BSEU JOURNAL OF ENGINEERING RESEARCH AND TECHNOLOGY, VOLUME I, ISSUE I, DECEMBER 2020

Multi-Objective Harris Hawks Optimizer for Multiobjective Optimization Problems

Meryem Kuşoğlu[1], Uğur Yüzgeç[2*]

Abstract— In this paper, a multi-objective version of the Harris Hawk Optimizer algorithm (HHO) is proposed, which is called Multi-Objective Harris Hawk Optimization (MOHHO). In the MOHHO algorithm, preserving the structure of the HHO algorithm, an archive repository has been added to the HHO algorithm to save and retrieve the Pareto optimal results. This repository is used for simulating the positions and solutions of the hawks. The archive member in the least populated area from this archive is selected using the roulette wheel process. This archive member is utilized as the rabbit in the proposed MOHHO algorithm. To show the performance of the MOHHO algorithm, we have taken unconstrained test functions known as ZDT from the literature. For the multi objective benchmarks, the MOHHO algorithm was compared with MOALO (Multi-objective AntLion optimizer) and MODA (Multi-objective Dragonfly optimizer) algorithms. Inverted Generational Distance (IGD) metric was used for ZDT benchmark comparison studies. The comparison results show that the proposed algorithm gives better results than the MOALO and MODA algorithms in terms of IGD metric for all test functions.

Keywords — Harris Hawks Optimizer (HHO), Multi-objective optimization, Multi-objective Harris Hawks optimizer.

I INTRODUCTION

Optimization, it is to find the best solution to a problem. Different algorithms can be used when solving problems with bearing a single goal. In general, optimization method is divided into two groups: heuristic method and mathematical method. Algorithms called meta-heuristics have been developed with the combination of basic heuristic methods. Optimization algorithms are used effectively in developing and hybrid studies to provide effective solutions to encountered problems [1].

Problems encountered in real life are multiple objective optimization problems. The single objective optimization methods have been insufficient to solve these problems. As a result of this, multi-objective optimization algorithms have been developed. These algorithms which works for multiple objectives, is quite successful compared to single-purpose optimization methods. In multi-objective optimization problems, as increasing the number of conflicting objective functions, the difficulty level of the problem increases [2].

Multi-objective optimization involves minimizing or maximizing multiple objectives subject to several constraints. The multi objective optimization problems basically include optimizing the other target while trying to optimize one of the goals. As an example from real life, analyzing the part to be compromised in a design, selecting the most suitable product and process designs, and other applications where an optimum solution between two or more conflicting goals is needed [3].

The problems in the world are mostly continuous, discrete, limited, or unlimited. It is difficult to make a few classifications using traditional mathematical programming techniques due to these features [4]. Some studies have shown that methods such as equality constraint are not efficient or generally not sufficient for large scale problems against nonlinear and indistinguishable problems [5]. Accordingly, meta-heuristic algorithms have been designed and used as an alternative solution due to their convenience. The common shortcoming of meta-heuristic algorithms is quite sensitive in tuning user-defined variables. Another disadvantage is that meta-heuristic algorithms can't always approach the global optimum [6]. There are two types of meta-heuristic algorithms, one-solution, and population-based [7]. While one solution is obtained in single solution optimization, a new solution is developed at every repetition in population-based optimization. Population-based meta-heuristic algorithms are often inspired by the nature [8 [9][10]. In these algorithms, each individual in the population represents a candidate solution in the search space. The population is updated iteratively and the new population is obtained. A stop criterion is determined and optimization is enabled to work up to this criterion [11][12].

Harris hawk optimizater (HHO) is a fresh algorithm developed by Heidari et al. [4]. Harris hawks are modeled after their actions to catch their prey and cooperative behavior. Harris hawks can choose different chase models depending on the types of flights they make for hunting. There are three main steps in HHO: first, to discover the prey, second, the surprise attack (claw), and lastly, the selection and implementation of one of the four different

[1] First author mail: meryemkusoglu26@gmail.com (https://orcid.org/0000-0003-2260-4318)
Department of Computer Engineering, Bilecik Seyh Edebali University, 11210 Bilecik, Turkey
[2*] Corresponding author mail: ugur.yuzgec@bilecik.edu.tr (https://orcid.org/0000-0002-5364-6265)
Department of Computer Engineering, Bilecik Seyh Edebali University, 11210 Bilecik, Turkey

attack strategies of Harris hawks. HHO is a population-based, gradient-free algorithm. Therefore, it can be applied to any problem with the appropriate formulation. Although it is a fairly new algorithm, it is observed that it is used in many studies when the literature is examined. The studies that have a context with this paper are summarized below.

