Lewis, 2016; Mirjalili, 2015a, 2015b). WOA is inspired by humpback whale hunting method using bubble-net strategy. ALO is inspired by hunting behaviour for the insect called antlions. MFO mimics the moth flying nature in the presence of artificial light.

Other animal-inspired metaheuristics that attracted the attention of researchers include Salp Swarm Algorithm (Mirjalili *et al.*, 2017), Dragonfly Algorithm (Mirjalili, 2016), Grasshopper Algorithm (Saremi *et al.*, 2017), Lion Optimisation Algorithm (Yazdani and Jolai, 2016), Social Spider Algorithm (Yu and Li, 2015), Elephant Herding Algorithm (Wang *et al.*, 2016), Bird Swarm Algorithm (Meng *et al.*, 2016) and Spotted Hyena Optimiser (Dhiman and Kumar, 2017). These metaheuristics were introduced from 2015 to 2018 and obtained high publication citation. Few animal-based metaheuristics have been proposed in 2019 and early 2020. These metaheuristics include Harris Hawks Optimisation (HHO) (Heidari *et al.*, 2019), Butterfly Optimisation Algorithm (BOA) (Arora and Singh, 2019), Squirrel Search Algorithm (Jain *et al.*, 2019), Fitness Dependent Optimiser (FDO) (Abdullah and Ahmed, 2019), Side-Blotched Lizard Algorithm (O. Maciel *et al.*, 2020) and Pathfinder Algorithm (PFA) (Yapici and Cetinkaya, 2019). Although these algorithms are relatively new, they are widely used by researchers, especially HHO, BOA and PFA.

HHO is inspired by surprise pounce strategy of Harris hawks to chase the prey. Prior to surprise pounce, it cooperatively besieges the prey to make it exhausted. BOA is based on the foraging behaviour of butterflies seeking for flower source. The BOA implements a cooperative movement, where every butterfly will emit fragrance to attract each other. The butterfly will then fly towards the best fragrance and is affected by the search space landscape. PFA mimics the cooperative movement of an animal group to find the food source or prey. A structured leadership hierarchy is implemented to guide the search process.

The second most popular metaheuristics proposed between 2015–2020 are biological-based algorithms. The metaheuristics that fall in this category are inspired by nature and biological systems such as weather, water cycle, air system, human body and tree growth. The popular algorithm in this category is Water Wave Optimisation (WWO) (Zheng, 2015).

WWO mimics the shallow water wave concept consists of propagation, refraction and breaking phases. Another popular biological-based metaheuristic is Lightning Search Algorithm that is inspired by step leader propagation in lightning phenomenon (Shareef *et al.*, 2015). Other popular metaheuristics in this category are Virus Colony Search (Li *et al.*, 2016), Water Evaporation Optimisation (Kaveh and Bakhshpoori, 2016), Kidney-Inspired Algorithm (Jaddi *et al.*, 2017) and Color Harmony Algorithm (Zaeimi and Ghoddosian, 2020).

Physical-based metaheuristics are also a popular category among researchers from 2015 to 2020. Multi-Verse Optimiser (MVO) has the highest citation in this category (Mirjalili *et al.*, 2016). This algorithm is inspired by a cosmological concept that multiple universes interact with each other via white hole, black hole and worm hole concepts. Heat Transfer Search and Thermal Exchange Optimisation are algorithms inspired by thermodynamics (Patel and Savsani, 2015; Kaveh and Dadras, 2017). These algorithms use the thermal equilibrium concept in their mechanisms, where the thermal imbalance exists in the system and its surrounding. Ideal gas optimisation is derived from thermodynamics and kinetics of ideal gas law (Shams *et al.*, 2017). In this algorithm, gas molecules with different pressures and temperatures represent the candidate solutions. Atom Search Optimisation (ASO) is a physical-based metaheuristic that mimics molecular dynamics (Zhao *et al.*, 2019). The solution is represented by atoms and measured by their masses, with the heavier atom representing a better fitness. The solution reproduction is modelled on the basis of atomic movements, where the heavy atom moves towards the light ones.

Apart from the abovementioned categories, metaheuristics inspired by sport activities, especially football (or soccer) are another category. However, these metaheuristics have received less attention compared with animal, biological and physical-based algorithms. To date, several football-inspired algorithms have been published. A League Championship Algorithm (LCA) based on the league system in sports team competition was proposed in 2009 (Kashan, 2009). In the LCA, the candidate solution is represented by a sports team and the strategy formation represents the parameter values in the solution. The second football-based metaheuristic is Football Optimisation Algorithm (FOA) (Hatamzadeh and Khayyambashi, 2012). In the FOA, the players are ranked and selected in terms of fitness. Soccer Game Optimisation (SGO) is another football-based algorithm inspired by soccer player movement during the game (Purnomo and Wee, 2013). In SGO, the player represents the candidate solution and the ball dribbler represents the best solution. Golden Ball Algorithm (GBA) is another metaheuristic that simulates the concepts in a football game (Osaba *et al.*, 2013). In the GBA, a football tournament consists of several teams, where each player represents the candidate solution. Each team works individually and competes with one another to become the best.

Soccer League Optimisation (SLO) simulates the football league system in European countries, where each country has several leagues with different club performance (Khaji, 2014). The wealthier team has a greater chance to sign a better player, whereas the poorer team has the ability to purchase only young players. Football Game Algorithm (FGA) is another football-based metaheuristic. This algorithm mimics the behaviour of football players to find the best position in scoring a goal (Fadakar and Ebrahimi, 2016). The player in a better position has a better chance to get the ball. World cup optimisation (WCO) is inspired by the International Federation of Association Football (FIFA) tournament of national football teams to reach the FIFA World Cup championship (Razmjooy *et al.*, 2016). The new population is established on the basis on the previous cup champion and ranking of teams.

The existing metaheuristics inspired by a football game can be divided into two groups. The first group of metaheuristics such as GBA, SLO, LCA and WCO algorithms, mimics the football league or championship system. These algorithms normally consider the overall team performance rather than individual fitness to obtain candidate solutions with better average fitness. The second group, including SGO, FGA and FOA, simulates the football game itself by modelling the player movement, ball position and substitutes. The exploitation mechanisms in most algorithms clearly represent the actual football game, including the player movement. Majority of these algorithms have applied substitution strategy to enhance exploration. These algorithms only simulate a general football game behaviour without implementing specific tactics. Different football playing styles such as samba by the South American team and total football by the Dutch team, are used by football teams. This study proposed a novel football-based algorithm that is inspired by a football-playing style called tiki-taka. Apart from achieving high-quality result, the challenge in the proposed metaheuristic is to balance the exploration and exploitation capabilities throughout the optimisation (Dokeroglu *et al.*, 2019).

2. Tiki-taka algorithm

2.1 Inspiration

Tiki-taka refers to a football playing style that is associated with the Spanish national team and Barcelona football club (BCF). This playing style is characterised by short passing, player movement and possession control. The tiki-taka approach allows a football team to slowly build the attack movements from the defend position. This tactic was introduced by Johan Cruyff and popularised by Pep Guardiola. The BCF won 14 out of 19 trophies from 2008 to 2012 by using this tactic (Hayward, 2015). Apart from that, the tiki-taka tactic also led to the success of the Spanish national team in winning the Union of European Football Associations (UEFA) football championship in 2008 (EURO 2008) under its manager, Luis Aragonés and FIFA World Cup 2010 under its manager, Vicente del Bosque.

The tiki-taka style is contradicted by physical football that favours the physical strength, running ability and man-mark ability of the opponent. In tiki-taka, few smart players are needed to read the game, with quick movements and precise player positions. They are the key players for the team who will determine the game pace. In the BCF, these players include Lionel Messi, Andrés Iniesta and Xavi Hernández. During the game, players consistently look for a chance to pass the ball to the key players for them to build the attacking movements. Tiki-taka tactics aim to dominate the ball possession using tactical superiority and fluidity to overcome opponents.

2.2 Mathematical formulation

Tiki-taka Algorithm (TTA) is inspired by two main characteristics in the tiki-taka tactic, which are short passing and player movement. In a real game, players will form a triangle of three players who will keep passing the ball among them. The players will form another triangle by finding a better space when the opponents enter the triangle.

In the TTA, the short passing strategy is adopted, where the player (potential solution) will pass the ball to the nearby player, as shown in Figure 1. The player will then find a better position in accordance with the ball and key player positions. In the TTA, the concept of multiple key players (leaders) is adopted, similar to the real tiki-taka strategy. The aim of multiple leaders is to enhance solution divergence and avoid algorithm trapping in the local optimum.

Figure 2 shows a flowchart of the proposed TTA. The algorithm begins by initializing the player position and its parameters. The player position is evaluated by using a fitness function. Next, the key players will be updated in accordance with their fitness level. The algorithm will update the ball position before the player position is updated.

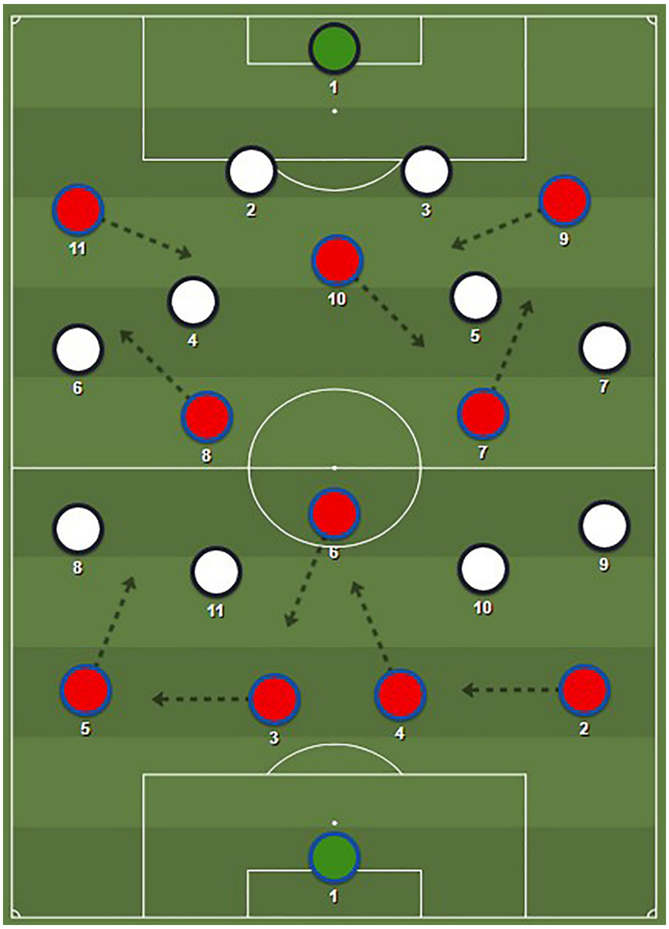

Figure 1.
Short passing in tiki-taka

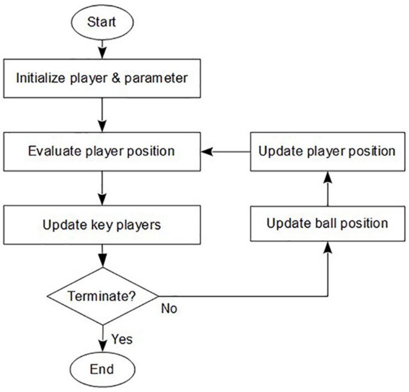

Figure 2.
Flowchart of the TTA

2.2.1 Initialisation. Consider a football team with n players. The player positions that represent the possible solutions are randomly created with d dimensions within the bound limit. At the same time, the number of key players, n_k, is determined, which is approximately 10% of the total players or a minimum of three key players. Apart from simulating the real tiki-taka play, the concept of multiple key players (leading solutions) in the TTA enables it to maintain the diversity because the candidate solution is influenced by different leaders. Another matrix that represents ball position, B, is established. In this algorithm, the ball position represents the vector that will guide the player movement in the exploitation phase. Note that for initial solution, $B = P$:

$$P = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,d} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n,1} & p_{n,2} & \cdots & p_{n,d} \end{pmatrix} \tag{1}$$

$$B = \begin{pmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,d} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n,1} & b_{n,2} & \cdots & b_{n,d} \end{pmatrix} \tag{2}$$

The initial player position, P, is evaluated in accordance with the objective function. The top n_k players are updated in the key player archive, h. The key player archive will be updated in every iteration and only contains the current values.

