JOURNAL of
MATERIALS and MECHATRONICS:A

e-ISSN 2717-8811
JournalMM, 2023, 4(2), 424-445
https://doi.org/10.55546/jmm.1291032

Araştırma Makalesi / Research Article

Improved Tuna Swarm Optimization Algorithm for Engineering Design Problems

Harun GEZİCİ

Kırklareli University, Electronics and Automation Department, Kırklareli, Turkey,
ORCID ID: https://orcid.org/0000-0003-1604-1416, harun.gezici@klu.edu.tr

Geliş/ Received: 02.05.2023; **Kabul / Accepted:** 20.09.2023

ABSTRACT: Tuna Swarm Optimization (TSO) which is developed by being inspired by the hunting strategies of the tuna fish is a metaheuristic optimization algorithm (MHA). TSO is able to solve some optimization problems successfully. However, TSO has the handicap of having premature convergence and being caught by local minimum trap. This study proposes a mathematical model aiming to eliminate these disadvantages and to increase the performance of TSO. The basic philosophy of the proposed method is not to focus on the best solution but on the best ones. The Proposed algorithm has been compared to six current and popular MHAs in the literature. Using classical test functions to have a preliminary evaluation is a frequently preferred method in the field of optimization. Therefore, first, all the algorithms were applied to ten classical test functions and the results were interpreted through the Wilcoxon statistical test. The results indicate that the proposed algorithm is successful. Following that, all the algorithms were applied to three engineering design problems, which is the main purpose of this article. The original TSO has a weak performance on design problems. With optimal costs like 1.74 in welded beam design problem, 1581.47 in speed reducer design problem, and 38.455 in I-beam design problem, the proposed algorithm has been the most successful one. Such a case leads us to the idea that the proposed method of this article is successful for improving the performance of TSO.

Keywords: Tuna Swarm Optimization, Swarm-Based Metaheuristic Algorithm, Engineering Design Problems.

*Sorumlu yazar / Corresponding author: harun.gezici@klu.edu.tr

Bu makaleye atıf yapmak için /To cite this article

Gezici, H. (2023). Improved Tuna Swarm Optimization Algorithm for Engineering Design Problems. Journal of Materials and Mechatronics: A (JournalMM), 4(2), 424-445.

1.INTRODUCTION

Today, real world problems are identified through complex mathematical equations which include many parameters. In the field of optimization, these mathematical equations are named objective functions (Noureddine, 2015). Depending on the kind of the problem, the output of the objective function may be required to be minimum or maximum (Mareli and Twala, 2018). At the same time, these problems have many limitations. These limitations are generally about the interrelation of the parameters in the objective functions. The overall purpose of optimization is to optimally determine the parameters in the objective function under certain limitations (Hashim et al., 2022). In the early stages of optimization studies, gradient descent (GD) methods were used. GD is unlikely to be preferred by researches because of its incapability in solving nonlinear design problems. Besides, for engineering problems with wide search space, computation times are long and they are not able to present optimum solutions (S. Kumar et al., 2023). As a result of such disadvantages of GD, researchers focused on metaheuristic algorithms (MHA) (Feng et al., 2021). Depending on the improvement procedures within their structure, MHAs aim to find the most reasonable result within the most reasonable period of time without scanning the search space. MHAs are classified into four subgroups depending on their source of inspiration. These are

- Evolution-based Algorithms: They are improved by being inspired by the biological behaviours of living creatures. They are based on evolutionary laws like crossover and mutation. Primary evolution-based algorithms are genetic algorithms (Mirjalili, 2019), differential evolution (Deng et al., 2021), genetic programming (F. Zhang et al., 2021), evolutionary strategies (Rosso et al., 2022), and evolutionary programming (Gul et al., 2021).
- Swarm-based Algorithms: They are improved by being inspired by the social behaviours of animals like insects and birds within their group. Particle swarm optimization (PSO) (Gad, 2022), ant colony optimization (Wu et al., 2023), grey wolf optimizer (GWO) (Mirjalili et al., 2014), monarch butterfly optimization (G.-G. Wang et al., 2019), earthworm optimizer (G.-G. Wang et al., 2018), moth search algorithm (G.-G. Wang, 2018), firefly algorithm (V. Kumar and Kumar, 2021), artificial bee colony (Öztürk et al., 2020), bat algorithm (BA) (Y. Wang et al., 2019) and Tuna swarm optimization (Xie et al., 2021) are some examples of MHA in this group.
- Physical-based Algorithms: They are improved through various physic laws. Simulated annealing (Amine, 2019), gravitational search algorithm (Rashedi et al., 2009), nuclear reaction optimization (Wei et al., 2019), water cycle algorithm (Korashy et al., 2019), sine cosine algorithm (SCA) (Abualigah and Diabat, 2021), big bang-big crunch (Mbuli and Ngaha, 2022), black hole (Abdulwahab et al., 2019) and harmony search (Abualigah et al., 2020) are the example of physics-based algorithms.
- Human-based algorithms: They are improved by being inspired by the social behaviours of humans. Teaching-learning-based optimization (Li et al., 2019), social evolution and learning optimization (M. Kumar et al., 2018), group teaching optimizer (Y. Zhang and Jin, 2020), heap-based optimizer (Askari, Saeed, et al., 2020), political optimizer (Askari, Younas, et al., 2020), taboo search (Prajapati et al., 2020), Exchange market algorithm (Jafari et al., 2020) and brain storm optimizer (Xue et al., 2022) are examples of this group.

MHAs are stochastic. They have two search procedures; exploration and exploitation (Raja et al., 2022). During the exploration phase, MHAs determine the promising sections of search space.

During exploitation phase, the determined sections are surveyed in detail. In order to succeed, one of the most significant characteristic of MHAs is the balance between exploitation and exploration (S. Kumar et al., 2023). In some MHAs, this balance is constructed by a probability key that is determined randomly and ranges between 0 and 1 (Ramachandran et al., 2022). Besides, in some MHAs, exploration is performed in early iteration numbers and exploitation is performed in advanced iteration numbers (Xie et al., 2021). Some optimization problems have one local minimum, which is the global minimum at the same time. Some problems have more than one local minimum of which only one is global minimum. Therefore, it is more difficult to solve these problems. Most of the MHAs improved to solve this kind of problems have the disadvantages of premature convergence and being caught by local minimum trap. Moreover, as it is stated by no free lunch theorem, an optimization algorithm cannot solve all optimization problems (Wolpert and Macready, 1997). Hence, researchers tend either to improve new optimizers or to increase the productivity of the available ones.

This study presents an improved version of recently published swarm-based TSO. The proposed algorithm is named Improved TSO (ITSO). It specifically focuses on the premature convergence problem of TSO. In addition to that, local search procedure is improved in order to prevent it from being trapped by local minimum. The improvement is about focusing on the three best points of the search space rather than focusing on the best one. This method eliminates the problems caused by the premature convergence problem by increasing the efficiency of TSO's global search capability. Furthermore, as this method focuses on the three-best solution, it helps to avoid the local minimum trap. The contribution of this study is as follows.

- It introduces a method that allows TSO to escape from premature convergence and local minimum trap.
- It makes betterments in the local search procedure of TSO and presents an improved version of it.
- The proposed algorithm is tested through 10 classical test functions and 3 engineering design problems. The results are evaluated through Wilcoxon test.
- The proposed algorithm is compared to the popular MHAs in the literature.

In earlier studies, TSO is proven to be successful for the solution of optimization problems. However, once it is applied to real world problems, it presents some failures. Hence, researchers conduct works in order to increase its performance. While doing literature review, this study concentrates on works using methods to increase the performance of TSO. In a study on parameter identification of photovoltaic cells, the researchers propose the chaotic variant of TSO (C. Kumar and Magdalin Mary, 2022). In this study, two parameters determined by number of iterations and other randomly determined parameters are assigned through tent chaotic map. The researchers state that the results are more successful than the results of the competitive algorithms. However, this study does not enable us with the information on how other chaotic maps effect the performance of TSO. Besides, no change is made on the mathematical model of TSO. In another study on parameter estimation of photovoltaic batteries, the researchers present a hybrid algorithm made of TSO and differential evolution algorithm (Tan et al., 2022). In order to increase population diversity and convergence efficiency of the proposed algorithm, this study concentrates on strategies such as mutation, crossover factor ranking, and linear reduction of the population. The researchers inform us that the improved algorithm outperforms its competitors. Neither this study makes a change on the mathematical model of TSO. In another study that focuses on estimating the speed of the wind, the modified TSO is hybridized with long short-term memory strategy (Tuerxun et al., 2022). In this study, in order to increase the diversity of the initial population of TSO, tent chaotic map is used. Moreover, TSO is used for image segmentation as well (J. Wang et al., 2022). Like the previous one, this study too uses tent chaotic map in order to increase the diversity of the initial population of TSO. TSO is also used in another study that deals with path planning of autonomous underwater vehicle (Yan et al., 2023). This study presents TSO based on reinforcement learning. It is emphasized that reinforcement learning improves the weak determination of TSO. In another study that regards the problems of TSO's premature convergence and being caught by local optimum trap, the researchers adapt circle chaotic map and levy flight to TSO (W. Wang and Tian, 2022). The Circle chaotic map is used to increase the diversity of the initial population while Levy flight is integrated to mathematical model of TSO. It is reported that these changes increase the performance of TSO. It is highly common to use PID method for controlling the engine revolution speed. TSO is used to determine the PID coefficient (Guo et al., 2022). The researchers indicate that TSO has a better performance compared to conventional methods (Ashraf et al., 2022; Fu and Zhang, 2022).

Having studied all these methods, it is observed that two methods are used in order to increase the performance of TSO. The first one is using chaotic maps to diversify the initial population. The second one is determining the parameters within the mathematical model of TSO in various ways. On the other hand, in some studies, TSO is used as is. In this study, the mathematical model of TSO is changed. The main objective of such a change is to focus not on the best solution but on the best ones. This approach leads TSO to escape from premature convergence and local minimum trap.