By looking at Lefebvre L.'s proposal in 1997 [13] and his studies [14] [15] [16], falcons can be listed among the most intelligent birds in nature. One of the features that distinguish Harris hawks from other predators is that they search for food with other members of the family. When they set a goal, they try to get to know other members of the family and be aware of their actions. Harris hawks' hunting tactic is the "Surprise attack" strategy known as "Seven kills" [4].

Essam H. Houssein[17] in his work on drug design and discovery proposed a classification called HHO-SVM that hybridizes Harris hawk optimizer (HHO) with support vector machines for chemical descriptor selection and chemical compound activities. As a result of his study, he concluded that he achieved high accuracy and performance against other algorithms (PSO, DA, BOA, MFO, GWO, and SCA) he compared.

Huiling Chen [18] designed the HHO algorithm, which integrates the OBL mechanism and the CLS strategy together to estimate the solar cell model and the parameters of the photovoltaic modular. The system he designed has taken more stable and better results compared to other studies.

Dieu Tien Bui [19] used the HHO algorithm by synthesizing it with an artificial neural network (ANN) to overcome the computational deficiency in the spatial modeling of the landslide susceptibility map. The landslide susceptibility map produced by the HHO-ANN algorithm has shown that it is more successful than the ANN map in terms of predicting unseen landslide events. Hao Chen [20] added chaos strategy, multi-population mechanism, and differential evolution (DE) strategies to the HHO algorithm to improve performance in his article. Thus, it increased the diversity of the HHO population (CMDHHO). Overall, CMDHHO has significantly enhanced HHO's core global and local search capabilities.

N. A. Golilarz [21] proposed a new automated method based on the proposed ConvNet (a convolutional neural network) and optimization algorithm for the recognition of CCP (nine control chart models). HHO-ConvNet has compared several experiments and other methods performed to see the quality and performance analysis. His proposed CCP system has recorded 99.80 correct classifications. N. Amiri Golilarz [22] used the HHO algorithm to remove noise on the image and noise in his study. He compared his results with alternative results. He noted that the HHO algorithm visually and quantitatively performed much better than the JADE algorithm.

Vikram Kumar Kamboj [23] has successfully upgraded the existing HHO algorithm using the Sine-Cosine algorithm. He tested the hybrid hHHO-SCA algorithm he developed on continuous, discrete, constrained, nonlinear and convex engineering design. He observed that it was successful against other algorithms compared in his study.

Ahmed A. Ewees[24] used chaotic maps to optimize the parameters of MVO in his study to solve engineering problems, and HHO used the search field of the MVO to make a local search. Proposed CMVHHO has successfully applied to solve the four selected engineering problems. X. Bao [25] proposed an alternative method for color image segmentation, inspired by the hybridization of HHO and DE algorithms. He used the Otsu method and Kapur entropy and universality values. He showed that his work on various images had satisfactory results.

Pei Du [26] developed the MOHHO algorithm to estimate air pollutant concentrations and to adjust the parameters of the ELM model to obtain high accuracy and stability at the same time. The hybrid model he developed according to the results of his study showed that it gives high accuracy and stable results compared to other models used in the comparison.

Z. M. Elgamal [27] developed the CHHO algorithm by including chaotic maps in the HHO algorithm in his study for feature selection in the medical field and adding SA (Simulated Annealing) algorithm at the stage of use. He compared the CHHO algorithm with other optimization algorithms (GOA, GA, PSO, BOA, ALO). As a result of his comparison, he found that the CHHO algorithm was more successful.

Akdağ O [28] applied the HHO algorithm to the Optimum Load Flow problem in his study for the minimization of active power losses. In his study, he reduced the active power loss to 22.68MW. Islam, M. Z. [29] tried to compensate for total fuel cost, active power loss, and environmental emission cost in his study on optimal power flow that takes HHO-based single and multi-purpose environmental emission into account. He recorded that the HHO algorithm was more successful when he compared his results with SSA, WOA, MF, and GWO algorithms.

Houssein, E. H. [30] aimed to select the most important features in his study and to classify the information in the kimformatics data sets. He proposed the CHHO-CS method, combining the HHO algorithm with CS (cuckoo search) and C (chaotic map) operators to improve the performance of HHO. He combined this system he proposed with the chemical descriptor selection and SVM to handle its chemical activities. At the end of his study, he revealed that CHHO-CS gave successful results in the comparison with standard algorithms.

