Journal of Ambient Intelligence and Humanized Computing (2023) 14:8045–8065
https://doi.org/10.1007/s12652-022-04347-1

ORIGINAL RESEARCH

A new hybrid method based on Aquila optimizer and tangent search algorithm for global optimization

Sinem Akyol[1]

Received: 24 November 2021 / Accepted: 11 July 2022 / Published online: 8 August 2022

Abstract
Since no single algorithm can provide the optimal solutions for all problems, new metaheuristic methods are always being proposed or developed by combining current algorithms or creating adaptable versions. Metaheuristic methods should have a balanced exploitation and exploration stages. One of these two talents may be sufficient in some metaheuristic methods, while the other may be insufficient. By integrating the strengths of the two algorithms and hybridizing them, a more efficient algorithm can be formed. In this paper, the Aquila optimizer-tangent search algorithm (AO-TSA) is proposed as a new hybrid approach that uses the intensification stage of the tangent search algorithm (TSA) instead of the limited exploration stage to improve the Aquila optimizer's exploitation capabilities (AO). In addition, the local minimum escape stage of TSA is applied in AO-TSA to avoid the local minimum stagnation problem. The performance of AO-TSA is compared with other current metaheuristic algorithms using a total of twenty-one benchmark functions consisting of six unimodal, six multimodal, six fixed-dimension multimodal, and three modern CEC 2019 benchmark functions according to different metrics. Furthermore, two real engineering design problems are also used for performance comparison. Sensitivity analysis and statistical test analysis are also performed. Experimental results show that hybrid AO-TSA gives promising results and seems an effective method for global solution search and optimization problems.

Keywords Aquila optimizer · Global optimization · Hybrid method · Tangent search algorithm

1 Introduction

Optimization is the task of searching the best among all candidate solutions for a problem under certain conditions. An optimization problem can be expressed as any problem that aims to find unknown variable values, provided that certain constraints are met (Murty 2003). While the solution method mostly depends on the type of variables (integer, real, etc.), objectives, and constraints (linear, non-linear, etc.) used in modeling the problem in classical optimization algorithms, its effectiveness also depends on the solution space (concave, convex, etc.), the number of constraints, and the number of decision variables. Furthermore, they do not provide general solution strategies that can be applied to problem formulations in the presence of different types of decision variables, objectives, and constraints. That is, most algorithms solve models with certain types of objective and constraint functions. However, the formulation of optimization problems in many different fields simultaneously needs various types of decision variables, objective functions, and constraint functions. Therefore, classical and general-purpose metaheuristic optimization algorithms are proposed. These techniques have gained considerable popularity in recent years because to their high computational efficiency and simplicity in transformation (Akyol and Alatas 2012, 2017; Alatas 2007). In most real-life problems, the search space is infinite or so large that all solutions cannot be evaluated. Thus, it is necessary to find a good solution by evaluating the solutions in a reasonable time. For such problems, assessment of solutions in a reasonable time is essentially the same thing as assessment of "some solutions" in the entire search space. What and how some solutions are selected vary based on the metaheuristic technique.

General-purpose metaheuristic methods can be grouped into eight categories as biology-based, physics-based, swarm-based, music-based, social-based, chemistry-based, sports-based, and mathematics-based. There are also hybrid

✉ Sinem Akyol
sakyol@firat.edu.tr

[1] Software Engineering Department, Engineering Faculty, Firat University, 23319 Elazig, Turkey

methods that are combinations of these (Akyol and Alatas 2012, 2017; Alatas 2007). Tunicate Swarm Algorithm (Kaur et al. 2020), Reptile Search Algorithm (Abualigah et al. 2022), Spotted Hyena Optimizer (Dhiman and Kumar 2017), Emperor Penguin Optimizer (Dhiman and Kumar 2018), Seagull Optimization Algorithm (Dhiman and Kumar 2019), and Sooty Tern Optimization Algorithm (Dhiman and Kaur 2019) are biology-based; the Parliamentary Optimization Algorithm (Borji 2007) and the Imperialist Competitor Algorithm (Atashpaz-Gargari and Lucas 2007) are social-based; Artificial Chemical Reaction Algorithm (Alatas 2011) is chemistry-based; Melody Search Algorithm (Ashrafi and Dariane 2011) is music-based; Archimedes Optimization Algorithm (Hashim et al. 2021) and Spring Search Algorithm (Dehghani et al. 2020b) are physics-based; Bonobo Optimizer (Das and Pratihar 2019) and Rat Swarm Optimizer (Dhiman et al. 2021) are swarm-based; Most Valuable Player Algorithm (Bouchekara 2020) and Darts Game Optimizer (Dehghani et al. 2020c) are sports-based and Arithmetic Optimization Algorithm (Abualigah et al. 2021a) is mathematics-based algorithms and models. Multi Leader Optimizer (Dehghani et al. 2020a) can also be classified as both a swarm-based and social-based algorithm (Akyol 2018; Akyol and Alatas 2012, 2020; Alatas 2007). In some studies, algorithms inspired by plant intelligence were examined in a separate group as plant-based methods (Akyol and Alatas 2017) and algorithms inspired by the law of reflection and refraction of light were categorized as light-based methods (Alatas and Bingol 2020).

AO (Abualigah et al. 2021b) is swarm-based and TSA (Layeb 2021) is mathematical-based metaheuristic algorithms. AO was developed with inspiration from the hunting skills of the Aquila and TSA was developed based on the tangent function. In the literature, the number of studies with AO and TSA is few. Only a few problem-oriented studies were performed for AO and there is not any study on the improvement of TSA. Using AO, AlRassas et al. developed the Adaptive Neuro-Fuzzy Inference System model and used this model to predict oil production for two different locations in Yemen and China (AlRassas et al. 2021). Elaziz et al. proposed a hybrid version of deep learning and AO and applied it to images of Covid-19 data (Abd Elaziz et al. 2021). Ma and Zhao proposed an improved version of the AO algorithm using wavelet mutation and quasi-opposite learning strategies. Then, they used this improved version of the AO and the Bernoulli model together to estimate the rural community population in China (Ma et al. 2021). Vashishtha and Kumar used AO to adjust the length of the minimum deconvolution filter used to detect bearing defects in the Francis turbine (Vashishtha and Kumar 2021).

Metaheuristic algorithms need to have exploration and exploitation capabilities. These two abilities must work in a balanced way. In some metaheuristic algorithms, while the exploitation capability works well, the exploration capability may be lacking, or the exploration capability may work well, but the exploitation capability may not be sufficient due to the stochastic nature of the method (Dhiman 2021). By hybridizing the algorithms, a more efficient algorithm can be obtained by combining the strengths of the two algorithms along with eliminating the weaknesses. Many optimization algorithms suffer from early convergence due to the local optimum in the exploitation phase (Upadhyay and Chhabra 2021). Hybrid algorithms, which are more preferred than individual metaheuristic algorithms, are used in solving broader optimization problems (Verma and Parouha 2021). When compared to classical forms of metaheuristic methods, their hybrid versions show substantial gains (Dokeroglu et al. 2019). Recent research suggests that hybrid metaheuristic algorithms can provide more efficient behavior and better flexibility (Blum et al. 2011). The fundamental purpose of hybrid algorithms is to combine the characteristics of various metaheuristic techniques and reap the benefits of synergy.

AO and TSA are two of the newest metaheuristic methods and there is not any hybrid version of TSA. Only two hybrid versions of AO exist in the literature. Wang et al. proposed a new hybrid algorithm by addressing the strengths of AO's good exploration ability and the exploitation phase of Harris Hawks Optimizer (Wang et al. 2021). Mahajan et al. stated that the AO converges early due to the direct addition of the global best location in the location update, making the search phase strong. However, this will cause it to remain at the local optimum. In addition, they also stated that the convergence speed of the Arithmetic Optimization Algorithm (AOA) is reported as low and the search capability as weak. To overcome this deficiency of AOA, they proposed the hybrid AOAAO algorithm (Mahajan et al. 2022).

Based on these, in this article, a new hybrid algorithm, AO-TSA, is proposed for complex solution search and optimization problems. While the exploration phase is applied at 2/3 of the number of iterations in the AO method, the exploitation phase is generally insufficient. To give more space to exploitation capability, the effective intensification phase of TSA is applied instead of the narrowed exploration phase of the AO. In addition, TSA's local minimum escape steps are also applied to avoid the local minimum stagnation problem. According to various metrics, the performance of AO-TSA is compared to those of other existing metaheuristic algorithms utilizing a total of twenty-one benchmark functions, including six unimodal, six multimodal, six fixed-dimension multimodal, and three modern CEC 2019 benchmark functions. In addition, two real engineering design problems are used to compare performance. Statistical test analysis and sensitivity analysis are also performed. The proposed hybrid method seems to achieve better results at an earlier time than AO and TSA. In the second part of this study, AO and TSA are explained in detail and flow diagrams are given. In the third part, the hybrid AO-TSA algorithm is explained and how the two algorithms are combined in the flow diagram is shown. The fourth part introduces the used benchmark functions and real engineering design problems. In the fifth part, the performances of standard AO, standard TSA, AO-TSA, and metaheuristic algorithms with promising results in the literature are presented by comparing the experimental results obtained in benchmark functions. The paper is concluded along with possible future research directions in the sixth part.

2 Standard Aquila optimizer and tangent search algorithm

In this section, the AO (Abualigah et al. 2021b), which is inspired by the hunting skills of Aquila, and TSA (Layeb 2021), which is based on the mathematical tangent function, are explained in detail and flow diagrams are given.

2.1 Aquila optimizer

Aquilas are the most common species of eagles and are known as the most popular birds of prey in the Northern Hemisphere. The four hunting methods used by Aquilas are described as follows: The first method, vertically inclined high-flying, is used by the Aquila to hunt birds while flying high above the ground. After discovering its prey and getting closer and closer to it, the wings take a long, low-angle glide that rises rapidly. The second method, the perimeter flight with a short soaring attack, is the flight in which the Aquila rises low above the ground. In the third method, the Aquila slowly attacks its victim, trying to land on the prey it has chosen on the ground. This method is called slow landing attack and flight action. The last method is walking and catching. The Aquila tries to attract prey by walking on the ground in this method. The AO algorithm was inspired by these four hunting methods of Aquila.

As in all population-based methods, in AO, the initial population (X) starts with randomly generated values between the lower and upper limits. The optimal result is the best solution obtained from the candidate solutions as a result of iterations. X consists of a set of randomly generated candidate solutions using Eq. (1). N represents the total number of candidate solutions (population size), X_{ij} represents jth decision values (positions) of the ith solution, and D represents the size of the problem. $rand$ is a random number, UB_j is the jth upper limit, and LB_j, is the jth lower limit (Abualigah et al. 2021b).

$$X_{ij} = rand \times \left(UB_j - LB_j\right) + LB_j, i = 1, 2, \dots, Nj = 1, 2, \dots, D \tag{1}$$

Based on the four behavior patterns that the Aquila simulates while hunting, the AO algorithm is represented by a method for each of these behaviors: high glide search area selection with vertical slope, short glide attack, and reconnaissance within a different search area with the boundary line, low flight exploitation within a convergence search area with slow landing attack, and catching prey on foot. In the AO algorithm, where T is the maximum number of iterations and t is the number of current iteration, exploration steps are applied when the condition $t \leq \left(\frac{2}{3}\right) \times T$ is met, otherwise, exploitation steps are started.

The behavior of Aquila is modeled as a mathematical optimization paradigm, which tries to find the best solution according to the certain constraints (Abualigah et al. 2021b).

