Operational Research in Engineering Sciences: Theory and Applications
Vol. 6, Issue 1, 2023, pp. 115-142
ISSN: 2620-1607
eISSN: 2620-1747
DOI: https://doi.org/10.31181/oresta/060106

COMPARATIVE ANALYSIS OF METAHEURISTIC OPTIMIZERS IN THE PERFORMANCE OPTIMIZATION OF WIRE ELECTRIC DISCHARGE MACHINING PROCESSES

Subhankar Saha[1, 2], Saikat Ranjan Maity[1*], Sudip Dey [1]

[1] Department of Mechanical Engineering, National Institute of Technology Silchar, India
[2] Department of Mechanical Engineering, St. Martin's Engineering College, Telangana, India.

Received: 25 July 2022
Accepted: 29 October 2022
First online: 24 November 2022

Research Paper

***Abstract**: WEDM is an intricate process whereby improper selection of machine parameters often leads to undesirable performances. Therefore, the extraction of optimal machining parameters is pivotal for achieving better performances in WEDM. Metaheuristic optimizers have gained immense popularity due to their capability of providing global optimal solutions. The application of recently reported metaheuristic optimizers in non-traditional machining processes is rarely being explored. In light of the above, the current paper examines the use of six recently reported metaheuristic optimizers, namely the ant lion optimization (ALO), chimp optimization algorithm (ChoA), moth flame optimization (MFO), spotted hyena optimization (SHO), Harris Hawk optimization algorithm (HHO), Marine predator algorithm (MPA) to optimize WEDM performances in three WEDM processes. Particle swarm optimization (PSO) and Teaching learning-based optimization (TLBO), two well-known existing optimization approaches, are also included in this study to enable a reasonable comparison of the algorithms' performance. The algorithms are compared with parameters such as the quality of optimal solutions, convergence behavior, and average computational time. HHO algorithm is found to be robust amongst the eight competitors in terms of culminating the global optimal solution and propensity to quickly converge to the global optimal solution which corroborates the high exploration and exploitation capability of the algorithm. Therefore, HHO optimizer can be exploited in future to determine the optimal operating conditions for other manufacturing processes.*

***Keywords**: WEDM, optimization, metaheuristic algorithms, two sample t-test, sensitivity analysis.*

[*]Corresponding Author: sahamech90@gmail.com (S. Saha) saikatjumtech@gmail.com (S.R. Maity), infodrsudip@gmail.com (S. Dey)

1. Introduction

In the era of technological advancements, there is a growing demand for advanced materials which are hard and difficult-to-machine. The machining of such advanced materials with high geometrical accuracy using traditional machining approaches is an impossible task. As a result, a number of non-traditional machining (NTM) techniques are available to meet the requirement for high geometrical accuracy. WEDM is a non-traditional machining approach that has garnered a lot of interest in the industry because of its ability to create intricate curves with high geometrical accuracy (Majumder & Maity, 2018). The material removal in the WEDM process commences when an electric spark emerges amid the wire-workpiece interface. The spark liquidifies the material, and subsequently the molten debris is cleaned by the dielectric fluid injected from the top and bottom nozzles. The simplified view of the WEDM procedure is portrayed in Figure 1 (Absil et al., 2021).

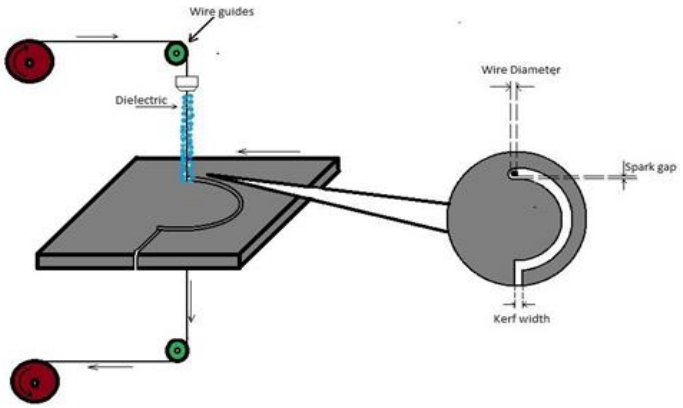

Figure 1. Simplified view of WEDM process

The performance attributes in WEDM are not always acquired at the envisioned level due to the process's intrinsic nature and a number of processing parameters (pulse duration, pulse interval, servo voltage, wire feed, wire speed, and so on), i.e., each process performance enhancement comes at the expense of another. As a result, machining under optimal operating conditions guarantees that a trade-off between the process performances is adequately maintained. In light of the preceding, researchers looked at several optimization techniques for selecting the best combination of process attributes. The next section provides a concise description of the optimization techniques that have been reported in WEDM operation, the importance of metaheursitic optimization algorithms and the advantages and limitations of different metaheuristic optimizers.

2. Literature Review

Mandal et al. (2016) derived the optimal operating conditions by the desirability function while processing Nimonic C-263 superalloy through WEDM (Mandal et al., 2016). In a recent investigation, the research group adopted a hybrid strategy i.e., signal-to-noise ratio and the Taguchi methodology to optimize the performance variables of the WEDM process concurrently (Ramakrishnan & Karunamoorthy, 2008). A group of researchers optimized the performances using RSM in WEDM of

Inconel 718 (Tonday & Tigga, 2019). Khan et al. (2014) implemented grey relational strategy to optimize microhardness and surface roughness simultaneously during the WEDM process (Khan et al., 2014). It is worth pointing that conventional techniques such as the Taguchi approach and grey relational analysis do not guarantee global optimum solutions as they commence the optimization with the specific level of process parameters. As a result, researchers are intrigued to adopt metaheuristic algorithms in WEDM process since they provide global optimum solutions or the best solutions. Metaheuristic algorithms are algorithms that are used in tackling a spectrum of complicated optimization problems without requiring to substantially adapt to each problem. The greek word "meta," which appears in the term, denotes that these algorithms are "higher level" heuristics, as contrary to problem-specific heuristics.

Metaheuristic algorithms are commonly used to address problems for which no appropriate problem-specific algorithm exists. The following traits are shared by almost all metaheuristics: They are nature-inspired; they use stochastic components (involving random variables); they do not evaluate the gradient or Hessian matrix of the objective function (Yoshioka et al., 2019).

The research community has been using metaheuristic algorithms in WEDM for the past thirty years. One pioneer contribution is the proposition of the simulated annealing method in WEDM to discover the optimal operating condition for the cutting rate and surface roughness (Tarng et al., 1995). In a similar manner, Sadeghi et al. (2011) explored the Tabu-search algorithm for optimizing the performance parameters (Sadeghi et al., 2011). In WEDM of Inconel-690, a modified version of cuckoo search algorithm is proposed to assess the optimal outcomes (Rao et al., 2011). A group of two researcher performed optimization of the performance parameters employing bat algorithm in taper formation in Inconel 718 exploiting WEDM. In a research effort, NSGA methodology is executed to track the various optimal parametric combinations (Pareto set) for two performance parameters in WEDM of Ti6Al4V (Nayak & Mahapatra, 2016). In a similar manner, Garg et al. (2012) found a set of Pareto optimal solutions in WEDM of Ti6Al-4V alloy employing the NSGA-II algorithm (Garg et al., 2012). In view of the above, it is observed that there are limited research in WEDM which have documented the use of metaheuristic optimizers in WEDM processes. It is worth emphasizing that the No-Free-Lunch (NFL) theorem asserts that one specific algorithm cannot solve all sorts of optimization problems (Wolpert & Macready, 1997).

Furthermore, previous studies have not documented the use of recently developed metaheuristic optimizers in WEDM. As a result, we plan to investigate the algorithmic performance of six recently reported metaheuristic optimizers, as well as two popular state-of-the-art metaheuristic optimizers, while optimizing WEDM performances either individually (single-objective) or collectively (multi-objective) for three WEDM processes. The six recently reported metaheuristic optimizers are ant lion optimization (ALO), chimp optimization algorithm (ChoA), moth flame optimization (MFO), spotted hyena optimization (SHO), Harris Hawk optimization (HHO), and Marine predator algorithm (MPA). Whereas, the two popular metaheuristic optimizers are particle swarm optimization and teaching learning- based optimization. We discussed the typical characteristics of each representative algorithm as follows to support the decisions made during the algorithm selection process. The PSO optimizer is simpler (Lee & Park, 2006). However, the main loophole

of this optimizer is the quick convergence of all solutions which undermines the diversity in the population (Juneja & Nagar, 2016). The TLBO optimizer has fewer tuning parameters than other optimizers, doesn't get stuck in local optima like other optimizers, and provides an accurate global optimal solution in minimum time (Uzlu et al., 2014). The limitation of this optimizer is that it ends with near-optimal solution in minimum iterative step (Sultana & Roy, 2014). SHO requires a low computational effort when tackling problems with high dimensions (Krishna et al., 2021). However, it is found that the problem space remains partially explored using SHO because of the concentrated search around the current optimal solution which might be a local optimum (Sabahno & Safara, 2022). ChoA has ample of advantages such as high exploration, a semi-deterministic feature of chaotic maps assists in high exploitation, local optima avoidance is very high, few parameters to tune, ease in implementation due to the parallel structure of independent groups (Khishe & Mosavi, 2020).