The rest of the study is organized as follows. In the second part, TSO is introduced, and information about the proposed algorithm is given. In the third section, computational results are presented. In the last section, the results of the study are evaluated.

2. MATERIALS AND METHODS

2.1 Tuna Swarm Optimization

Tuna is a carnivorous sea creature (Xie et al., 2021). Thanks to their anatomical structure, they can swim really fast. They also have high manoeuvrability. Compared to their sizes, their preys are smaller. Such a case enables preys to swim and manoeuvre faster. Therefore, Tuna fish hunt in groups. Hunting behaviours of tuna fish have two significant strategies (C. Kumar and Magdalin Mary, 2022). The first strategy is the spiral foraging behaviour that directs prey to shallow waters. The second strategy is that each tuna fish swim following another one and form parabolic shapes (Tuerxun et al., 2022). TSO is a swarm-based MHA improved being inspired by these two hunting strategies of tuna (J. Wang et al., 2022). Like other MHAs, the initial population is given randomly (Equation (1)).

$$X\_i=rand(ub-lb)+lb, \quad i=1,2,...,NP \tag{1}$$

Where, X_i is the initial population, ub,lb is the lower and upper bounds of the search space, and NP is the number of the population.

2.1.1 Spiral foraging

Once small fish packs fish encounter predators such as tuna fish, in order to distract the predators, they continuously change their swimming direction. In order to deal with such a challenge, tuna fish generate a spiral area around their prey. While performing this spiral movement, each Tuna fish follows another one before it. It means that there is an exchange of information between tuna fish. The mathematical model of the spiral motion of TSO is given in Equation (1). Some parameters in this equation are calculated by Equation (2), (3), (4) and (5).

$$X_i^{t+1} = \begin{cases} \alpha_1 \cdot (X_{best}^t + \beta \cdot |X_{best}^t - X_i^t|) + \alpha_2 \cdot X_i^t, & i = 1 \\ \alpha_1 \cdot (X_{best}^t + \beta \cdot |X_{best}^t - X_i^t|) + \alpha_2 \cdot X_{i-1}^t, & i = 2, \dots, NP \end{cases} \tag{1}$$

$$\alpha_1 = a + (1 - a) \cdot \frac{t}{t_{max}} \tag{2}$$

$$\alpha_2 = (1 - a) - (1 - a) \cdot \frac{t}{t_{max}} \tag{3}$$

$$\beta = e^{bl} \cdot cos(2\pi b) \tag{4}$$

$$l = e^{3\,cos\left(((t_{max}+1/t)-1)\pi\right)} \tag{5}$$

Where, t is the current iteration, t_{max} is the maximum iteration, and b is a random number evenly distributed between 0 and 1. α_1 and α_2 are weight coefficients controlling the tendency of tuna fish to follow each other. The constant a determines the characteristic of this tendency. i^{th} within the X_i^{t+1} $t+1$ is an individual. β is the equation of spiral movement and l is the parameter of this equation. The most important disadvantage of the spiral movement is the hunting failure of the followed Tuna fish. In such a case, tuna fish continue hunting by choosing a random location. This eases each tuna fish to scan a wider area. It also enables TSO to have a more advanced global search capability. The mathematical model of this hunting strategy is given in Equation (6).

$$X_i^{t+1} = \begin{cases} \alpha_1 \cdot (X_{rand}^t + \beta \cdot |X_{rand}^t - X_i^t|) + \alpha_2 \cdot X_i^t, & i = 1 \\ \alpha_1 \cdot (X_{rand}^t + \beta \cdot |X_{rand}^t - X_i^t|) + \alpha_2 \cdot X_{i-1}^t, & i = 2, \dots, NP \end{cases} \tag{6}$$

Here X_{rand}^t is a randomly picked individual from the group. While some MHAs conduct global searches at the early stages of their searching processes, they conduct local searches at the further stages. While improving TSO, this approach is embraced. Hence, as the number of iteration increases, TSO changes the reference point of spiral movement from random individuals to the best one. The final mathematical model of spiral food searching strategy is as follows (Equation (7)).

$$X_i^{t+1} = \begin{cases} \begin{cases} \alpha_1 \cdot (X_{rand}^t + \beta \cdot |X_{rand}^t - X_i^t|) + \alpha_2 \cdot X_i^t, & i = 1 \\ \alpha_1 \cdot (X_{rand}^t + \beta \cdot |X_{rand}^t - X_i^t|) + \alpha_2 \cdot X_{i-1}^t, & i = 2, \dots, NP \end{cases}, rand \geq \frac{t}{t_{max}} \\ \begin{cases} \alpha_1 \cdot (X_{best}^t + \beta \cdot |X_{best}^t - X_i^t|) + \alpha_2 \cdot X_i^t, & i = 1 \\ \alpha_1 \cdot (X_{best}^t + \beta \cdot |X_{best}^t - X_i^t|) + \alpha_2 \cdot X_{i-1}^t, & i = 2, \dots, NP \end{cases}, rand < \frac{t}{t_{max}} \end{cases} \tag{7}$$

2.1.2 Parabolic foraging

Tuna fish hunt also by having parabolic movements around their preys. This movement could be around the prey regarded as the best solution as well as it could be around itself. The probability of picking either of these two moves is equal. The mathematical model of parabolic motion is given in Equation (8) and (9). The pseudo-code of TSO is given in Algorithm 1.

$$X_i^{t+1} = \begin{cases} X_{best}^t + rand \cdot (X_{best}^t - X_i^t) + TF \cdot p^2 \cdot (X_{best}^t - X_i^t), & rand < 0.5 \\ TF \cdot p^2 \cdot X_i^t, & rand \geq 0.5 \end{cases} \tag{8}$$

$$p = \left(1 - \frac{t}{t_{max}}\right)^{(t/t_{max})} \tag{9}$$

where TF is a random number with a value of 1 or -1.

2.2 Improved Tuna Swarm Optimization

In TSO, the best solution is the location of the fish to be caught. Tuna fish try to approach the prey by following each other. This prevents search space from being scanned efficiently. Especially, TSO's focusing only on the best solution at advanced iteration numbers leads it to be caught by local optimum trap. In order to improve the performance of TSO, this study proposes a new local search procedure that is inspired by GWO.

In order to represent the hierarchical order of the wolves in GWO, Alpha, Beta, and Gamma wolves are identified (Mirjalili et al., 2014). Alfa wolf leads the pack. Beta ones are the best Alpha candidates. Besides, Beta wolves enable communication between the pack and the Alpha wolf. Gamma wolves are tertiary wolves and they assist alpha and beta ones to manage the pack. In GWO, the three best solutions are represented by Alpha, Beta, and Gamma wolves. The positions of all other wolves are updated with respect to the positions of these three wolves (A. Kumar et al., 2017).

There is no evidence presenting that tuna fish have a hierarchical order. However, during hunting, the hunting school could make sudden changes in their directions. This occurs especially when the hunters are close to the prey. This act of the hunters leads us to the idea that local search procedure of TSO could be improved. Depending on the position of the prey, the hunting school has countless probability of changing direction. However, since the number of this probability is so high and it will increase the solution time of the algorithm, it should be limited at a reasonable number. In this study, being inspired by GWO, the three best solution vectors are used to update the location of the tuna fish. The new mathematical model of the proposed ITSO is given in Equation (10) and (11).

$$X_i^{t+1} = \begin{cases} \begin{cases} \alpha_1 \cdot (X_{rand}^t + \beta \cdot |X_{rand}^t - X_i^t|) + \alpha_2 \cdot X_i^t, & i = 1 \\ \alpha_1 \cdot (X_{rand}^t + \beta \cdot |X_{rand}^t - X_i^t|) + \alpha_2 \cdot X_{i-1}^t, & i = 2, \dots, NP \end{cases}, rand \geq \dfrac{t}{t_{max}} \\ \begin{cases} X_1 = \alpha_1 \cdot (X_\alpha^t + \beta \cdot |X_\alpha^t - X_i^t|) + \alpha_2 \cdot X_i^t, \\ X_2 = \alpha_1 \cdot \left(X_\beta^t + \beta \cdot \left|X_\beta^t - X_i^t\right|\right) + \alpha_2 \cdot X_i^t, \\ X_3 = \alpha_1 \cdot \left(X_\gamma^t + \beta \cdot \left|X_\gamma^t - X_i^t\right|\right) + \alpha_2 \cdot X_i^t, \\ \dfrac{X_1 + X_2 + X_3}{3}, \end{cases} \quad i = 1, \dots, NP, rand < \dfrac{t}{t_{max}} \end{cases} \tag{10}$$

$$X_i^{t+1} = \begin{cases} \begin{cases} X_1 = \alpha_1 \cdot (X_\alpha^t + \beta \cdot |X_\alpha^t - X_i^t|) + \alpha_2 \cdot X_i^t, \\ X_2 = \alpha_1 \cdot \left(X_\beta^t + \beta \cdot \left|X_\beta^t - X_i^t\right|\right) + \alpha_2 \cdot X_i^t, \\ X_3 = \alpha_1 \cdot \left(X_\gamma^t + \beta \cdot \left|X_\gamma^t - X_i^t\right|\right) + \alpha_2 \cdot X_i^t, \\ \dfrac{X_1 + X_2 + X_3}{3}, \end{cases} & i = 1, \dots, NP, rand < 0.5 \\ \{TF \cdot p^2 \cdot X_i^t, & i = 1, \dots, NP, rand \geq 0.5 \end{cases} \tag{11}$$

In the equations, X_α, X_β, and X_γ respectively, represent the best, the second best, and the third best solution. X_1, X_2, and X_3 are respectively the location vectors acquired by X_α, X_β, and X_γ. The new position vector is determined by the mean of X_1, X_2, and X_3. Other parameters in the equations are calculated as in section 2.2. The pseudocode of ITSO is given in Algorithm 1.