2.1.1 Step 1: extended exploration (X_1)

In this method (X_1), Aquila discovers the hunting ground and chooses the best hunting area with a high glide on a vertical slope. Here, AO conducts extensive expeditions by flying high to determine the search area where prey is located. This behavior is mathematically presented as in Eq. (2).

$$X_1(t+1) = X_{best}(t) \times \left(1 - \frac{t}{T}\right) + \left(X_M(t) - X_{best}(t) \times rand\right) \tag{2}$$

Here, $X_1(t+1)$ represents the solution of the $(t+1)$th iteration produced by the first method (X_1). $X_{best}(t)$ represents the best solution achieved up to the tth iteration and with this, the proximate location of the prey is determined. By using the $\left(1 - \frac{t}{T}\right)$ equation, it is decided whether to apply extended search steps. $X_M(t)$ calculated using Eq. (3) indicates the mean value of the positions of the existing solutions in the tth iteration. $rand$ is a randomly generated number in [0, 1] (Abualigah et al. 2021b).

$$X_M(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t), \forall i = 1, 2, \dots, D \tag{3}$$

2.1.2 Step 2: narrowed exploration (X_2)

In this method (X_2), the hunting ground is found with a high flight. In the short glide attack and contour flight method, the Aquila prepares the attack area by drawing circles on the prey before attacking the prey it has determined. In this method, AO searches in detail around its chosen prey before attacking. This behavior is mathematically presented as in Eq. (4).

$$X_2(t+1) = X_{best}(t) \times Levy() + X_R(t) + (y - x) \times rand \tag{4}$$

$X_2(t+1)$ represents the solution of the $(t+1)$ th iteration, produced by the second method (X_2). Levy flight distribution

$Levy()$ is calculated as given in Eq. (5). The random solution $X_R(t)$ is in the interval $[1, N]$ in the tth iteration (Abualigah et al. 2021b).

$$Levy() = s \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}} \tag{5}$$

Here, s is a constant with a value of 0.01. u and v are random numbers between 0 and 1. σ is computed by Eq. (6).

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin \frac{\pi\beta}{2}}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right) \tag{6}$$

Here, β is a constant with a value of 1.5. In Eq. (4), x and y are utilized for representing the spiral shape in the search and are calculated using Eqs. (7), (8), (9), (10), and (11).

$$y = r \times \cos(\theta) \tag{7}$$

$$x = r \times \sin(\theta) \tag{8}$$

$$r = r_1 + U \times D_1 \tag{9}$$

$$\theta = -\omega \times D_1 + \theta_1 \tag{10}$$

$$\theta_1 = \frac{3 \times \pi}{2} \tag{11}$$

r_1 takes a value from 1 to 20 to determine the number of search cycles, and U is a constant with a value of 0.00565. D_1 is an integer from 1 to the length of the search field (D) and ω is a small constant of 0.005 (Abualigah et al. 2021b).

2.1.3 Step 3: extended exploitation (X_3)

In this method (X_3), when the area of prey is correctly found and the Aquila is ready to land and attack, it descends steeply towards to prey with a frontal attack to explore its response. This method, in which the Aquila approaches and attacks by using the environment of the prey it chooses, is called low flight with landing attack. The mathematical expression of this behavior is shown in Eq. (12).

$$\begin{aligned} X_3(t+1) = {} & (X_{best}(t) - X_M(t)) \times \alpha - rand \\ & + ((UB - LB) \times rand + LB) \times \delta \end{aligned} \tag{12}$$

$X_3(t+1)$ is the solution of the $(t+1)$ th iteration produced by the third search method (X_3). α and δ represent parameters of the exploitation tuning that are set to a small value of 0.1 (Abualigah et al. 2021b).

2.1.4 Step 4: narrowed exploitation (X_4)

In the last method (X_4), when the Aquila approaches its prey on land, it attacks according to the stochastic movements of its prey. This step is called walking and catching prey. Finally, the AO goes after the prey in the final location. This phenomenon is mathematically presented as in Eq. (13).

$$\begin{aligned} X_4(t+1) = {} & QF \times X_{best}(t) - (G_1 \times X(t) \times rand) \\ & - G_2 \times Levy() + rand \times G_1 \end{aligned} \tag{13}$$

$X_4(t+1)$ is the solution of the $(t+1)$ th iteration produced by the fourth search method (X_4). QF is calculated using Eq. (14) and represents a quality function used to balance search strategies. G_1 calculated using Eq. (15) represents the different movements of the AO for tracking the prey during evasion. G_2, calculated using Eq. (16), is a decreasing value from 2 to 0, denoting the slope of the flight for the AO for tracking prey during evasion from the first position to the final position. $X(t)$ is the current solution in the t th iteration (Abualigah et al. 2021b).

$$QF(t) = t^{\frac{2 \times rand - 1}{(1-T)^2}} \tag{14}$$

$$G_1 = 2 \times rand - 1 \tag{15}$$

$$G_2 = 2 \times \left(1 - \frac{t}{T}\right) \tag{16}$$

$QF(t)$ is the quality function value of the tth iteration. The flowchart of the AO is given in Fig. 1.

2.2 Tangent search algorithm

The Tangent Search Algorithm is developed based on the tangent function used to explore the search area well. In TSA, the equations of motion are performed with a spherical step "$step \times \tan(\theta)$". As in Levy flight, it is called tangent flight because the tangent function performs the flight function.

Having a better balance between exploration and intensification (exploitation) will make the optimization algorithm more successful. Too much intensification causes algorithms to converge quickly to the local minimum, and too much exploration slows down and sometimes differentiates the algorithm too much. To reach a balance between intensification and exploration, TSA is built from 3 main components: intensification, exploration, and escape from local minimum. Also, the local minimum escape procedure is applied to a

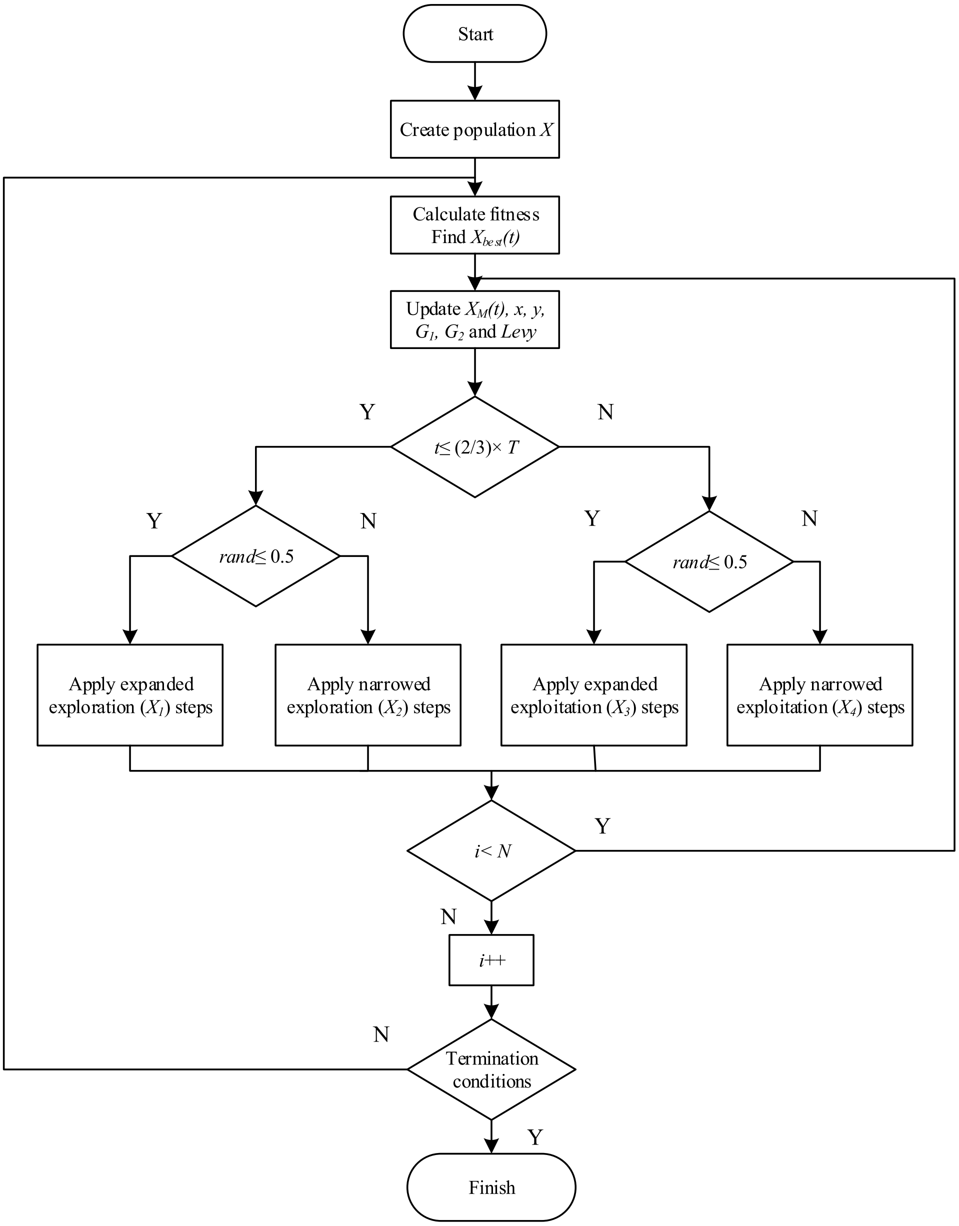

Start
Create population X
Calculate fitness
Find $X_{best}(t)$
Update $X_M(t)$, x, y, G_1, G_2 and $Levy$
Y
N
$t \leq (2/3) \times T$
Y
N
$rand \leq 0.5$
Y
N
$rand \leq 0.5$
Apply expanded exploration (X_1) steps
Apply narrowed exploration (X_2) steps
Apply expanded exploitation (X_3) steps
Apply narrowed exploitation (X_4) steps
$i < N$
Y
N
i++
N
Termination conditions
Y
Finish

Fig. 1 Flowchart of AO

random individual (solution) in each iteration to avoid getting stuck in the local minimum. As in other population-based algorithms, in TSA, individuals in the first population are calculated using Eq. (1) to be evenly distributed within the boundaries of the solution space (Layeb 2021).

2.2.1 Intensification phase

In this phase, TSA first performs a random local walk according to Eq. (17) or Eq. (18). Some of the decision variables of the solution obtained (for the problems with size larger than four, the variable rate is 20%, while for the problems with four dimensions or less, it is 50%) are replaced with the values of the corresponding variables in the current best solution.

$$X_i(t+1) = X_i(t) + step \times \tan(\theta) \times \left(X_i(t) - X_{best}(t)\right) \quad (17)$$

$$X_i(t+1) = X_{best}(t), \text{ if value}i\text{is selected} \quad (18)$$

As a result, the resulting solution $X_i(t+1)$ has a similarity rate of less than 50% with the optimal available solution. If the values of the found solution exceed the LB and UB limits of the problem, Eq. (19) is used for correction (Layeb 2021).

$$X = rand \times (UB - LB) + LB \quad (19)$$

2.2.2 Exploration phase

Unlike local search methods, it has a large exploration capacity due to the global random walk. TSA uses tangent flight and variable step size for global random walking. By using the tangent function, the search area is explored more efficiently. In fact, θ being close to $\pi/2$ will increase the tangent value and the resulting solution will be far from the current solution. With a value of θ near to 0, the tangent function will have small values, and the solution will be similar to the current solution. Therefore, Eq. (20) belonging to the exploration phase converges between local and global random walk. For exploratory search, this equation used is applied with probability $1/D$ to each variable (Layeb 2021).

$$X_i(t+1) = X_i(t) + step \times \tan(\theta) \tag{20}$$

2.2.3 Escape from the local minimum

TSA includes a mechanism that uses a certain procedure to escape the local minimum stagnation problem. The procedure consists of two parts executed with the *Pesc* probability value. A random agent search is selected at each iteration, and then either Eq. (21) or Eq. (22) is used. In addition, the new random solutions can replace the worst solutions with probability 1% (Layeb 2021). In Eq. (21), *sign*() represents the signum function that returns the sign of given number.