In contrary, it has few limitations such as premature convergence, a slow rate of convergence, discovering local minima rather than global minima, and a low balance between exploitation and exploration (Kaur et al., 2022). The advantage of using MFO is that it is simple, and can be easily hybridized with other algorithms (Shehab et al., 2020). But, it may easily fall into the local optima because it emphasizes on exploitation more than exploration which causes premature convergence, and the search ability is insufficient (Shan et al., 2021). Population diversity, strong optimization ability, and fewer adjustment factors are the typical advantages of ALO algorithm (Yao et al., 2021). Due to the roulette wheel selection technique, ALO algorithm suffers from rapid convergence (Abualigah et al., 2021). MPA optimizer has limited number of algorithmic variables. Moreover, the procedures are simple and converge fast with the added benefits of flexibility, and robustness (Yakout et al., 2021). However, it exhibits premature convergence in complex and high dimensional problems, and falls in local optima (Houssein et al., 2021). HHO optimizer is simple and has a few exploratory and exploitative mechanisms (Mansoor et al., 2020). But it has the major limitation of displaying finite exploration behavior as the exploration behavior depends on the equal perching chance, and in the mid-flight, the escape energy gets limited within unity (Naik et al., 2021).

In the present study, the goal is to compare the considered algorithms' performances based on several parameters such as the quality of optimal solutions, convergence behavior, and average computing time. The motive behind the comparative analysis is to find the most reliable optimization algorithm. Performance stability of the optimizers are retrieved exploiting the sensitivity analyses. Lastly, we tested the performance of the eight competing optimizers on benchmark test functions (i.e., the Sphere function and the Generalized Rastrigin's function) to determine the robust algorithm. The strategy adopted to accomplish the goal of the current research is delineated with a flowchart (see in Figure 2).

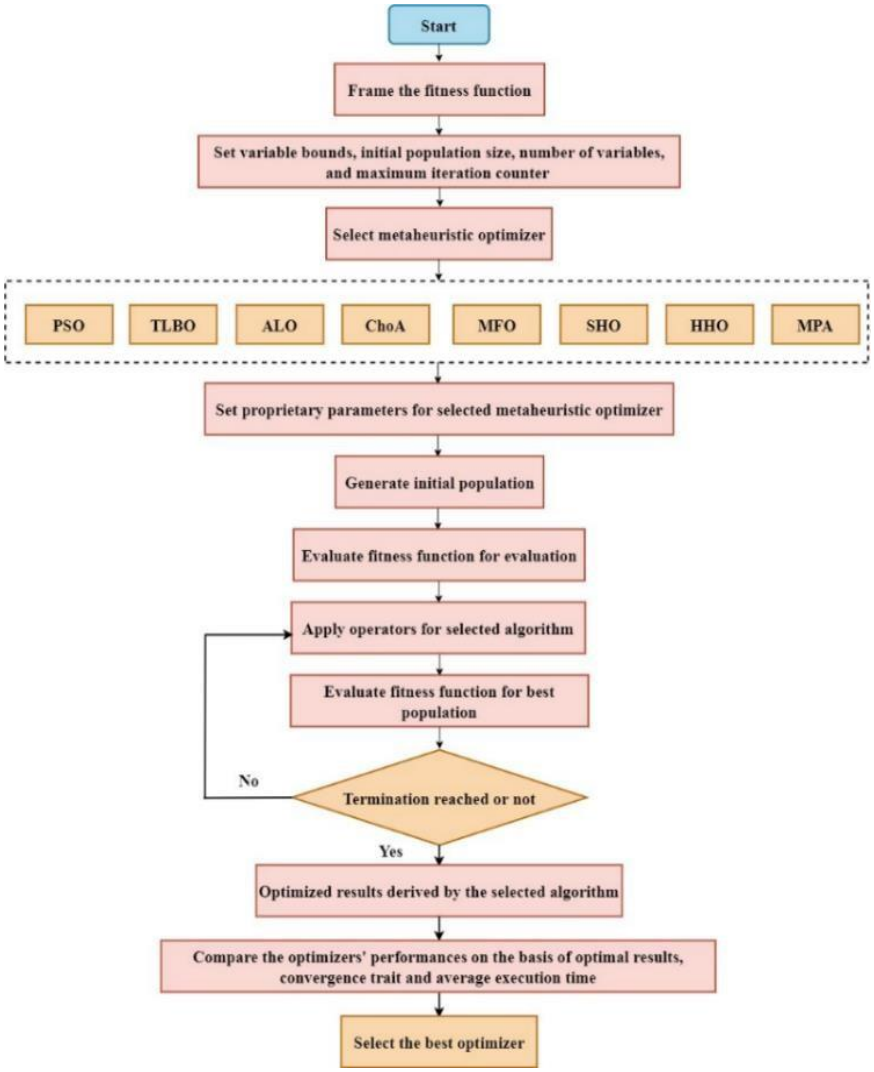

Figure 2. Flowchart showing the strategy adopted for the present work

3. Metaheuristic optimization algorithms

3.1. Teaching Learning Based Optimization (TLBO)

The fundamental concept of TLBO is to simulate a two-stage learning process in a traditional classroom setting (Rao et al., 2011). The communication of knowledge between a teacher and students occurs in the first stage, known as the Teacher Phase. The amount of knowledge gained by students is proportional to the amount of teacher's knowledge. However, in practice, the likelihood of the teaching to become successful is distributed according to Gaussian law.

Only a small percentage of students can comprehend everything indicated by the right end of the Gaussian distribution. However, the chances of learning new things aren't entirely eliminated. A student can understand from the fellow students at the

second stage, known as the Learner Phase. Overall, the amount of knowledge conveyed to a student is determined by his or her teacher and by peer learning exchanges.

3.1.1. Teacher Phase

In this phase, a teacher intends to improve the average performance in the subject being taught. The teaching job is first assigned to the best individual in the population $X_{teacher}$, after which the algorithm improves other individuals X_i by adjusting their positions towards that of the teacher $X_{teacher}$. The current mean value of the individuals X_{mean} is used to create each individual's position, which symbolizes the traits of all learners in the current generation. The disparity amid the teacher's knowledge and the students' knowledge is simulated in Eq. (1), which shows how the difference in student performance is affected by the difference in teacher's knowledge and the students' knowledge.

$$X_{new} = X_i + r(X_{teacher} - (T_F X_{mean})) \quad (1)$$

The *TF* in Eq. (1) refers to a teaching factor which depicts the altered mean value, and *r* refers a random number in [0,1]

3.1.2. Learner Phase

Increasing an individual's knowledge (X_i) is done in this phase through peer learning from any student X_{ii} and interaction amid the individual and other learners. Two states can arise based on the relative knowledge levels of these two students: if X_{ii} is better than X_i, X_i will move towards X_{ii} (shown in Eq. (2)), and if X_{ii} is worse than X_i , X_i will be moved away from X_{ii} (shown in Eq. (3). Student will be allowed into the population if he or she performs better by using Eq. (2), and Eq. (3). The algorithm will iterate till the end condition is reached.

$$X_{new} = X_i + r(X_{ii} - (X_i) \quad (2)$$

$$X_{new} = X_i + r(X_i - (X_{ii}) \quad (3)$$

3.2. Particle Swarm Optimization

PSO is led by swarm intelligence behavior which takes advantage of the social information sharing model. Individuals (i.e., particles) fly across a higher-dimensional search space in PSO (Poli et al., 2007). The individuals' tendency to imitate the success of others in population leads to changes in particle positions within the search space (called swarm). The knowledge, of a particle's surroundings thereby affects its modification within the swarm. The search characteristic of a particle is affected by the search characteristic of other particles in the swarm.

The particle keeps track of its location in the problem space, which is related to its best solution so far, known as p_{best} , and the overall best value is the best value recorded by the particle swarm optimizer when globally treated. Furthermore, its current location, as determined by any particle in the population, is known as g_{best} . Each particle's velocity is changed as it moves toward its p_{best} and g_{best} positions in the particle swarm optimization process. Separate random values are created for

acceleration towards the p_{best} and g_{best} positions, which are weighted by random terms. The PSO method adjusts the particle's velocities and positions as shown in the equations below:

$$vel_i(t+1) = \lambda\left[vel_i(t) + c_1 r_1\left(p_{best} - z_i(t)\right) + c_2 r_2\left(g_{best} - z_i(t)\right)\right], \lambda = \frac{2}{\left|\sqrt{2-\varphi-\sqrt{\varphi^2-4\varphi}}\right|} \quad (4)$$

Where $\varphi = c_1 + c_2,\ \varphi > 4$

$$z_i(t+1) = z_i + vel_i(t+1) \quad (5)$$

where, *c1* and *c2* are the positive constants which represent the cognitive learning factor, and social learning factor respectively, *r1 and r2* are random numbers in range [0,1]. $z_i = [z_{i1}, z_{i2}, \ldots., z_{idim}]^T$ depicts the ith particle position in the search space of dimension *dim*, and $vel_i = [vel_{i1}, vel_{i2}, \ldots., vel_{in}]^T$ depicts the ith particle velocity.

3.3. Spotted Hyena optimizer

Spotted Hyena Optimizer is a new bio-inspired optimization algorithm, which simulates the collaborative behavior of a group of spotted hyenas during encircling, hunting, and attacking the prey (Dhiman & Kumar, 2017).

3.3.1 Prey encircling

During prey encircling, the target prey is assumed to be the best solution, and the other spotted hyenas change their positions by following the best solution. This behavior is mathematically modeled as follows:

$$\vec{D}_{\mathrm{h}} = |\vec{A}.\vec{P}_{\mathrm{prey}}(t) - \vec{P}_{\mathrm{SH}}(t)| \quad (6)$$

$$\vec{P}_{\mathrm{SH}}(t+1) = \vec{P}_{\mathrm{Prey}}(t) - \vec{B}.\vec{D}_{\mathrm{h}} \quad (7)$$

where $\vec{D}_{\mathrm{h}}$ denotes the separation between the spotted hyena and prey, t denotes the present iteration, $\vec{A}$ and $\vec{B}$ are co-efficient vectors, $\vec{P}_{\mathrm{Prey}}$ represent prey's position vector, and $\vec{P}_{\mathrm{SH}}$ represent spotted hyena's position vector.

$$\vec{A} = 2 \times \vec{r}_1 \quad (8)$$

$$\vec{B} = 2\vec{s} \times \vec{r}_2 - \vec{s} \quad (9)$$

$$\vec{s} = 5 - \left(t \times \frac{5}{MaxIter}\right) \quad (10)$$

where $\vec{s}$ reduces linearly from 5 to 0, and $\vec{r}_1$, $\vec{r}_2$ and randomly changes between 0 and 1.