In this study, a version of Harris Hawk optimizer adapted to the multiobjective optimization problem is presented. The proposed MOHHO algorithm was tested for the unconstrained ZDT1, ZDT2, ZDT3, ZDT4, ZDT5, and ZDT6 test problems from the literature. IGD performance metric was used to see better its accuracy and convergence. MOHHO’s performance was compared with MOALO, MOPSO, and MODA algorithms. According to the comparison results, MOHHO has the best performance.

II HARRIS HAWK OPTIMIZER (HHO)

Harris hawk optimizer is a herd-based algorithm without gradients [31]. It has various activities, time-varying stages of exploration and exploitation [4]. It is flexible and at the same time gives high performance and quality results. The main logic of the HHO algorithm was inspired by the cooperative behavior of hawks in nature and the hunting style called "surprise jump" [32]. The HHO algorithm process for optimization problems is basically based on two phases: exploration and exploitation [33].

1. Exploration Phase

Falcons have keen eyes for detecting and tracking its prey. But sometimes it is not easy to find prey. Therefore, hawks may have to perch on a place and wait patiently for hours. The response of this behavior in HHO is modeled as the discovery stage as follows [4]:

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1 |X_{rand}(t) - 2r_2 X(t)| & q \geq 0.5 \\ (X_{rabbit}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)) & q < 0.5 \end{cases} \quad (1)$$

where $X(t+1)$ is the position vector of hawks in the next iteration, $X(t)$ is the current position of hawks, $X_{rabbit}(t)$ is the position of the rabbit, that is, the prey. $r_1 - r_4$ and q are the random numbers between [0,1]. LB and UB are the upper and lower limits of the optimization problem. $X_{rand}(t)$ is a hawk chosen randomly. X_m represents the mean position of the population. It is calculated as follows:

$$X_m(t) = \frac{1}{N} \sum_{i=1}^{N} X_i(t) \quad (2)$$

where N denotes the population size.

2. Transition from Exploration to Exploitation

This stage is important for the performance of meta-heuristic algorithms. The escaped energy of the rabbit (target), called E in HHO, is used for the transformation between exploration and exploitation. The value of E decreases after each iteration [4]. Mathematical model is given below:

$$E = 2E_0 * \left(1 - \frac{t}{T}\right) \quad (3)$$

where E_0 is a random number in the range [-1,1], t represents the current iteration and T stands for the maximum iteration. $|E| \geq 1$ exploration phase is used to search for prey; $|E| < 1$ is used to take advantage of promising space.

3. Exploitation phase

When it comes to this stage, Harris hawks attack the hunt they have determined in the previous stage. While the hawks attack, the prey tries to escape. Therefore, different tracking styles emerge from the real situation. There are four strategies to model the attack phase according to the chasing strategies of the hawks [4]. A random number (r) is used to determine whether the prey can escape. A condition of $r < 0.5$ indicates that the prey has escaped, and $r \geq 0.5$ the prey cannot escape. E (the escape energy of the prey) affects the behavior of the hawk. If $|E| \geq 0.5$, soft siege; If $|E| < 0.5$, they perform a hard siege [34]. In Fig. 1, these phases of HHO algorithm are shown.

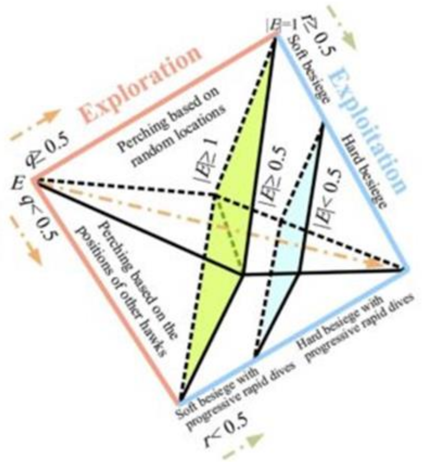


Figure 1. Stages of Harris hawk optimization (HHO) [4].

4. Soft Besiege

At this stage, Harris hawks make attacks that mislead the prey, reducing the energy of its prey. This method of attack is called soft siege and its mathematical equivalent is as follows [4].

$$X(t+1) = \Delta X(t) - E|JX_{rabbit}(t) - X(t)| \quad (4)$$

$$\Delta X(t) = X_{rabbit}(t) - X(t) \quad (5)$$

$\Delta X(t)$ denotes the difference between the rabbit's position and the current position in t iteration. J represents the rabbit's movement in nature and it is randomly calculated in each iteration [32].