$$\begin{aligned} X_i(t+1) =& X_i(t) + (15 \times sign(rand - 0.5)/\log(1+t)) \\ & \times \left(X_{best}(t) - rand \times \left(X_{best}(t) - XX_i(t)\right)\right) \end{aligned} \tag{21}$$

$$X_i(t+1) = X_i(t) + \tan(\theta) \times (UB - LB) \tag{22}$$

Pswitch, *Pesc*, *step*, and θ are the basic parameters used to highlight intensification and exploration search. The balance between global and local random walks is aimed at controlling the parameter of switching, $Pswitch\epsilon[0,1]$. The parameter $Pesc\epsilon[0,1]$ is the escape procedure probability. For guiding and highlighting the intensification and exploration capability, the *step* parameter is used. A variable step size is used in TSA to get a good approximation of the best solution and avoid lack of precision. At the beginning of the search, a large step size is adopted as the search process progresses, and the step size is non-linearly decreased during iterations. By this adaptive size behavior, a good balance between intensification and exploration is aimed in TSA. The step size is influenced by the tangent flights that give an oscillating and periodic behavior to the search. This method utilizes a nonlinear reduction scheme for logarithm function-based adaptive step size to adapt the exploration and intensification search process. A fine convergence is aimed with the help of the logarithm function. On the other hand, it is seen that better results are obtained when different step size functions are used (Layeb 2021). Therefore, to be more efficient, two step size variants are used in TSA. In the intensification search, the first variant is used and is calculated as given in Eq. (23). The second step size is calculated using Eq. (24) in the exploration search.

$$step1 = 10 \times sign(rand - 0.5) \times norm\left(X_{best}(t)\right) \times \log(1 + 10 \times D/t) \tag{23}$$

$$step2 = sign(rand - 0.5) \times norm\left(X_{best}(t) - X_i(t)\right)/\log(20 + t) \tag{24}$$

Here, *norm*() is a specific mathematical norm. The $sign(-,+)$ component controls the direction of the exploration and intensification phase (Layeb 2021). The flowchart of TSA is given in Fig. 2.

3 Hybrid Aquila optimizer-tangent search algorithm

High exploration and exploitation abilities are required in solution search and optimization methods. The exploration phase's goal is to thoroughly investigate the search space and identify the most promising potential solutions. The exploitation phase is designed to guide the search process to the best solution feasible for the population. The accuracy and speed of convergence of a metaheuristic method can be enhanced by properly balancing exploitation and exploration

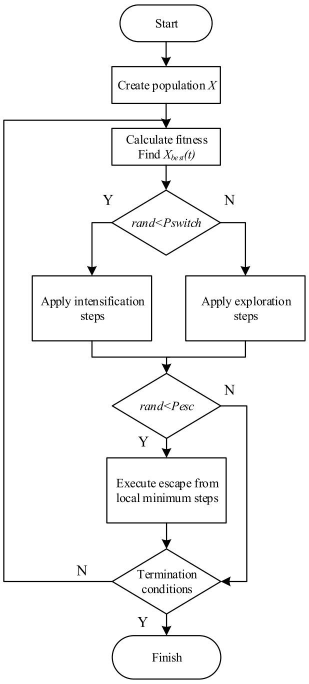


Fig. 2 Flowchart of TSA

performance. A new hybrid algorithm that is stronger than these two algorithms can be obtained by hybridizing an algorithm with strong exploitation ability, but weak exploration ability and an algorithm with strong exploration ability but weak exploitation ability. A more powerful new algorithm is aimed by combining the strong exploitation capability of the first algorithm with the powerful exploration capability of the second algorithm. Thus, the hybridization of the two methods can be used to combine the strengths of each method in a single approach and take advantage of the advantages while eliminating the disadvantages of each method.

In the AO method, while the exploration phase is applied at 2/3 of the iteration, the exploitation phase is insufficient. To give more importance to exploitation capability, the effective intensification phase of TSA is applied instead of the narrowed exploration phase of the AO. Therefore, it is aimed to reach the optimum solution earlier by strengthening the exploitation stage of the hybrid AO-TSA. In this paper, a hybrid AO-TSA is proposed by integrating the intensification steps of TSA into the narrowed exploration phase of the AO. This proposed hybrid algorithm finds the global solution faster without getting stuck in the local optimum, by properly balancing the exploitation and exploration phase. Finally, in order to escape from the local minimum stagnation problem, the escape from local minimum steps used by TSA are also implemented in this new hybrid algorithm. The flow diagram of the proposed hybrid AO-TSA is shown in Fig. 3.

4 Test functions and engineering design problems

Generally, well-defined and complex functions are used to define standard measuring of the optimization methods. Standard problems and interfaces to search and optimization problems are already specified in order to compare different optimization algorithms on different types of search and optimization problems. Six unimodal (Sphere, Rosenbrock, Quartic, Schwefel's 1.20, Schwefel's 2.21, and Schwefel's 2.22), six multimodal (Schwefel, Levy Function, Ackley, Griewank, Penalized, and Rastrigin), and six fixed-dimension multimodal (Foxholes, Kowalik, Goldstein-Price, Shekel 7, Shekel 10, and Six Hump Camel) benchmark functions were used to compare the performances of the AO-TSA, AO, TSA, SCA (Mirjalili 2016), WOA (Mirjalili and Lewis 2016), I-GWO (Nadimi-Shahraki et al. 2021), CSA (Askarzadeh 2016), and BO (Das and Pratihar 2019) algorithms. In addition, three of the CEC 2019 benchmark functions (Storn's Chebyshev Polynomial Fitting Problem, Lennard–Jones Minimum Energy Cluster, and Inverse Hilbert Matrix Problem) were used. The equations, parameters, minimum values, and problem size of these twenty-one test functions are shown in Table 1.

Two engineering design problems (three bar truss design and tension/compression spring design) were also used to test the efficiency of the proposed method. In tension/compression spring design, which is a continuous constrained problem, it is aimed to find the best values of the mean coil diameter, number of active coils, and wire diameter parameters in order to minimize the tension/compression spring weight. The mathematical expression of the problem is shown in Eq. (25).

$$
\begin{aligned}
&\min f(x) = \left(x_3 + 2\right) x_2 x_1^2 \\
&g_1(x) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \le 0 \\
&g_2(x) = \frac{4x_2^2 - x_1 x_2}{12566\left(x_2 x_1^3 - x_1^4\right)} + \frac{1}{5108 x_1^2} - 1 \le 0 \\
&g_3(x) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \le 0 \\
&g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \le 0 \\
&2 \le x_1 \le 15, 0.25 \le x_2 \le 1.3, 0.05 \le x_3 \le 2
\end{aligned}
\tag{25}
$$

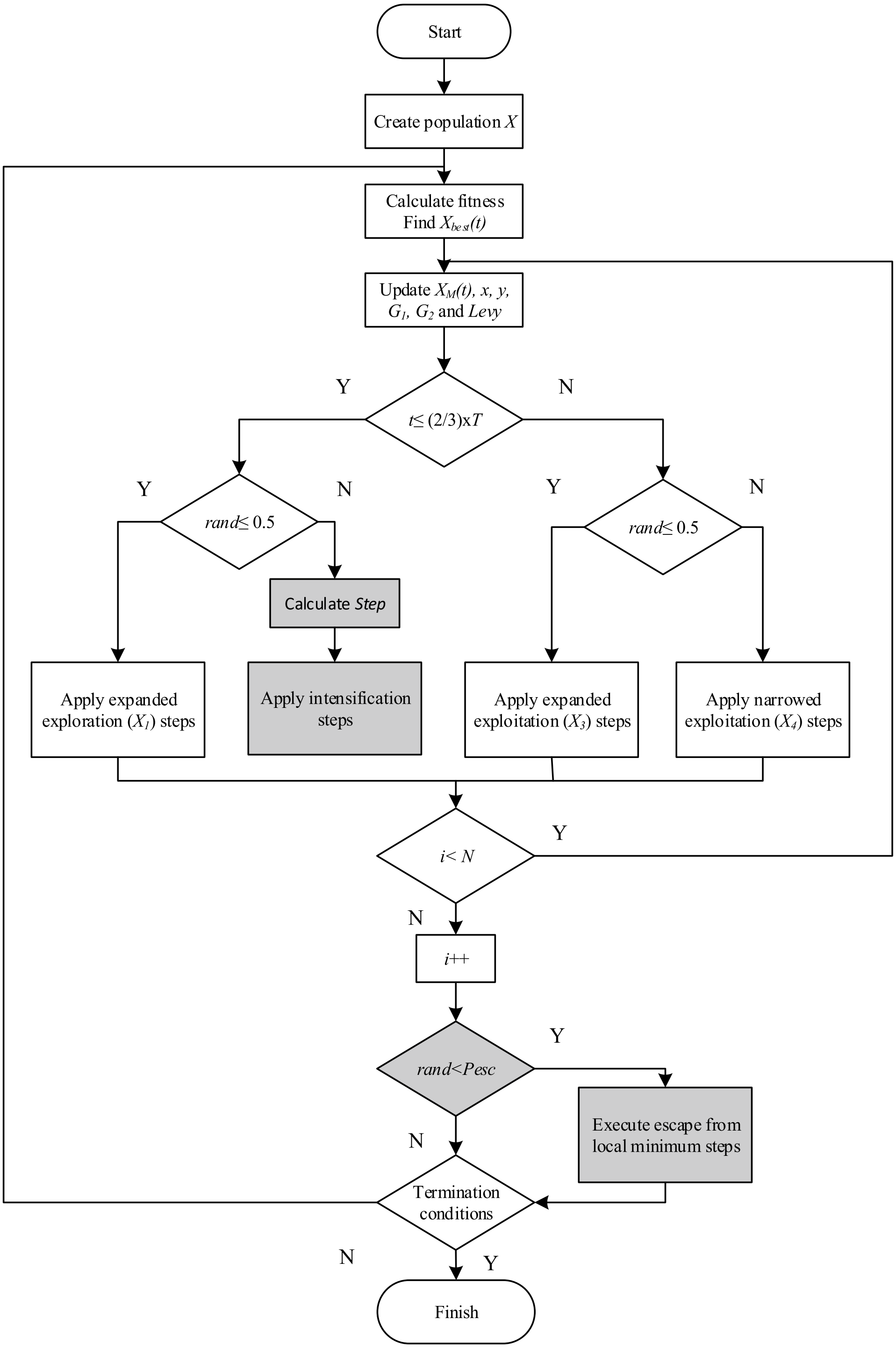
In the problem of three bar truss design, which aims to minimize the weight of a three-bar truss, the objective function is simple. However, as with other structural design problems, there are many constraints. These constraints are buckling, deflection, and stress. Equation (26) depicts the problem's mathematical expression.

$$
\begin{aligned}
&\min f(x) = \left(2\sqrt{2}x_1 + x_2\right) \times l \\
&g_1(x) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2} P - \sigma \le 0 \\
&g_2(x) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1 x_2} P - \sigma \le 0 \\
&g_3(x) = \frac{1x_2}{\sqrt{2x_2 + x_1}} P - \sigma \le 0 \\
&0 \le x_1, x_2 \le 1, \\
&l = 100cm, P = 2\frac{KN}{cm^2}, \sigma = KN/cm^2
\end{aligned}
\tag{26}
$$

5 Experimental results

In this study, for all experiments, the initial population size of the algorithms used was 30. All algorithms were started and run under equal conditions. The number of function

Fig. 3 Flowchart of hybrid AO-TSA


Start
Create population X
Calculate fitness
Find $X_{best}(t)$
Update $X_M(t)$, x, y, G_1, G_2 and $Levy$
$t \leq (2/3) \times T$
Y
N
$rand \leq 0.5$
Y
N
$rand \leq 0.5$
Y
N
Calculate *Step*
Apply expanded exploration (X_1) steps
Apply intensification steps
Apply expanded exploitation (X_3) steps
Apply narrowed exploitation (X_4) steps
$i < N$
Y
N
i++
$rand < Pesc$
Y
N
Execute escape from local minimum steps
Termination conditions
N
Y
Finish

evaluations was used as the termination condition of the algorithms. Accordingly, the algorithm was terminated when the number of function evaluations reaches 10,000. Each algorithm was run 20 times for each test function and obtained results are presented. Standard values for the parameters of AO and TSA were used in the experiments.