3.3.2 Prey hunting and Prey searching

During prey hunting, the hunting strategy adopted by spotted hyenas in the SHO algorithm is modeled mathematically as follows:

$$\vec{D}_{\mathrm{K}} = |\vec{A}.\vec{P}_{\mathrm{bSH}} - \vec{P}_{\mathrm{K}} \quad (11)$$

$$\vec{P}_{\mathrm{K}} = \vec{P}_{\mathrm{bSH}} - \vec{B} \times \vec{D}_{\mathrm{K}} \quad (12)$$

$$\vec{C}_{\mathrm{h}} = \vec{P}_{\mathrm{K}} + \vec{P}_{\mathrm{K+1}} + \ldots. + \vec{P}_{\mathrm{K+M}} \quad (13)$$

where, $\vec{P}_{\mathrm{bSH}}$ is the initial best position of spotted hyena, $\vec{P}_{\mathrm{K}}$ represents the position of other spotted hyenas, and M represents the number of spotted hyenas (shown in

Eq.(13)).

$$\mathrm{M}=\mathrm{count}(\vec{P}_{\mathrm{bSH}}, \vec{P}_{\mathrm{bSH+1}}, \vec{P}_{\mathrm{bSH+2}}, \ldots(\vec{P}_{\mathrm{bSH}}+\vec{R})) \tag{14}$$

$\vec{R}$ randomly varies between 0.5 and 1, and $\vec{C}_{\mathrm{h}}$ is a cluster of M number of optimal solutions. During prey hunting, spotted Hyenas attack the prey in a way that is mathematically expressed below:

$$\vec{P}_{\mathrm{SH}}(t+1)=\frac{\overrightarrow{C_h}}{M} \tag{15}$$

where, $\vec{P}_{\mathrm{SH}}(t+1)$ saves the best solutions, and the positions of the remaining spotted hyenas' changes relative to the best-spotted hyena's position. During prey searching, the vector A in Eq. (11) provides random values during the iteration process, which aids in exploration.

3.4. Chimp Optimization algorithm

The Chimp optimization algorithm, a metaheuristic optimizer, is motivated by the intelligence behavior exhibited by the chimps during hunting in their communities (Khishe & Mosavi, 2020). There are four categories of chimps, i.e., attacker, chaser, barrier, and driver, with different capabilities. The chaser, barrier, and driver lead the exploration process while the function of the attacker leads the exploitation process. The behavior of chimps modeled as follows:

$$\mathrm{d}=c.P_{Prey}(t)-o.P_{chimp}(t) \tag{16}$$

$$P_{chimp}(t+1)=P_{Prey}(t)-a.d \tag{17}$$

where P_{Prey} and P_{chimp} indicate the position vectors of the prey and the chimp respectively. The co-efficient vectors c, o, and a are determined as follows:

$$a=2.g.r_1-g \tag{18}$$

$$c=2.r_2 \tag{19}$$

$$o=\text{chaotic\_value} \tag{20}$$

where g diminishes from 2.5 to 0 through the iteration process, o is a chaotic vector determined using chaotic maps. The generation of stochastic population of chimps is the initial step in the chimp optimization algorithm. Then in the next step, chimps are classified into four varying categories: driver, barrier, attacker, and chaser. The best chimps are the initial attacker, barrier, driver, and chaser as they are aware of the prey's position.

Therefore, amongst the entire set of best solutions, four best solutions are used to represent them. The rest of the chimps are compelled to change their locations on the basis of the best chimp locations. This behavior can be mathematically expressed as below:

$$d_{\mathrm{Attacker}}=|c_1.P_{\mathrm{Attacker}}(t)-o_1.P| \tag{21}$$

$$d_{\mathrm{Barrier}}=|c_2.P_{\mathrm{Barrier}}(t)-o_2.P| \tag{22}$$

$$d_{\mathrm{Chaser}}=|c_3.P_{\mathrm{Chaser}}(t)-o_3.P| \tag{23}$$

$$d_{\mathrm{Driver}}=|c_4.P_{\mathrm{Driver}}(t)-o_4.P| \tag{24}$$

$$P_1=P_{\mathrm{Attacker}}(t)-a_1.d_{\mathrm{Attacker}} \tag{25}$$

$$P_2 = P_{\text{Barrier}}(t) - a_2 . d_{\text{Barrier}} \quad (26)$$
$$P_3 = P_{\text{Chaser}}(t) - a_3 . d_{\text{Chaser}} \quad (27)$$
$$P_4 = P_{\text{Driver}}(t) - a_4 . d_{\text{Driver}} \quad (28)$$
$$P(t+1) = (P_1 + P_2 + P_3 + P_4)/4 \quad (29)$$

where, $d_{Attacker}$, $d_{Barrier}$, d_{Chaser}, d_{Driver} analogs with d in Eq. (16).

For updating the location of the chimps during the searching period, a probability of 50% is chosen between two alternatives, i.e., the usual updating rule and the chaotic model, which is mathematically expressed below:

$$P_{\text{chimp}}(t+1) = \begin{cases} P_{\text{Prey}}(t) - a.d \ if \mu < 0.5 \\ \text{Chaotic\_value} \ if \mu \geq 0.5 \end{cases} \quad (30)$$

3.5. Moth Flame Optimization

Transverse orientation for navigation of moths at night using moonlight forms the motivation of this MFO algorithm (Mirjalili, 2015b). In the MFO algorithm, the candidate solutions are the moths and the problem variables refers to their positions in the search space. The set of moths is represented as a matrix with *n* moths and *dim* dimensions which is shown below:

$$M = \begin{bmatrix} m_{1,1} & m_{1,2} & \dots & m_{1,dim} \\ m_{2,1} & m_{2,2} & \dots & m_{2,dim} \\ \vdots & \vdots & \vdots & \vdots \\ m_{n,1} & m_{n,1} & \cdots & m_{n,dim} \end{bmatrix} \quad (31)$$

We further suppose that the fitness values for all the moths are stored in an array, as follows:

$$OM = \begin{bmatrix} OM_1 \\ OM_2 \\ \vdots \\ OM_n \end{bmatrix} \quad (32)$$

Another key part of the MFO algorithm is flames. The following is a matrix that is identical to the moth matrix:

$$M = \begin{bmatrix} F_{1,1} & F_{1,2} & F_{1,dim} \\ F_{2,1} & F_{2,2} & F_{2,dim} \\ \vdots & \vdots & \vdots \\ F_{n,1} & F_{n,1} & F_{n,dim} \end{bmatrix} \quad (33)$$

where *n* is the number of moths and *dim* is the dimension. The dimension of the flame matrix is the same as the dimension of the moth matrix. Both the moth and the flame are solutions, but the moth is the search agent and the flame is the moth's best position. Flames are the flags that moths drop during the search process, and the moths travel around the flags and update accordingly. As a result of this, the moths never lose their best solution. According to the equation below, moths update their position in relation to flame.

$$M_i = S(M_i, F_j) \quad (34)$$

where M_i represents the ith moth, F_j represents the jth flame, and the spiral function is represented by S. The logarithmic spiral motion of the moth is given below:

$$S(M_i, F_j) = D_i . e^{bt} . \cos(2\pi r) + F_j \quad (35)$$

$$\text{Where, } D_i = |F_j - M_i \quad (36)$$

b is a constant that determines the form of spiral motion, r refers to random number within [-1, 1]. Flame gets updated over the course of iterations as follows:

$$\text{flame no} = \text{round}\left(\text{N} - \text{t} * \frac{\text{N}-1}{\text{MaxIter}}\right) \quad (37)$$

where N is the maximum number of flame

3.6. Ant Lion Optimization

Ant Lion Optimizer is a metaheuristic optimizer which is conceptualized based on the chasing strategy of antlions in catching their prey (Mirjalili, 2015a). The ant lions hide underneath the base of the cone-shaped cavities in the sand and then wait for the ants to get captured in the hole. They throw sand at the tip of the trap so that the ants fail to escape and slide down to the bottom of the trap. In this manner, the ants get captured by the ant lions. The pits are rebuilded to capture other ants. The positions of the ants are stored in the matrix M_{Ant} (shown in Eq. (38)) which is employed during the optimization.

$$M = \begin{bmatrix} A_{11} & \cdots & A_{1dim} \\ \vdots & \ddots & \vdots \\ A_{n1} & \cdots & A_{ndim} \end{bmatrix} \quad (38)$$

where n refers to the quantity of ants, and dim refers to the dimension of the problem. Fitness function f is utilized for the evaluation of the fitness of each ant during optimization; the fitness values are stored in the matrix M_{OAnt} as shown below:

$$M_{OAnt} = \begin{bmatrix} \text{f}([A_{1,1}A_{1,2}, \ldots, A_{1,dim}]) \\ \text{f}([A_{2,1}A_{2,2}, \ldots, A_{2,dim}]) \\ \vdots \\ \vdots \\ \text{f}([A_{n,1}A_{n,2}, \ldots, A_{n,dim}]) \end{bmatrix} \quad (39)$$

Apart from ants, the ant lions also have their hideouts in the search domain. The matrices M_{Antlion} and M_{OAntlion} save the positions and fitness values of the ant lions, respectively.