Algorithm 1. TSO and ITSO pseudocode

TSO pseudocode	**ITSO pseudocode**

Input: *NP:* Population size, t_{max}: maximum iteration
Output: X_{best}: The best individual, f_{best}: Its fitness value
Initialize the random population of tunas ($X_i, i = 1,2,\ldots,NP$) and assign parameters a and z
While ($t < t_{max}$)
  Calculate the fitness values and update X_{best}
  For (each tuna) **do**
  Update α_1, α_2, p using equation (2), (3), (9)
  If ($rand < z$) **then**
 Update X_i^{t+1} using equation (1)
  Else if ($rand \geq z$) **then**
    If ($rand < 0.5$) **then**
      If ($t/t_{max} < rand$) **then**
        Update X_i^{t+1} using equation (6)
      Else if ($t/t_{max} \geq rand$) **then**
        Update X_i^{t+1} using equation (1)
    Else if ($rand \geq 0.5$) **then**
      Update X_i^{t+1} using equation (8)
  $t = t + 1$
Return: X_{best}, f_{best}
	Input: *NP:* Population size, t_{max}: maximum iteration
	Output: X_α: The best individual, f_α: Its fitness value
	Initialize the random population of tunas ($X_i, i = 1,2,\ldots,NP$) and assign parameters a and z
	While ($t < t_{max}$)
	  Calculate the fitness values and update X_α, X_β ve X_γ
	  For (each tuna) **do**
	  Update α_1, α_2, p using equation (2), (3), (9)
	  If ($rand < z$) **then**
	    Update X_i^{t+1} using equation (1)
	  Else if ($rand \geq z$) **then**
	    If ($rand < 0.5$) **then**
	      If ($t/t_{max} < rand$) **then**
	        Update X_i^{t+1} using equation (10)
	      Else if ($t/t_{max} \geq rand$) **then**
	        Update X_i^{t+1} using equation (10)
	    Else if ($rand \geq 0.5$) **then**
	      Update X_i^{t+1} using equation (11)
	  $t = t + 1$
	Return: X_α, f_α

3. RESULTS AND DISCUSSION

This section provides us with the results that the proposed algorithm and competitor algorithms presents from a series of optimization problems. The results are also interpreted through the Wilcoxon statistical test. All algorithms were first applied to 10 well-known classical test functions. Then it was applied to 3 engineering design problems.

3.1 Compared Algorithms and Experimental Setup

The proposed algorithm is an improved version of TSO. Therefore, TSO was chosen as one of the competing algorithms. The proposed algorithm is inspired by GWO while improving. Therefore, GWO is defined as one of the competing algorithms. Moreover, the proposed algorithm is compared to Cuckoo Search (CS), BAT, SCA, and Covariance Matrix Adaptation Evolution Strategy (CMA-ES). The reason why these MHAs are chosen is that they are popular and validated. Besides, these algorithms have been used for the solutions of many optimization problems in medicine, economy, and engineering. The structure of these algorithms is simple and they generate consistent results. All algorithms are coded in Python language. Tests are conducted on a computer with Windows 10 64-bit Professional and 64GB of RAM. The stopping criteria of the algorithms is the number of iterations. The results of 30 independent runs of all algorithms are recorded at every different population number of all the algorithms. All MHAs are sensitive to initial parameters. Hence, for a just comparison, the parameters of the competing algorithms are adjusted in the way that they are given in their original articles. Table 1 presents information about the parameter settings, iteration and population numbers of all algorithms.

Table 1. Parameter settings for algorithms

Algorithm	Parameters	Iteration	Population
CMA-ES (Hansen et al., 2003)	$m_0 = 0, \sigma_0 = 0.5, C_0 = 1, p_0 = 0, s_0 = 0, k = 0, \mu = \lambda/2$	200	50, 100, 200
SCA (Mirjalili, 2016)	$a = 2, r_1 = rand[0, 2\pi]$		
BAT (Gandomi et al., 2013)	$A = 0.5, r = 0.5, Q_{min} = 0, Q_{max} = 2$		
CS (Rajabioun, 2011)	$p_a = 0.25, \alpha = 0.1, \beta = 1.5$		
GWO (Mirjalili et al., 2014)	$a = 2$		
TSO (Xie et al., 2021)	$z = 0.5, a = 0.7$		
ITSO	$z = 0.5, a = 0.7$		

3.2 Classic Test Functions

In this section, in order to evaluate its performance, the proposed algorithm is tested through 10 well-known functions. 6 of these functions (F1-F6) are unimodal and 4 of them (F7-F10) are multimodal functions. Unimodal functions have one global minimum. Hence, it is used to test the local search capability of MHAs. Multimodal functions have more than one local minimum. Only one of these local minimums is the global minimum. This kind of functions are used to examine the global search capability of MHAs and their ability to avoid the local minimum trap. Descriptive information about classical test functions is given in Table 2.

While performing the first experiment, population is set to be 50. In Table 3, all algorithms' the minimum, average and worst results by the classical test functions are given. In addition, the convergence curves of the algorithms are given in Figure 1. In minimum value metric, the proposed algorithm is the most successful one by generating the most successful results in 7 of 10 functions (F1-F4, F7, F9, F10). On the other hand, CMA-ES generates the best results in 3 functions which leads it to be the second most successful algorithm (F5, F6, F8). In mean value metric, the proposed algorithm generates the best results in 8 of 10 functions (F1-F4, F7, F10) while CMA-ES is successful in 2 functions (F5, F6). In the worst value metric, the proposed algorithm happens to be the most successful one by successfully solving 8 functions (F1-F4, F7-F10) while CMA-ES is successful in 2 functions (F5, F6).

Table 2. Description of benchmark functions

Function	Range	Dim	f_{min}
$F_1(z) = \sum_{i=1}^{dim} z_i^2$	[-100, 100]	30	0
$F_2(z) = \sum_{i=1}^{dim} \lvert z_i \rvert + \prod_{i=1}^{dim} \lvert z_i \rvert$	[-10, 10]	30	0
$F_3(z) = \sum_{i=1}^{dim} \left(\sum_{j=1}^{i} z_j \right)^2$	[-100, 100]	30	0
$F_4(z) = max_i\{\lvert z_i \rvert, 1 \le i \le dim\}$	[-100, 100]	30	0
$F_5(z) = \sum_{i=1}^{dim-1} \left[100\left(z_{i+1} - z_i^2\right)^2 + (z_i - 1)^2 \right]$	[-30, 30]	30	0
$F_6(z) = \sum_{i=1}^{dim} i z_i^4 + rand\,[0,1]$	[-1.28, 1.28]	30	0

$F_7(z) = \sum_{i=1}^{11}\left[a_i - \frac{z_1(b_i^2 + b_i z_2)}{b_i^2 + b_i z_3 + z_4}\right]^2$	[-5, 5]	4	$\approx 0{,}0003075$
$F_8(z) = -\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0, 10]	4	$-10{,}1532$
$F_9(z) = -\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0, 10]	4	$-10{,}4028$
$F_{10}(z) = -\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0, 10]	4	$-10{,}536$

Table 3. Results on benchmark functions (Population: 50)

Obj. Func.		**ITSO**	**TSO**	**GWO**	**CMA-ES**	**CS**	**BAT**	**SCA**
F1	Min.	***1.68E-46***	2.99E-35	1.12E-13	3.49E-24	1.20E+03	7.60E+03	8.13E+00
	Mean	***3.12E-32***	1.73E-15	1.35E-12	9.77E-12	1.81E+03	1.54E+04	4.20E+02
	Worst	***8.48E-31***	4.57E-14	7.81E-12	1.86E-10	3.21E+03	3.10E+04	1.93E+03
F2	Min.	***1.72E-24***	2.81E-18	1.40E-08	1.17E-15	3.10E+01	3.19E+01	8.59E-02
	Mean	***5.67E-18***	1.61E-08	5.43E-08	6.93E-07	4.67E+01	2.97E+04	6.75E-01
	Worst	***9.24E-17***	2.70E-07	1.34E-07	4.83E-06	6.72E+01	8.03E+05	2.15E+00
F3	Min.	***4.39E-43***	9.48E-41	2.84E-02	5.64E-12	7.54E+03	1.42E+04	6.87E+03
	Mean	***4.34E-23***	4.27E-11	9.12E-01	7.03E-05	1.23E+04	4.78E+04	1.51E+04
	Worst	***1.25E-21***	1.20E-09	3.46E+00	1.17E-03	1.62E+04	8.11E+04	2.71E+04
F4	Min.	***1.28E-23***	3.35E-16	2.58E-03	1.10E-11	2.89E+01	3.13E+01	2.31E+01
	Mean	***3.36E-17***	2.26E-09	1.17E-02	6.92E-06	3.35E+01	4.61E+01	4.51E+01
	Worst	***7.02E-16***	2.87E-08	3.43E-02	5.16E-05	4.02E+01	6.48E+01	7.56E+01
F5	Min.	4.00E-03	2.82E-03	2.60E+01	***1.21E-05***	1.03E+05	2.93E+06	1.14E+04
	Mean	4.49E+00	8.01E-01	2.72E+01	***1.39E-02***	3.92E+05	1.31E+07	7.41E+05
	Worst	2.87E+01	6.25E+00	2.88E+01	***1.01E-01***	8.01E+05	3.17E+07	2.77E+06
F6	Min.	4.78E-06	2.71E-04	1.43E-04	***8.49E-08***	1.01E+03	8.24E+03	1.49E+01
	Mean	9.44E-03	7.10E-03	5.92E-01	***1.70E-04***	1.81E+03	1.56E+04	4.44E+02
	Worst	1.84E-02	6.85E-02	1.30E+00	***7.88E-04***	3.52E+03	2.63E+04	3.02E+03
F7	Min.	***1.88E-05***	3.03E-05	1.24E-03	5.35E-05	2.62E-01	1.35E+00	3.37E-02
	Mean	***2.65E-04***	1.56E-03	4.69E-03	7.47E-04	5.73E-01	7.36E+00	5.56E-01
	Worst	***8.62E-04***	5.07E-03	8.58E-03	3.42E-03	1.16E+00	1.84E+01	3.19E+00
F8	Min.	-1.02E+01	-1.02E+01	-1.02E+01	***-1.02E+01***	-1.02E+01	-1.02E+01	-4.99E+00
	Mean	***-1.02E+01***	-1.01E+01	-8.32E+00	-6.07E+00	-1.01E+01	-5.57E+00	-2.82E+00