Tables 2 and 3 show the best, worst, mean, and standard deviation values as "Best", "Worst", "Mean", and "Std" respectively obtained after the Hybrid AO-TSA, AO, TSA, WOA, SCA, I-GWO, BO, and CSA were run 20 times for each test function. According to these tables, the AO-TSA algorithm gives the best results for Sphere, Rosenbrock, Schwefel's 2.21, Schwefel, Ackley, Griewank, Rastrigin, Foxholes, Kowalik, Shekel 7, Shekel 10, Six Hump Camel, and Lennard–Jones Minimum Energy Cluster test functions. The worst results were obtained from CSA in general. The

Table 1 Test functions

Name	Function Equation	Parameters	MIN	Problem Size (D)
Sphere	$f_1(x) = \sum_{i=1}^{N} x_i^2$	$-100 < x_i < 100$	0	10
Rosenbrock	$f_2(x) = \sum_{i=1}^{N-1}[100\left(x_i^2 - x_{i+1}\right)^2 + (1 - x_i)^2]$	$-30 < x_i < 30$	0	10
Quartic	$f_3(x) = \sum_{i=0}^{N} ix_i^4 + rand[0, 1)$	$-1.28 < x_i < 1.28$	0	10
Schwefel's 1.20	$f_4(x) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j\right)^2$	$-100 < x_i < 100$	0	10
Schwefel's 2.21	$f_5(x) = max_i\{\lvert x_i\rvert, 1 \le i \le N\}$	$-100 < x_i < 100$	0	10
Schwefel's 2.22	$f_6(x) = \sum_{i=0}^{N} \lvert x_i\rvert + \prod_{i=0}^{N} \lvert x_i\rvert$	$-10 < x_i < 10$	0	10
Schwefel	$f_7(x) = \sum_{i=1}^{N}(-x_i\sin(\sqrt{\lvert x_i\rvert}))$	$-500 < x_i < 500$	$-418.9829 \times \mathrm{n}$	10
Levy Function	$f_8(x) = 0.1\left(sin^2\left(3\pi x_1\right) + \sum_{i=1}^{n}\left(x_i - 1\right)^2\left[1 + sin^2\left(3\pi x_i + 1\right)\right] + \left(x_n - 1\right)^2\left[1 + sin^2\left(2\pi x_n\right)\right]\right)$ $+ \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	$-50 < x_i < 50$	0	10
Ackley	$f_9(x) = -20exp\left[-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right] - exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + e + 20$	$-32 < x_i < 32$	0	10
Griewank	$f_{10}(x) = 1 + \frac{1}{4000}\sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N}\cos\left(\frac{x_i}{\sqrt{i}}\right)$	$-600 < x_i < 600$	0	10
Penalized	$f_{11}(x) = \frac{\pi}{N}\left\{10\sin\left(\pi y_1\right)\right\} + \sum_{i=1}^{N-1}(y_i - 1)^2\left[1 + 10sin^2\left(\pi y_{i+1}\right) + \sum_{i=1}^{N}u\left(x_i, 10, 100, 4\right)\right],$ $where y_i = 1 + \frac{x_i+1}{4}, u(x_i, a, k, m)\left\{\begin{array}{cc} K(x_i - a)^m & if x_i > a \\ 0 & -a \le x_i \ge a \\ K(-x_i - a)^m & -a \le x_i \end{array}\right.$	$-50 < x_i < 50$	0	10
Rastrigin	$f_{12}(x) = \sum_{i=1}^{N}\left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	$-5.12 < x_i < 5.12$	0	10
Foxholes	$f_{13}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{i,j})^6}\right)^{-1}$	$-65.536 < x_i < 65.536$	1	2
Kowalik	$f_{14}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2+b_i x_2)}{b_i^2+b_i x_3+x_4}\right]^2$	$-5 < x_i < 5$	0.00030	4
Goldstein-Price	$f_{15}(x) = \left[1 + \left(x_1 + x_2 + 1\right)^2\left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)\right]\left[30 + \left(2x_1 - 3x_2\right)^2\left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)\right]$	$-2 < x_i < 2$	3	2
Shekel 7	$f_{16}(x) = -\sum_{i=1}^{7}\left[\left(X - a_i\right)\left(X - a_i\right)^T + c_i\right]^{-1}4$	$0 < x_i < 1$	4	-10.4028
Shekel 10	$f_{17}(x) = -\sum_{i=1}^{10}\left[\left(X - a_i\right)\left(X - a_i\right)^T + c_i\right]^{-1}4$	$0 < x_i < 1$	4	-10.5363
Six Hump Camel	$f_{18}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	$-5 < x_i < 5$	2	-1.0316

Table 1 (continued)

Name	Function Equation	Parameters	MIN	Problem Size (D)
Storn's Chebyshev Polynomial Fitting Problem	$f_{19}(x) = p_1 + p_2 + p_3$ $p_1 = \begin{cases} (u-d)^2 ifu < d, \\ 0otherwise; \end{cases} \quad u = \sum_{j=1}^{D} x_j (1.2)^{D-j}$ $p_2 = \begin{cases} (v-d)^2 ifv < d, \\ 0otherwise; \end{cases} \quad v = \sum_{j=1}^{D} x_j (-1.2)^{D-j}$ $p_k = \begin{cases} \left(w_k - 1\right)^2 ifw_k > 1 \\ \left(w_k + 1\right)^2 ifw_k < 1 \\ 0otherwise; \end{cases} \quad w_k = \sum_{j=1}^{D} x_j \left(\frac{2k}{m} - 1\right)^{D-j}$ $p_3 = \sum_{k=0}^{m} p_k, k = 0, 1, \ldots, m, m = 32D.$ $d = 72.661$	$-8192 < x_i < 8192$	1	9
Inverse Hilbert Matrix Problem	$f_{20}(x) = \sum_{i=1}^{n} \sum_{k=1}^{n} \left\lvert w_{i,k} \right\rvert$ $(w_{i,k}) = W = HZ - I,\ I = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \ldots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$ $H = \left(h_{i,k}\right), h_{i,k} = \frac{1}{i+k-1}, i, k = 0, 1, \ldots, n, n = \sqrt{D}$ $Z = \left(z_{i,k}\right), z_{i,k} = x_{i+n(k-1)}$	$-16384 < x_i < 16384$	1	16
Lennard–Jones Minimum Energy Cluster	$f_{21}(x) = 12.7120622568 + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(\frac{1}{d_{i,j}^2} - \frac{2}{d_{i,j}}\right),$ $d_{i,j} = \left(\sum_{k=0}^{2} \left(x_{3i+k-2} - x_{3j+k-2}\right)^2\right)^3, n = D/3$	$-4 < x_i < 4$	1	18

best results according to mean values for Sphere, Rosenbrock, Quartic, Schwefel's 1.20, Schwefel's 2.21, Schwefel's 2.22, Schwefel, Levy Function, Ackley, Griewank, Penalized, Rastrigin, Shekel 7, Shekel 10, Six Hump Camel, and Kowalik test functions were obtained from the hybrid AO-TSA. According to results obtained for unimodal benchmark functions, the proposed method seems to achieve the best results in 3 out of 6 functions with respect to the best values while it gives the best results in all unimodal functions concerning mean values. In multimodal benchmark functions, AO-TSA gives the best results in 4 out of 6 functions in terms of the best values, and the best results in all functions in terms of mean values are obtained by the proposed method.

As seen in Table 3, in fixed-dimension multimodal benchmark functions, AO-TSA achieves the best results in 5 out of 6 functions in terms of the best values, and the proposed technique achieves the best results in 4 out of 6 functions in terms of mean values. In CEC 2019 benchmark functions, the high performance obtained for other types of benchmark functions is not achieved by the proposed method. Generally, it is seen that the mean values obtained from AO-TSA for other test functions are better than other algorithms. When the standard deviation values were examined, AO-TSA gave the minimum value for all 6 test functions used. Considering the experiments, it is seen that AO-TSA gives promising results in terms of the standard deviation values compared to other algorithms.

Figure 4 shows the change in the mean fitness value/number of function evaluations of the test results of Hybrid AO-TSA, AO, TSA, WOA, SCA, I-GWO, BO, and CSA in Sphere, Rosenbrock, Quartic, Schwefel's 1.20, Schwefel's 2.21, Schwefel's 2.22, Schwefel, Levy Function, Ackley, Griewank, Penalized, and Rastrigin test functions.

According to the convergence curve demonstrated in Fig. 4, the hybrid AO-TSA reaches the best solution very rapidly for all test functions. Especially in Schwefel 2.21, Schwefel, Ackley, Griewank, and Rastrigin test functions, it gave good results in a very short time compared to other algorithms. Again, according to Fig. 4, the algorithm that reached the best solution in the test functions used was AO-TSA.

Figure 5 shows the change in the mean fitness value/number of function evaluations of the test results of Hybrid AO-TSA, AO, TSA, WOA, SCA, I-GWO, BO, and CSA in Foxholes, Kowalik, Goldstein-Price, Shekel 7, Shekel 10, Six Hump Camel, Storn's Chebyshev Polynomial Fitting Problem, Lennard–Jones Minimum Energy Cluster, and Inverse Hilbert Matrix Problem test functions. According to the convergence curve shown in Fig. 5, it is seen that the convergence rate of AO-TSA is quite good. Especially for Schekel 7 and Schekel 10 test functions, the proposed method converged much faster than other algorithms. In general, it is seen that AO-TSA converges rapidly and achieves better results.

On many benchmark functions, the effect of population size (N) is investigated. Numerous numbers of population size (20, 30, and 50) are investigated across 10,000 function evaluation number to properly analyze the AO-TSA's parameter sensitivity. Sensitivity analysis for population size is shown in Table 4. The results presented in the table indicate that in general, performance of the method increases with higher N values.

Furthermore, the effect of t parameter is investigated on many benchmark functions. 1/2, 2/3, and 3/4 values of the t parameter are used for parameter analysis of the algorithm. Results of sensitivity analysis for t parameter are shown in Table 5. The best results are obtained with the value of $\left(\frac{2}{3}\right) \times T$ for t parameter.

Box plots of the experimental results of the hybrid AO-TSA proposed in this study and the other seven algorithms used for comparison are shown in Figs. 6 and 7. When the box plots in Figs. 6 and 7 are examined, it is seen that the lower quartile, median, and upper quartile values obtained based on 20 runs from AO-TSA in Rosenbrock, Quartic, Schwefel's 1.20, Schwefel's 2.21, Schwefel's 2.22, Levy Function, Griewank, Penalized, Rastrigin, Foxholes, Kowalik, and Storns Chebyshev Polynomial test functions, especially the Schwefel test function, are smaller than the values obtained from other algorithms. It is observed that the lower quartile, median, and upper quartile values obtained from the AO-TSA for the Sphere, Ackley, Golden Stein Price, and Lennard Jones Minimum Energy Cluster test functions are close and small compared to those obtained from other algorithms.

For the problems of three bar truss design and tension/compression spring design, all algorithms were started and run under equal conditions. The optimal results are obtained when each algorithm is run 20 times and the parameter values that give these results are listed in Tables 6 and 7. When the tables are examined, it is seen that for these two problems, the best results are obtained from the proposed AO-TSA. Afterward, BO and I-GWO algorithms give better results, respectively.