$$M_{Antlion} = \begin{bmatrix} A_{1,1} & \cdots & A_{1,m} \\ \vdots & \ddots & \vdots \\ A_{n,1} & \cdots & A_{n,m} \end{bmatrix} \quad (40)$$

$$M_{OAntlion} = \begin{bmatrix} f([AL_{1,1}AL_{1,2}, \ldots, AL_{1,m}]) \\ f([AL_{2,1}AL_{2,2}, \ldots, AL_{2,m}]) \\ \vdots \\ \vdots \\ f([AL_{n,1}AL_{n,2}, \ldots, AL_{n,m}]) \end{bmatrix} \quad (41)$$

While searching for food, ants move in a stochastic fashion; thus, a random walk is selected to simulate ants' movement as below:

$$X(t) = [0, csum(2r(t_1) - 1), csum(2r(t_2) - 1) \ldots, csum(2r(t_{MaxIter}) - 1)] \quad (42)$$

where $csum$ reveals the cumulative sum, t reveals the random walk steps, and r (t) is

a stochastic function which is enumerated as follows:

$$r(t) = \begin{cases} 1 \text{ if rand} > 0.5 \\ 0 \text{ if rand} \leq 0.5 \end{cases} \quad (43)$$

To limit the movement within the search space in a random fashion, Eq. (42) is normalized exploiting the Eq. (44)

$$X_i^t = \frac{(X_i^t - a_i) \times (b_i - c_i^t)}{(d_i^t - a_i)} + c_i \quad (44)$$

Eq. (45), and Eq. (46) show how ants' slide down into pits.

$$c^t = \frac{c^t}{I} \quad (45)$$

$$d^t = \frac{d^t}{I} \quad (46)$$

When the ant reaches the pit bottom, the antlion snatches it and consumes it. To improve its chances of obtaining new prey, an antlion must update its position to the most recent position of the chased ant. Eq. (47) represents this procedure

$$\text{Antlion}_j^t = \text{Ant}_i^i \text{ if } f(\text{Ant}_i^t) > f(\text{Antlion}_j^t) \quad (47)$$

In every step, elitism is employed to keep the best solutions. The fittest antlion is treated as elite, which is the best antlion achieved. In every step, the elite should have an impact on the antlion (random movement). For this, every ant is assumed to associate with an antlion by Roulette wheel and elite, which Eq. (48) gives.

$$\text{Ant}_i^t = \frac{R_A^t + R_E^t}{2} \quad (48)$$

R_A^t, and R_E^trepresent the random walk around the selected antlion and elite at tth iteration respectively.

3.7. Marine Predator Algorithm (MPA)

The marine predator algorithm is a novel nature-inspired metaheuristic algorithm that replicates the biological interaction between marine predators and prey (Faramarzi et al., 2020). Predators are inspired in this algorithm to use the widespread foraging methods known as the Brownian and Levy random movement in the marine ecosystem.

Predators utilize the Brownian approach if there exists a large concentration of prey in the hunting region, and the Levy method when there is a low concentration of prey. However, environmental factors namely eddy generation and the effects of fish aggregating devices (FADs) are among the elements that influence marine predator behavior. The steps of the algorithm are enumerated as follows:

3.7.1. Initialization

Both the Prey (P) and Elite (E) matrices are formed during the initialization phase. In accordance with the survival of the fittest argument, the skilled foragers are the top predators in nature. Thus, in order to construct the Elite matrix, the fittest solution is designated as a top predator. Prey is another matrix with the same dimension as Elite, and predators use it to update their positions.

3.7.2. Phase 1

This phase commences during the one-third of iterations and is implemented by a large velocity ratio (v ≥10) for an adequate exploration ability, wherein the movement of the prey is faster than the predator. The prey moves quickly to guard their food. Whereas the fittest predators are stationary during this stage. This stage is mathematically illustrated with the help of equations (Eq. 49 & Eq. 50).

$$\text{While t} < \frac{\text{MaxIter}}{3}$$

$$\vec{S}_i = \overrightarrow{R_B} \otimes \left(\vec{E}_i - \left(\overrightarrow{R_B} \otimes \vec{P}_i\right)\right) i = 1,2, \ldots, \text{n} \quad (49)$$

$$\vec{P}_i = \vec{P}_i + \left(0.5\vec{R} \otimes \vec{S}_i\right) \quad (50)$$

where $\vec{S}_i$ indicate the step size of the predator, $\overrightarrow{R_B}$ is the random vector based on normally distributed Brownian motion, $\vec{R}$ indicate a uniformly distributed random variable, and n indicates the search agents per population. The notation ⊗ indicate entry-wise multiplications.

3.7.3. Phase 2

In this phase, there is a transient transformation from exploration to exploitation. Here, the velocity ratio of unity (v ≈ 1) indicates that both the predator and the prey moves at an identical speed.

$$\text{While } \frac{\text{MaxIter}}{3} < \text{t} < \frac{2}{3}\text{MaxIter}$$

The first half population gets updated based on Levy strategy as follows

$$\vec{S}_i = \overrightarrow{R_L} \otimes \left(\vec{E}_i - \left(\overrightarrow{R_L} \otimes \vec{P}_i\right)\right) i = 1,2, \ldots, \text{n}/2 \quad (51)$$

$$\vec{P}_i = \vec{P}_i + \left(0.5\vec{R} \otimes \vec{S}_i\right) \quad (52)$$

where $\overrightarrow{R_L}$ is a uniformly distributed random vector based on Levy motion. On contrary, the second half population is updated using Brownian strategy as follows (shown in Eq. (53) & Eq. (54):

$$\vec{S}_i = \overrightarrow{R_B} \otimes \left(\left(\vec{R}_B \otimes \vec{E}_i\right) - \vec{P}_i\right) i = n/2, \ldots, \text{n} \quad (53)$$

$$\vec{P}_i = \vec{E}_i + \left(0.5X_f \otimes \vec{S}_i\right) \quad (54)$$

where X_f is a variable that monitor the predator's step size and is evaluated by the following Eq. (55)

$$X_f = \left[1 - \left(\frac{t}{MaxItr}\right)\right]^{(2 \times t/MaxIter)} \quad (55)$$

3.7.4 Phase 3

This phase is usually marked with a high level of exploitation capacity. This phase is marked by a low velocity ratio (v = 0.1), in which the predator runs past the prey. This phase is based on Levy movement which is mathematically expressed as follows

$$\text{While t} > \frac{2}{3}\text{MaxIter}$$

$$\vec{S}_i = \overrightarrow{R_L} \otimes \left(\left(\vec{R}_B \otimes \vec{E}_i\right) - \vec{P}_i\right) i = 1,2, \ldots, \text{n} \quad (56)$$

$$\vec{P}_i = \vec{E}_i + \left(0.5X_f \otimes \vec{S}_i\right) \tag{57}$$

3.7.5. Finishing

After each iteration, the best solutions gets stored in the Elite (E) matrix. The final solution is then achieved after the last iteration.

3.8. Harris Hawk Optimization

Harris Hawk optimization (HHO) is a new nature-inspired optimizer that imitates the chasing trait of Harris hawks in order to catch their prey (rabbit), which are the best solutions in the search space (Heidari et al., 2019). HHO goes through two stages: the first is looking for prey with a group of hawks, and this stage is referred to as the exploration phase in the algorithm. The second stage involves hunting the prey in order to catch it, which is depicted in the optimization algorithm as the exploitation phase. The balance between exploitation and exploration of search space is determined by the rabbit's energy escape, with hawks having the potential to explore for large energy and exploitation for small energy (Kuriakose & Shunmugam, 2005). In the exploration phase, the HHO algorithm employs two alternative search strategies. These strategies are chosen based on α; if α is greater than 0.5, the first strategy is employed to search near one of the other hawks at random, but if α is lesser than 0.5, the second strategy, stated in Eq. (59) is employed for the search operation.

$$X(t+1) = \{X_{rand}(t) - s_1|X_{rand}(t) - 2s_2X(t)|\}\alpha \geq 0.5 \tag{58}$$

$$X(t+1) = \left\{\left(X_{rabbit}(t) - X_m(t)\right) - s_3(LB + s_4(UB - LB))\alpha < 0.5\right\} \tag{59}$$

$$\text{Where } X_m(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t)$$

The mathematical model to demonstrate the mechanism which is exploited to get transformed from the exploration phase to the exploitation phase is shown in Eq. (60).

$$E = 2E_0\left(1 - \frac{t}{T}\right) \tag{60}$$

The algorithm arrives the exploration phase when |E| ≥1 whereas the algorithm arrives the exploitation phase when |E| ≤1. E diminishes when the iteration count increases. In the exploitation phase, the HHO algorithm utilizes four different ways to conduct optimization operations.

If E is greater than 0.5, two techniques are used: besiege and soft besiege with increasing quick dives. If E is less than 0.5, two techniques are used: besiege and hard besiege with progressive quick dives. The illustration of the strategies can be found in the literature.