2.2.2 Update ball position. The basic characteristic of tiki-taka is short passing. The proposed algorithm adopts this concept. The player will pass the ball to the next nearby player. Although the successful passing rate in tiki-taka is high, losing the ball to the opponent is possible. In this work, the probability to lose the ball ($prob_{lose}$) is between 10%–30% of the overall passes.

The new ball position, b_i', is expressed as:

$$b_i' = \begin{cases} rand(b_i - b_{i+1}) + b_i, & r_p > prob_{lose} \\ b_i - (c_1 + rand)(b_i - b_{i+1}), & r_p \leq prob_{lose} \end{cases} \tag{3}$$

where r_p is a random number (0, 1). For a successful pass ($r_p > prob_{lose}$), the ball will be passed to the nearest player, with some random factors. For an unsuccessful pass, the ball is assumed to be blocked and delivered behind the player, as shown in equation (3). In this equation, c_1 is a coefficient that influences the ball reflection magnitude in the unsuccessful pass. The term $(b_i - b_{i+1})$ represents the distance between the i-th ball position to the i-th+1 ball. For the last ball position, b_n, the term b_{i+1} is replaced with b_1.

2.2.3 Update player position. After passing the ball, the player needs to move and find a better position in the formation. In the tiki-taka tactic, the player movement is influenced by the ball, key player and opponent player positions. However, this algorithm only considers the key player position (h) and ball position, as shown in Figure 3. The selection of key players for the updating process is random because the number of key players is more than one. This strategy allows the algorithm to maintain its diversity because the player movement is independent on a single best player's position.

The player updating procedure adopts the following formula:

$$p_i' = p_i + rand^* c_2^* (b_i' - p_i) + rand^* c_3^* (h - p_i) \tag{4}$$

where h represents the key player's position. c_2 and c_3 are the coefficients that balance the player position between the ball and the key player. The updated player position is evaluated and the key player position is updated in the archive.

The pseudocode of the TTA algorithm is presented as follows:

```
Procedure of TTA
 Initialise TTA parameter
              $d$ = problem dimension
              $n$ = player number
              $m$ = maximum iteration
              $prob_{lost}$ = probability of ball lost [0.1 ~ 0.3]
              $c_1$, $c_3$ = coefficients [0.5 ~ 1.5],
              $c_2$ = coefficient [1.0 ~ 2.5]
 Generate initial player position, $P = \{p_i, p_{i+1}, p_{i+2}, \ldots, p_n\}$
 Evaluate initial player position, $fp = f(p)$
 Save key players' position, $h$
 $iter = 0$
 While $iter < m$
        $iter = iter + 1$
        for $i = 1$ to $n$
            Update ball position, $B$ using equation (3)
 end
 for $j = 1$ to $n$
            Update player position, $P$ using equation (4)
 end
 Evaluate $P'$
            $fp = f(p')$
 Update historical best position, $h$
 End
```

The uniqueness of the proposed algorithm is the short passing mechanism for the exploitation of the solution. In the TTA, the solution updating process is influenced by the next nearby player and multiple key players to guide the search direction.

To estimate the algorithm complexity, Big-O notation is used to show the highest order of algorithm complexity in each step. The proposed TTA's complexity relies on the number of maximum iterations, m and the number of players (or population size), n. During the

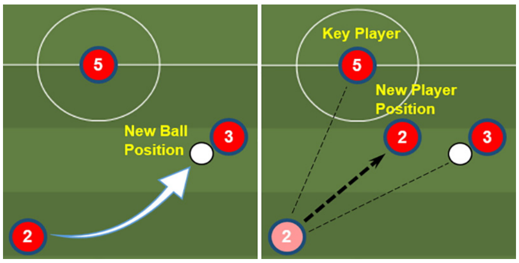

Figure 3.
Update ball and player positions

initialisation stage, a loop dependent on *n* size is used to generate initial players and its complexity is O(*n*). The algorithm's complexity increases to O(*m***n*) when it enters the main iteration loop for evaluation, update key player, update ball position and update player position steps. The algorithm only involves a constant O(1) for termination. The highest Big-O order in the main iteration loop is O(*m***n*). The population size in metaheuristic applications typically ranges from 20 to 40. However, the maximum iteration can reach to hundred thousands, indicating that *n* is relatively small compared with *m*. Therefore, the complexity of the TTA is low because it linearly increases when the value of *m* is large enough compared with *n*.

3. Results and discussion

Nineteen standard benchmark mathematical test functions with known global optima were applied to test the performance of TTA (Suganthan, 2005; Liang *et al.*, 2005). The test problems were divided into three categories, namely, unimodal, multimodal and composite functions. All the functions were tested with 50 dimensions. The unimodal test function consisted of F1 to F7, as shown in Table 1. These functions were used to test the exploitation ability of the algorithm. These functions had only one optimum solution, without local optimum.

The second category, known as multimodal, comprised six functions (F8 to F13), as shown in Table 2. The multimodal functions contained multiple optimum solutions. However, only one solution was the global optimum and the rest were local optima. Therefore, these problems were more challenging compared with the unimodal function. The multimodal problems were used to test the exploration ability of the proposed algorithm. These problems were also used to test the ability of the proposed algorithm in avoiding trapping in the local optimum.

The third category was the composite function, where the unimodal and multimodal functions were modified by combining, rotating, biasing and shifting the original functions. The composite function was used to test the balance between exploitation and exploration in the algorithm for optimizing real problems. This condition was because the composite function simulates the search space in real problems, with complex and large numbers of local optimum. These functions were repeated until the desired number of dimensions was achieved. The composite test function is presented in Table 3.

3.1 Effects of tiki-taka algorithm parameters

Four parameters, namely, c_1: coefficient for ball reflection magnitude, c_2: coefficient of ball position, c_3: coefficient of key player position and $prob_{lost}$: probability of unsuccessful ball pass, influenced the TTA performance. $prob_{lost}$ parameter function maintains the

Table 1.
Unimodal test function

Test function	Range
$F_1(x) = \sum_{i=1}^{d} x_i^2$	(−100, 100)
$F_2(x) = \sum_{i=1}^{d} \lvert x_i \rvert + \prod_{i=1}^{d} \lvert x_i \rvert$	(−10, 10)
$F_3(x) = \sum_{i=1}^{d} \left(\sum_{j-1}^{i} x_j \right)^2$	(−100, 100)
$F_4(x) = \max_i \{\lvert x_i \rvert,\ 1 \le i \le d\}$	(−100, 100)
$F_5(x) = \sum_{i=1}^{d-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2 \right]$	(−30, 30)
$F_6(x) = \sum_{i=1}^{d} ([x_i + 0.5])^2$	(−100, 100)
$F_7(x) = \sum_{i=1}^{n} i x_i^4 + \mathrm{random}[0, 1]$	(−1.28, 1.28)

Test function	Range
$F_8(x) = \sum_{i=1}^{d} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	(−500, 500)
$F_9(x) = \sum_{i=1}^{d} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	(−5.12, 5.12)
$F_{10}(x) = -20e\left(-0.2\sqrt{\frac{1}{d}\sum_{i}^{d} x_i^2}\right) - e(1\ d \sum_{i}^{d} \cos(2\pi x_i)) + 20 + e$	(−32, 32)
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	(−600, 600)
$F_{12}(x) = \frac{\pi}{d}\left\{10\sin(\pi y_1) + \sum_{i=1}^{d-1} (y_i - 1)^2[1 + 10^{\sin 2}(\pi y_{i+1})] + (y_d - 1)^2\right\} + \sum_{i=1}^{d} u(x_i, 10, 100, 4)$	(−50, 50)
$y_i = 1 + \frac{x_i + 1}{4}$	
$F_{13}(x) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{d} (x_i - 1)^2\left[1 + \sin^2(3\pi x_i + 1)\right] + (x_d - 1)\ 2\left[1 + \sin^2(2\pi x\ d)\right]\right\} + \sum_{i=1}^{d} u(x_i, 5, 100, 4)$	(−50, 50)

Table 2. Multimodal test function

Test function	Range
F_{14}(CF1): $f_1, f_2, f_3, \ldots, f_d = \text{Sphere Function}$; $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_d] = [1, 1, 1, \ldots, 1]$; $[\lambda_1, \lambda 2, \lambda_3, \ldots, \lambda_d] = [5/100, 5/100, 5/100, \ldots, 5/100]$	(−65.536, 65.536)
F_{15}(CF2): $f_1, f_2, f_3, \ldots, f_d = \text{Griewank's Function}$; $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_d] = [1, 1, 1, \ldots, 1]$; $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_d] = [5/100, 5/100, 5/100, \ldots, 5/100]$	(−5, 5)
F_{16}(CF3): $f_1, f_2, f_3, \ldots, f_d = \text{Griewank's Function}$; $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_d] = [1, 1, 1, \ldots, 1]$; $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_d] = [1, 1, 1, \ldots, 1]$	(−5, 5)
F_{17}(CF4): $f_1, f_2 = \text{Ackley's Function}$; $f_3, f_4 = \text{Rastrigin's Function}$; $f_5, f_6 = \text{Weierstrass's Function}$; $f_7, f_8 = \text{Griewank's Function}$; $f_9, f_{10} = \text{Sphere Function}$; $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1]$; $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = \left[\frac{5}{32}, \frac{5}{32}, 1, 1, \frac{5}{0.5}, \frac{5}{0.5}, \frac{5}{100}, \frac{5}{100}, \frac{5}{100}, \frac{5}{100}\right]$	(−5, 5)
F_{18}(CF5): $f_1, f_2 = \text{Rastrigin's Function}$; $f_3, f_4 = \text{Weierstrass's Function}$; $f_5, f_6 = \text{Griewank's Function}$; $f_7, f_8 = \text{Ackley's Function}$; $f_9, f_{10} = \text{Sphere Function}$; $[\sigma_1, \sigma_2, \sigma 3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1]$; $[\lambda_1, \lambda_2, \lambda 3, \ldots, \lambda_{10}] = \left[\frac{1}{5}, \frac{1}{5}, \frac{5}{0.5}, \frac{5}{0.5}, \frac{5}{100}, \frac{5}{100}, \frac{5}{32}, \frac{5}{32}, \frac{5}{100}, \frac{5}{100}\right]$	(–2, 2)
F_{19}(CF6): $f_1, f_2 = \text{Rastrigin's Function}$; $f_3, f_4 = \text{Weierstrass's Function}$; $f_5, f_6 = \text{Griewank's Function}$; $f_7, f_8 = \text{Ackley's Function}$; $f_9, f_{10} = \text{Sphere Function}$; $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]$; $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = \left[0.1*\frac{1}{5}, 0.2*\frac{1}{5}, 0.3*\frac{5}{0.5}, 0.4*\frac{5}{0.5}, 0.5*\frac{5}{100}, 0.6*\frac{5}{100}, 0.7*\frac{5}{32}, 0.8 * \frac{5}{32}, 0.9*\frac{5}{100}, 1*\frac{5}{100}\right]$	(0, 1)

Table 3.
Composite test function

population diversity, where a specified percentage of the solution will undergo different ball position updating procedures. c_1 determines the reflection magnitude of unsuccessful ball pass. High c_1 coefficient results in far ball distance from the original position. This mechanism will assist in maintaining the solution diversity for the entire iterations. c_2 and c_3 help balance the influence of ball and key player positions.