5. Hard Besiege

It is the strategy applied after the energy of the prey decreases ($r \geq 0.5, |E| \leq 0.5$). Harris hawks hardly make any siege to catch prey. Its mathematical expression is given below:

$$X(t+1) = X_{rabbit}(t) - E|\Delta x(t)| \quad (6)$$

An example of this step with a hawk is illustrated in Fig. 2.

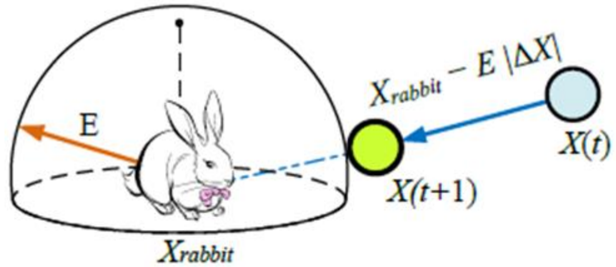


Figure 2. Hard besiege of HHO [4].

6. Soft Besiege with Progressive Fast Dives

At this stage where the prey still has the energy to escape, Harris hawk makes a soft siege before the surprise paw move. For a soft encirclement, it is assumed that they decide their next move based on the rule in Eq. (7).

$$Y = X_{rabbit}(t) - E|JX_{rabbit}(t) - X(t)| \tag{7}$$

It is estimated whether they will dive according to Levy Flight (LF)-based models as given the following rule:

$$Z = Y + S \times LF(D) \tag{8}$$

where D denotes the size of the problem, S stands for a random vector with the $1xD$ size. LF is calculated using the levy fight function:

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}$$
$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}} \tag{9}$$

where β is a constant (1.5), u, v represent random numbers. Eq. (10) is used to update the location of the hawks during the soft siege phase.

$$X(t+1) = \begin{cases} Y \; if F(Y) < F(X(t)) \\ Z \; if F(Z) < F(X(t)) \end{cases} \tag{10}$$

While calculating the Y and Z equations, the 7th and 8th equations are used. For hawk, this step transformed into a simple visual is shown in Fig.3.

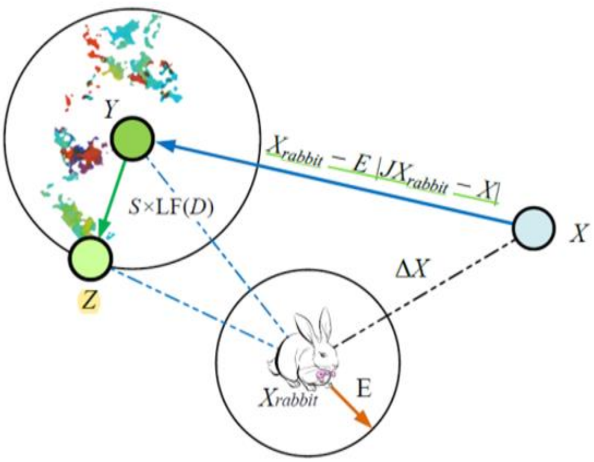


Figure 3. Soft Besiege with progressive fast dives [4].

7. Hard Besiege with Progressive Fast Dives

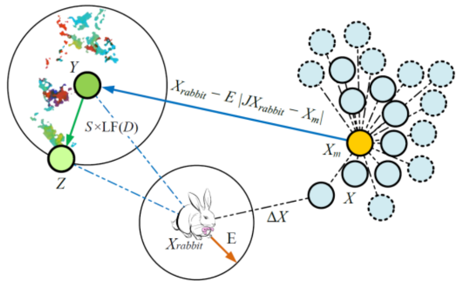
In this phase, the prey has no energy. The Harris hawk makes a fierce siege before the surprise paw action to catch its prey. In hard besiege condition, the position vector of hawks are determined as the following rule:

$$X(t+1) = \begin{cases} Y \; if F(Y) < F(X(t)) \\ Z \; if F(Z) < F(X(t)) \end{cases} \tag{11}$$

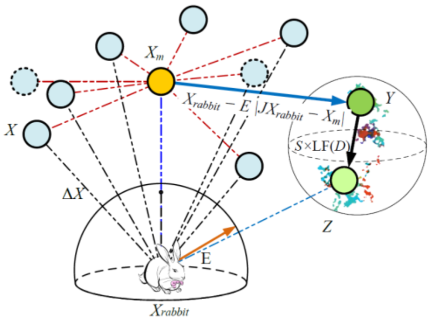
The Y and Z values are obtained using the new rules in Eq. (12) and Eq. (13) as given below.

$$Y = X_{rabbit}(t) - E|JX_{rabbit}(t) - X_m(t)| \tag{12}$$
$$Z = Y + S \times LF(D) \tag{13}$$

Here, the value of $X_m(t)$ is obtained by using Eq. (2). A simple example of the operations mentioned at this stage is shown in Fig. 4 for 2D and 3D spaces. The pseudo-code of the original HHO algorithm is given in Algorithm 1.