4. WEDM performance optimization

To assess the efficacy of the eight metaheuristic optimization techniques, single and multiple objective optimization is carried out for two WEDM processes (elaborated in case 1, and case 2). The codes for the eight optimizers are built in MATLAB R2018a and executed on Windows 10 OS, Intel(R) Core™ i5 processor, and 8.00 GB RAM. For unprejudiced comparison amid the performances of the considered optimizers, population size, and maximum generation is kept at 50 and 100 respectively for all the considered algorithms

4.1. Case 1 Performance optimization in WEDM of A286 superalloy

We assessed the WEDM performances such as material removal rate (*MRR*, in mm2/min) and surface roughness (*SR* in μm) for an A286 Superalloy. Machining of the samples are accomplished using an Ultra cut F1 model, a variant of the WEDM machine tool. Twenty-seven sets of experiments are undergone under the L27 scheme. Five parameters are tuned in three subsequent levels (depicted in Table 1) within the stipulated bounds while machining. Finally, multiple performances are optimized simultaneously using a multi-objective evolutionary algorithm and a decision making tool (Saha et al., 2021). In this research, we use the eight different metaheuristic optimizers to optimize individual performances as well as two performances at the same time. The goal is to compare the performances of the optimizers. To accomplish the task, we intend to exploit the mathematical expressions used for devising the correlation between the response variables and the explanatory variables in the previous investigation (Saha et al., 2021). The mathematical models are shown below:

$$MRR = 567 - 20.39\,x_1 - 2.91x_2 + 102.8x_3 + 28.4x_4 + 2.83x_5 + 0.0497x_1^2 + 0.0354x_2^2 - 8.192x_3^2 - 0.037x_4^2 - 0.0302x_5^2 + 0.0117x_1 * x_2 + 0.858x_1 * x_3 - 0.1511x_1 * x_4 - 0.0034x_1 * x_5 - 0.244x_2 * x_3 - 0.1140x_2 * x_4 + 0.0196x_2 * x_5 - 0.326x_3 * x_4 - 0.145x_4 * x_5 \quad (61)$$

$$SR = -177.3 + 0.59\,x_1 - 0.889x_2 + 27.26x_3 + 1.82x_4 + 0.110x_5 - 0.00276x_1^2 + 0.00934x_2^2 - 1.2656x_3^2 - 0.0323x_4^2 - 0.00642x_5^2 + 0.00014x_1 * x_2 + 0.0177x_1 * x_3 - 0.0157x_1 * x_4 - 0.0034x_1 * x_5 - 0.244x_2 * x_3 - 0.1140x_2 * x_4 + 0.0196x_2 * x_5 - 0.326x_3 * x_4 - 0.145x_3 * x_5 \quad (62)$$

Table 1. Process variables and levels

Process variables	Level 1	Level 2	Level 3
(x_1) Pulse on period (μs)	120	125	130
(x_2) Pulse off period (μs)	48	52	56
(x_3) Peak current (A)	10	11	12
(x_4) Wire feed rate (m/min)	5	7	9
(x_5) Servo voltage (v)	30	35	40

4.1.1. Single-objective optimization

The optimization of MRR and SR is accomplished under a set of constraints i.e.,$\mathbf{120 \le x_1 \le 130, 48 \le x_2 \le 56, 10 \le x_3 \le 12, 5 \le x_3 \le 9, and\ 30 \le x_4 \le 40}$. The results of the different optimizers for the two performance attributes is demonstrated in Table 2. It is evident that ChoA, MFO, HHO, MPA, and PSO are able to produce the optimized MRR of 37.527 mm2/min which is close to the maximum MRR present in the experimental dataset (Saha et al., 2021). However, ALO, SHO, and TLBO

produces optimized MRR of 35.701 mm2/min, 29.614 mm2/min and 6.718 mm2/min respectively, which is relatively poor.

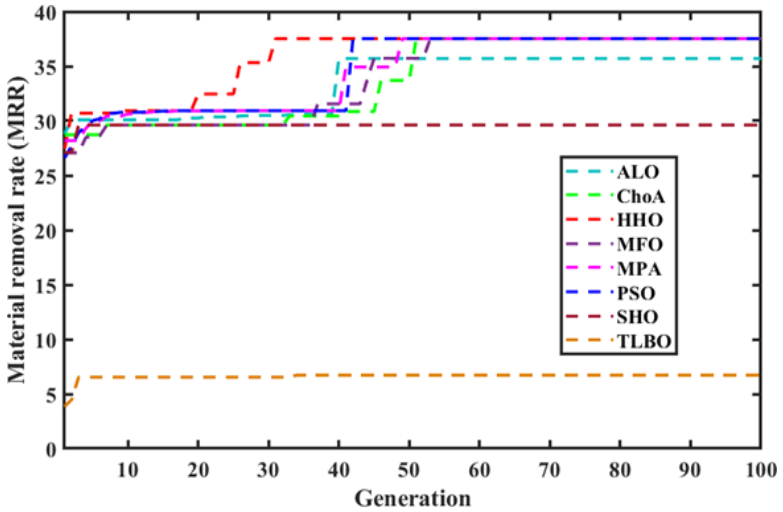

Figure 3. Convergence behavior of ALO, ChoA, HHO, MFO, MPA, PSO, SHO, and TLBO for MRR

For SR, unlike the SHO, we detect that all the algorithms have produced similar results, but better than all the results reported in the previous investigation (Saha et al., 2021). To realize the convergence traits of the optimizers, we plotted the convergence history of the competing optimizers while optimizing the response MRR (shown in Figure 3). It is noted that HHO algorithm rapidly converges to the global optimal solution which exposes the algorithm's outstanding exploitation capability.

Table 2. Single-objective optimization outcomes

Optimizer	Response	Optimal Value	Pulse on period	Pulse off period	Peak Current	Wire feed rate	Servo voltage
ALO	*MRR*	35.701	130	48	11.51	5.01	31.14
	SR	0.4776	120	56	10	5	40
ChoA	*MRR*	37.527	130	48	11.63	7	30
	SR	0.4776	120	56	10	5	40
MFO	*MRR*	37.527	130	48	11.59	7	30
	SR	0.4776	120	56	10	5	40
SHO	*MRR*	29.614	130	48	12	5	30
	SR	0.5320	120	56	10	9	40
MPA	*MRR*	37.527	130	48	11.59	7	30
	SR	0.4776	120	56	10	5	40
HHO	*MRR*	37.527	130	48	11.59	7	30
	SR	0.4776	120	56	10	5	40
PSO	*MRR*	37.527	130	48	11.59	5	30
	SR	0.4776	120	56	10	5	40
TLBO	*MRR*	6.718	120	56	10	9	40
	SR	0.4776	120	56	10	5	40

4.1.2 Multiple objective optimization

To perform optimization of the two performance attributes (MRR and SR) simultaneously, we formed the objective function using weighted-sum method as follows:

$$\text{Min(Y)} = w_1 \frac{Y(\text{SR})}{\text{SR}_{\min}} - w_2 \frac{Y(\text{MRR})}{\text{MRR}_{\max}} \tag{63}$$

where w_1 and w_2 are the preference weights assigned to SR and MRR, respectively. Here, equal weights for all the responses are considered, i.e., $w_1 = w_2 = 0.5$. SR_{min} is the minimum surface roughness and $\text{MRR}_{\max}$ is the maximum material removal rate, which are procured from the single-objective optimization outcomes. Table 3 shows the optimal MRR and SR values recommended using the eight competitor techniques (ALO, MFO, ChoA, MPA, SHO, HHO, PSO, and TLBO).

HHO portrays the superior performance i.e., it produces the global optimal responses of MRR and SR at minimal value of the combined fitness function (Y = 0.306708). The corresponding optimal process parameters are pulse on period = 130 s, pulse off period = 52 s, peak current = 10 A, wire feed rate = 5 m/min, and servo voltage = 30 volt. Figure 4 analyzed the convergence traits of the eight competing algorithms. The objective function of the HHO algorithm approaches the least fitness value (global optimum) in the fewest generations possible, demonstrating the algorithm's excellent exploitation capability. Table 4 shows the average computing time (seconds) consumed by the optimizers while optimizing the multiple performances. As seen in Table 4, TLBO has the shortest average computation time.

Table 3. Multiple objective optimization outcomes

Optimizer	Response	Optimal Value	Y	Pulse on period	Pulse off period	Current	Wire Speed	Servo voltage
(Saha et al., 2021)	MRR SR	36.04 3.49		130	52	10	5	30
ALO	MRR SR	1.51 0.96	0.47967	120	56	10	5	39
MFO	MRR SR	1.20 0.79	0.47984	120	56	10	5	40
SHO	MRR SR	1.20 0.79	0.47984	120	56	10	5	40
PSO	MRR SR	1.20 0.79	0.47984	120	56	10	9	40
TLBO	MRR SR	1.20 0.79	0.47984	120	56	10	5	40
HHO	MRR SR	36.04 3.49	0.30670	130	52	10	5	30
MPA	MRR SR	1.20 0.79	0.47984	120	56	10	5	40
CHoA	MRR SR	1.20 0.79	0.47984	120	56	10	5	40

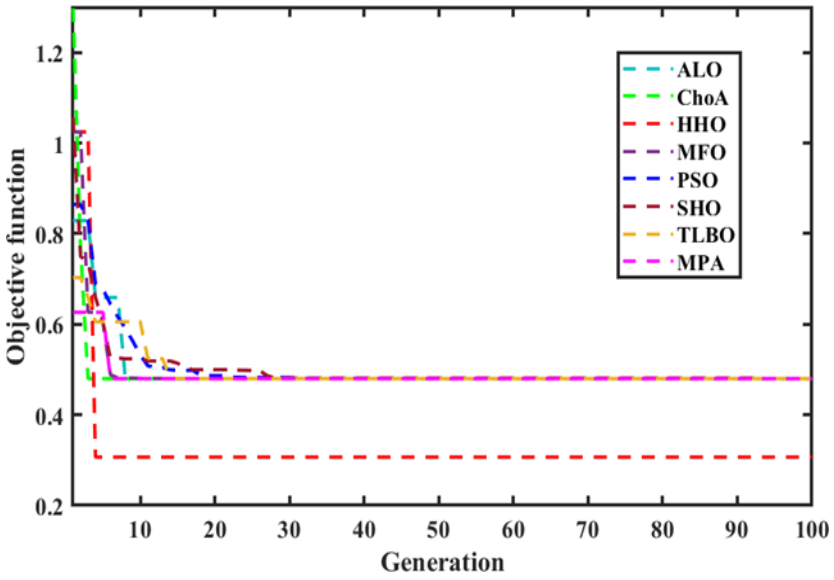

Figure 4. Convergence behavior of ALO, ChoA, MFO, SHO, HHO, MPA, PSO, and TLBO for multi-objective function.