A design of experiment based on Taguchi was constructed to identify the effects of these parameters on the TTA's performance. Each parameter was set to three levels, as shown in Table 4 and L27 orthogonal array was used. The average rank obtained by each experiment set was used as an output parameter for analysis. Figure 4 shows the main effect plot of signal-to-noise ratio in the experiment.

As shown in Figure 4, c_3 has the highest effect, followed by c_2. The performance of TTA is low when c_3 is at a high level because of the high dependencies on the key player. This condition causes the solutions to be trapped in the local optimum. $prob_{lost}$ poorly performs at a high level because the solution becomes extremely diversified. This condition disrupts the algorithm convergence because excessive solutions are randomly delivered far from the original position. On the basis of the main effect plot for signal-to-noise ratio in Figure 4, the optimum level for the TTA parameters are $c_1 = 1.2$, $c_2 = 2.5$, $c_3 = 1.0$ and $prob_{lost} = 0.2$.

3.2 Comparison of performance

The TTA's performance to optimise the benchmark functions was compared with 11 metaheuristic algorithms through computational experiments. The comparison algorithms used were mainly sport-based algorithms, including SLO, LCA, GBA and FGA (Kashan, 2009; Osaba *et al.*, 2013; Fadakar and Ebrahimi, 2016). The TTA was also compared with the popular swarm-based algorithm, PSO and a highly cited algorithm in the past five years, MFO (Kennedy and Eberhart, 1995; Mirjalili, 2015a). The proposed TTA was compared with the recent metaheuristics published in early 2019, namely, ASO (Zhao *et al.*, 2019), BOA (Arora and Singh, 2019), FDO (Abdullah and Ahmed, 2019), HHO (Heidari *et al.*, 2019) and PFA (Yapici and Cetinkaya, 2019). In the experiment, all algorithms used 30 search agents

Level	c_1	c_2	c_3	$prob_{lost}$
Low	0.8	1.5	0.5	0.1
Medium	1.2	2.0	1.0	0.2
High	1.5	2.5	1.5	0.3

Table 4. Parameter levels for taguchi design

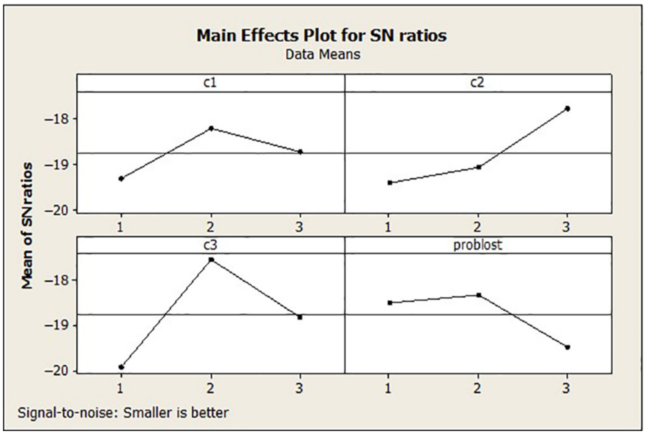

Figure 4. Main effect plot of signal-to-noise ratios

and 1,000 iterations. The optimisation was repeated for 30 times to reduce the pseudo-random effect in the solution.

A statistical test was then conducted to test the significance of the results obtained by TTA compared with the comparison algorithms. A nonparametric test was chosen because the results (i.e. best fitness) obtained from optimisation did not have a normal distribution. For this purpose, a Wilcoxon-signed rank test was conducted for each set of results. The null hypothesis in this test was that the median of the two samples are equal. The null hypothesis is accepted when the p-value is larger than the significance level (0.05) and vice versa. In this case, the null hypothesis was rejected because it showed that a significant difference was found between the TTA and comparison algorithm results.

Table 5 shows the mean and standard deviation of fitness for benchmark test problems. The underlined data represent the best mean for a particular test function. On the basis of the results for unimodal test problem, the TTA outperformed other algorithms in five out of seven problems (i.e. F1, F2, F3, F4 and F7). The TTA ranked third for the rest of the test functions (F5, F6). In addition, 92% of the p-value obtained from the Wilcoxon-signed rank test was smaller than 0.05, as shown in Table 6. The data marked with an asterisk (*) show the equal median of two samples or the comparison algorithm with a significant performance compared with the TTA. The data with double asterisks (**) indicate that the comparison algorithm is better and has significant performance compared with the TTA.

For the multimodal test problems, the TTA performed best in three out of six test problems. The best TTA performance was observed in F9, F10 and F11 problems. The TTA was behind HHO for F8 and F12. In the F13 problem, the TTA ranked sixth behind the comparison algorithms, which was its worst performance for all test problems. The significance of the results obtained by the TTA was supported by statistical test, where more than 81% of the p-values were smaller than 0.05 (Table 6).

In the composite benchmark problems, few algorithms showed outstanding performance by producing similar minimum mean fitness. In F17 and F18, 11 out of 12 metaheuristics reached the minimum mean fitness. The results obtained by the TTA for this particular class of problem were excellent. The TTA obtained the minimum mean in all test problems, except for F16. For this problem, TTA was behind BOA. However, only 58% of the cases with significant p-values are observed in Table 6. The percentage was relatively low compared with the unimodal and multimodal classes, which was because the TTA obtained similar mean fitness in a few problems compared with other algorithms.

Table 7 presents the count of test problems where the TTA performs better, equal or worse when a head-to-head comparison of mean fitness is conducted. Compared with LCA, the TTA displayed better performance in 14 test problems, whereas it exhibited in equal and worse performance in four and one test problem, respectively.

As shown in Table 7, the TTA showed better performance in 77% of the cases compared with the comparison algorithms. The TTA concomitantly revealed equivalent performance in 16% cases. These numbers collectively underlined TTA and its equal performance at least with the best metaheuristics in 93% cases. The TTA performed worse than the comparison algorithms, especially the new metaheuristics of ASO and HHO, in the remaining 7% cases.

Figures 5, 6 and 7 show the convergence plots for the selected unimodal, multimodal and composite test problems, respectively. For the unimodal test problem, the algorithm kept converging until the last segment of iterations, indicating that the TTA continuously improved the solution through exploitation until the end. The convergence curve of the multimodal and composite test problems showed that the TTA had a fast convergence rate.

Table 5. Optimisation results of benchmark test functions

Function		SLO	LCA	GBA	FGA	PSO	MFO
Unimodal function							
F1	Mean	3.33E-13	199.5008	1.08E-15	0.8262	0.2426	9014.504
	SD	6.44E-13	154.0497	1.07E-16	2.5057	0.2024	9940.33
F2	Mean	9.3E-08	4.9825	1.81E-08	0.9343	3.2715	78.6241
	SD	6.96E-08	2.7206	1.01E-09	0.5497	0.8478	26.0911
F3	Mean	2.34E-12	111975	100.2237	1138.645	1.4748	64283.15
	SD	3.41E-12	23348.08	60.5912	901.1192	0.6656	33746.01
F4	Mean	2.56E-07	71.4791	38.8617	6.8691	0.3466	81.6061
	SD	2.4E-07	7.0291	5.4642	1.5288	5.30E-2	2.7860
F5	Mean	48.7309	5.05E + 4	84.3481	153.2752	93.8767	1.60E + 7
	SD	0.1353	8.01E + 4	37.1771	61.3972	25.9966	3.37E + 7
F6	Mean	7.0548	168.124	9.81E-16	0.0467	0.4449	2019.726
	SD	0.5260	155.6126	1.71E-16	0.0944	0.2638	4210.849
F7	Mean	02.10E-3	1.6183	0.0297	0.0592	2.1033	19.9954
	SD	1.93E-3	0.5768	0.0103	0.0156	0.8618	22.7090
Multi-modal function							
F8	Mean	−5555.2	−13472.4	−13110.6	−10874	−10932.2	−13602.9
	SD	1185.434	423.8492	1219.321	1258.415	1094.256	1183.551
F9	Mean	1.71E-12	120.2755	171.4306	100.2026	56.8671	297.8231
	SD	3E-12	21.4257	43.4829	31.8876	13.0353	29.9294
F10	Mean	2.05E-07	5.5854	6.15E-09	2.8829	1.0173	19.4308
	SD	2.18E-07	1.9273	5.98E-10	0.6397	0.4580	0.5310
F11	Mean	3.44E-13	3.3693	3.69E-3	0.2185	7.63E-3	100.1102
	SD	3.81E-13	2.4581	8.31E-3	0.1666	5.78E-3	137.0857
F12	Mean	0.0849	1.1598	1.4088	0.8289	6.55E-3	1613.76
	SD	0.0222	0.5047	1.5294	0.9573	3.95E-3	4963.621
F13	Mean	4.9057	48.4498	9.88E-3	11.7883	0.9373	8.20E + 7
	SD	0.2272	74.3521	0.0131	8.8911	0.4639	1.73E + 06
Composite function							
F14	Mean	3.0621	0.998	2.1825	4.0494	12.6705	1.5920
	SD	3.1969	2.16E-16	2.0691	3.0738	1.6E-13	1.2523
F15	Mean	2.32E-03	2.61E-03	6.92E-04	2.54E-03	3.82E-04	3.08E-03
	SD	6.33E-03	1.40E-03	3.78E-04	6.28E-03	2.43E-04	6.08E-03
F16	Mean	−1.0316	−1.0297	−1.0316	−1.0316	−1.0316	−1.0316
	SD	1.28E-16	3.96E-03	2.09E-16	0	6.78E-16	0
F17	Mean	0.39788	0.3979	0.39788	0.39788	0.39788	0.39788
	SD	0	0	0	0	0	0
F18	Mean	3.00	3.00	3.00	3.00	3.00	3.00
	SD	3.64E-15	1.04E-14	6.45E-16	1.07E-15	1.21E-15	1.72E-15
F19	Mean	−3.86278	−3.86278	−3.86278	−3.86278	−3.86278	−3.86278
	SD	4.68E-16	3.31E-16	3.31E-16	9.36E-16	2.71E-15	9.36E-16

(continued)

Table 5.