	Worst	***-1.01E+01***	-9.77E+00	-2.29E+00	-5.05E+00	-1.01E+01	-2.63E+00	-8.78E-01
F9	Min.	***-1.04E+01***	-1.04E+01	-1.04E+01	-1.04E+01	-1.04E+01	-1.04E+01	-9.42E+00
	Mean	***-1.04E+01***	-1.04E+01	-1.00E+01	-5.43E+00	-1.02E+01	-6.25E+00	-3.73E+00
	Worst	***-1.03E+01***	-9.99E+00	-5.09E+00	-1.84E+00	-5.09E+00	-1.84E+00	-9.02E-01
F10	Min.	***-1.05E+01***	-1.05E+01	-1.05E+01	-5.13E+00	-1.05E+01	-1.05E+01	-9.11E+00
	Mean	***-1.05E+01***	-1.05E+01	-1.03E+01	-5.04E+00	-1.05E+01	-5.90E+00	-4.49E+00
	Worst	***-1.05E+01***	-9.98E+00	-4.92E+00	-2.42E+00	-1.03E+01	-1.68E+00	-2.00E+00

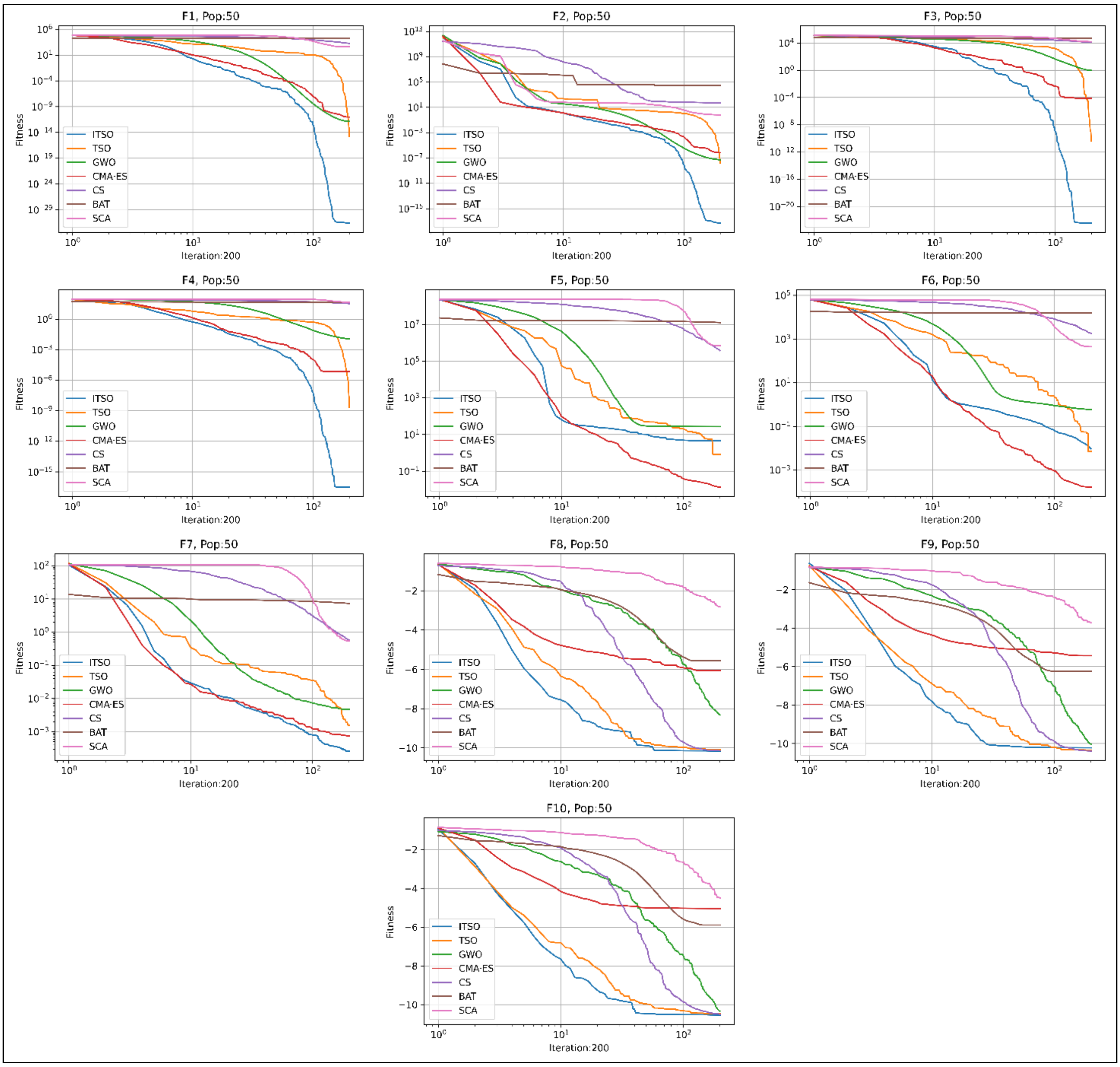

Figure 1. Convergence curve ITSO and other optimizers (Population: 50)

While performing the second experiment, population is set to be 100. In Table 4, all algorithms' the minimum, average and worst results by the classical test functions are given. In addition, the convergence curves of the algorithms are given in Figure 2. In minimum value metric, the proposed algorithm is the most successful one in 6 of 10 functions (F1-F4, F6, F7) while BAT is successful in

2 functions (F8-F10). On the other hand, TSO and CMA-ES are successful in one function each (F5, F9). In the mean value metric, the proposed algorithm is the most successful one in 8 out of 10 functions (F1-F4, F7-F10). CMA-ES is successful in two functions (F5, F6). In the worst value metric, the proposed algorithm is the most successful one by successfully solving 8 functions (F1-F4, F7-F10). CMA-ES is successful in two functions (F5, F6).

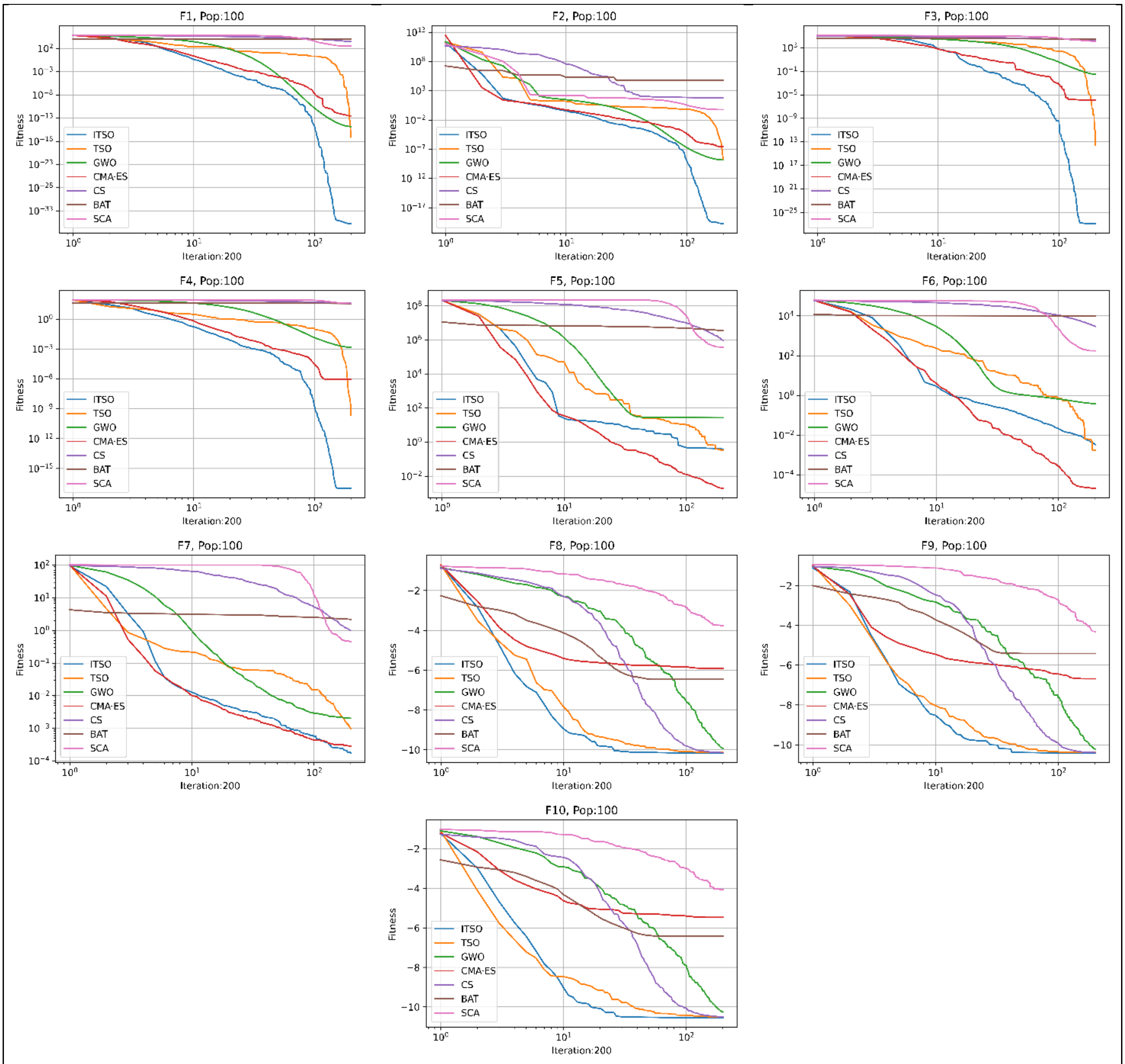

Figure 2. Convergence curve ITSO and other optimizers (Population: 100)