Table 4. Average computation time for the eight optimizers

Optimizer	Average Computational time (secs)
ALO	2.20241
ChoA	1.83478
MFO	1.53465
SHO	1.85662
HHO	2.53421
MPA	1.25715
PSO	1.23427
TLBO	1.20124

4.2. Case 2 Performance optimization in WEDM of Ti-6Al-4V alloy

Devarajaiah & Muthumari, (2018) conducted WEDM machining on Ti-6Al-4V employing wire EDM machine tool of Model: DK 7732 (Devarajaiah & Muthumari, 2018). The machine tool used in this work is based on reusable wire technology and doesn't need air-conditioning below 40 degrees centigrade. Molybdenum wire electrode (diameter of 0.18 mm) is employed as wire electrode. Four parameters, i.e., pulse duration, pulse off period, applied current, and wire-speed, were selected as control variables. The levels considered for the four control variables are revealed in Table 5. Two vital process performance measures were considered as responses (i.e., material removal rate (MRR in g/min) and power consumption (PC in kW)).

The experimental trials were carried out as per Taguchi L16 design, and each trial is repeated thrice to capture the variability in the WEDM responses. Furthermore, regression analysis was employed by Devarajaiah & Muthumari, (2018) to correlate

the performance measures with the control variables (Devarajaiah & Muthumari, 2018). The regression models for MRR and PC are shown below (Eq. (64) & Eq. (65)):

$$MRR = 0.00249\,x_1 - 0.0056x_2 + 0.0151x_3 - 0.000011x_4 - 0.000065x_1^2 + 0.00039x_2^2 - 0.00008x_3^2 + 0.000001x_1x_4 - 0.00088x_2x_3 \quad (64)$$

$$PC = 0.0756 + 0.002\,x_1 - 0.0569x_2 + 0.0133x_3 + 0.000045x_4 + 0.000036x_1^2 + 0.00273x_2^2 + 0.002x_3^2 \quad (65)$$

Table 5. Process variables with levels

Process Variables	Level 1	Level 2	Level 3	Level 4
(x_1) Pulse on period (μs)	13	20	27	36
(x_2) Pulse off period (μs)	4	6	8	10
(x_3) Current (A)	1	2	4	5
(x_4) Wire speed (rpm)	350	700	1050	1400

4.2.1 Single objective optimization

In this case, two responses, i.e., MRR and PC, are optimized separately engaging the eight metaheuristic optimization algorithms. In other words, we intend to discover the optimal parametric condition for both the responses separately using the competing algorithms. The goal is to maximize the MRR and minimize the PC subjected to the imposed constraints as follows: $13 \leq \boldsymbol{x_1} \leq 36$, $4 \leq \boldsymbol{x_2} \leq 10$,$1 \leq \boldsymbol{x_3} \leq 5$, and $350 \leq \boldsymbol{x_4} \leq 1400$. Table 6 exhibits the single objective optimization solutions derived by the eight metaheuristic optimizers. It is observed that all the competing optimizers furnished improved optimal MRR than the optimal MRR derived by (Devarajaiah & Muthumari, 2018).

Unlike the SHO algorithm, all the algorithms have drastically improved the MRR from its initial value of 0.0647gm/min (Devarajaiah & Muthumari, 2018). From the optimal PC values as registered by the eight competitor algorithms (shown in Table 6), it is worth pointing that all the algorithms deliver almost similar performance. Besides, the optimal PC provided by the optimizers are found to be relatively better than the optimal PC endorsed by (Devarajaiah & Muthumari, 2018). When the convergence traits of the eight competitor algorithms are analyzed (shown in Figure 5), it is noted that HHO, and TLBO have accelerated tendency to converge faster to global optimal solution implying better exploitation potential of the two algorithms.

Table 6. Single objective optimization outcomes.

Optimizer	Response	Optimal Value	Pulse duration	Pulse off period	Current	Wire Speed
(Devarajaiah & Muthumari, 2018)	MRR PC	0.0647 0.589	27 27	4 8	4 1	1400 700
ALO	MRR PC	0.0825 0.523	29.92 13	4 10	5 1	1400 350
ChoA	MRR PC	0.0825 0.523	29.92 13	4 10	5 1	1400 350
HHO	MRR PC	0.0825 0.523	29.92 13	4 10	5 1	1400 350
MPA	MRR PC	0.0825 0.52313	29.92 13	4 10	5 1	1400 350
SHO	MRR PC	0.0669 0.52313	21.17 13	4 10	5 1	350 350
MFO	MRR PC	0.0825 0.52313	29.92 13	4 10	5 1	1400 350
PSO	MRR PC	0.0825 0.52313	29.92 13	4 10	5 1	1400 350
TLBO	MRR PC	0.0825 0.52313	29.92 13	4 10	5 1	1400 350

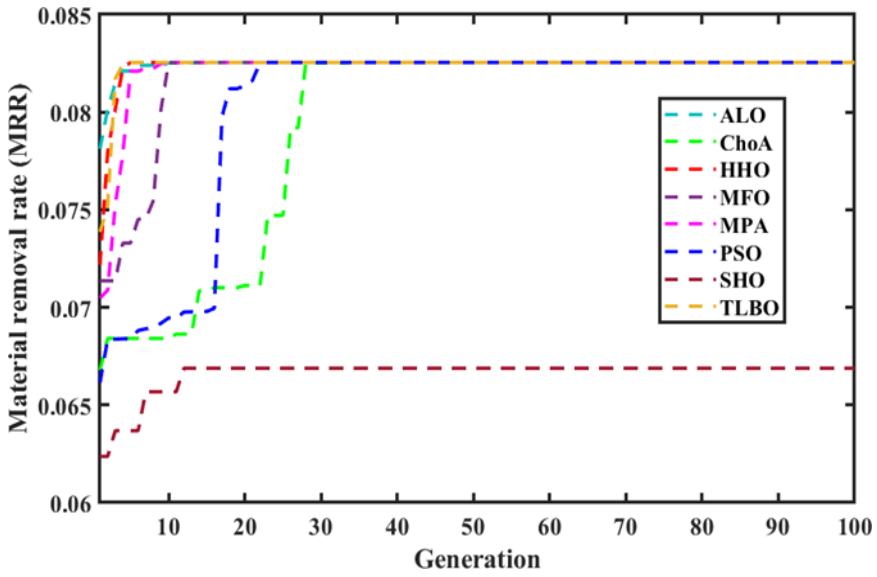

Figure 5. Convergence behavior of ALO, ChoA, MFO, SHO, HHO, MPA, PSO, and TLBO for MRR.

4.2.2. Multiple objective optimization

For multiple performance optimization, Eq. (66) is exploited as the objective

function which is displayed below:

$$\text{Min(Y)} = w_1 \frac{Y(\text{PC})}{\text{PC}_{\text{min}}} - w_2 \frac{Y(\text{MRR})}{\text{MRR}_{\text{max}}} \tag{66}$$

where w_1 and w_2 are the preference weights to PC and MRR respectively. In the present paper, we have assigned equal weights to PC and MRR, respectively. PC_{min} is the minimum power consumption, and MRR_{max} is the maximum material removal rate.

The values are attained from the single-objective optimization results. Table 7 reported the findings of multiple performance optimization exploiting eight metaheuristic optimization algorithms. ALO, MFO, ChoA, HHO, TLBO, and MPA have been discovered to seek the best trade-off condition for both the performance attributes as corroborated when compared with the reported results by (Devarajaiah & Muthumari, 2018). Substantial improvement in MRR with a marginal decrement in the performance of PC is evident utilizing these algorithms. Conversely, SHO and PSO algorithms are found to deliver mediocre optimal performances.

When the convergence traits of the eight competing algorithms are compared, it is discovered that the HHO algorithm rapidly converges to the minimal function value at minimal generation (as shown in Figure 6), confirming the HHO algorithm's exceptional exploitation capability. The comparison of average computational time for the eight algorithms for multiple performance optimization is exhibited in Table 8. It is noted from Table 8 that the TLBO algorithm requires the least computational time to reach the optimality condition.

Table 7. Multiple-objective optimization outcomes.

Optimizer	Response	Optimal Value	Y	Pulse on time	Pulse off period	Current	Wire Speed
(Devarajaiah & Muthumari, 2018)	MRR PC	0.0348 0.625		20	6	2	1050
ALO	MRR PC	0.049 0.670	0.53072	16.12	6.71	5	350
MFO	MRR PC	0.048 0.669	0.53228	16.09	6.77	5	350
ChoA	MRR PC	0.049 0.670	0.53072	16.12	6.71	5	350
HHO	MRR PC	0.049 0.670	0.53072	16.12	6.71	5	350
SHO	MRR PC	0.044 0.676	0.53089	15.75	6.40	5	350
PSO	MRR PC	0.0255 0.572	0.53199	16.12	10	2.88	350
TLBO	MRR PC	0.049 0.670	0.53072	16.12	6.71	5	350
MPA	MRR PC	00.049 0.670	0.53072	16.12	6.71	5	350