Function	ASO	FDO	BOA	HHO	PFA	TTA
Unimodal function						
F1	65.6419	5.82E-54	1.87E-14	2.8E-186	7.03E-14	3.7E-198
	59.0205	1.38E-53	9.17E-16	0	9.59E-14	0
F2	7.0785	3.91E-4	8.83E-17	5.44E-96	6.4E-10	2.5E-103
	3.0728	8.39 E-4	7.22E-17	2.79E-95	3.73E-10	1.4E-102
F3	12237.89	2.46E-18	1.78E-14	1.4E-132	100.3365	4E-177
	3211.396	5.78E-18	1.79E-15	7.6E-132	81.21773	0
F4	19.6681	3.78E-15	1.19E-11	4.94E-92	6.4880	2.6E-101
	4.6398	7.06E-15	6.14E-13	2.61E-91	2.1834	1.2E-100
F5	1978.779	10.486	48.9293	4.21E-3	67.0471	48.1064
	1270.149	23.0549	3.97E-2	7.21E-3	37.4660	0.7742
F6	68.2213	5.27E-32	10.4500	2.63E-05	0.01677	1.95E-05
	144.2584	5.03E-32	0.8208	3.53E-05	0.0645	0.9091
F7	0.9075	0.8157	8.76E04	6.55E-05	0.0601	6.23E-05
	0.4023	0.2995	3.11E-4	7.07E-05	0.0116	4.84E-05
Multi-modal function						
F8	−8448.39	−10612.6	−5089.19	−20948.8	−13555	−14580.8
	1039.226	8631.474	429.1594	0.6353	718.4408	841.7898
F9	90.3189	5.2907	1.48E-08	0.00	176.9247	0.00
	19.4826	2.0400	4.68E-08	0	18.5633	0
F10	5.6558	7.28E-15	1.23E-11	8.88E-16	7.1870	8.88E-16
	1.6434	1.5E-15	5.57E-13	0.00	8.1186	0
F11	1.5657	0.1364	6.62E-15	0.00	8.09E-3	0.00
	0.6776	0.0716	5.55E-15	0	9.91E-3	0
F12	9.1160	4.1696	0.9618	9.35E-07	4.1577	4.40E-06
	2.2620	1.9089	0.1164	9.6E-07	2.3998	9.86E-06
F13	54.0699	0.8585	4.9954	4.77E-05	0.1464	4.0056
	17.2337	1.7756	0.0041	7.94E-05	0.1254	0.2408
Composite function						
F14	1.3952	2.779	1.1052	1.1954	2.4397	0.998
	0.6940	1.7240	0.31194	0.9122	2.7907	3.39E-16
F15	7.71E-04	6.95E-04	3.57E-04	3.58E-04	0.008452	3.37E-04
	4.18E-05	5.63E-04	4.83E-05	1.78E-04	0.010072	6.15E-05
F16	−1.0316	−1.0316	−10236.2	−1.0316	−1.0316	−1.0316
	1.48E-16	4.96E-12	7088.526	5.82E-11	1.2E-10	5.61E-16
F17	0.39788	0.39788	0.41058	0.39788	0.39788	0.39788
	0	5.44E-15	0.01946	1.63E-07	3.22E-09	0
F18	3.00	3.00	3.3135	3.00	3.00	3.00
	2.36E-15	2.21E-11	0.3080	3.56E-09	1.98E-09	6.79E-08
F19	−3.86278	−3.86042	−3.86278	−3.86161	−3.86278	−3.86278
	6.94E-16	0.003807	6.64E-13	0.001652	1.72E-08	4.2E-16

Table 6. Results of wilcoxon signed rank test (p-value) for benchmark test functions

Function	SLO	LCA	GBA	FGA	PSO	MFO	ASO	FDO	BOA	HHO	PFA
Unimodal function											
F1	2.94E-11	2.94E-11	2.94E-11	2.94E-11	3.02E-11	2.94E-11	2.94E-11	2.94E-11	2.94E-11	2.44E-09	2.99E-11
F2	2.94E-11	2.94E-11	2.94E-11	2.94E-11	3.02E-11	2.94E-11	2.94E-11	2.94E-11	2.94E-11	1.69E-09	2.99E-11
F3	2.94E-11	2.94E-11	2.94E-11	2.94E-11	3.02E-11	2.94E-11	2.94E-11	2.94E-11	2.94E-11	4.98E-11	2.99E-11
F4	2.94E-11	2.94E-11	2.94E-11	2.94E-11	3.02E-11	2.94E-11	2.94E-11	2.94E-11	2.94E-11	3.2E-09	2.99E-11
F5	8.22E-08	2.94E-11	7.92E-03	2.94E-11	3.02E-11	2.94E-11	2.94E-11	1.05E-07**	2.94E-11	3.02E-11**	0.02761
F6	2.94E-11	2.94E-11	2.94E-11**	2.94E-11	3.02E-11	2.94E-11	2.94E-11	2.48E-11**	2.94E-11	0.9705*	2.42E-09
F7	2.94E-11	2.94E-11	2.94E-11	2.94E-11	3.02E-11	2.94E-11	2.94E-11	2.94E-11	2.94E-11	0.5894*	2.99E-11
Multi-modal function											
F8	2.94E-11	5.03E-06	9.41E-06	2.94E-11	9.51E-06	2.25E-03	2.94E-11	6.7E-05	2.94E-11	3.02E-11**	2.12E-05
F9	1.12E-12	1.17E-12	1.17E-12	1.17E-12	1.21E-12	1.17E-12	1.17E-12	1.17E-12	0.0814*	N/A*	1.2E-12
F10	1.17E-12	1.17E-12	1.17E-12	1.17E-12	1.21E-12	1.17E-12	1.17E-12	1.55E-13	1.15E-12	N/A*	1.2E-12
F11	1.17E-12	1.17E-12	1.07E-12	1.17E-12	1.21E-12	1.17E-12	1.17E-12	1.17E-12	1.17E-12	N/A*	1.2E-12
F12	2.94E-11	2.94E-11	1.05E-07	2.94E-11	3.02E-11	2.94E-11	2.94E-11	2.94E-11	2.94E-11	0.0006**	2.99E-11
F13	9.69E-11	6.94E-03	2.09E-11**	7.92E-03**	3.02E-11**	2.94E-11	2.94E-11	1.05E-07	2.94E-11	3.02E-11**	2.99E-11**
Composite function											
F14	4.35E-08	N/A*	0.0013	1.71E-09	1.69E-14	0.0108	0.0013	1.65E-09	1.17E-12	0.0408	0.0006
F15	1.69E-05	2.94E-11	7.91E-09	6.66E-03	1.69E-05	2.85E-11	2.94E-11	8.72E-07	9.7E-05	0.0138	0.6608*
F16	N/A*	8.75E-07	N/A*	N/A*	N/A*	N/A*	N/A*	N/A*	1.17E-12**	N/A*	N/A*
F17	N/A*	N/A*	N/A*	N/A*	N/A*	N/A*	N/A*	N/A*	1.17E-12	6.24E-10	1.83E-10
F18	1.3E-07	1.3E-07	1.3E-07	1.3E-07	1.3E-07	1.3E-07	1.3E-07	1.3E-07	2.63E-11	8.87E-03	0.9225*
F19	N/A*	N/A*	N/A*	N/A*	N/A*	N/A*	N/A*	4.64E-08	1.17E-12	1.21E-12	1.18E-12

Notes: *No significant difference; **comparison algorithm has significant performance over TTA

In most test problems, the convergence completed at the 200th iteration. This result explained the superiorities of the TTA to obtain global optimum with a small number of iterations.

3.3 Algorithm analyses

An additional test with only five players and 100 iterations was conducted for the selected test problems to verify the search characteristic of the TTA. Figure 8 presents the search history, trajectory and average fitness for the selected test problems. The player positions for 100 iterations were marked with black marks. The solutions based on the search history were fairly distributed in the search space during the early stage of iteration. The solutions then moved towards the promising global optimum.

The trajectory illustrated in the third column of Figure 8 shows the value of the first variable for the first player, indicating the change in the variable value during the iteration. In most problems, rapid changes occur during the early stage of iteration. The changes gradually reduce with time. In F8, F12, F16 and F17, the variable values experienced a small fluctuation after rapid changes, indicating that the exploration of the solution continued to occur. Figure 9 shows the percentage of exploitation and exploration in the TTA for the

Table 7.
Head-to-head comparison of the TTA performance

TTA performance	SLO	LCA	GBA	FGA	PSO	MFO	ASO	FDO	BOA	HHO	PFA	(%)
Better	16	14	14	16	14	16	16	14	17	10	15	77.0
Equal	3	4	3	3	4	3	3	2	1	5	3	16.3
Worse	0	1	2	0	1	0	0	3	1	4	1	6.7