In addition, experimental results were evaluated by the Friedman test and it was examined whether there was a statistically significant difference between the results obtained from the algorithms used in this study. This non-parametric

Table 2 Statistical results for unimodal and multimodal test functions

	AO-TSA	AO	TSA	WOA	SCA	I-GWO	CSA	BO
Unimodal								
Sphere								
Best	**4.7E–154**	1E–149	1.21E–39	1.98E–33	7.55E–07	6.77E–37	2.39E–11	2.52E–12
Worst	8.1E–147	9.88E–57	1.07E–26	1.06E–22	0.110803	5.57E–32	0.000498	0.912853
Mean	**8.7E–148**	4.94E–58	9.32E–28	8.99E–24	0.016279	3E–33	3.79E–05	0.047794
Std	**2.1E–147**	2.21E–57	2.88E–27	2.81E–23	0.031767	1.24E–32	0.000112	0.203758
Rosenbrock								
Best	**8.86E–05**	0.000187	0.085826	4.117923	7.827973	5.092199	8.677987	0.109465
Worst	0.034303	0.086325	8.852858	6.736209	23.32113	6.418753	9.042301	18.8698
Mean	**0.007296**	0.008634	6.397952	6.231087	10.22525	5.888638	8.937261	4.448394
Std	**0.01028**	0.018897	2.156126	0.587402	3.964981	0.278459	0.077351	4.216027
Quartic								
Best	1.12E–07	5.75E–07	1.03E–14	0.000175	0.449391	0.000603	0.045067	**1.12E–15**
Worst	2.61E–05	0.000129	0.043949	0.053844	1.366444	0.00371	0.179443	0.043949
Mean	**5.41E–06**	2.99E–05	0.01303	0.006058	0.716828	0.00113	0.102158	0.00879
Std	**6.91E–06**	3.50E–05	0.012952	0.013652	0.223815	0.000689	0.036993	0.009827
Schwefel's 1.20								
Best	4.28E–85	4.93E–52	1.29E–57	6.80E–03	6.58E+03	4.12E–01	2.40E–17	**0**
Worst	1.25E–74	9.31E–43	7.82E–38	2.31E+03	3.90E+04	3.16E+02	5.35E–05	0.075742
Mean	**1.16E–75**	5.10E–44	5.33E–39	4.88E+02	1.62E+04	3.25E+01	1.37E–05	1.19E–02
Std	**3.55E–75**	2.08E–43	1.79E–38	5.85E+02	6.72E+03	6.99E+01	1.86E–05	1.99E–02
Schwefel's 2.21								
Best	**1.60E–45**	1.94E–28	1.08E–33	4.81E–14	4.96E+01	2.78E–01	4.69E–08	3.32E–04
Worst	1.33E–40	3.01E–25	6.00E–19	7.31E+01	8.11E+01	2.38E+00	2.12E–02	4.45E+00
Mean	**1.24E–41**	4.63E–26	3.06E–20	2.04E+01	6.52E+01	8.35E–01	1.93E–03	6.21E–01
Std	**2.98E–41**	7.36E–26	1.34E–19	2.51E+01	9.88E+00	6.44E–01	4.77E–03	1.08E+00
Schwefel's 2.22								
Best	2.83E–45	7.29E–28	1.70E–27	9.18E–43	2.32E+01	2.16E–04	6.08E–06	**0**
Worst	1.73E–42	6.81E–25	6.53E–18	6.47E–40	9.51E+02	3.78E–03	2.67E–02	0.000086
Mean	**5.86E–43**	1.23E–25	3.28E–19	4.84E–41	1.87E+02	1.06E–03	2.65E–03	4.39E–06
Std	**5.40E–43**	1.89E–25	1.46E–18	1.46E–40	2.80E+02	1.08E–03	6.55E–03	1.93E–05
Multimodal								
Schwefel								
Best	**– 4189.83**	– 3382.47	– 837.966	– 4188.63	– 2366.05	– 3723.08	– 38.4806	– 4071.39
Worst	– 4189.16	– 1888.33	– 837.966	– 2248.8	– 1868.56	– 2582.79	– 33.9503	– 3479.2
Mean	**– 4189.68**	– 2560.64	– 837.966	– 3199.73	– 2074.72	– 3275.2	– 36.9236	– 3799.89
Std	0.204603	461.0238	**2.13E–07**	665.1424	121.6722	285.8669	1.216046	152.9926
Levy Function								
Best	1.12E–07	5.75E–07	1.03E–14	0.000175	0.449391	0.000603	0.045067	**1.12E–15**
Worst	2.61E–05	0.000129	0.043949	0.053844	1.366444	0.00371	0.179443	0.043949
Mean	**5.41E–06**	2.99E–05	0.01303	0.006058	0.716828	0.00113	0.102158	0.00879
Std	**6.91E–06**	3.5E–05	0.012952	0.013652	0.223815	0.000689	0.036993	0.009827
Ackley								
Best	**8.88E–16**	8.88E–16	8.88E–16	4.44E–15	3.6E–06	4.44E–15	8.58E–05	1.11E–12
Worst	8.88E–16	4.44E–15	2.316849	4.1E–12	1.599691	7.99E–15	0.010407	1.155149
Mean	**8.88E–16**	2.84E–15	0.115874	3.42E–13	0.13784	4.62E–15	0.002403	0.057759
Std	**0**	1.81E–15	0.518056	9.38E–13	0.384317	7.94E–16	0.00282	0.258299
Griewank								
Best	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.60E+01	2.84E–01	4.39E–11	5.17E–02
Worst	0.00E+00	8.72E–01	2.51E–01	8.37E–01	1.68E+02	7.31E–01	4.33E–03	3.27E–01

Table 2 (continued)

	AO-TSA	AO	TSA	WOA	SCA	I-GWO	CSA	BO
Mean	**0.00E + 00**	4.36E–02	4.66E–02	1.36E–01	1.16E + 02	5.73E–01	5.80E–04	1.56E–01
Std	**0.00E + 00**	1.95E–01	6.99E–02	2.88E–01	3.22E + 01	1.19E–01	1.14E–03	7.96E–02
Penalized								
Best	9.09E–11	1.36E–09	5.12E–09	1.20E–02	3.13E + 04	1.15E–03	9.77E–02	**3.47E–15**
Worst	1.60E–07	2.28E–05	2.07E–01	2.87E–01	2.07E + 08	3.68E–01	3.36E–01	6.22E–01
Mean	**6.17E–08**	2.46E–06	2.07E–02	6.72E–02	8.23E + 07	5.09E–02	1.96E–01	1.55E–01
Std	**5.29E–08**	5.29E–06	5.42E–02	7.30E–02	6.04E + 07	1.06E–01	6.49E–02	2.14E–01
Rastrigin								
Best	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	7.06E + 01	2.56E + 01	2.14E–11	**0.00E + 00**
Worst	0.00E + 00	1.42E–14	1.59E + 01	2.84E–14	1.44E + 02	5.08E + 01	9.25E–04	3.09E + 00
Mean	**0.00E + 00**	7.11E–16	3.33E + 00	2.84E–15	1.18E + 02	4.08E + 01	1.37E–04	1.10E + 00
Std	**0.00E + 00**	3.18E–15	5.11E + 00	8.75E–15	1.81E + 01	6.34E + 00	2.90E–04	1.03E + 00

test is used to describe the differences in the behavior of multiple algorithms. Test cases are expressed in rows while the outcomes of comparing algorithms are shown in columns in the Friedman test. In the experiments, the Null Hypothesis (H_0) is "There is not a meaningful difference between the fitness function values of the compared algorithms and those of the proposed hybrid AO-TSA". Alternative Hypothesis (H_1) is "There is a meaningful difference between the fitness function values of the compared algorithms and those of the proposed hybrid AO-TSA". Analysis results with the Friedman test are shown in Table 8. The alpha value was determined as 0.05 and the degrees of freedom (Df) (number of samples compared-1) was 7. According to the alpha value and the degrees of freedom, it is seen that the value of x_F^2 is 14.067, from the chi-square distribution table. When Table 8 is examined, it is seen that the x^2 value is greater than the x_F^2 value $(x^2 > x_F^2)$ for all test functions. Accordingly, the Null Hypothesis (H_0) is rejected and the Alternative Hypothesis (H_1) is accepted. In other words, there is a statistically significant difference between the fitness function values of the compared algorithms and those of the proposed hybrid AO-TSA according to the Friedman test.

6 Conclusions

Metaheuristic approaches have grown in popularity in recent years as a result of their high computational power and ease of conversion. There is, however, no algorithm that can provide the optimal solution to all problems. As a result, new metaheuristic algorithms are being proposed and current algorithms are being improved. Exploration and exploitation skills should be included in metaheuristic algorithms. One of these two capabilities may be sufficient in some metaheuristic algorithms, while the other may be insufficient. In this study, a new hybrid method, AO-TSA, was proposed by combining the TSA intensification phase with the limited exploration stage to improve AO's exploitation ability. In addition, the local minimum escape steps of TSA are also applied in the new hybrid algorithm. By combining the strengths of these two metaheuristic algorithms, a new more efficient general-purpose hybrid algorithm was presented as a solution search methodology for complex optimization problems.

Six unimodal, six multimodal, six fixed-dimension multimodal, and three modern CEC 2019 benchmark functions were used to compare the performances of AO-TSA, AO, TSA, WOA, SCA, I-GWO, CSA, and BO. When the mean results obtained by running 20 times were examined; it was observed that, generally, hybrid AO-TSA reaches better solutions earlier than the other algorithms according to the convergence curves. The best or near best values for the Best, Mean, and Standard Deviation criteria in multimodal test functions were found by AO-TSA. Furthermore, in unimodal benchmark functions, the best mean values are obtained from the proposed hybrid AO-TSA. The best results for multimodal benchmark functions are achieved by the proposed method. The high performance obtained for other types of benchmark functions is not achieved by the proposed algorithm in CEC 2019 benchmark functions. Finally,

Table 3 Statistical results for fixed-dimension multimodal and CEC 2019 test functions