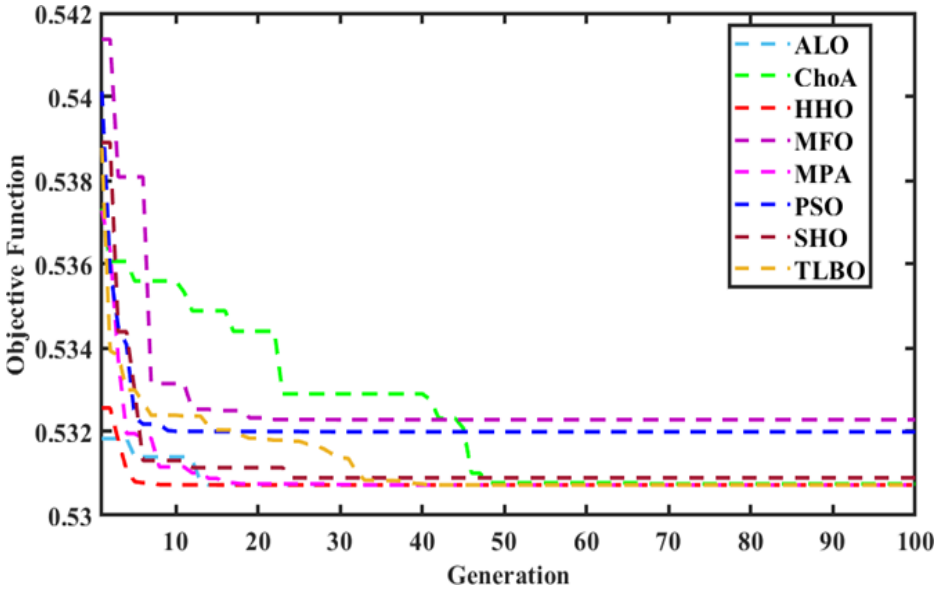

Figure 6. Convergence behavior of ALO, ChoA, MFO, SHO, HHO, MPA, PSO, and TLBO for multi-objective function

Table 8. Average computation time for the eight optimizers

Optimizer	Average computational time (secs)
ALO	5.070851
ChoA	1.664359
MFO	1.119026
HHO	1.052527
MPA	0.720689
SHO	0.1390570
PSO	0.2506701
TLBO	0.3054067

5. Statistical and sensitivity Analysis

To summarize the robustness and performance stability of the metaheuristic optimizers, we retrieved the statistical data evaluated for all the optimizers while dealing with multi-objective optimization problems in the two considered cases. The statistical metrics such as the mean and co-efficient of variation are evaluated after the execution of the optimizers for 30 number of runs.

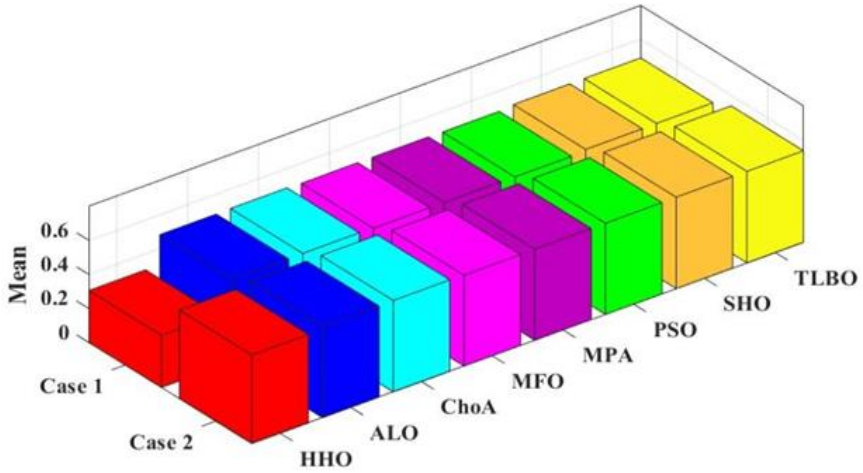

Figure 7. Bar plots revealing mean value of objective function derived using optimizers for the two cases. Case-1 Performance optimization in WEDM of A286 superalloy; Case-2 Performance optimization in WEDM of Ti-6Al-4V superalloy

The mean value of objective function procured using the optimizers for the two cases are plotted in the form of 3D bar plots in Figure (7). From Figure (7), it is clear that HHO provides the least mean value of objective function for all the cases corroborating the robustness of HHO over the other competing algorithms in terms of tracking the global optimal solution. Figure (8) portrays the coefficient of variation procured using the optimizers for the two cases in the form of 3D bar plots. It is noted that HHO exhibits the least coefficient of variation for all the cases corroborating that HHO optimizer has the maximum stability over its other competitors. To summarize, it can also be inferred that HHO exhibits robustness in bolstering an adequate balance between two phases (i.e., exploration and exploitation).

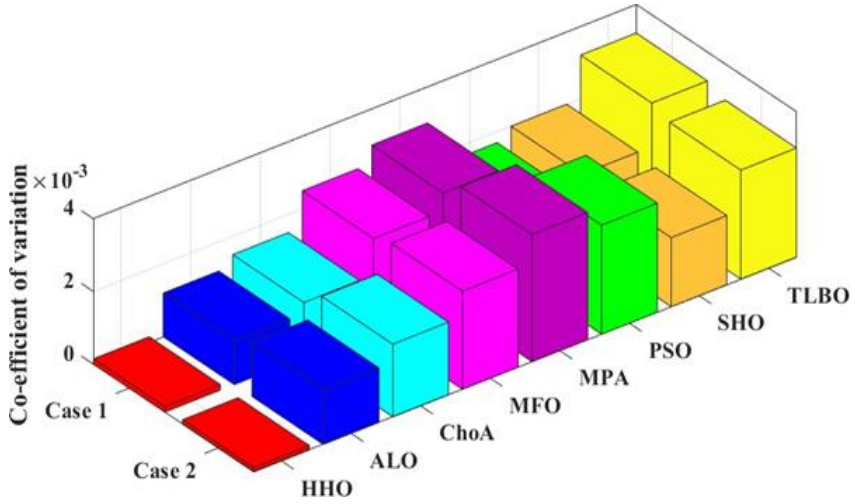

Figure 8. Bar plots revealing coefficient of variation derived using optimizers for the twocases. Case-1 Performance optimization in WEDM of A286 superalloy; Case-2 Performance optimization in WEDM of Ti-6Al-4V superalloy

6. Performance of optimizers on benchmark test functions

As corroborated from the different cases investigated in WEDM on the superiority of HHO over the other competing algorithms, we further intend to investigate the robustness of HHO by comparing its performance with the other competing optimizers on standard test functions. The details of the standard test functions can be found in the literature (Zhu & Kwong, 2010). The standard test functions considered in the present work are the Sphere function, and the Generalized Rastrigin's function, respectively. Sphere function is a unimodal function which contains only one optimum point. Whereas Generalized Rastrigin's function is a multimodal function which contains many local optima but only one global optimum. The mathematical description of the functions is illustrated below:

Sphere function

$$F(x) = \sum_{i=1}^{dim=30} x_i^2 \quad -100 \le x_i \le 100 \tag{67}$$

where *dim* denotes the dimension of the solution space, and $-100 \le \boldsymbol{x_i} \le 100$ depicts the initial range of $\boldsymbol{x_i}$.

Generalized Rastrigin's function

$$\boldsymbol{F(x) = \sum_{i=1}^{dim=30} \left[x_i^2 - 10\cos(2\pi x_i + 10\right] \quad -5.12 \le x_i \le 5.12} \tag{68}$$

where *dim* refers the dimension of the solution space, and $-5.12 \le \boldsymbol{x_i} \le 5.12$ depicts the initial range of $\boldsymbol{x_i}$.

Table 9 depicts the results (Average ± standard deviation) of the eight optimizers on the optimization of benchmark test functions. HHO algorithm is found to be superior over the other optimizers in terms of robustness.

Table 9. Performance comparison on benchmark test functions

Optimizer	ALO	ChoA	MFO	HHO	MPA	SHO	PSO	TLBO
Sphere Function	5.30E-09 ±3.4035 E-09	2.60E-06 ±6.43 E-06	3.38E-13 ±5.21 E-13	0 ± 0	2.76 E-21 ±3.66 E-21	6.61 E-100 ±1.5 E-99	3.39 E-07 ±8.40871 E-07	2.83E-89 ±6.78 E-89
Generalized Rastrigin's Function	23.4809 8± 12.6612 3	10.63647 ± 10.56995	20.702215 ± 11.504771 2	0 ± 0	0 ± 0	0 ± 0	12.43697 ± 6.52438	2.45E+00 ± 2.1543798 09

7. Conclusions

WEDM is a complicated machining process. Machining in WEDM at any parametric combination does not result in enhanced performance outcomes. For improved performances in WEDM, machining must be carried out in compliance with the optimal parametric settings. Metaheuristic optimizers have grown in prominence due to their potential to provide global optimal solutions. However, the use of recently reported metaheuristic optimizers in non-traditional machining techniques has received little attention.

The novelty of the present article is to explore the six recently reported metaheuristic optimizers namely the ant lion optimization (ALO), chimp optimization algorithm (ChoA), moth flame optimization (MFO), spotted hyena optimization (SHO), Harris Hawk optimization algorithm (HHO), and Marine predator algorithm (MPA) in the optimization of WEDM performances for two WEDM processes.

Two well-known existing optimization approaches (i.e., PSO and TLBO), are also included in this study to allow a fair evaluation of the algorithms' performance. The comparison between the eight algorithms are carried out in terms of the optimal solutions, convergence rate, and average computational time. The goal of the comparative analyses is to select the robust optimizer. It is observed that the HHO algorithm is extremely robust in yielding global optimal solutions.

Moreover, HHO algorithm surpasses other competitors in terms of rapid convergence. Thus, HHO portrays high exploration and exploitation potential. TLBO algorithm shows the least average computation time. Future research might focus on the exploitation of HHO to determine the optimal operating condition in other areas of manufacturing.