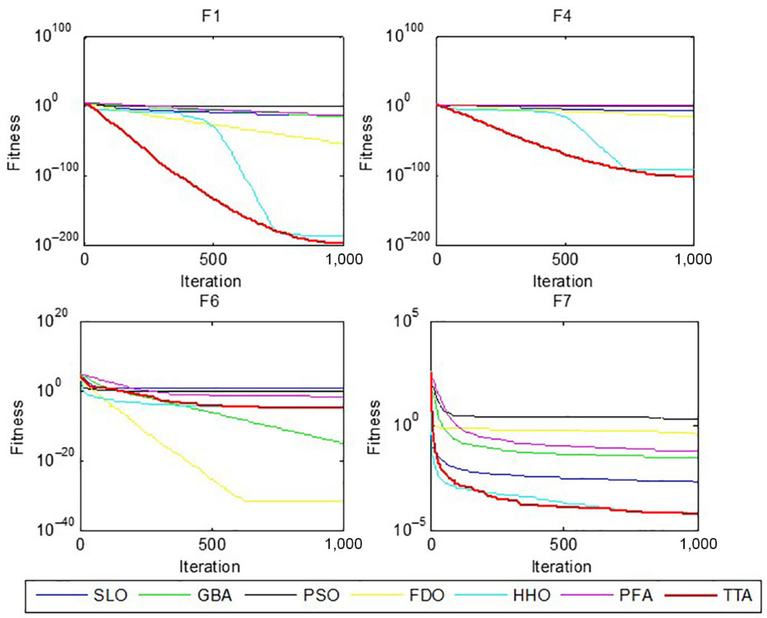

Figure 5.
Convergence plot of unimodal problems

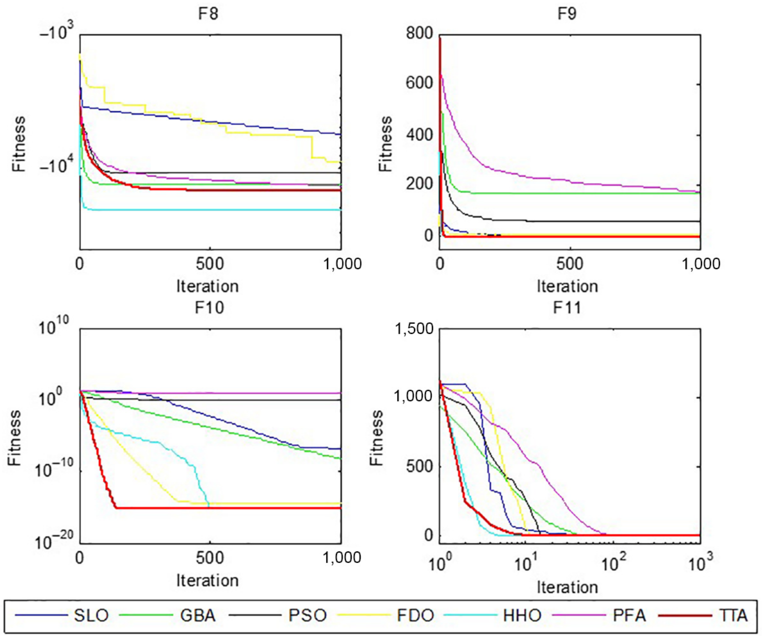

Figure 6.
Convergence plot of multimodal problems

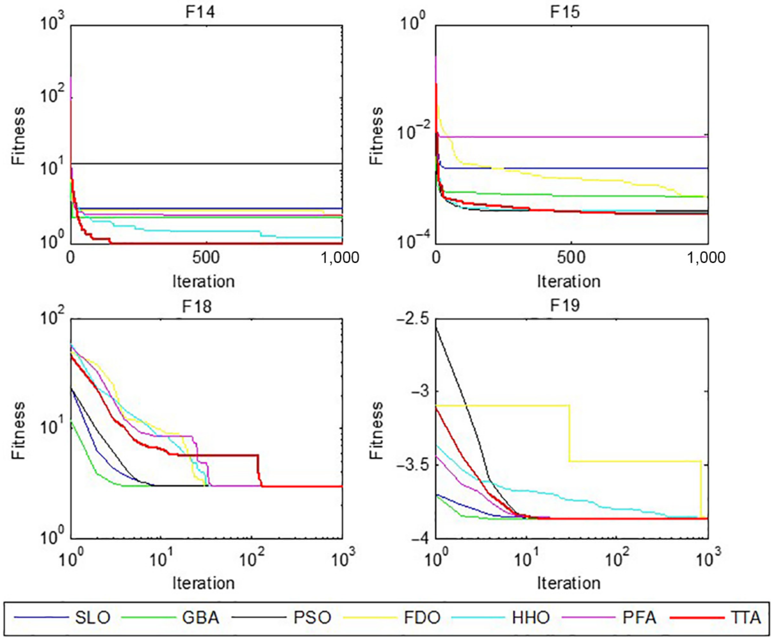

Figure 7.
Convergence plot of composite problems

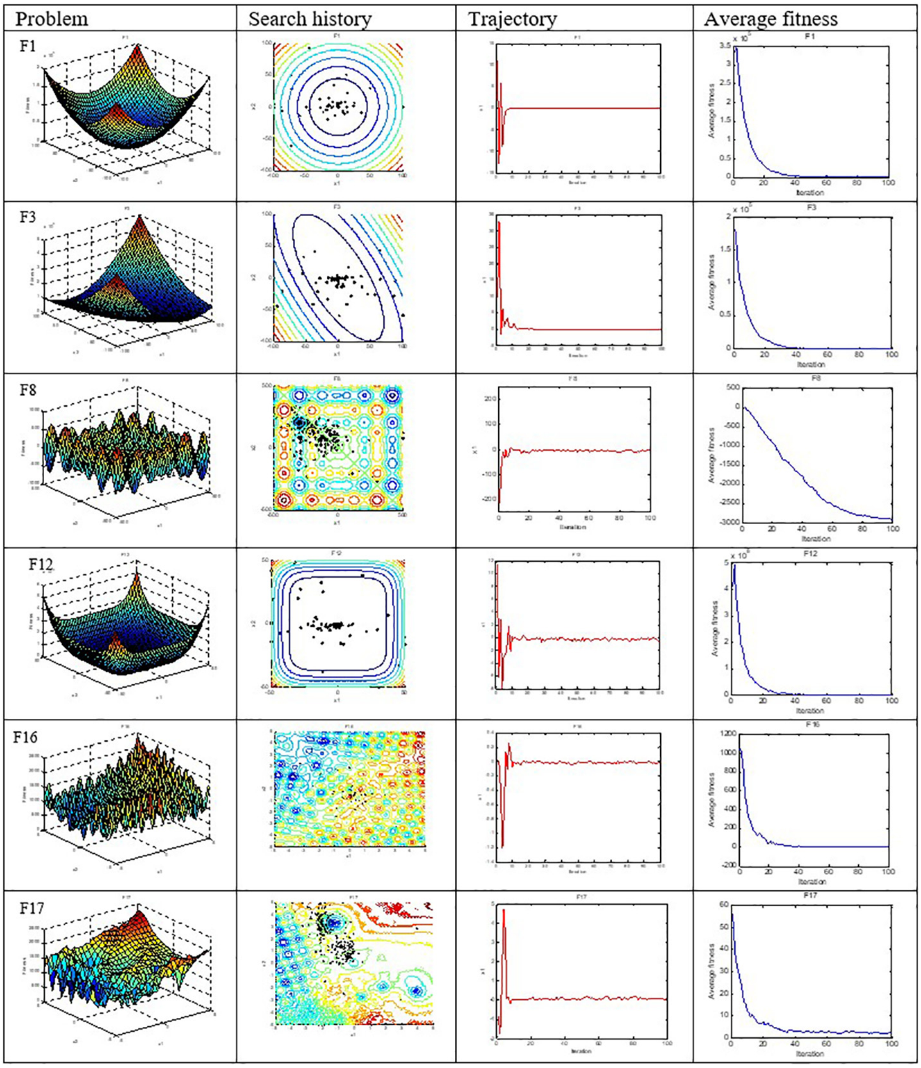

Figure 8.
Search history, trajectory and average fitness for the selected problems

selected benchmark functions. Population diversity was used to measure the exploration and exploitation (Cheng *et al.*, 2014; Salleh *et al.*, 2018). As shown in Figure 9, the TTA maintained the exploration and exploitation ratio at an average of 26%: 74%. These plots clearly proved that the exploration occurred until the end of the iteration.

The average fitness measures the mean for all solutions in every iteration. The plot of average fitness in Figure 8 provides evidence that the overall population fitness is improved throughout the iterations. This finding showed the effectiveness of exploitation feature in the TTA, where it improved the player position for the entire population. The convergence rate for average fitness was consistent with Figure 7, where the TTA experienced a fast convergence rate.

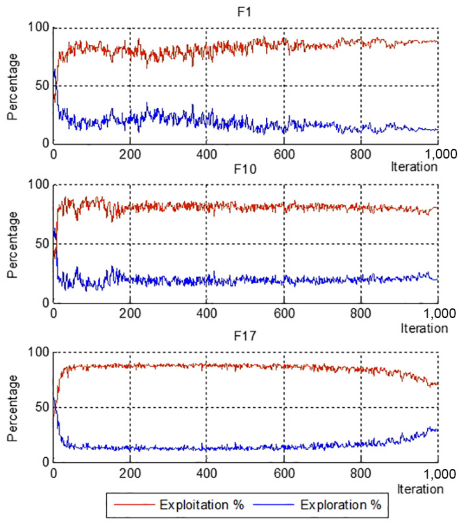

Figure 9.
Exploration and exploitation for the selected problems

4. Engineering design optimisation

In this section, TTA is implemented to optimise classical engineering design problems. In this work, five engineering design problems with constraints are considered. The problems are welded beam, spring, pressure vessel, gear train and three-bar designs. Apart from the variable bounds, these problems have several inequality constraints to be fulfiled. A death penalty approach is used to deal with these constraints. In this approach, an extremely large penalty will be given to the fitness when any of the constraints is violated.

4.1 Welded beam design

This problem aims to design a welded beam with minimum cost (Ragsdell and Phillips, 1976). The welded beam design in Figure 10 is subjected to shear stress (τ), bending stress (σ), buckling

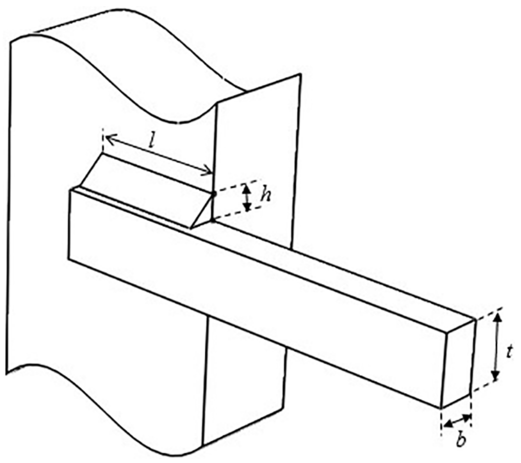

Figure 10.
Welded beam design

load (P_c) and beam deflection (δ). Four design variables need to be optimised in this problem. The variables comprise weld thickness (h), length of the bar being welded (l), bar height (t) and bar thickness (b). For this problem, seven inequality constraints, namely g_1 to g_7, are found:

$$\text{Minimize} f(X) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$$

$$\text{Subjected to}: \; g_1(X) = \tau(X) - \tau_{max} \leq 0$$

$$g_2(X) = \sigma(X) - \sigma_{max} \leq 0$$

$$g_3(X) = x_1 - x_4 \leq 0$$

$$g_4(X) = 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0$$

$$g_5(X) = 0.125 - x_1 \leq 0$$

$$g_6(X) = \delta(X) - \delta_{max} \leq 0$$

$$g_7(X) = P - P_c(X) \leq 0$$

$$0.1 \leq x_i \leq 2.0, \;\; i = 1, \;\; 4$$

$$0.1 \leq x_i \leq 10.0, \;\; i = 2, \;\; 3$$

$$\tau(X) = \sqrt{(\tau')^2 + 2\tau'\tau''}, \;\; \tau' = \frac{P}{\sqrt{2}x_1x_2}, \;\; \tau'' = \frac{MR}{J}, \;\; M = P\left(L + \frac{x_2}{2}\right)$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}, \;\; J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$

$$\sigma(X) = \frac{6PL}{x_4x_3^2}, \;\; \delta(X) = \frac{6PL^3}{Ex_3^2x_4}$$

$$P_c(X) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$$P = 6000\ \text{lb},\ L = 14\ \text{in.},\ E = 3\times10^6\ \text{psi},\ G = 12\times10^6\ \text{psi}$$

$$\tau_{max} = 13,600\ \text{psi},\ \sigma_{max} = 30,000\ \text{psi},\ \delta_{max} = 0.25\ \text{in.}$$

The welded beam design problem has been solved by many optimisation algorithms in the literature. Table 8 presents the best optimisation results of welded beam design using the TTA. The obtained results were compared with the results published in various journals using different algorithms. For this problem, the results were compared with the GA, Co-Evolutionary PSO (CPSO), Improved ACO (IACO), MFO, Grey Wolf Optimiser (GWO), Artificial Bee Colony (ABC), WOA and MVO. The TTA performance was compared with the latest algorithms, including Cohort Intelligent (CI), Sailfish Optimiser (SFO), HHO and Marine Predators Algorithm (MPA). The results indicated that the TTA obtained the best solution compared with the comparison algorithms with a fitness value of 1.695247. The result showed the superior performance of the TTA over major algorithms in optimizing real problems.