Table 4. Results on benchmark functions (Population: 100)

Obj. Func.		ITSO	TSO	GWO	CMA-ES	CS	BAT	SCA
F1	Min.	***1.32E-50***	4.97E-36	2.13E-16	3.53E-22	1.33E+03	4.74E+03	3.87E+00
	Mean	***2.05E-36***	7.85E-18	1.48E-15	3.06E-13	2.76E+03	8.69E+03	3.24E+02
	Worst	***5.96E-35***	1.33E-16	5.67E-15	5.89E-12	3.96E+03	1.39E+04	3.86E+03
F2	Min.	***2.23E-27***	8.28E-22	3.54E-10	7.99E-20	3.85E+01	1.07E+01	4.57E-02
	Mean	***2.05E-20***	2.23E-09	1.46E-09	2.48E-07	5.42E+01	5.73E+04	5.86E-01
	Worst	***3.40E-19***	3.26E-08	3.81E-09	2.31E-06	7.49E+01	1.71E+06	2.14E+00
F3	Min.	***1.56E-42***	7.71E-37	9.12E-04	2.88E-16	9.99E+03	1.12E+04	3.91E+03
	Mean	***1.51E-27***	2.64E-14	2.96E-02	1.21E-06	1.57E+04	2.77E+04	1.25E+04
	Worst	***2.29E-26***	7.61E-13	2.61E-01	1.61E-05	2.23E+04	7.81E+04	2.80E+04
F4	Min.	***5.55E-26***	6.91E-21	4.11E-04	2.28E-10	3.02E+01	2.78E+01	1.97E+01
	Mean	***1.07E-17***	2.34E-10	1.46E-03	8.66E-07	3.95E+01	3.81E+01	3.90E+01
	Worst	***1.88E-16***	4.99E-09	6.71E-03	5.08E-06	4.48E+01	4.88E+01	6.00E+01
F5	Min.	2.01E-04	***2.02E-09***	2.60E+01	5.04E-07	5.10E+05	6.01E+05	3.91E+03
	Mean	3.79E-01	3.45E-01	2.67E+01	***1.98E-03***	9.41E+05	3.46E+06	3.83E+05
	Worst	1.46E+00	1.18E+00	2.94E+01	***2.16E-02***	1.50E+06	7.29E+06	2.17E+06
F6	Min.	***1.84E-07***	8.53E-05	1.34E-04	2.88E-07	1.82E+03	4.27E+03	1.33E+01
	Mean	3.37E-03	1.71E-03	3.83E-01	***2.14E-05***	2.99E+03	9.56E+03	1.76E+02
	Worst	7.36E-03	3.06E-02	1.01E+00	***2.26E-04***	4.66E+03	1.63E+04	4.96E+02
F7	Min.	***7.01E-07***	6.57E-05	6.00E-04	2.03E-06	3.41E-01	5.45E-01	2.44E-02
	Mean	***1.78E-04***	9.98E-04	2.04E-03	2.82E-04	1.00E+00	2.20E+00	4.60E-01
	Worst	***8.15E-04***	6.53E-03	4.86E-03	2.90E-03	1.60E+00	5.12E+00	2.27E+00
F8	Min.	-1.02E+01	-1.02E+01	-1.02E+01	-1.02E+01	-1.02E+01	***-1.02E+01***	-4.96E+00
	Mean	***-1.02E+01***	-1.01E+01	-9.92E+00	-5.90E+00	-1.01E+01	-6.46E+00	-3.75E+00
	Worst	***-1.02E+01***	-1.01E+01	-3.41E+00	-5.06E+00	-1.01E+01	-2.63E+00	-8.79E-01
F9	Min.	-1.04E+01	-1.04E+01	-1.04E+01	***-1.04E+01***	-1.04E+01	-1.04E+01	-1.03E+01
	Mean	***-1.04E+01***	-1.04E+01	-1.02E+01	-6.68E+00	-1.04E+01	-5.42E+00	-4.32E+00
	Worst	***-1.04E+01***	-1.03E+01	-5.09E+00	-5.09E+00	-1.03E+01	-1.84E+00	-9.08E-01
F10	Min.	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01	***-1.05E+01***	-6.54E+00
	Mean	***-1.05E+01***	-1.05E+01	-1.03E+01	-5.47E+00	-1.05E+01	-6.41E+00	-4.07E+00
	Worst	***-1.05E+01***	-1.04E+01	-2.42E+00	-2.81E+00	-1.04E+01	-1.68E+00	-9.41E-01

While performing the third experiment, population is set to be 200. In Table 5, all algorithms' the minimum, average and worst results by the classical test functions are given. In addition, the convergence curves of the algorithms are given in Figure 3**.** In the minimum value metric, the proposed algorithm is the most successful one in 6 out of 10 functions (F1-F5, F7) while CMA-ES is successful in 4 functions (F6, F8-F10). In the mean value metric, the proposed algorithm is the most successful algorithm in 8 out of 10 functions (F1-F4, F7-F10). CMA-ES is successful in two functions (F5, F6). In the worst value metric, the proposed algorithm is the most successful one by presenting the best results in 8 out of 10 functions (F1-F4, F7-F10). CMA-ES solves 2 functions successfully (F5, F6).

Table 5. Results on benchmark functions (Population: 200)

Obj. Func.		ITSO	TSO	GWO	CMA-ES	CS	BAT	SCA
F1	Min.	***2.59E-52***	2.87E-35	2.72E-19	2.27E-24	2.21E+03	1.97E+03	1.87E+00
	Mean	***2.87E-41***	2.21E-19	2.18E-18	1.22E-14	3.81E+03	4.55E+03	1.69E+02
	Worst	***5.87E-40***	6.27E-18	1.00E-17	2.03E-13	5.17E+03	7.27E+03	1.37E+03
F2	Min.	***1.07E-25***	3.02E-20	7.77E-12	7.86E-20	4.66E+01	1.68E+00	6.39E-02
	Mean	***1.32E-21***	5.16E-12	3.03E-11	3.26E-08	6.22E+01	2.76E+02	4.08E-01
	Worst	***1.65E-20***	1.06E-10	1.16E-10	1.21E-07	7.81E+01	5.70E+03	1.14E+00
F3	Min.	***9.02E-44***	1.60E-38	4.05E-06	8.54E-13	1.28E+04	6.01E+03	1.90E+03
	Mean	***1.54E-30***	2.40E-17	3.22E-04	4.45E-08	1.65E+04	1.49E+04	7.33E+03
	Worst	***4.05E-29***	6.77E-16	1.33E-03	7.71E-07	2.04E+04	3.51E+04	1.59E+04
F4	Min.	***6.86E-27***	1.60E-22	2.56E-05	3.13E-11	3.87E+01	2.19E+01	1.65E+01
	Mean	***1.54E-20***	3.44E-12	1.08E-04	1.46E-07	4.36E+01	2.82E+01	3.39E+01
	Worst	***2.54E-19***	6.05E-11	5.03E-04	1.18E-06	4.85E+01	3.78E+01	6.23E+01
F5	Min.	***1.47E-08***	3.97E-05	2.52E+01	1.95E-07	7.83E+05	7.82E+04	7.69E+02
	Mean	9.25E-02	1.74E-01	2.63E+01	***4.56E-04***	1.76E+06	5.60E+05	9.83E+04
	Worst	4.06E-01	1.18E+00	2.79E+01	***9.14E-03***	2.81E+06	2.14E+06	1.33E+06
F6	Min.	1.62E-04	1.19E-07	8.34E-05	***1.29E-08***	2.39E+03	7.31E+02	1.08E+01
	Mean	1.67E-03	4.58E-04	1.27E-01	***1.45E-06***	3.69E+03	4.52E+03	7.92E+01
	Worst	4.20E-03	2.20E-03	5.11E-01	***9.50E-06***	4.88E+03	7.08E+03	2.15E+02
F7	Min.	***8.35E-06***	8.73E-05	4.14E-04	9.61E-06	3.68E-01	4.10E-01	8.97E-03
	Mean	***1.10E-04***	4.47E-04	9.89E-04	1.41E-04	1.42E+00	1.04E+00	1.76E-01
	Worst	***6.21E-04***	1.18E-03	1.84E-03	7.08E-04	2.23E+00	2.27E+00	8.30E-01
F8	Min.	-1.02E+01	-1.02E+01	-1.02E+01	***-1.02E+01***	-1.02E+01	-1.02E+01	-6.67E+00
	Mean	***-1.02E+01***	-1.01E+01	-9.47E+00	-6.92E+00	-1.01E+01	-6.64E+00	-4.38E+00
	Worst	***-1.02E+01***	-1.01E+01	-5.06E+00	-5.06E+00	-1.01E+01	-2.63E+00	-8.79E-01
F9	Min.	-1.04E+01	-1.04E+01	-1.04E+01	***-1.04E+01***	-1.04E+01	-1.04E+01	-8.81E+00
	Mean	***-1.04E+01***	-1.04E+01	-1.02E+01	-6.15E+00	-1.04E+01	-6.90E+00	-4.99E+00

	Worst	***-1.04E+01***	-1.04E+01	-5.12E+00	-5.09E+00	-1.04E+01	-1.84E+00	-9.09E-01
F10	Min.	-1.05E+01	-1.05E+01	-1.05E+01	***-1.05E+01***	-1.05E+01	-1.05E+01	-8.64E+00
	Mean	***-1.05E+01***	-1.05E+01	-1.05E+01	-5.64E+00	-1.05E+01	-6.07E+00	-5.11E+00
	Worst	***-1.05E+01***	-1.05E+01	-1.05E+01	-1.68E+00	-1.05E+01	-2.42E+00	-3.09E+00