Acknowledgment

S Saha is grateful for the financial support from MoE, India, during this work**.**

References

Absil, P.-A., Diao, O., & Diallo, M. (2021). Assessment of COVID-19 hospitalization forecasts from a simplified SIR model. Letters in Biomathematics. https://doi.org/10.30707/LiB8.1.1682013528.154572

Abualigah, L., Shehab, M., Alshinwan, M., Mirjalili, S., & Elaziz, M. A. (2021). Ant lion optimizer: a comprehensive survey of its variants and applications. Archives of Computational Methods in Engineering, 28, 1397-1416. https://doi.org/10.1007/s11831-020-09420-6

Devarajaiah, D., & Muthumari, C. (2018). Evaluation of power consumption and MRR in WEDM of Ti–6Al–4V alloy and its simultaneous optimization for sustainable production. Journal of the Brazilian Society of Mechanical Sciences and Engineering, 40, 1-18. https://doi.org/10.1007/s40430-018-1318-y

Dhiman, G., & Kumar, V. (2017). Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications. Advances in Engineering

Software, 114, 48-70. https://doi.org/10.1016/j.advengsoft.2017.05.014

Faramarzi, A., Heidarinejad, M., Mirjalili, S., & Gandomi, A. H. (2020). Marine Predators Algorithm: A nature-inspired metaheuristic. Expert Systems with Applications, 152, 113377. https://doi.org/10.1016/j.eswa.2020.113377

Garg, M. P., Jain, A., & Bhushan, G. (2012). Modelling and multi-objective optimization of process parameters of wire electrical discharge machining using non-dominated sorting genetic algorithm-II. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 226(12), 1986-2001. https://doi.org/10.1177/0954405412462778

Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. Future generation computer systems, 97, 849-872. https://doi.org/10.1016/j.future.2019.02.028

Houssein, E. H., Mahdy, M. A., Fathy, A., & Rezk, H. (2021). A modified Marine Predator Algorithm based on opposition based learning for tracking the global MPP of shaded PV system. Expert Systems with Applications, 183, 115253. https://doi.org/10.1016/j.eswa.2021.115253

Juneja, M., & Nagar, S. (2016). Particle swarm optimization algorithm and its parameters: A review. In 2016 International Conference on Control, Computing, Communication and Materials (ICCCCM) (pp. 1-5). IEEE. https://doi.org/10.1109/ICCCCM.2016.7918233

Kaur, M., Kaur, R., Singh, N., & Dhiman, G. (2022). Schoa: a newly fusion of sine and cosine with chimp optimization algorithm for hls of datapaths in digital filters and engineering applications. Engineering with Computers, 38(Suppl 2), 975-1003. https://doi.org/10.1007/s00366-020-01233-2

Khan, N. Z., Khan, Z. A., Siddiquee, A. N., & Chanda, A. K. (2014). Investigations on the effect of wire EDM process parameters on surface integrity of HSLA: A multi-performance characteristics optimization. Production & Manufacturing Research, 2(1), 501-518. https://doi.org/10.1080/21693277.2014.931261

Khishe, M., & Mosavi, M. R. (2020). Chimp optimization algorithm. Expert Systems with Applications, 149, 113338. https://doi.org/10.1016/j.eswa.2020.113338

Krishna, M. M., Panda, N., & Majhi, S. K. (2021). Solving traveling salesman problem using hybridization of rider optimization and spotted hyena optimization algorithm. Expert Systems with Applications, 183, 115353. https://doi.org/10.1016/j.eswa.2021.115353

Kuriakose, S., & Shunmugam, M. (2005). Multi-objective optimization of wire-electro discharge machining process by non-dominated sorting genetic algorithm. Journal of materials processing technology, 170(1-2), 133-141. https://doi.org/10.1016/j.jmatprotec.2005.04.105

Lee, K. Y., & Park, J.-B. (2006). Application of particle swarm optimization to economic

dispatch problem: advantages and disadvantages. In 2006 IEEE PES power systems conference and exposition (pp. 188-192). IEEE. https://doi.org/10.1109/PSCE.2006.296295

Majumder, H., & Maity, K. (2018). Prediction and optimization of surface roughness and micro-hardness using grnn and MOORA-fuzzy-a MCDM approach for nitinol in WEDM. Measurement, 118, 1-13. https://doi.org/10.1016/j.measurement.2018.01.003

Mandal, A., Dixit, A. R., Das, A. K., & Mandal, N. (2016). Modeling and optimization of machining nimonic C-263 superalloy using multicut strategy in WEDM. Materials and Manufacturing Processes, 31(7), 860-868. https://doi.org/10.1080/10426914.2015.1048462

Mansoor, M., Mirza, A. F., & Ling, Q. (2020). Harris hawk optimization-based MPPT control for PV systems under partial shading conditions. Journal of Cleaner Production, 274, 122857. https://doi.org/10.1016/j.jclepro.2020.122857

Mirjalili, S. (2015a). The ant lion optimizer. Advances in Engineering Software, 83, 80-98. https://doi.org/10.1016/j.advengsoft.2015.01.010

Mirjalili, S. (2015b). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. Knowledge-Based Systems, 89, 228-249. https://doi.org/10.1016/j.knosys.2015.07.006

Naik, M. K., Panda, R., Wunnava, A., Jena, B., & Abraham, A. (2021). A leader Harris hawks optimization for 2-D Masi entropy-based multilevel image thresholding. Multimedia Tools and Applications, 1-41. https://doi.org/10.1007/s11042-020-10467-7

Nayak, B. B., & Mahapatra, S. S. (2016). Optimization of WEDM process parameters using deep cryo-treated Inconel 718 as work material. Engineering Science and Technology, an International Journal, 19(1), 161-170. https://doi.org/10.1016/j.jestch.2015.06.009

Poli, R., Blackwell, T., & Kennedy, J. (2007). Particle swarm optimization. Swarm Intelligence, 1(1), 33-57. https://doi.org/10.1007/s11721-007-0002-0

Ramakrishnan, R., & Karunamoorthy, L. (2008). Modeling and multi-response optimization of Inconel 718 on machining of CNC WEDM process. Journal of materials processing technology, 207(1-3), 343-349. https://doi.org/10.1016/j.jmatprotec.2008.06.040

Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Computer-aided design, 43(3), 303-315. https://doi.org/10.1016/j.cad.2010.12.015

Sabahno, M., & Safara, F. (2022). ISHO: improved spotted hyena optimization algorithm for phishing website detection. Multimedia Tools and Applications, 81(24),

34677-34696. https://doi.org/10.1007/s11042-021-10678-6

Sadeghi, M., Razavi, H., Esmaeilzadeh, A., & Kolahan, F. (2011). Optimization of cutting conditions in WEDM process using regression modelling and Tabu-search algorithm. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 225(10), 1825-1834. https://doi.org/10.1177/0954405411406639

Saha, S., Maity, S. R., Dey, S., & Dutta, S. (2021). Modeling and combined application of MOEA/D and TOPSIS to optimize WEDM performances of A286 superalloy. Soft Computing, 25, 14697-14713. https://doi.org/10.1007/s00500-021-06264-5

Shan, W., Qiao, Z., Heidari, A. A., Chen, H., Turabieh, H., & Teng, Y. (2021). Double adaptive weights for stabilization of moth flame optimizer: Balance analysis, engineering cases, and medical diagnosis. Knowledge-Based Systems, 214, 106728. https://doi.org/10.1016/j.knosys.2020.106728

Shehab, M., Abualigah, L., Al Hamad, H., Alabool, H., Alshinwan, M., & Khasawneh, A. M. (2020). Moth–flame optimization algorithm: variants and applications. Neural Computing and Applications, 32(14), 9859-9884. https://doi.org/10.1007/s00521-019-04570-6

Sultana, S., & Roy, P. K. (2014). Multi-objective quasi-oppositional teaching learning based optimization for optimal location of distributed generator in radial distribution systems. International Journal of Electrical Power & Energy Systems, 63, 534-545. https://doi.org/10.1016/j.ijepes.2014.06.031

Tarng, Y., Ma, S., & Chung, L. (1995). Determination of optimal cutting parameters in wire electrical discharge machining. International Journal of Machine Tools and Manufacture, 35(12), 1693-1701. https://doi.org/10.1016/0890-6955(95)00019-T

Tonday, H. R., & Tigga, A. M. (2019). An empirical evaluation and optimization of performance parameters of wire electrical discharge machining in cutting of Inconel 718. Measurement, 140, 185-196. https://doi.org/10.1016/j.measurement.2019.04.003

Uzlu, E., Kankal, M., Akpınar, A., & Dede, T. (2014). Estimates of energy consumption in Turkey using neural networks with the teaching–learning-based optimization algorithm. Energy, 75, 295-303. https://doi.org/10.1016/j.energy.2014.07.078

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. IEEE transactions on evolutionary computation, 1(1), 67-82. https://doi.org/10.1109/4235.585893

Yakout, A. H., Sabry, W., Abdelaziz, A. Y., Hasanien, H. M., AboRas, K. M., & Kotb, H. (2021). Enhancement of frequency stability of power systems integrated with wind energy using marine predator algorithm based PIDA controlled STATCOM. Alexandria Engineering Journal, 61(8), 5851-5867. https://doi.org/10.1016/j.aej.2021.11.011

Yao, Y., Li, Y., Xie, D., Hu, S., Wang, C., & Li, Y. (2021). Coverage enhancement strategy for WSNs based on virtual force-directed ant lion optimization algorithm. IEEE

sensors journal, 21(17), 19611-19622. https://doi.org/10.1109/JSEN.2021.3091619

Yoshioka, H., Yaegashi, Y., Yoshioka, Y., & Tsugihashi, K. (2019). A short note on analysis and application of a stochastic open-ended logistic growth model. Letters in Biomathematics, 6(1), 67-77. https://doi.org/https://doi.org/10.30707/LiB6.1Yoshioka

Zhu, G., & Kwong, S. (2010). Gbest-guided artificial bee colony algorithm for numerical function optimization. Applied mathematics and computation, 217(7), 3166-3173. https://doi.org/10.1016/j.amc.2010.08.049