4.2 Tension/compression spring design

The tension/compression spring design (Figure 11) aims to minimise the weight, subjected to shear stress, surge frequency and minimum deflection (Coello, 2000). Three continuous design variables, namely, wire diameter (*d*), mean coil diameter (*D*) and number of active coils (*N*), are used in this problem. This problem is another popular engineering design problem used in optimisation. Several algorithms, including GA, PSO, ABC, MFO and Improved Harmony Search (IHS), are used in the literature to optimise this problem:

$$\text{Minimize}\, f(X) = (x_3 + 2)x_2x_1^2$$

$$\text{Subjected to}:\ g_1(X) = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0$$

Algorithm	x_1 (*h*)	x_2 (*l*)	x_3 (*t*)	x_4 (*b*)	*f*(X)
GA (Carlos and Coello, 2000)	0.1829	4.0483	9.3666	0.2059	1.8242
CPSO (He and Wang, 2007)	0.202369	3.544214	9.04821	0.205723	1.728024
IACO (Kaveh and Talatahari, 2010)	0.2057	3.471131	9.036683	0.205731	1.724918
MFO (Mirjalili, 2015a)	0.2057	3.4703	9.0364	0.2057	1.72452
GWO (Mirjalili *et al.*, 2014)	0.205676	3.478377	9.03681	0.205778	1.72624
ABC (Akay and Karaboga, 2012)	0.20573	3.470489	9.036624	0.20573	1.724852
WOA (Mirjalili and Lewis, 2016)	0.205396	3.484293	9.037426	0.206276	1.730499
MVO (Mirjalili *et al.*, 2016)	0.205463	3.473193	9.044502	0.205695	1.72645
CI (Shastri *et al.*, 2019)	0.2057	3.4704	9.0366	0.2057	1.7248
SFO (Shadravan *et al.*, 2019)	0.2038	3.6630	9.0506	0.2064	1.73231
HHO (Heidari *et al.*, 2019)	0.204039	3.531061	9.027463	0.206147	1.73199057
MPA (Faramarzi *et al.*, 2020)	0.205728	3.470509	9.036624	0.205730	1.724853
TTA	0.205727	3.253169	9.036624	0.20573	*1.695247*

Table 8. Optimisation results of welded beam design

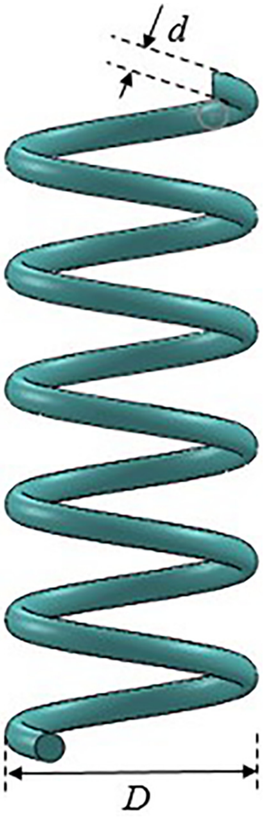

Figure 11.
Tension/compression spring

$$g_2(X) = \frac{4x_2^2 - x_1x_2}{12566\left(x_2x_1^3 - x_1^4\right)} + \frac{1}{5108x_1^2} - 1 \le 0$$

$$g_3(X) = 1 - \frac{140.45x_1}{x_2^2x_3} \le 0$$

$$g_4(X) = \frac{x_1 + x_2}{1.5} - 1 \le 0$$

$$0.05 \le x_1 \le 2.00, \;\; 0.25 \le x_2 \le 1.30, \;\; 2.0 \le x_1 \le 15.0$$

Table 9 shows the optimisation results of tension/compression spring adopted from different sources. The best fitness was obtained by the IACO algorithm with a fitness of 0.012643. The TTA together with the ABC and MPA ranked second. The three algorithms shared a similar fitness value of 0.012665 but with different design variable values. In this problem, the TTA outperformed major algorithms such as GA, CPSO, IACO and MFO.

4.3 Pressure vessel

The pressure vessel is another popular problem in engineering design optimisation, as displayed in Figure 12 (Sandgren, 1990). This problem aims to minimise the overall cost of the vessel. However, the design must consider few constraints. The design variables for this problem are shell thickness ($T_s = x_1$), head thickness ($T_h = x_2$), inner radius ($R = x_3$) and cylindrical section length ($L = x_4$). The values of x_1 and x_2 variables range from 0.0625 to 6.1875, with an increment of 0.0625. x_3 and x_4 are continuous variables ranging from 10 to 200:

$$\text{Minimize } f(X) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

$$\text{Subjected to : } g_1(X) = -x_1 + 0.0193x_3 \leq 0$$

$$g_2(X) = -x_3 + 0.00954x_3 \leq 0$$

$$g_3(X) = -\pi x_3^2x_4 + 1{,}296{,}000 \leq 0$$

$$g_4(X) = -x_4 + 240 \leq 0$$

$$1 \times 0.0625 \leq x_1, x_2 \leq 99 \times 0.0625$$

Algorithm	x_1 (*d*)	x_2 (*D*)	x_3 (*N*)	*f*(X)
GA (Carlos and Coello, 2000)	0.05148	0.351661	11.6322	0.012705
CPSO (He and Wang, 2007)	0.051728	0.357644	11.24454	0.012675
IHS (Akay and Karaboga, 2012)	0.05115438	0.34987116	12.0764321	0.0126706
IACO (Kaveh and Talatahari, 2010)	0.051865	0.3615	11	*0.012643*
MFO (Mirjalili, 2015a)	0.051994	0.364109	10.86842	0.012667
GWO (Mirjalili *et al.*, 2014)	0.05169	0.356737	11.28885	0.012666
ABC (Akay and Karaboga, 2012)	0.051749	0.358179	11.20376	0.012665
WOA (Mirjalili and Lewis, 2016)	0.051207	0.345215	12.00403	0.012676
CI (Shastri *et al.*, 2019)	0.05157	0.35418	11.43864	0.012667
HHO (Heidari *et al.*, 2019)	0.051796393	0.359305355	11.138859	0.012665443
MPA (Faramarzi *et al.*, 2020)	0.051724477	0.35757003	11.2391955	0.012665
TTA	0.051669	0.356226	11.31784	0.012665

Table 9. Optimisation results of tension/compression spring design

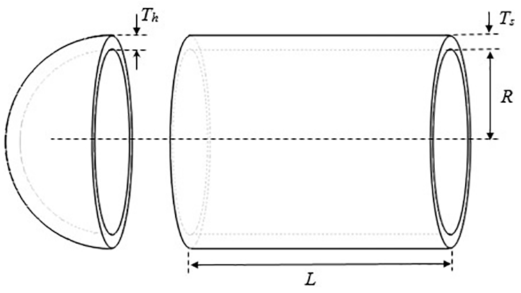

Figure 12. Pressure vessel

$$10 \leq x_3, x_4 \leq 200$$

As shown in the results in Table 10, the best fitness for pressure vessel (6,051.564) was obtained by the GWO algorithm (Mirjalili *et al.*, 2014). For this problem, the TTA ranked second with a fitness of 6,059.714. The result obtained by the TTA was exactly the same with MFO, CI and ABC. In this problem, x_1 and x_2 were treated as discrete variables, making it incomparable with other studies that assumed x_1 and x_2 as continuous variables.

4.4 Three-bar truss design

This problem consists of a planar truss with three bars, as depicted in Figure 13 and has been introduced with the objective to minimise the volume of the truss (Gandomi *et al.*, 2013; Yang and Gandomi, 2012). Two design variables, namely, cross-sections of A1 (or x_1) and A2 (or x_2), are involved in this problem. This problem has three nonlinear inequality constraints:

$$\text{Minimize} f(X) = \left(2\sqrt{2}x_1 + x_2\right)*l$$

Table 10.
Optimisation results of pressure vessel design

Algorithm	x_1 (T_s)	x_2 (T_h)	x_3 (R)	x_4 (L)	f(X)
GA (Carlos and Coello, 2000)	0.8125	0.4345	40.3239	200	6288.745
CPSO (He and Wang, 2007)	0.8125	0.4375	42.09127	176.7465	6061.078
IACO (Kaveh and Talatahari, 2010)	0.8125	0.4375	42.09835	176.6378	6059.726
MFO (Mirjalili, 2015a)	0.8125	0.4375	42.09845	176.6366	6059.714
GWO (Mirjalili *et al.*, 2014)	0.8125	0.4345	42.08918	176.7587	*6051.564*
ABC (Akay and Karaboga, 2012)	0.8125	0.4375	42.09845	176.6366	6059.714
WOA (Mirjalili and Lewis, 2016)	0.8125	0.4375	42.09827	176.639	6059.741
MVO (Mirjalili *et al.*, 2016)	0.8125	0.4375	42.09074	176.7387	6060.807
CI (Shastri *et al.*, 2019)	0.81249	0.4375	42.09844	176.636	6059.714
MPA (Faramarzi *et al.*, 2020)	0.8125	0.4375	42.098445	176.636607	6059.7144
TTA	0.8125	0.4375	42.09845	176.6366	6059.714

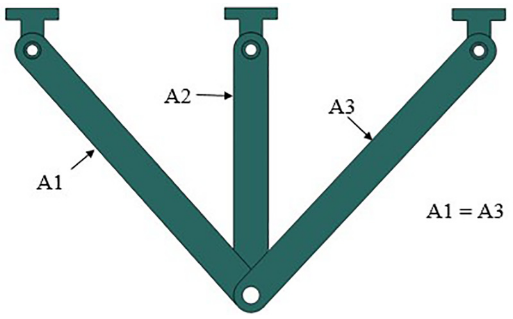

Figure 13.
Three-bar truss

$$\text{Subjected to}: g_1(X) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0$$

$$g_2(X) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0$$

$$g_3(X) = \frac{1}{\sqrt{2}x_2 + x_1}P - \sigma \leq 0$$

$$l = 100\ cm,\ \ P = 2\ kN/cm^2,\ \ \sigma = 2\ kN/cm^2$$

$$0 \leq x_1, x_2 \leq 1$$

The TTA's performance in Table 11 was compared with MFO and MVO. The following algorithms, including, Cuckoo Search Algorithm (CSA), Hybridised PSO with Differential Evolution (PSO-DE), Flower Pollination Algorithm (FPA), ALO, Bat Algorithm (BA) and HHO, were also used. The results of the three-bar truss design in Table 11 indicated that the TTA obtained a roughly similar performance with other major algorithms. However, the results showed that the TTA's fitness outperformed other algorithms, except PSO-DE, when they were closely inspected up to eight decimals. This result showed the effectiveness of the TTA in optimising real-world problems.