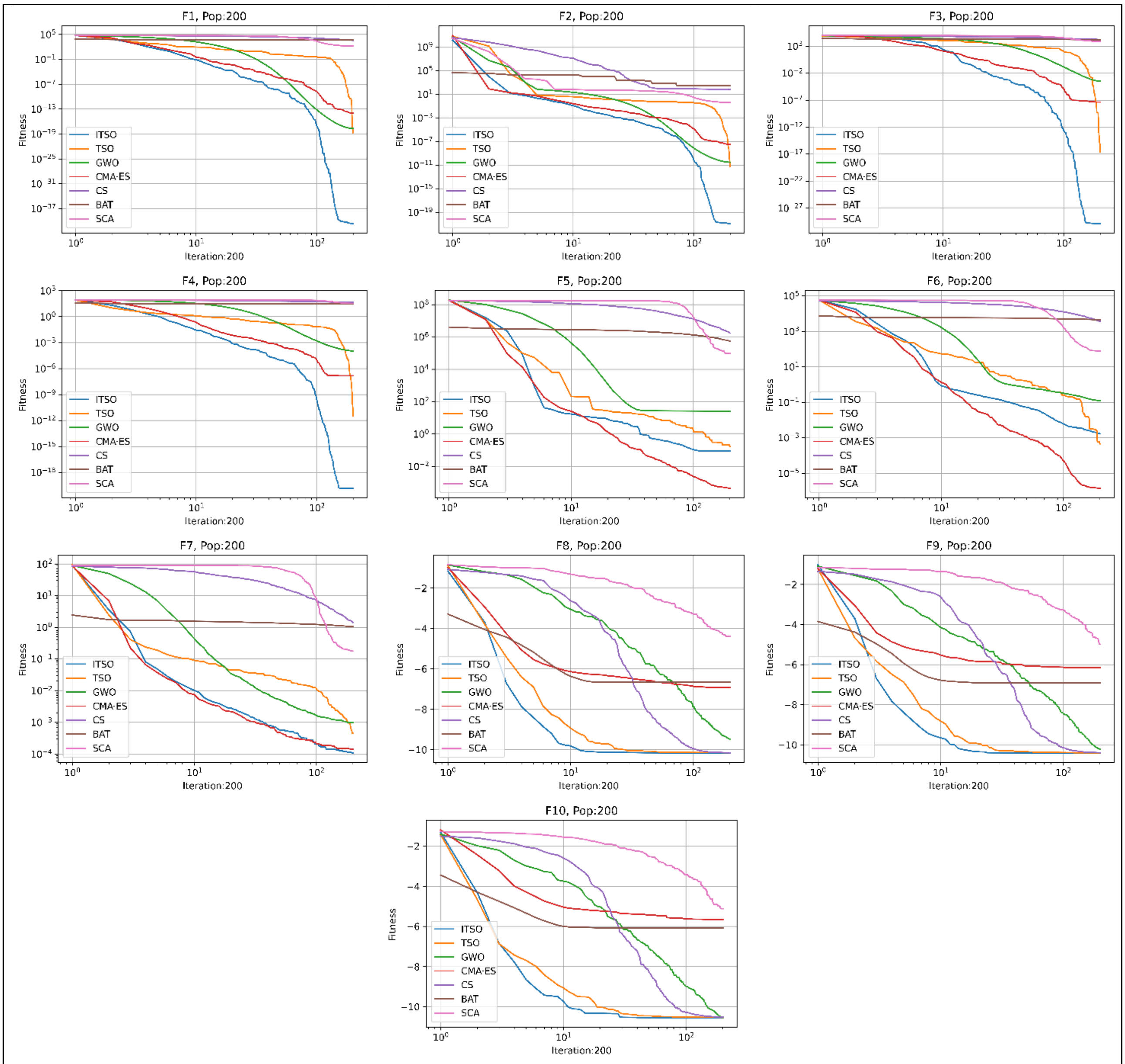

Figure 3. Convergence curve ITSO and other optimizers (Population: 200)

The results of an MHA informs us about its success. However, whether its success is statistically meaningful or not should be indicated. The Wilcoxon test is a non-parametric test that is frequently used for the comparison of optimization algorithms. The significance level of Wilcoxon test is set to be 5%. In Table 6, the results of the proposed algorithm's results of its comparison to its competitors are given. In the tables, "+" symbolizes that the proposed algorithm is better than its competitors while "-" indicates that it is worse.

Table 6. Wilcoxon rank-sum test

	Pop: 50						Pop: 100						Pop: 200					
Fnc.	**TSO**	**GWO**	**CMA-ES**	**CS**	**BAT**	**SCA**	**TSO**	**GWO**	**CMA-ES**	**CS**	**BAT**	**SCA**	**TSO**	**GWO**	**CMA-ES**	**CS**	**BAT**	**SCA**
F1	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
F2	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
F3	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
F4	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
F5	-	+	-	+	+	+	-	+	-	+	+	+	+	+	-	+	+	+
F6	-	+	-	+	+	+	-	+	-	+	+	+	-	+	-	+	+	+
F7	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
F8	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
F9	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
F10	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+

3.3 Engineering Design Problems

In this section, the proposed algorithm and competitor algorithms are applied to 3 engineering design problems. These problems have more than one local minimum and constraints. Results of 30 independent runs of all algorithms are recorded. Moreover, the population number and the number of iterations are respectively set to be 100 and 200.

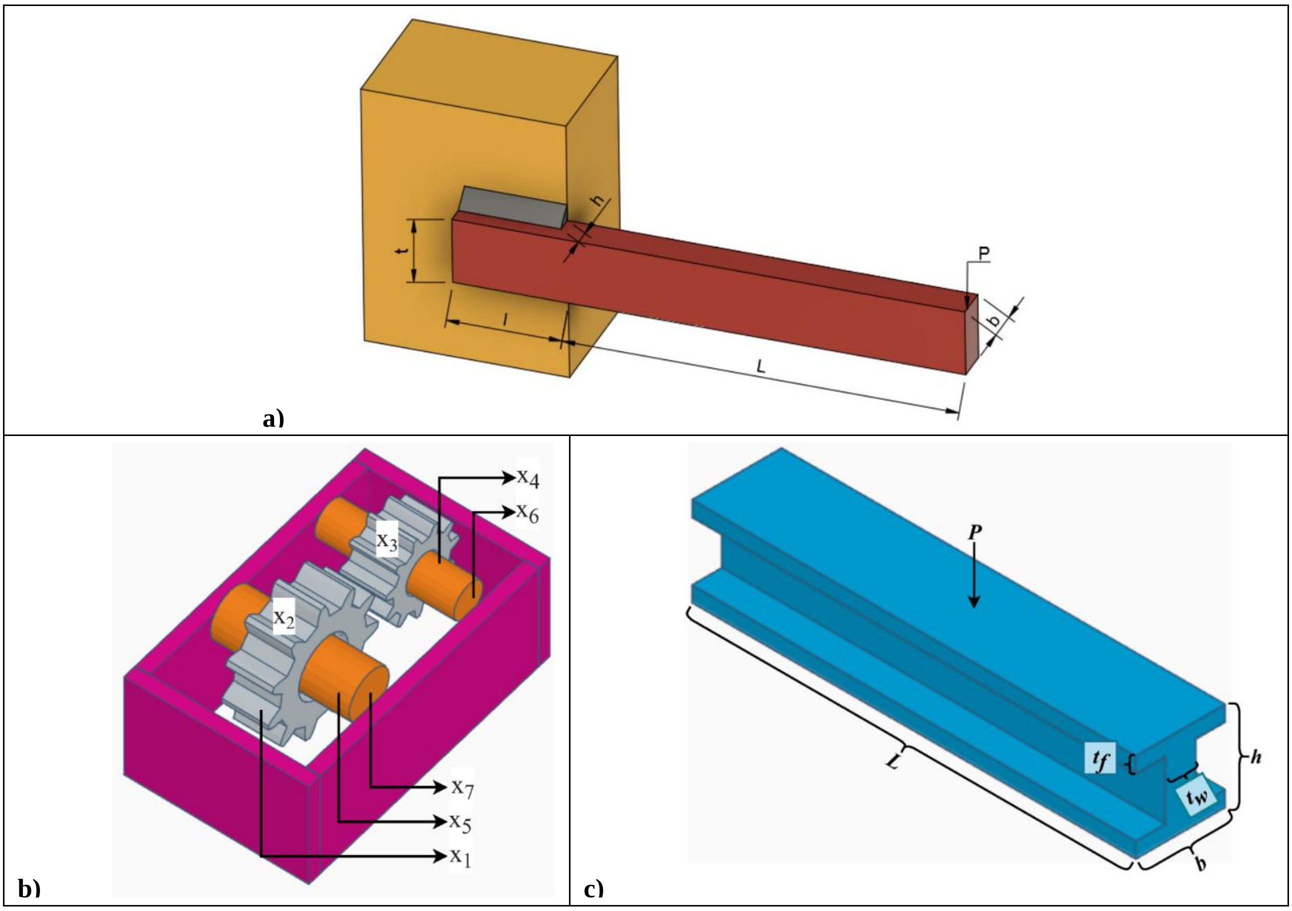

Figure 4. Engineering *d*esign problems a) Welded beam design problem b) Speed reducer design problem c) I-beam design problem

3.3.1 Welded beam design problem

The aim of the welded beam design problem is to minimize the production cost (Xie et al., 2021). The parameters and schematic representation of the problem are given in Figure 4a. The design is expected to stand the P load. The problem has 4 design parameters: welding thickness (h), length of welded joint (l), bar height (t) and bar thickness (b). The parameters (Equation (12)), objective function (Equation (13)) and constraints (Equation (14), (15), (16)) of the problem are given in the following equations.