Fixed–Dimension Multimodal								
Foxholes								
Best	0.998004	0.998004	0.998004	0.998004	0.998033	0.998004	12.67051	0.998004
Worst	1.103655	12.67051	21.07269	10.76318	2.982105	0.998004	12.67051	0.998004
Mean	1.014715	3.817256	8.887892	3.68741	1.855165	**0.998004**	12.67051	**0.998004**
Std	0.034192	4.414279	5.043994	4.236413	0.909152	1.79E–11	**4.35E–13**	9.88E–10
Kowalik								
Best	**0.000342**	0.000438	0.000628	0.000349	0.000575	0.000501	0.000739	0.000424
Worst	0.003498	0.012201	0.020363	0.002237	0.001861	0.001247	0.002193	**0.020363**
Mean	**0.000624**	0.001942	0.005693	0.000859	0.001209	0.00074	0.001297	0.008681
Std	0.00068	0.002452	0.008691	0.000534	0.000381	**0.000295**	0.000344	0.009791
Golden Stein Price								
Best	3.004974	3.007641	**3**	3.000016	3.000311	**3**	3.001175	**3**
Worst	3.147442	9.765201	30	3.000065	3.007065	3	3.043565	3
Mean	3.073239	4.194818	6.857143	3.000033	3.001866	**3**	3.013311	**3**
Std	0.037193	1.577673	9.804686	1.35E–05	0.001758	9.43E–13	0.012615	**4.89E–15**
Shekel 7								
Best	**–1.04E + 01**	**–1.04E + 01**	**–1.04E + 01**	–1.03E + 01	–1.20E + 00	–1.03E + 01	–5.34E + 00	**–1.04E + 01**
Worst	–1.04E + 01	–1.01E + 01	–1.04E + 01	–8.83E–01	–4.42E–01	–5.13E + 00	–1.23E + 00	–1.84E + 00
Mean	**–1.04E + 01**	**–1.04E + 01**	**–1.04E + 01**	–5.62E + 00	–6.78E–01	–7.47E + 00	–3.04E + 00	–5.17E + 00
Std	3.46E–04	7.65E–02	**0.00E + 00**	2.99E + 00	2.32E–01	2.08E + 00	1.07E + 00	3.22E + 00
Shekel 10								
Best	**–1.05E + 01**	**–1.05E + 01**	**–1.05E + 01**	–1.04E + 01	–1.62E + 00	**–1.05E + 01**	–5.01E + 00	**–1.05E + 01**
Worst	–1.05E + 01	–1.05E + 01	–1.05E + 01	–2.37E + 00	–5.70E–01	–5.12E + 00	–1.38E + 00	–2.42E + 00
Mean	**–1.05E + 01**	–1.05E + 01	–1.05E + 01	–5.68E + 00	–8.66E–01	–7.34E + 00	–2.94E + 00	–6.42E + 00
Std	2.52E–03	1.24E–02	**5.47E–15**	2.56E + 00	2.22E–01	2.33E + 00	9.77E–01	3.85E + 00
Six Hump Camel								
Best	**–1.03E + 00**	–1.03E + 00	–1.03E + 00	–1.03E + 00	–1.03E + 00	–1.03E + 00	–1.03E + 00	–1.03E + 00
Worst	–1.03E + 00	–9.26E–01	–2.15E–01	–1.03E + 00	2.23E + 00	–1.03E + 00	–1.03E + 00	–1.03E + 00
Mean	**–1.03E + 00**	–1.02E + 00	–9.50E–01	–1.03E + 00	–1.10E–01	–1.03E + 00	–1.03E + 00	–1.03E + 00
Std	1.62E–04	2.54E–02	2.51E–01	5.30E–08	9.45E–01	2.88E–12	5.60E–05	0.00E + 00
CEC 2019								
Storns Chebyshev Polynomial								
Best	53,579.84	75,397.47	57,505.87	2.13E + 09	1.19E + 10	3.19E + 08	**50,262.18**	27,231,871
Worst	127,977.1	407,644.3	832,996.4	1.88E + 11	1.53E + 11	5.96E + 09	60,547.57	5.9E + 09
Mean	83,329.26	132,294.5	344,530.3	6.98E + 10	4.33E + 10	2.99E + 09	**54,517.57**	6.24E + 08
Std	20,358.19	85,240.25	270,357.9	6.13E + 10	3.12E + 10	1.6E + 09	**3269.862**	1.28E + 09
Inverse Hilbert Matrics Problem								
Best	17.39732	17.61456	17.631	17.3477	17.51053	17.34405	17.40818	**17.34286**
Worst	17.85994	17.89831	20.6038	17.35847	17.92798	17.34618	17.4459	17.34286
Mean	17.60954	17.74257	18.6201	17.35081	17.6632	17.34455	17.4237	**17.34286**
Std	0.111446	0.072811	0.688797	0.002725	0.113481	0.000475	0.010425	**1.58E–11**
Lennard Jones Min. Energy Cluster								
Best	**12.7024**	12.70267	**12.7024**	**12.7024**	12.70245	12.70241	12.70262	**12.7024**
Worst	12.7028	12.70586	12.7024	12.70241	12.70312	12.70242	12.70393	12.7024
Mean	12.7025	12.70373	**12.7024**	**12.7024**	12.7026	12.70241	12.70304	**12.7024**
Std	8.29E–8	0.000907	**6.59E–14**	2.91E–07	0.000181	4.38E–06	0.00037	1.74E–07

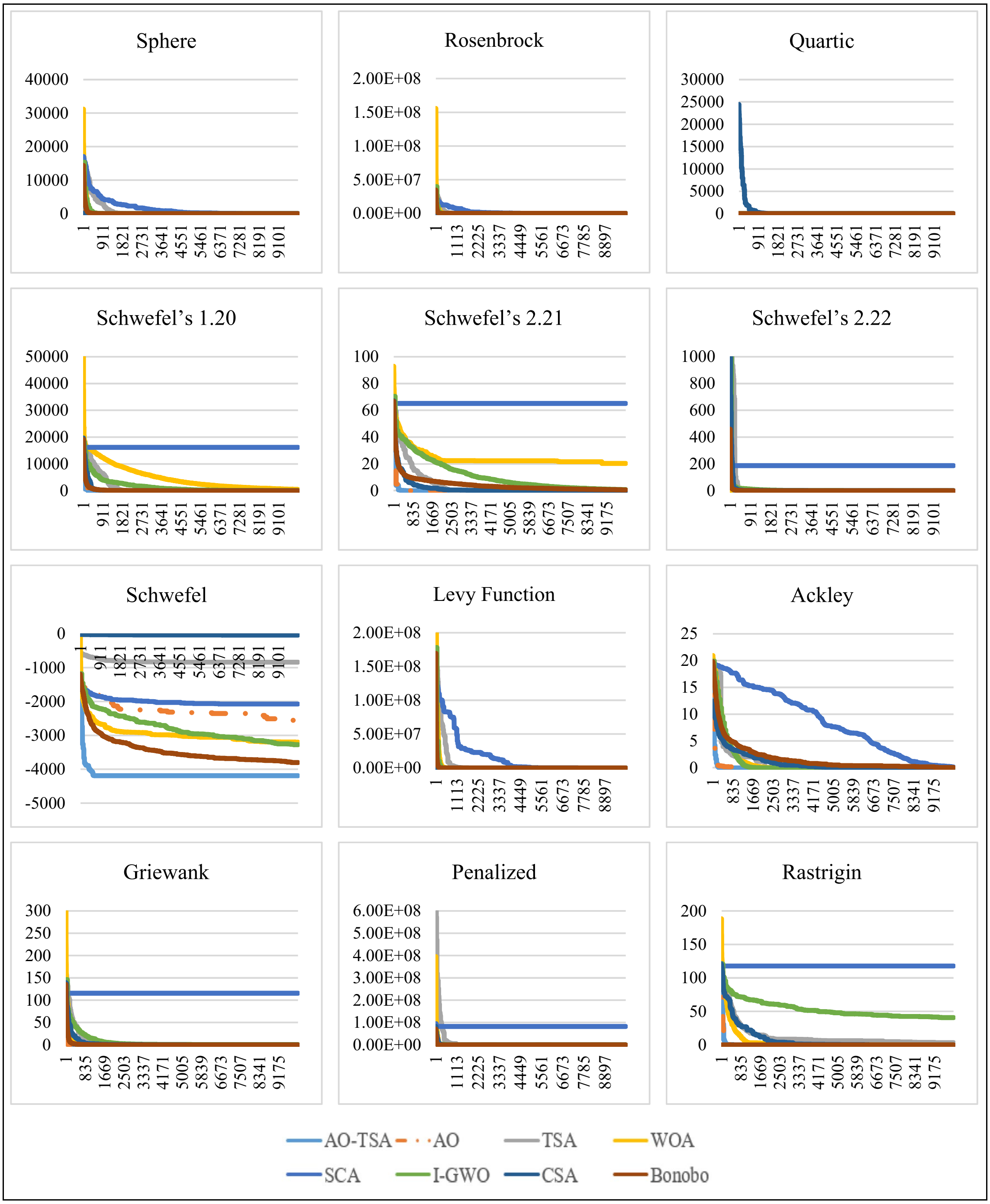


Fig. 4 Change of the fitness values according to the number of function evaluations

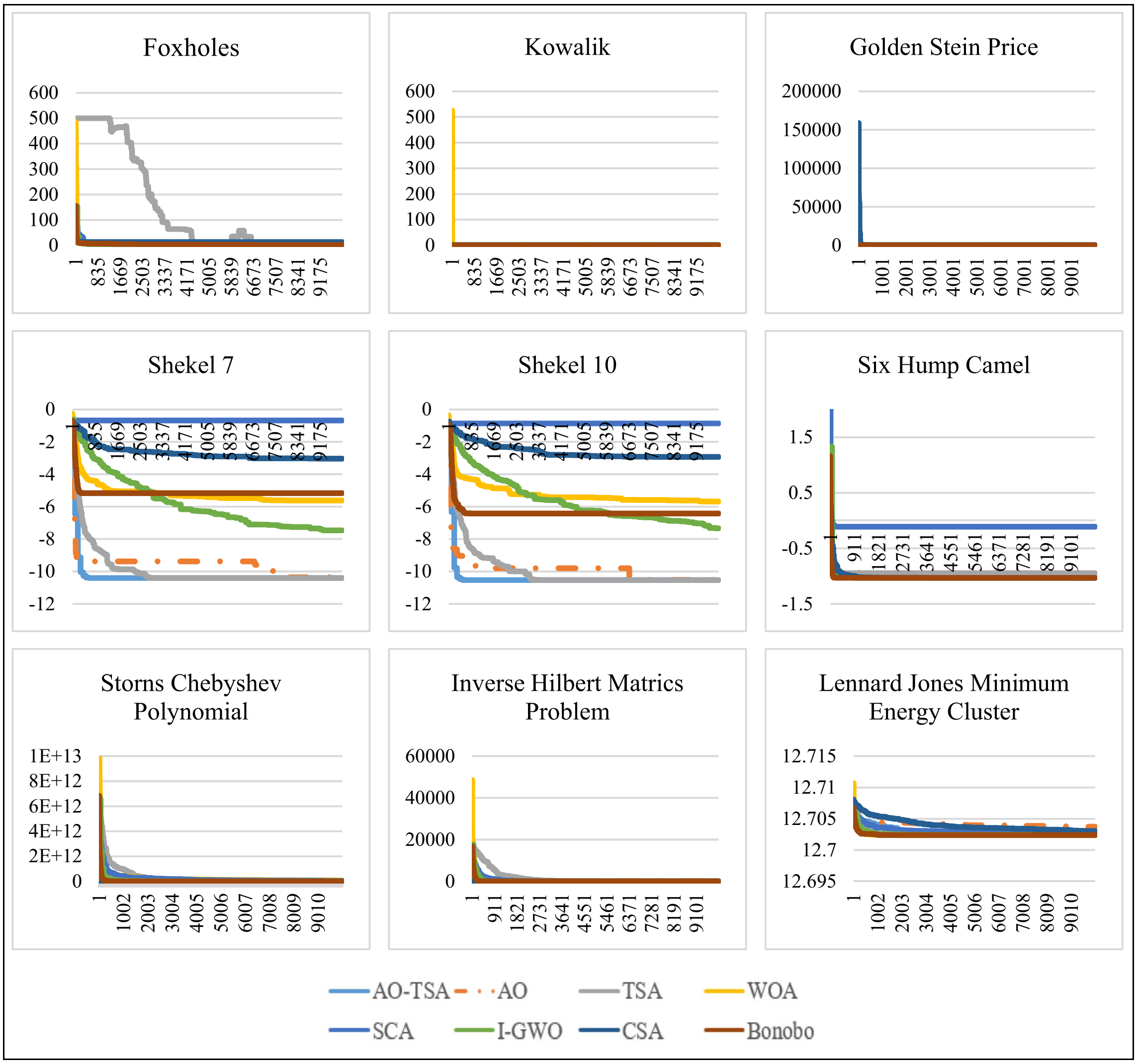


Fig. 5 Change of the fitness values according to the number of function evaluations

when Friedman analysis of the results was performed, statistically significant differences were found. Despite the fact that the results show that hybrid AO-TSA is an effective method for global optimization, the requirement for parameters for the hybrid method a priori appears to be an AO-TSA limitation. However, in order to eliminate this limitation, new approaches can be proposed for adaptively specifying the algorithm parameters.

In future studies, it is planned to use different versions of this algorithm in real-world problems by proposing and making it multi-objective. Different local search methods can be integrated into this method to increase the accuracy and search power of hybrid AO-TSA. In addition, distributed and parallel versions of this algorithm can be developed for future studies.