4.5 Cantilever beam design

Cantilever beam design is a civil engineering problem that aims to minimise the total weight of the beam (Gandomi *et al.*, 2013; Yang and Gandomi, 2012). The cantilever beam consists of five hollow square blocks with different sizes, as shown in Figure 14. The first square block is fixed at the end of Block 1 and a vertical force (F) is applied to the end of Block 5. Five design variables, which are the width/height of five different blocks, are used in this problem:

$$\text{Minimize} f(X) = 0.6224(x_1 + x_2 + x_3 + x_4 + x_5)$$

Algorithm	x_1	x_2	*f*(X)
MFO (Mirjalili, 2015a)	0.788245	0.409467	263.89597972
CSA (Gandomi *et al.*, 2013)	0.78867	0.40902	263.97156204
MVO (Mirjalili *et al.*, 2016)	0.788603	0.408453	263.89585070
PSO-DE (Liu *et al.*, 2010)	0.788675	0.408248	*263.89584337*
FPA (Nigdeli *et al.*, 2016)	0.78853	0.40866	263.89596407
ALO (Mirjalili, 2015b)	0.788663	0.408283	263.89584351
BA (Yang and Gandomi, 2012)	0.78863	0.40838	263.89624833
SFO (Shadravan *et al.*, 2019)	0.7884562	0.40886831	263.89592128
HHO (Heidari *et al.*, 2019)	0.788662816	0.408283133832900	263.89584348
TTA	0.7886711002	0.4082597017	263.89584340

Table 11. Optimisation results of three-bar truss design

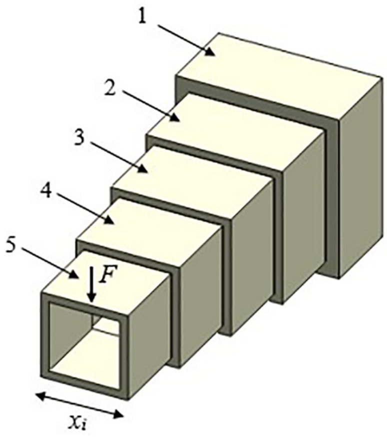

Figure 14.
Cantilever beam

$$\text{Subjected to}: \; g(X) = \frac{61}{x_1^3} + \frac{27}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0$$

$$0.01 \leq x_1, x_2, x_3, x_4, x_5 \leq 100$$

The results of cantilever beam design optimisation are presented in Table 12. As shown in the best fitness in the last column, the TTA result clearly outperformed all other algorithms, including the latest algorithm, Social Mimic Optimisation (SMO) (Balochian and Baloochian, 2019) with a fitness value of 1.30326. The results of engineering design problems clearly indicated the ability of the TTA to optimise real-world problems with constraints, including the discrete parameter in the pressure vessel problem. The remarkable TTA performance obtained in the welded beam and cantilever designs explained the superiority of the proposed algorithm.

5. Discussions and conclusions

This study presented a novel metaheuristic inspired by a football-playing style known as tiki-taka. The tiki-taka style is associated with short passing, player positioning and possession maintaining. The proposed TTA adopted the short passing and player

Table 12.
Optimisation results of cantilever beam design

Algorithm	x_1	x_2	x_3	x_4	x_5	*f*(X)
MFO (Mirjalili, 2015a)	5.984872	5.316727	4.497333	3.513616	2.16162	1.339988086
MVO (Mirjalili *et al.*, 2016)	6.02394	5.306011	4.495011	3.496022	2.152726	1.3399595
ALO (Mirjalili, 2015b)	6.01812	5.31142	4.48836	3.49751	2.158329	1.33995
FPA (Nigdeli *et al.*, 2016)	6.0202	5.3082	4.5042	3.4856	2.1557	1.33997
LAPO (Nematollahi *et al.*, 2017)	6.012436	5.314871	4.495914	3.499394	2.151155	1.336521415
CSA (Gandomi *et al.*, 2013)	6.0089	5.3049	4.5023	3.5077	2.1504	1.33999
SMO (Balochian and Baloochian, 2019)	5.78691656	5.051692965	4.227690161	3.739055332	2.25739270	1.31095
TTA	5.969892	4.869666	4.474679	3.4855	2.139488	*1.30326*

positioning characteristics during optimisation. Compared with existing football-inspired algorithms, this algorithm is the first to mimic football tactics in its model.

The results of unimodal benchmark functions where the TTA performed best in five out of seven problems showed that the TTA had a remarkable exploitation ability. The player updating procedure characterised by short passing to a nearby player is a good strategy to collect information from all candidate solutions. In the multimodal problem, the TTA maintained its performance in the top three ranks in five out of six benchmark functions. These results explained the exploration capability of the TTA. The results showed the exploration and local optimum avoidance capabilities of the TTA because multimodal problems consisted of hundreds of local optima. The effective exploration capability of the TTA relied on multiple key players as leaders and the ball losing mechanism in the algorithm. The best TTA performance is found in the composite benchmark functions with five problems and ranks second in one problem. The results substantiated the TTA's capability to balance between exploitation and exploration in solving real-world problems.

The head-to-head comparison explained the capability of the TTA to maintain its performance at different difficulty levels. The percentage of TTA cases with better and equal performance sufficiently justified its superiority compared with well-established, highly cited and recent metaheuristics. Subsequent analysis on the characteristic of TTA revealed that this algorithm maintained the population diversity throughout the optimisation. At the same time, the proposed algorithm maintained the exploration and exploitation ratio at 26%:74%. The results of engineering design problem justified the capability of the TTA to optimise real-life problems with multiple constraints. Although a simple constraint handling approach (i.e. death penalty) was used, the proposed algorithm found the best solution in three problems and ranked second in the two remaining problems. The obtained results in benchmark functions and engineering designs problem clearly explained the superiority of the TTA.

The uniqueness of TTA depends on short passing strategy in the player position updating mechanism. In comparison with other metaheuristics, the TTA exploits the nearby solution in addition to a set of leading solutions (key players) to determine the next solution position. This feature makes every single solution contributes to the exploitation of other solutions. At the same time, this strategy allows the proposed algorithm to maintain the diversity because the influence of the nearby solution is higher than the leading solutions. The unsuccessful pass concept promotes the exploration in the search space. The ball with unsuccessful pass will be delivered away from original position to explore the new region.

Although the TTA exhibited exceptional performance, this algorithm had few limitations, as observed from the computational experiments. The first drawback is that the TTA's performance easily drops with the change in the coefficient. The experimental results using Taguchi design show that the changes in coefficient level significantly contribute to the overall algorithm performance. In future applications, the optimum coefficient value should be identified for a specific problem before making a decision. The second limitation is the exploration mechanism, namely, the unsuccessful ball pass, of the proposed algorithm. The ball will be reflected behind the player's team when an unsuccessful pass occurs. This strategy makes the exploration activity to concentrate on a particular region in the search space. The overall TTA's performance can be enhanced with an improved strategy to ensure the exploration can be evenly distributed.

References

Abdel-Basset, M., Abdel-Fatah, L., (2018), and., and Sangaiah, A.K. "Metaheuristic algorithms: a comprehensive review", in *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*, Sangaiah, A. K., Sheng, M. and Zhang, Z., (Eds), Academic Press, pp. 185-231.

Abdullah, J.M. and Ahmed, T. (2019), "Fitness dependent optimizer: inspired by the bee swarming reproductive process", *IEEE Access*, Vol. 7, pp. 43473-43486, doi: 10.1109/ACCESS.2019.2907012.

Akay, B. and Karaboga, D. (2012), "Artificial bee colony algorithm for large-scale problems and engineering design optimization", *Journal of Intelligent Manufacturing*, Vol. 23 No. 4, pp. 1001-1014, doi: 10.1007/s10845-010-0393-4.

Akçay, Y., Li, H. and Xu, S.H. (2006), "Greedy algorithm for the general multidimensional knapsack problem", *Annals of Operations Research*, Vol. 150 No. 1, pp. 17, doi: 10.1007/s10479-006-0150-4.

Arora, S. and Singh, S. (2019), "Butterfly optimization algorithm: a novel approach for global optimization", *Soft Computing*, Vol. 23 No. 3, pp. 715-734, doi: 10.1007/s00500-018-3102-4.

Balochian, S. and Baloochian, H. (2019), "Social mimic optimization algorithm and engineering applications", *Expert Systems with Applications*, Vol. 134, pp. 178-191, doi: 10.1016/j.eswa.2019.05.035.

Carlos, A. and Coello, C. (2000), "Constraint-handling using an evolutionary multiobjective optimization technique", *Civil Engineering Systems*, Vol. 17 No. 4, pp. 319-346, doi: 10.1080/02630250008970288.

Cheng, S., Shi, Y., Qin, Q., Zhang, Q. and Bai, R. (2014), "Population diversity maintenance in brain storm optimization algorithm", *Journal of Artificial Intelligence and Soft Computing Research*, Vol. 4 No. 2, pp. 83-97.

Coello, C.A.C. (2000), "Use of a self-adaptive penalty approach for engineering optimization problems", *Computers in Industry*, Vol. 41 No. 2, pp. 113-127, doi: 10.1016/S0166-3615(99)00046-9.

Dhiman, G. and Kumar, V. (2017), "Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications", *Advances in Engineering Software*, Vol. 114, pp. 48-70, doi: 10.1016/j.advengsoft.2017.05.014.

Dokeroglu, T., Sevinc, E., Kucukyilmaz, T. and Cosar, A. (2019), "A survey on new generation metaheuristic algorithms", *Computers and Industrial Engineering*, Vol. 137, pp. 106040, doi: 10.1016/j.cie.2019.106040.

Dorigo, M., Maniezzo, V. and Colorni, A. (1996), "The ant system: optimization by a colony of cooperating agents", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 26 No. 1, pp. 1-13.

Fadakar, E. and Ebrahimi, M. (2016), "A new metaheuristic football game inspired algorithm", in *2016 1st Conference on Swarm Intelligence and Evolutionary Computation (CSIEC)*, pp. 6-11, doi: 10.1109/CSIEC.2016.7482120.

Faramarzi, A., Heidarinejad, M., Mirjalili, S. and Gandomi, A.H. (2020), "Marine predators algorithm: a nature-inspired metaheuristic", *Expert Systems with Applications*, Vol. 152, pp. 113377, doi: 10.1016/j.eswa.2020.113377.

Gandomi, A.H., Yang, X.-S. and Alavi, A.H. (2013), "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems", *Engineering with Computers*, Vol. 29 No. 1, pp. 17-35, doi: 10.1007/s00366-011-0241-y.

Hatamzadeh, P. and Khayyambashi, M.R. (2012), "Football optimization: an algorithm for optimization inspired by football game", in *11th Intelligent Systems Conference*, Kharazmi University: Tehran, p. 261.

Hayward, B. (2015), "The evolution of barcelona's tiki taka", available at: www.goal.com/en/news/12/spanish-football/2015/09/29/15804882/the-evolution-of-barcelonas-tiki-taka (accessed 27 May 2019).

He, Q. and Wang, L. (2007), "An effective co-evolutionary particle swarm optimization for constrained engineering design problems", *Engineering Applications of Artificial Intelligence*, Vol. 20 No. 1, pp. 89-99, doi: 10.1016/J.ENGAPPAI.2006.03.003.

Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M. and Chen, H. (2019), "Harris hawks optimization: algorithm and applications", *Future Generation Computer Systems*, Vol. 97, pp. 849-872, doi: 10.1016/j.future.2019.02.028.

Jaddi, N.S., Alvankarian, J. and Abdullah, S. (2017), "Kidney-inspired algorithm for optimization problems", *Communications in Nonlinear Science and Numerical Simulation*, Vol. 42, pp. 358-369, doi: 10.1016/j.cnsns.2016.06.006.

Jain, M., Singh, V. and Rani, A. (2019), "A novel nature-inspired algorithm for optimization: Squirrel search algorithm", *Swarm and Evolutionary Computation*, Vol. 44, pp. 148-175, doi: 10.1016/j.swevo.2018.02.013.

Kashan, A.H. (2009), "League championship algorithm: a new algorithm for numerical function optimization", in *2009 International Conference of Soft Computing and Pattern Recognition*, pp. 43-48, 10.1109/SoCPaR.2009.21.

Kaveh, A. and Bakhshpoori, T. (2016), "Water evaporation optimization: a novel physically inspired optimization algorithm", *Computers and Structures*, Vol. 167, pp. 69-85, doi: 10.1016/j.compstruc.2016.01.008.

Kaveh, A. and Dadras, A. (2017), "A novel Meta-heuristic optimization algorithm: thermal exchange optimization", *Advances in Engineering Software*, Vol. 110, pp. 69-84, doi: 10.1016/j.advengsoft.2017.03.014.

Kaveh, A. and Talatahari, S. (2010), "An improved ant colony optimization for constrained engineering design problems", *Engineering Computations*, Vol. 27 No. 1, pp. 155-182, doi: 10.1108/02644401011008577.

Kennedy, J. and Eberhart, R. (1995), "Particle swarm optimization", in *Proceedings of International Conference on Neural Networks*, Vol. 4, pp. 1942-1948, 10.1109/ICNN.1995.488968.

Khaji, E. (2014), "Soccer league optimization: a heuristic algorithm inspired by the football system in European countries", in *3rd Recent Inovations Conference on Industrial Engineering and Mechanical Engineering*, p. 82.

Li, M.D., Zhao, H., Weng, X.W. and Han, T. (2016), "A novel nature-inspired algorithm for optimization: virus colony search", *Advances in Engineering Software*, Vol. 92, pp. 65-88, doi: 10.1016/j.advengsoft.2015.11.004.

Liang, J.-J., Suganthan, P.N. and Deb, K. (2005), "Novel composition test functions for numerical global optimization", in *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005*, pp. 68-75.

Liu, H., Cai, Z. and Wang, Y. (2010), "Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization", *Applied Soft Computing*, Vol. 10 No. 2, pp. 629-640, doi: 10.1016/j.asoc.2009.08.031.

Meng, X.-B., Gao, X.Z., Lu, L., Liu, Y. and Zhang, H. (2016), "A new bio-inspired optimisation algorithm: bird swarm algorithm", *Journal of Experimental and Theoretical Artificial Intelligence*, Vol. 28 No. 4, pp. 673-687, doi: 10.1080/0952813X.2015.1042530.

Mirjalili, S. (2015a), "Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm", *Knowledge-Based Systems*, Vol. 89, pp. 228-249, doi: 10.1016/j.knosys.2015.07.006.

Mirjalili, S. (2015b), "The ant lion optimizer", *Advances in Engineering Software*, Vol. 83, pp. 80-98, doi: 10.1016/j.advengsoft.2015.01.010.

Mirjalili, S. (2016), "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems", *Neural Computing and Applications*, Vol. 27 No. 4, pp. 1053-1073, doi: 10.1007/s00521-015-1920-1.

Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H. and Mirjalili, S.M. (2017), "Salp swarm algorithm: a bio-inspired optimizer for engineering design problems", *Advances in Engineering Software*, Vol. 114, pp. 163-191, 10.1016/j.advengsoft.2017.07.002.

Mirjalili, S. and Lewis, A. (2016), "The whale optimization algorithm", *Advances in Engineering Software*, Vol. 95, pp. 51-67, doi: 10.1016/j.advengsoft.2016.01.008.

Mirjalili, S., Mirjalili, S.M. and Hatamlou, A. (2016), "Multi-verse optimizer: a nature-inspired algorithm for global optimization", *Neural Computing and Applications*, Vol. 27 No. 2, pp. 495-513, doi: 10.1007/s00521-015-1870-7.

Mirjalili, S., Mirjalili, S.M. and Lewis, A. (2014), "Grey wolf optimizer", *Advances in Engineering Software*, Vol. 69, pp. 46-61, doi: 10.1016/j.advengsoft.2013.12.007.

Nematollahi, A.F., Rahiminejad, A. and Vahidi, B. (2017), "A novel physical based Meta-heuristic optimization method known as lightning attachment procedure optimization", *Applied Soft Computing*, Vol. 59, pp. 596-621, doi: 10.1016/j.asoc.2017.06.033.

Nigdeli, S.M., Bekdacs, G., (2016), and., and Yang, X.-S. "Application of the flower pollination algorithm in structural engineering", " in *Metaheuristics and Optimization in Civil Engineering*, Yang, X.-S., Bekda\cs, G. and Nigdeli, S. M., (Eds), Springer International Publishing, Cham, pp. 25-42.

O. Maciel, C., Cuevas, E., Navarro, M.A., Zaldívar, D. and Hinojosa, S. (2020), "Side-Blotched lizard algorithm: a polymorphic population approach", *Appl. Soft Comput. J*, Vol. 88, doi: 10.1016/j.asoc.2019.106039.

Osaba, E., Diaz, F., (2013), and and Onieva, E. "A novel Meta-heuristic based on soccer concepts to solve routing problems", in *Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation*, pp. 1743-1744, 10.1145/2464576.2480776.

Patel, V.K. and Savsani, V.J. (2015), "Heat transfer search (HTS): a novel optimization algorithm", *Information Sciences*, Vol. 324, pp. 217-246, doi: 10.1016/j.ins.2015.06.044.

Purnomo, H.D. and Wee, H.-M. (2013), "Soccer game optimization: an innovative integration of evolutionary algorithm and swarm intelligence algorithm", " in *Meta-Heuristics Optimization Algorithms in Engineering, Business, Economics, and Finance*, IGI Global, pp. 386-420.

Ragsdell, K.M. and Phillips, D.T. (1976), "Optimal design of a class of welded structures using geometric programming", *Journal of Engineering for Industry*, Vol. 98 No. 3, pp. 1021-1025.

Razmjooy, N., Khalilpour, M. and Ramezani, M. (2016), "A new Meta-Heuristic optimization algorithm inspired by FIFA world cup competitions: theory and its application in PID designing for AVR system", *Journal of Control, Automation and Electrical Systems*, Vol. 27 No. 4, pp. 419-440, doi: 10.1007/s40313-016-0242-6.

Salleh, M.N.M., Hussain, K., Cheng, S., Shi, Y., Muhammad, A., Ullah, G. and Naseem, R. (2018), "Exploration and exploitation measurement in swarm-based metaheuristic algorithms: an empirical analysis", in *International Conference on Soft Computing and Data Mining*, pp. 24-32.

Sandgren, E. (1990), "Nonlinear integer and discrete programming in mechanical design optimization", *Journal of Mechanical Design*, Vol. 112 No. 2, pp. 223-229.

Saremi, S., Mirjalili, S. and Lewis, A. (2017), "Grasshopper optimisation algorithm: theory and application", *Advances in Engineering Software*, Vol. 105, pp. 30-47, doi: 10.1016/j.advengsoft.2017.01.004.

Shadravan, S., Naji, H.R. and Bardsiri, V.K. (2019), "The sailfish optimizer: a novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems", *Engineering Applications of Artificial Intelligence*, Vol. 80, pp. 20-34, doi: 10.1016/j.engappai.2019.01.001.

Shams, M., Rashedi, E., Dashti, S.M. and Hakimi, A. (2017), "Ideal gas optimization algorithm", *International Journal of Artificial Intelligence*, Vol. 15 No. 2, pp. 116-130.

Shareef, H., Ibrahim, A.A. and Mutlag, A.H. (2015), "Lightning search algorithm", *Applied Soft Computing*, Vol. 36, pp. 315-333, doi: 10.1016/j.asoc.2015.07.028.

Shastri, A.S., Horat, E.V., Kulkarni, A.J. and Jadhav, P.S. (2019), "Optimization of constrained engineering design problems using cohort intelligence method", in *Proceedings of the 2nd International Conference on Data Engineering and Communication Technology, 2019*, pp. 1-11.

Suganthan, P. N., Hansen, N., Liang, J. J., Deb, K., Chen, Y. P., Auger, A. and Tiwari, S. (2005), "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization", *KanGAL Rep*,

Wang, G.-G., Deb, S., Gao, X.-Z. and Dos Santos Coelho, L. (2016), "A new metaheuristic optimisation algorithm motivated by elephant herding behaviour", *International Journal of Bio-Inspired Computation*, Vol. 8 No. 6, pp. 394-409, doi: 10.1504/IJBIC.2016.081335.

Yang, X. and Gandomi, A.H. (2012), "Bat algorithm: a novel approach for global engineering optimization", *Engineering Computations*, Vol. 29 No. 5, pp. 464-483, doi: 10.1108/02644401211235834.

Yapici, H. and Cetinkaya, N. (2019), "A new Meta-heuristic optimizer: pathfinder algorithm", *Applied Soft Computing*, Vol. 78, pp. 545-568, doi: 10.1016/j.asoc.2019.03.012.

Yazdani, M. and Jolai, F. (2016), "Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm", *Journal of Computational Design and Engineering*, Vol. 3 No. 1, pp. 24-36, doi: 10.1016/j.jcde.2015.06.003.

Yu, J.J.Q. and Li, V.O.K. (2015), "A social spider algorithm for global optimization", *Applied Soft Computing*, Vol. 30, pp. 614-627, doi: 10.1016/j.asoc.2015.02.014.

Zaeimi, M. and Ghoddosian, A. (2020), "Color harmony algorithm: an art-inspired metaheuristic for mathematical function optimization", *Soft Comput*, doi: 10.1007/s00500-019-04646-4.

Zhao, W., Wang, L. and Zhang, Z. (2019), "Atom search optimization and its application to solve a hydrogeologic parameter estimation problem", *Knowledge-Based Systems*, Vol. 163, pp. 283-304, doi: 10.1016/j.knosys.2018.08.030.

Zheng, Y.-J. (2015), "Water wave optimization: a new nature-inspired metaheuristic", *Computers and Operations Research*, Vol. 55, pp. 1-11, doi: 10.1016/j.cor.2014.10.008.

About the author

Mohd Fadzil Faisae Ab. Rashid received PhD from Cranfield University, UK in 2013. He was awarded scholarships to pursue Master and Doctoral degrees from Malaysian Government. Currently, he is an Associate Professor in College of Engineering, Universiti Malaysia Pahang. He is also a Chartered Engineer under the Institution of Mechanical Engineers. His research interests are in engineering optimisation, particularly focus on manufacturing system, metaheuristics and discrete event simulation techniques. Mohd Fadzil Faisae Ab. Rashid can be contacted at: ffaisae@ump.edu.my