Consider: $$\vec{X} = [x_1\ x_2\ x_3\ x_4] = [h\ l\ t\ b] \quad (12)$$

Minimize: $$f_{cost}(\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \quad (13)$$

Subject to: $$g_1(\vec{x}) = \tau(\vec{x}) - \tau_{max} \le 0,\ g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{max} \le 0,\ g_3(\vec{x}) = \delta(\vec{x}) - \delta_{max} \le 0,\ g_4(\vec{x}) = x_1 - x_4 \le 0,\ \ g_5(\vec{x}) = P - P_c(\vec{x}) \le 0,\ \ g_6(\vec{x}) = 0.125 - x_1 \le 0,\ g_7(\vec{x}) = 0.11047x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \le 0 \quad (14)$$

Range: $$0.1 \le x_1 \le 2,\ \ 0.1 \le x_2 \le 10,\ 0.1 \le x_3 \le 10,\ 0.1 \le x_4 \le 2 \quad (15)$$

Where: $$\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2},\ \tau' = \frac{P}{\sqrt{2}x_1x_2},\ \tau = \frac{MR}{J},\ \ M = P\left(L + \frac{x_2}{2}\right),\ R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2},\ J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2\right]\right\},\ \sigma(\vec{x}) = \frac{6PL}{x_4x_3^2},\ \delta(\vec{x}) = \frac{4PL^3}{Ex_3^3x_4},\ P_c(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right),\ P = 6{,}000\ lb,\ L = 14\ in,\ \delta_{max} = 0.25\ in,\ E = 30 \times 10^6\ \ psi,\ G = 12 \times 10^6\ \ psi,\ \tau_{max} = 13{,}600\ psi,\ \sigma_{max} = 30{,}000\ psi \quad (16)$$

The results of by all algorithms from the welded beam design problem are given in Table *7*. Examining the results, it is observed that with 1.74 optimal cost, the proposed algorithm is the most successful one. GWO is the second and the CMA-ES is the third most successful algorithm with optimal costs like 1.74 and 1.835. With an optimal cost of 2.25, TSO is one of the algorithms having the worst result.

Table 7. Comparison of results for welded beam design problem

Optimizer	Optimal values for variables				Optimal cost
	x_1	x_2	x_3	x_4	
ITSO	**0.204**	**3.520**	**9.110**	**0.205**	**1.740**
TSO	0.100	0.100	0.100	0.100	2.250
GWO	0.185	3.940	9.240	0.205	1.780
CMA-ES	0.200	3.633	9.105	0.206	1.835
CS	0.236	3.200	8.390	0.239	1.860
BAT	0.136	7.540	7.380	0.308	2.510
SCA	0.213	3.610	8.720	0.233	1.900

3.3.2 Speed Reducer Design

The main goal of the speed reducer design problem is to minimize the weight of the design (Ahmadianfar et al., 2020). Speed Reducer Design has 7 design variants which are face width (x_1), module of teeth (x_2), number of teeth on pinion (x_3), length of shaft 1 between bearings (x_4), length of shaft 2 between bearings (x_5), diameter of shaft 1 (x_6), and diameter of shaft 2 (x_7) (Figure 4b). The objective function (Equation (17)) and constraints (Equation (18), (19)) of the problem are given in the following equations.

Minimize: $$f_{cost}(\vec{x}) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2) \quad \textbf{(17)}$$

Subject to: $$g_1(\vec{x}) = \frac{27}{x_1x_2^2x_3} - 1 \le 0,\ g_2(\vec{x}) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \le 0,\ g_3(\vec{x}) = \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \le 0 \quad \textbf{(18)}$$

$$g_4(\vec{x}) = \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \le 0,\ g_5(\vec{x}) = \frac{1}{110x_6^3}\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 16.9\times 10^6} - 1 \le 0$$
$$g_6(\vec{x}) = \frac{1}{85x_7^3}\sqrt{\left(\frac{745x_5}{x_2x_3}\right)^2 + 157.5\times 10^6} - 1 \le 0,\ g_7(\vec{x}) = \frac{x_2x_3}{40} - 1 \le 0,$$
$$g_8(\vec{x}) = 5\frac{x_2}{x_1} - 1 \le 0, g_9(\vec{x}) = \frac{x_1}{12x_2} - 1 \le 0,\ g_{10}(\vec{x}) = \frac{1.5x_6+1.9}{x_4} - 1 \le 0,$$
$$g_{11}(\vec{x}) = \frac{1.1x_7 + 1.9}{x_5} - 1 \le 0$$

Range: $2.6 \le x_1 \le 3.6,\ 0.7 \le x_2 \le 0.8,\ 17 \le x_3 \le 28,\ 7.3 \le x_4 \le 8.3,$ $7.3 \le x_5 \le 8.3, 2.9 \le x_6 \le 3.9,\ 5 \le x_7 \le 5.5$ **(19)**

The results of all algorithms from the speed reducer design problem are given in Table 8. With 1581.47 optimal cost, the proposed algorithm is the most successful one while Bat is the second with 1581.494 optimal cost and CS is the third with 1582.539 optimal cost. Having the 1612.18 optimal cost, TSO is the one with the worst result.

Table 8. Comparison of results for speed reducer problem

Optimizer	Optimal values for variables							Optimal cost
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
ITSO	***3.500***	***0.700***	***17.000***	***8.015***	***8.008***	***3.777***	***5.297***	***1581.470***
TSO	2.605	0.700	17.079	7.293	7.296	2.903	4.994	1612.180
GWO	3.500	0.700	17.000	8.300	8.054	3.900	5.384	1584.598
CMA-ES	3.500	0.700	17.000	7.300	8.300	3.437	5.500	1589.722
CS	3.500	0.700	17.000	8.300	8.300	3.900	5.500	1582.539
BAT	3.500	0.700	17.000	8.300	8.070	3.900	5.413	1581.494
SCA	3.502	0.700	17.000	8.300	8.300	3.900	5.500	1588.522

3.3.3 I-beam design problem

The main purpose of this problem is to minimize the vertical deflection of the I beam (Ahmadianfar et al., 2020). I beam has 4 variants which are length (l), height (h), and two thickness (t_w, t_f) (Figure 4c). The parameters (Equation (20)), objective function (Equation (21)) and constraints (Equation (22), (23)) of the problem are given in the following equations.

Consider: $$\vec{X} = [x_1, x_2, x_3, x_4] = [h, l, t_w, t_f] \quad \textbf{(20)}$$

Minimize: $$f_{cost}(\vec{x}) = \frac{5000}{\frac{1}{12}t_w(h-2t_f)^3 + \frac{1}{6}lt_f{}^3 + 2lt_f\left(\frac{h-t_f}{2}\right)^2} \quad \textbf{(21)}$$

Subject to: $$g_1(\vec{x}) = 2lt_f + t_w(h-2t_f)^3 \le 300$$
$$g_2(\vec{x}) = \frac{180000h}{t_w(h-2t_f)^3 + 2lt_f[4t_f{}^2 + 3h(h-2t_f)]} + \frac{15000l}{(h-2t_f)t_w{}^3 + 2t_f l^3} \le 6 \quad \textbf{(22)}$$

Range: $$10 \le h \le 80,\ 10 \le l \le 50,\ 0.9 \le t_w \le 5,\ 0.9 \le t_f \le 5 \quad \textbf{(23)}$$

The results of all algorithms from the I-beam design problem are given in Table 9. With 38.455 optimal cost, the proposed algorithm is the most successful one while GWO is the second with 38.61 optimal cost and CMA-ES is the third with 38.668 optimal cost. TSO is algorithm with the worst result as it has 44.777 optimal cost.

Table 9. Comparison of results for I- beam design problem

Optimizer	Optimal values for variables				Optimal cost
	x_1	x_2	x_3	x_4	
ITSO	***13.583***	***25.857***	***0.900***	***5.000***	***38.455***
TSO	10.000	10.000	0.900	0.900	44.777
GWO	13.535	25.998	0.900	5.000	38.610
CMA-ES	13.561	25.934	0.900	5.000	38.668
CS	13.608	25.732	0.900	5.000	38.949
BAT	13.582	25.864	0.900	5.000	41.129
SCA	13.616	25.549	0.900	5.000	39.421

4. CONCLUSION

TSO is a swarm-based MHA that is improved by being inspired by the fishing strategies of tuna fish. The biggest disadvantage of TSO is that it gets caught by the local minimum trap. In order to solve this problem of TSO, this article proposes a new local search procedure. The main philosophy of the proposed approach is not focusing only on the best solution but on the best ones. The new proposed algorithm is applied to 10 classical test functions and welded beam design problem, speed reducer design problem and I-beam design problem. The results indicate the success of the proposed algorithm. The results of this study is as follows.

- Like many other MHAs, TSO focuses on the best results. Focusing not only on the best solution but on the best ones allows it to abstain from local minimum trap.
- The results of the classical test functions indicate that the proposed algorithm is successful at solving unimodal and multimodal functions.
- Besides, statistical results confirm that the improved algorithm’s success for the classical test results are meaningful.
- In engineering design problems, TSO is not able to present competitive results. In all design problems, the proposed algorithm is the most successful one though. Such a case indicates that the method proposed in this article to improve the performance of TSO is successful.
- As far as we know, this study is the first one proposing a change in the mathematical model of TSO. Considering the results, it could be suggested that TSO is an algorithm that is open to be improved.

For further studies, the following issues could be considered.

- Redesigning the mathematical model of TSO's spiral and parabolic search strategies. Moreover, TSO could be improved so that it might not need initial parameters.
- Analysing the parameters of TSO in a wide scale by determining them through 2 or 3 dimensional chaotic maps.
- Finding out its weakness by applying it to more real-world problems.
- By integrating TSO into machine learning and artificial intelligence, it could be applied in a field like image processing.
- In the optimization of unmanned air vehicles and the solution of mission planning problems, TSO could generate successful results.

5. CONFLICT OF INTEREST

Author approve that to the best of their knowledge, there is not any conflict of interest or common interest with an institution/organization or a person that may affect the review process of the paper.

6. REFERENCES

Abdulwahab H. A., Noraziah A., Alsewari A. A., Salih S. Q., An Enhanced Version of Black Hole Algorithm via Levy Flight for Optimization and Data Clustering Problems. IEEE Access 7, 142085-142096, 2019.

Abualigah L., Diabat A., Advances in Sine Cosine Algorithm: A comprehensive survey. Artificial Intelligence Review 54(4), 2567-2608, 2021.

Abualigah L., Diabat A., Geem Z. W., A Comprehensive Survey of the Harmony Search Algorithm in Clustering Applications. Applied Sciences 10(11), 3827, 2020.