Table 4 Sensitivity analysis of AO-TSA for population size (*N*)

	$N=20$	$N=30$	$N=50$
Sphere			
Best	0.00E+00	4.70E-154	5.76E-160
Worst	4.50E–263	8.10E–147	1.97E–143
Mean	2.37E–264	8.70E–148	9.88E–145
Std	0.00E+00	2.10E–147	4.41E–144
Rosenbrock			
Best	7.12E–04	8.86E–05	5.20E–08
Worst	7.06E+00	3.43E–02	2.16E+00
Mean	1.88E+00	7.30E–03	1.55E–01
Std	2.45E+00	1.03E–02	5.01E–01
Quartic			
Best	1.36E–05	1.12E–07	5.70E–06
Worst	6.42E–04	2.61E–05	6.59E–05
Mean	2.05E–04	5.41E–06	3.80E–05
Std	1.86E–04	6.91E–06	2.12E–05
Schwefel			
Best	– 4.19E+03	– 4.19E+03	– 4.19E+03
Worst	– 4.19E+03	– 4.19E+03	– 4.19E+03
Mean	– 4.19E+03	– 4.19E+03	– 4.19E+03
Std	9.12E–01	2.05E–01	4.60E–01
Levy Function			
Best	2.58E–08	1.12E–07	4.34E–09
Worst	9.10E–04	2.61E–05	8.22E–05
Mean	1.31E–04	5.41E–06	1.47E–05
Std	2.53E–04	6.91E–06	2.07E–05
Ackley			
Best	8.88E–16	8.88E–16	8.88E–16
Worst	8.88E–16	8.88E–16	8.88E–16
Mean	8.88E–16	8.88E–16	8.88E–16
Std	0.00E+00	0.00E+00	0.00E+00
Foxholes			
Best	9.98E–01	9.98E–01	9.98E–01
Worst	3.03E+00	1.10E+00	9.99E–01
Mean	1.18E+00	1.01E+00	9.98E–01
Std	5.62E–01	3.42E–02	1.66E–04
Kowalik			
Best	3.44E–04	3.42E–04	3.56E–04
Worst	4.05E–03	3.50E–03	2.54E–03
Mean	7.01E–04	6.24E–04	5.94E–04
Std	7.98E–04	6.80E–04	4.79E–04
Golden Stein Price			
Best	3.00E+00	3.00E+00	3.01E+00
Worst	3.68E+00	3.15E+00	3.38E+00
Mean	3.17E+00	3.07E+00	3.08E+00
Std	2.05E–01	3.72E–02	9.23E–02

Table 5 Sensitivity analysis of AO-TSA for *t* parameter

	$t \le \left(\frac{1}{2}\right) \times T$	$t \le \left(\frac{2}{3}\right) \times T$	$t \le \left(\frac{3}{4}\right) \times T$
Sphere			
Best	8.75E–155	4.70E–154	5.76E–160
Worst	2.20E–143	8.10E–147	1.97E–143
Mean	1.35E–144	8.70E–148	9.88E–145
Std	4.89E–144	2.10E–147	4.41E–144
Rosenbrock			
Best	6.67E–05	8.86E–05	5.70E–04
Worst	1.60E+00	3.43E–02	1.72E–01
Mean	1.27E–01	7.30E–03	1.63E–02
Std	3.96E–01	1.03E–02	3.89E–02
Quartic			
Best	6.49E–06	1.12E–07	2.29E–05
Worst	9.10E–04	2.61E–05	7.24E–04
Mean	2.29E–04	5.41E–06	2.08E–04
Std	2.59E–04	6.91E–06	2.14E–04
Schwefel			
Best	– 4.19E+03	– 4.19E+03	– 4.19E+03
Worst	– 4.19E+03	– 4.19E+03	– 4.19E+03
Mean	– 4.19E+03	– 4.19E+03	– 4.19E+03
Std	1.36E–01	2.05E–01	3.13E–01
Levy Function			
Best	5.87E–08	1.12E–07	6.55E–08
Worst	1.11E–03	2.61E–05	2.51E–04
Mean	8.21E–05	5.41E–06	5.17E–05
Std	2.50E–04	6.91E–06	6.37E–05
Ackley			
Best	8.88E–16	8.88E–16	8.88E–16
Worst	8.88E–16	8.88E–16	8.88E–16
Mean	8.88E–16	8.88E–16	8.88E–16
Std	0.00E+00	0.00E+00	0.00E+00
Foxholes			
Best	9.98E–01	9.98E–01	9.98E–01
Worst	1.26E+00	1.10E+00	2.98E+00
Mean	1.02E+00	1.01E+00	1.28E+00
Std	6.50E–02	3.42E–02	5.66E–01
Kowalik			
Best	3.18E–04	3.42E–04	3.32E–04
Worst	5.70E–03	3.50E–03	3.06E–03
Mean	7.42E–04	6.24E–04	6.51E–04
Std	1.17E–03	6.80E–04	5.93E–04
Golden Stein Price			
Best	3.00E+00	3.00E+00	3.00E+00
Worst	3.68E+00	3.15E+00	3.40E+00
Mean	3.19E+00	3.07E+00	3.10E+00
Std	1.94E–01	3.72E–02	9.70E–02

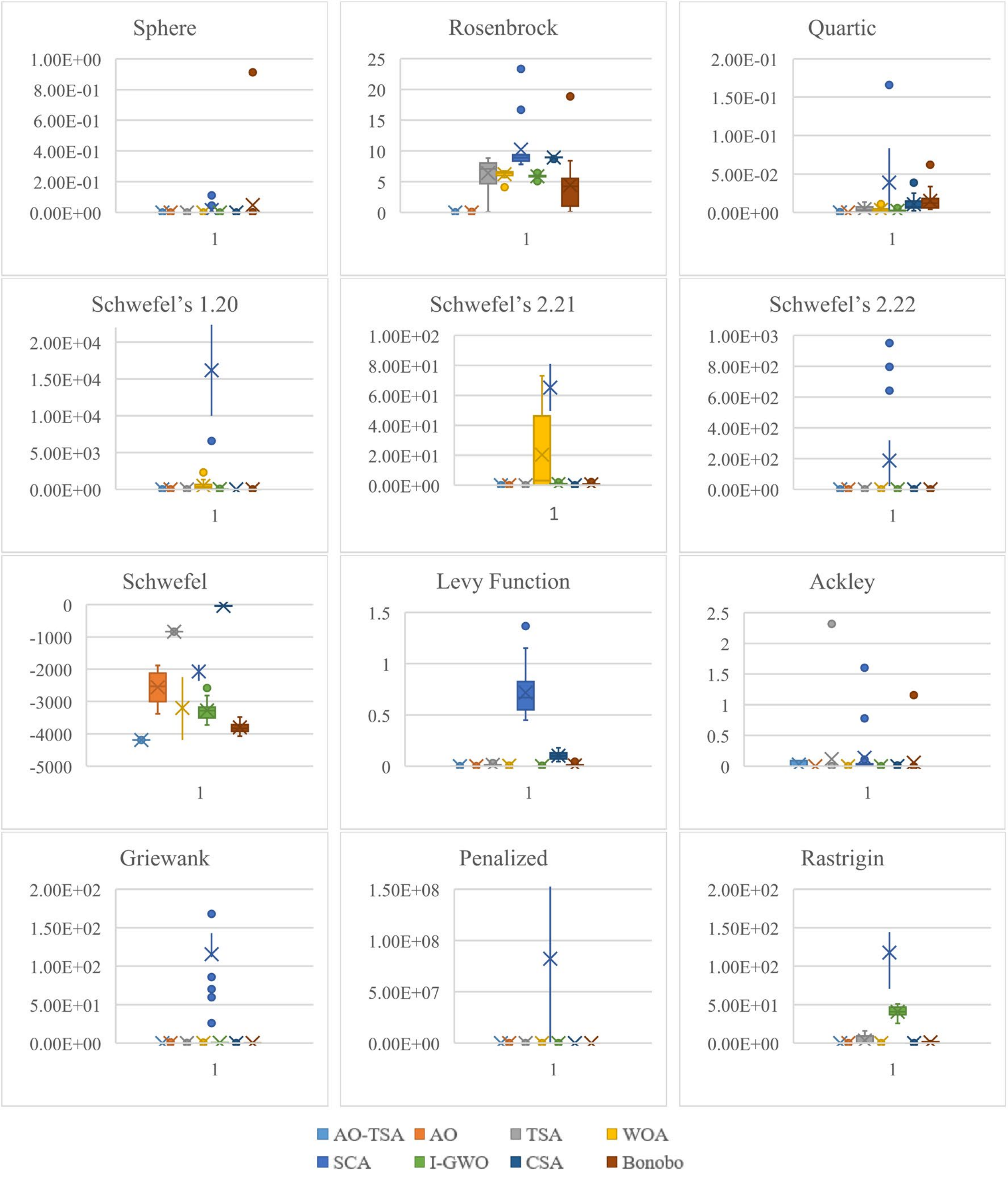


Fig. 6 Box plots for experimental results from unimodal and multimodal benchmark functions

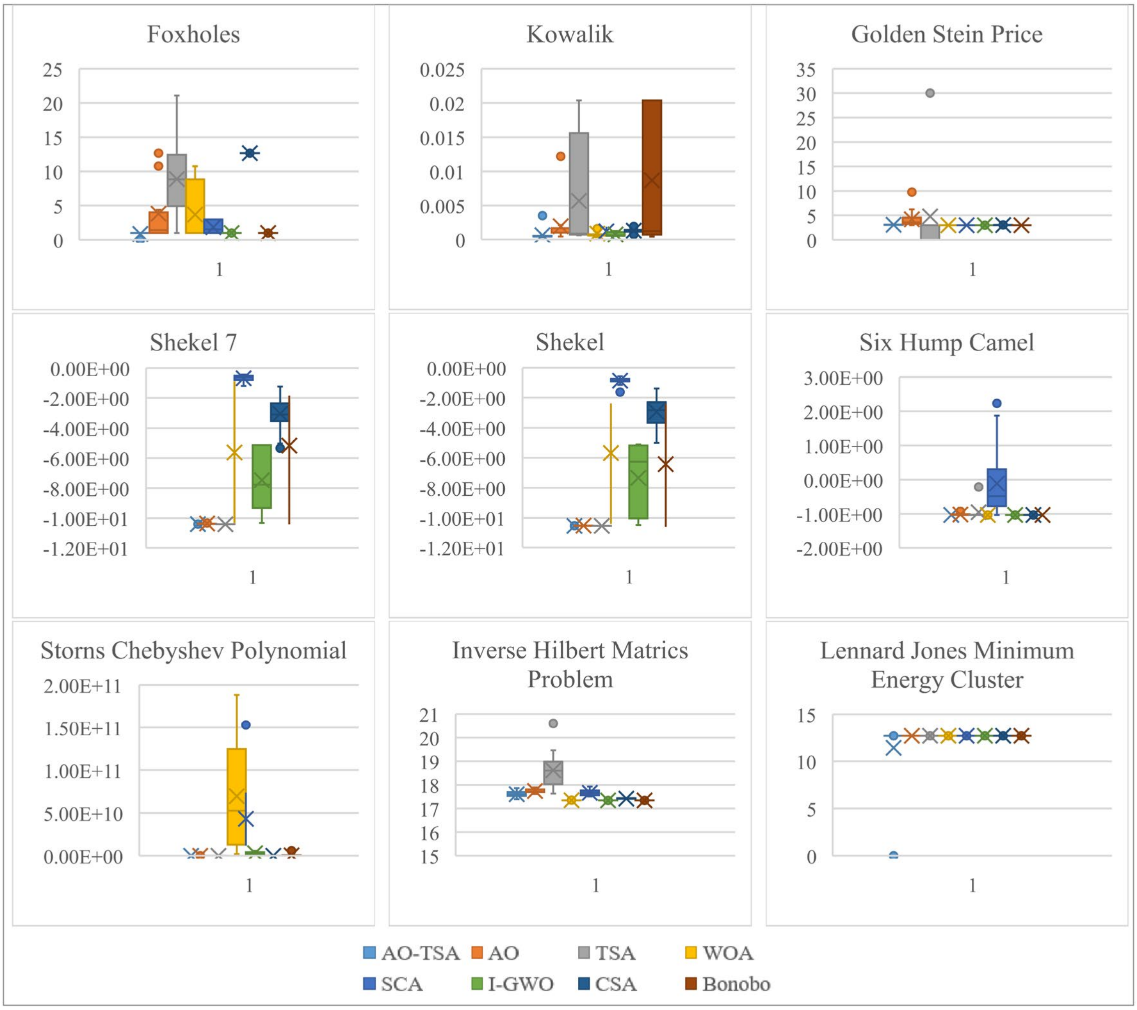


Fig. 7 Box plots for experimental results from fixed-dimension multimodal and CEC 2019 benchmark functions