Ahmadianfar I., Bozorg-Haddad O., Chu X., Gradient-based optimizer: A new metaheuristic optimization algorithm. Information Sciences 540, 131-159, 2020.

Amine K., Multiobjective Simulated Annealing: Principles and Algorithm Variants. Advances in Operations Research 2019, e8134674, 2019.

Ashraf H., Elkholy M. M., Abdellatif S. O., El‐Fergany A. A., Synergy of neuro-fuzzy controller and tuna swarm algorithm for maximizing the overall efficiency of PEM fuel cells stack including dynamic performance. Energy Conversion and Management:X 16, 100301, 2022.

Askari Q., Saeed M., Younas I., Heap-based optimizer inspired by corporate rank hierarchy for global optimization. Expert Systems with Applications 161, 113702, 2020.

Askari Q., Younas I., Saeed M., Political Optimizer: A novel socio-inspired meta-heuristic for global optimization. Knowledge-Based Systems 195, 105709, 2020.

Deng W., Shang S., Cai X., Zhao H., Song Y., Xu J., An improved differential evolution algorithm and its application in optimization problem. Soft Computing 25(7), 5277-5298, 2021.

Feng Y., Deb S., Wang G.-G., Alavi A. H., Monarch butterfly optimization: A comprehensive review. Expert Systems with Applications 168, 114418, 2021.

Fu C., Zhang L., A novel method based on tuna swarm algorithm under complex partial shading conditions in PV system. Solar Energy 248, 28-40, 2022.

Gad A. G., Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review. Archives of Computational Methods in Engineering 29(5), 2531-2561, 2022.

Gandomi A. H., Yang X.-S., Alavi A. H., Talatahari S., Bat algorithm for constrained optimization tasks. Neural Computing and Applications 22(6), 1239-1255, 2013.

Gul F., Rahiman W., Alhady S. S. N., Ali A., Mir I., Jalil A., Meta-heuristic approach for solving multi-objective path planning for autonomous guided robot using PSO–GWO optimization algorithm with evolutionary programming. Journal of Ambient Intelligence and Humanized Computing 12(7), 7873-7890, 2021.

Guo S.-M., Guo J.-K., Gao Y.-G., Guo P.-Y., Fu-Jun a H., Wang S.-C., Lou Z.-C., Zhang X., Research on Engine Speed Control Based on Tuna Swarm Optimization. Journal of Engineering Research and Reports 23(12), 272-280, 2022.

Hansen N., Müller S. D., Koumoutsakos P., Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). Evolutionary Computation 11(1), 1-18, 2003.

Hashim F. A., Houssein E. H., Hussain K., Mabrouk M. S., Al-Atabany W., Honey Badger Algorithm: New metaheuristic algorithm for solving optimization problems. Mathematics and Computers in Simulation 192, 84-110, 2022.

Jafari A., Khalili T., Babaei E., Bidram A., A Hybrid Optimization Technique Using Exchange Market and Genetic Algorithms. IEEE Access 8, 2417-2427, 2020.

Korashy A., Kamel S., Youssef A.-R., Jurado F., Modified water cycle algorithm for optimal direction overcurrent relays coordination. Applied Soft Computing 74, 10-25, 2019.

Kumar A., Pant S., Ram M., System Reliability Optimization Using Gray Wolf Optimizer Algorithm. Quality and Reliability Engineering International 33(7), 1327-1335, 2017.

Kumar C., Magdalin Mary D., A novel chaotic-driven Tuna Swarm Optimizer with Newton-Raphson method for parameter identification of three-diode equivalent circuit model of solar photovoltaic cells/modules. Optik 264, 169379, 2022.

Kumar M., Kulkarni A. J., Satapathy S. C., Socio evolution & learning optimization algorithm: A socio-inspired optimization methodology. Future Generation Computer Systems 81, 252-272, 2018.

Kumar S., Yildiz B. S., Mehta P., Panagant N., Sait S. M., Mirjalili S., Yildiz A. R., Chaotic marine predators algorithm for global optimization of real-world engineering problems. Knowledge-Based Systems 261, 110192, 2023.

Kumar V., Kumar D., A Systematic Review on Firefly Algorithm: Past, Present, and Future. Archives of Computational Methods in Engineering 28(4), 3269-3291, 2021.

Li S., Gong W., Yan X., Hu C., Bai D., Wang L., Gao L., Parameter extraction of photovoltaic models using an improved teaching-learning-based optimization. Energy Conversion and Management 186, 293-305, 2019.

Mareli M., Twala B., An adaptive Cuckoo search algorithm for optimisation. Applied Computing and Informatics 14(2), 107-115, 2018.

Mbuli N., Ngaha W. S., A survey of big bang big crunch optimisation in power systems. Renewable and Sustainable Energy Reviews 155, 111848, 2022.

Mirjalili S., SCA: A Sine Cosine Algorithm for solving optimization problems. Knowledge-Based Systems 96, 120-133, 2016.

Mirjalili S., Evolutionary Algorithms and Neural Networks, Springer International Publishing, First Edition, United States, pp. 43-55, 2019.

Mirjalili S., Mirjalili S. M., Lewis A., Grey Wolf Optimizer. Advances in Engineering Software 69, 46-61, 2014.

Noureddine S., An optimization approach for the satisfiability problem. Applied Computing and Informatics 11(1), 47-59, 2015.

Öztürk Ş., Ahmad R., Akhtar N., Variants of Artificial Bee Colony algorithm and its applications in medical image processing. Applied Soft Computing 97, 106799, 2020.

Prajapati V. K., Jain M., Chouhan L., Tabu Search Algorithm (TSA): A Comprehensive Survey, 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE), Jaipur/India, February 7-8, 2020, pp: 1-8.

Raja B. D., Patel V. K., Yildiz A. R., Kotecha P., Performance of scientific law-inspired optimization algorithms for constrained engineering applications. Engineering Optimization 55(10), 1798-1812, 2023.

Rajabioun R., Cuckoo Optimization Algorithm. Applied Soft Computing 11(8), 5508-5518, 2011.

Ramachandran M., Mirjalili S., Nazari-Heris M., Parvathysankar D. S., Sundaram A., Charles Gnanakkan C. A. R., A hybrid Grasshopper Optimization Algorithm and Harris Hawks Optimizer for Combined Heat and Power Economic Dispatch problem. Engineering Applications of Artificial Intelligence 111, 104753, 2022.

Rashedi E., Nezamabadi-pour, H., Saryazdi S., GSA: A Gravitational Search Algorithm. Information Sciences 179(13), 2232-2248, 2009.

Rosso M. M., Cucuzza R., Aloisio A., Marano G. C., Enhanced Multi-Strategy Particle Swarm Optimization for Constrained Problems with an Evolutionary-Strategies-Based Unfeasible Local Search Operator. Applied Sciences 12(5), 2285, 2022.

Tan M., Li Y., Ding D., Zhou R., Huang C., An Improved JADE Hybridizing with Tuna Swarm Optimization for Numerical Optimization Problems. Mathematical Problems in Engineering 2022, e7726548, 2022.

Tuerxun W., Xu C., Guo H., Guo L., Zeng N., Cheng Z., An ultra-short-term wind speed prediction model using LSTM based on modified tuna swarm optimization and successive variational mode decomposition. Energy Science & Engineering 10(8), 3001-3022, 2022.

Wang G.-G., Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems. Memetic Computing 10(2), 151-164, 2018.

Wang G.-G., Deb S., Coelho L. D. S., Earthworm optimisation algorithm: a bio-inspired metaheuristic algorithm for global optimisation problems. International Journal of Bio-Inspired Computation 12(1), 1-22, 2018.

Wang G.-G., Deb S., Cui Z., Monarch butterfly optimization. Neural Computing and Applications 31(7), 1995-2014, 2019.

Wang J., Zhu L., Wu B., Ryspayev A., Forestry Canopy Image Segmentation Based on Improved Tuna Swarm Optimization. Forests 13(11), 1746, 2022.

Wang W., Tian J., An Improved Nonlinear Tuna Swarm Optimization Algorithm Based on Circle Chaos Map and Levy Flight Operator. Electronics 11(22), 3678, 2022.

Wang Y., Wang P., Zhang J., Cui Z., Cai X., Zhang W., Chen J., A Novel Bat Algorithm with Multiple Strategies Coupling for Numerical Optimization. Mathematics 7(2), 135, 2019.

Wei Z., Huang C., Wang X., Han T., Li Y., Nuclear Reaction Optimization: A Novel and Powerful Physics-Based Algorithm for Global Optimization. IEEE Access 7, 66084-66109, 2019.

Wolpert D. H., Macready W. G., No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation 1(1), 67-82, 1997.

Wu L., Huang X., Cui J., Liu C., Xiao W., Modified adaptive ant colony optimization algorithm and its application for solving path planning of mobile robot. Expert Systems with Applications 215, 119410, 2023.

Xie L., Han T., Zhou H., Zhang Z.-R., Han B., Tang A., Tuna Swarm Optimization: A Novel Swarm-Based Metaheuristic Algorithm for Global Optimization. Computational Intelligence and Neuroscience 2021, e9210050, 2021.

Xue Y., Zhang Q., Zhao Y., An improved brain storm optimization algorithm with new solution generation strategies for classification. Engineering Applications of Artificial Intelligence 110, 104677, 2022.

Yan Z., Yan J., Wu Y., Cai S., Wang H., A novel reinforcement learning based tuna swarm optimization algorithm for autonomous underwater vehicle path planning. Mathematics and Computers in Simulation 209, 55-86 2023.

Zhang F., Mei Y., Nguyen S., Zhang M., Tan K. C., Surrogate-Assisted Evolutionary Multitask Genetic Programming for Dynamic Flexible Job Shop Scheduling. IEEE Transactions on Evolutionary Computation 25(4), 651-665, 2021.

Zhang Y., Jin Z., Group teaching optimization algorithm: A novel metaheuristic method for solving global optimization problems. Expert Systems with Applications 148, 113246, 2020.