Table 6 Optimal results obtained from the methods for tension/compression spring design

	x_1	x_2	x_3	$f(x)$
AO-TSA	0.051162	0.344061	12.08972	0.012689
AO	0.065023	0.632572	8.461970	0.027980
TSA	0.085277	0.901630	12.54581	0.095375
WOA	0.061224	0.632459	3.987035	0.014194
SCA	0.069467	0.677090	5.908206	0.025839
I-GWO	0.050773	0.334713	12.77824	0.012803
CSA	0.059300	0.548500	9.43E+00	0.022047
BO	0.053143	0.392721	9.453119	0.012703

Table 7 Optimal results obtained from the methods for three bar truss design

	x_1	x_2	$f(x)$
AO-TSA	0.790512	0.403105	263.9010
AO	0.763076	0.488572	264.6876
TSA	0.797520	0.387339	264.3067
WOA	0.759767	0.496860	264.5806
SCA	0.747313	0.581608	269.5328
I-GWO	0.784408	0.420579	263.9220
CSA	0.806047	0.370386	265.0231
BO	0.792187	0.398517	263.9159

Table 8 Analysis of the results with the Friedman test

	Test functions											
	f_1	f_2	f_3	f_7	f_8	f_9	f_{13}	f_{14}	f_{15}	f_{19}	f_{20}	f_{21}
Algorithms												
AO-TSA	1.05	1.50	1.40	1.00	1.85	1.43	4.50	1.70	6.65	2.45	5.60	5.45
AO	1.95	1.50	1.70	4.65	2.35	2.30	5.38	6.10	7.45	2.95	6.60	7.70
TSA	3.95	5.05	4.25	7.00	4.60	3.50	6.48	4.98	4.40	3.60	7.85	1.58
WOA	4.75	5.10	4.10	3.75	4.35	4.70	3.73	3.25	3.60	7.50	3.00	2.78
SCA	7.75	7.35	7.30	5.85	8.00	7.75	4.85	5.35	4.75	7.35	5.90	5.75
I-GWO	3.30	4.25	4.05	3.45	4.05	3.28	1.70	3.40	1.80	6.00	2.00	4.00
CSA	6.75	7.50	6.40	8.00	7.00	7.10	7.55	5.70	5.55	1.00	4.05	7.10
BO	6.50	3.75	6.80	2.30	3.80	5.95	1.83	5.53	1.80	5.15	1.00	1.65
N	20	20	20	20	20	20	20	20	20	20	20	20
Chi-Square (x^2)	132.783	121.367	115.383	130.333	103.269	125.348	101.994	55.470	103.069	131.533	132.517	133.534
Df	7	7	7	7	7	7	7	7	7	7	7	7
Asymp. Sig	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Data deposition statement No new data were generated for this study.

References

Abd Elaziz M, Dahou A, Alsaleh NA, Elsheikh AH, Saba AI, Ahmadein M (2021) Boosting COVID-19 image classification using MobileNetV3 and Aquila optimizer algorithm. Entropy 23(11):1383. https://doi.org/10.3390/e23111383

Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH (2021a) The arithmetic optimization algorithm. Comput Methods Appl Mech Eng 376:113609. https://doi.org/10.1016/j.cma.2020.113609

Abualigah L, Yousri D, Abd Elaziz M, Ewees AA, Al-qaness MA, Gandomi AH (2021b) Aquila optimizer: a novel meta-heuristic optimization algorithm. Comput Ind Eng 157:107250. https://doi.org/10.1016/j.cie.2021.107250

Abualigah L, Abd Elaziz M, Sumari P, Geem ZW, Gandomi AH (2022) Reptile search algorithm (RSA): a nature-inspired meta-heuristic optimizer. Expert Syst Appl 191:116158. https://doi.org/10.1016/j.eswa.2021.116158

Akyol S (2018) Güncel akıllı optimizasyon algoritmalarıyla duygu sınıflandırılması. Doktora Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elazığ.

Akyol S, Alatas B (2012) Güncel sürü zekası optimizasyon algoritmaları. Nevşehir Üniversitesi Fen Bilimleri Enstitüsü Dergisi 1(1):36–50

Akyol S, Alatas B (2017) Plant intelligence based metaheuristic optimization algorithms. Artif Intell Rev 47(4):417–462. https://doi.org/10.1007/s10462-016-9486-6

Akyol S, Alatas B (2020) Sentiment classification within online social media using whale optimization algorithm and social impact theory based optimization. Phys A 540:123094. https://doi.org/10.1016/j.physa.2019.123094

Alatas B (2007) Kaotik haritalı parçacık sürü optimizasyon algoritmaları geliştirme. Doktora Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elazığ.

Alatas B (2011) ACROA: artificial chemical reaction optimization algorithm for global optimization. Expert Syst Appl 38(10):13170–13180. https://doi.org/10.1016/j.eswa.2011.04.126

Alatas B, Bingol H (2020) Comparative assessment of light-based intelligent search and optimization algorithms. Light Eng. https://doi.org/10.33383/2019-029

AlRassas AM, Al-qaness MA, Ewees AA, Ren S, Abd Elaziz M, Damaševičius R, Krilavičius T (2021) Optimized ANFIS model using Aquila optimizer for oil production forecasting. Processes 9(7):1194. https://doi.org/10.3390/pr9071194

Ashrafi SM, Dariane AB (2011) A novel and effective algorithm for numerical optimization: Melody search (MS). In: 2011 11th International Conference on Hybrid Intelligent Systems (HIS) pp. 109–114. IEEE. https://doi.org/10.1109/HIS.2011.6122089

Askarzadeh A (2016) A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. Comput Struct 169:1–12. https://doi.org/10.1016/j.compstruc.2016.03.001

Atashpaz-Gargari E, Lucas C (2007) Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. In: IEEE Congress on Evolutionary Computation, CEC 2007, pp 4661–4667. https://doi.org/10.1109/CEC.2007.4425083

Blum C, Jakob P, Raidl GR, Roli A (2011) Hybrid metaheuristics in combinatorial optimization: a survey. Appl Soft Comput 11(6):4135–4151. https://doi.org/10.1016/j.asoc.2011.02.032

Borji A (2007) A new global optimization algorithm inspired by parliamentary political competitions. Lect Notes Comput Sci 4827(2007):61–71. https://doi.org/10.1007/978-3-540-76631-5_7

Bouchekara HREH (2020) Most valuable player algorithm: a novel optimization algorithm inspired from sport. Oper Res Int Journal 20(1):139–195. https://doi.org/10.1007/s12351-017-0320-y

Das AK, Pratihar DK (2019) A new bonobo optimizer (BO) for real-parameter optimization. In: 2019 IEEE Region 10 Symposium (TENSYMP), IEEE, pp 108–113. https://doi.org/10.1109/TENSYMP46218.2019.8971108

Dehghani M, Montazeri Z, Dehghani A, Ramirez-Mendoza RA, Samet H, Guerrero JM, Dhiman G (2020a) MLO: Multi leader optimizer. Int J Intell Eng Syst 13:364–373. https://doi.org/10.22266/ijies2020a.1231.32

Dehghani M, Montazeri Z, Dhiman G, Malik OP, Morales-Menendez R, Ramirez-Mendoza RA, Parra-Arroyo L (2020b) A spring search algorithm applied to engineering optimization problems. Appl Sci 10(18):6173. https://doi.org/10.3390/app10186173

Dehghani M, Montazeri Z, Givi H, Guerrero JM, Dhiman G (2020c) Darts game optimizer: a new optimization technique based on darts game. Int J Intell Eng Syst 13(5):286–294. https://doi.org/10.22266/ijies2020c.1031.26

Dhiman G (2021) ESA: A hybrid bio-inspired metaheuristic optimization approach for engineering problems. Eng Comput 37(1):323–353. https://doi.org/10.1007/s00366-019-00826-w

Dhiman G, Kaur A (2019) STOA: A bio-inspired based optimization algorithm for industrial engineering problems. Eng Appl Artif Intell 82:148–174. https://doi.org/10.1016/j.engappai.2019.03.021

Dhiman G, Kumar V (2017) Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications. Adv Eng Softw 114:48–70. https://doi.org/10.1016/j.advengsoft.2017.05.014

Dhiman G, Kumar V (2018) Emperor penguin optimizer: a bio-inspired algorithm for engineering problems. Knowl-Based Syst 159:20–50. https://doi.org/10.1016/j.knosys.2018.06.001

Dhiman G, Kumar V (2019) Seagull optimization algorithm: theory and its applications for large-scale industrial engineering problems. Knowl-Based Syst 165:169–196. https://doi.org/10.1016/j.knosys.2018.11.024

Dhiman G, Garg M, Nagar A, Kumar V, Dehghani M (2021) A novel algorithm for global optimization: rat swarm optimizer. J Ambient Intell Humaniz Comput 12(8):8457–8482. https://doi.org/10.1007/s12652-020-02580-0

Dokeroglu T, Sevinc E, Kucukyilmaz T, Cosar A (2019) A survey on new generation metaheuristic algorithms. Comput Ind Eng 137:106040. https://doi.org/10.1016/j.cie.2019.106040

Hashim FA, Hussain K, Houssein EH, Mabrouk MS, Al-Atabany W (2021) Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. Appl Intell 51(3):1531–1551. https://doi.org/10.1007/s10489-020-01893-z

Kaur S, Awasthi LK, Sangal AL, Dhiman G (2020) Tunicate swarm algorithm: a new bio-inspired based metaheuristic paradigm for global optimization. Eng Appl Artif Intell 90:103541. https://doi.org/10.1016/j.engappai.2020.103541

Layeb A (2021) The tangent search algorithm for solving optimization problems. arXiv preprint arXiv:2104.02559.

Ma L, Li J, Zhao Y (2021) Population forecast of China's rural community based on CFANGBM and improved Aquila optimizer algorithm. Fractal Fracti 5(4):190. https://doi.org/10.3390/fractalfract5040190

Mahajan S, Abualigah L, Pandit AK, Altalhi M (2022) Hybrid Aquila optimizer with arithmetic optimization algorithm for global optimization tasks. Soft Comput 26(10):4863–4881. https://doi.org/10.1007/s00500-022-06873-8

Mirjalili S (2016) SCA: a Sine Cosine Algorithm for solving optimization problems. Knowl-Based Syst 96:120–133. https://doi.org/10.1016/j.knosys.2015.12.022

Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008

Murty KG (2003) Optimization models for decision making. Internet edition, vol 1. http://www-personal.umich.edu/~murty/books/opti_model/. Accessed 23 Oct 2021

Nadimi-Shahraki MH, Taghian S, Mirjalili S (2021) An improved grey wolf optimizer for solving, engineering problems. Expert Syst Appl 166:113917. https://doi.org/10.1016/j.eswa.2020.113917

Upadhyay P, Chhabra JK (2021) Multilevel thresholding based image segmentation using new multistage hybrid optimization algorithm. J Ambient Intell Humaniz Comput 12:1081–1098. https://doi.org/10.1007/s12652-020-02143-3

Vashishtha G, Kumar R (2021) Autocorrelation energy and Aquila optimizer for MED filtering of sound signal to detect bearing defect in Francis turbine. Meas Sci Technol 33(1):015006. https://doi.org/10.1088/1361-6501/ac2cf2

Verma P, Parouha RP (2021) An advanced hybrid algorithm for nonlinear function optimization with real world applications. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-021-03588-w

Wang S, Jia H, Abualigah L, Liu Q, Zheng R (2021) An improved hybrid Aquila optimizer and harris hawks algorithm for solving industrial engineering optimization problems. Processes 9(9):1551. https://doi.org/10.3390/pr9091551

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.