(a) In 2D space



(b) In 3D space

Figure 4. Soft Besiege with progressive fast dives [4].

Algorithm 1: Pseudo-code of the HHO algorithm

Inputs: The population size N and maximum number of iterations T
Outputs: The location of rabbit and its fitness value
Initialize the random population X_i(i = 1, 2, …, N)
while (stopping condition is not met) do
    Calculate the fitness values of hawks
    Set X_{rabbit} as the location of rabbit (best location)
    for (each hawk (X_i)) do
        Update the initial energy E_0 and jump strength J
        E_0=2rand()-1, J=2(1-rand())
        Update the E using Eq. (3)
        if (|E| ≥ 1) then
            Update the location vector using Eq. (1)
        if (|E| < 1) then
            if (r ≥0.5 and |E|≥0.5) then
                Update the location vector using Eq. (4)
            else if (r ≥0.5 and |E < 0.5) then
                Update the location vector using Eq. (6)
            else if (r <0.5 and |E| ≥ 0.5) then
                Update the location vector using Eq. (10)
            else if (r <0.5 and |E| < 0.5) then
                Update the location vector using Eq. (11)
Return X_{rabbit}

III Multi Objective Harris Hawk Optimization Algorithm (MOHHO)

The aim of multi-objective optimization [35] is to find the most diverse Pareto optimal solutions accurately [36]. As can be understood from the name of multi-objective optimization, it is used for optimization with more than one objective [37], but before multi-objective optimization, it was solved by gathering in a possible target to solve multi-objective optimization problems [38].

The solution is easy to compare as there is only one goal in single objective optimizations and relational operators (≥, ≤, etc.) are used. Since there is more than one goal in multi-objective optimization problems, the best stability must be found among all objectives. The mathematical formula of a minimization problem is as given below [39]:

$$min\ F(x) = \{f_1(x), f_2(x), f_3(x), \dots, f_m(x)\} \quad (14)$$

$$Subject\ to \quad q_i(x) \geq 0, i = 1, 2, \dots, z \quad (15)$$

$$q_i(x) = 0, i = 1, 2, 3, \dots, k \quad (16)$$

$$L_i \leq x_i \leq U_i, i = 1, 2, 3, \dots., p \quad (17)$$

where m denotes the number of objective functions, q_i represents i^{th} inequality constraint, z stands for the number of the inequality constraints, k is the number of equality constraints, and p denotes the number of variables. L_i and U_i represent the lower and upper limits.

In this section, the MOHHO algorithm has been developed to solve multi-objective optimization problems using the HHO algorithm. An archive repository and a selection of roulette wheels method were used for the proposed multi-objective optimizer. The archive first stores the selected non-dominant Pareto optimal solutions in the current iteration. The important thing in this part is that the maximum size of the archive is predetermined. In each iteration, non-dominant solutions are compared with archived members, and the archive is updated. It is assumed that when the archive is full, the solutions with the most populated neighborhoods are removed from the archive to store new solutions. The following equation is given for the probability of removing a solution from the archive.

$$P_i = \frac{N_i}{c}, c > 1 \quad (18)$$

In this formulation, c is a constant, N_i is the number of solutions near the *i-th* solution. The steps of the proposed MOHHO algorithm are briefly as follows:

Step 1: N (population size), T (maximum iteration), and S (archive size) values are given. It is started randomly within variable ranges.

Step 2: Objective values $f_1(x)$ and $f_2(x)$ are calculated for each individual (hawk), X (t).

Step 3: Non-dominant Pareto optimal solutions in the population are selected in the current iteration; Thus, in $X_{rabbit}(t)$, the position of the rabbit is obtained by the leader selection mechanism that uses crowd distance to select solutions from a less populated area of the archive by the roulette wheel method.

Step 4: Every X(t) is updated.

Step 5: New objective value is calculated for each hawk, then non-dominant solutions are found; so better solutions are archived.

Step 6: When the archive is full, the data in the archive is deleted by the roulette wheel method to create space in the archive.

Step 7: The solutions in the archive are printed.

The pseudo-code of the proposed MOHHO algorithm is given in Algorithm 2.

Algorithm 2: Pseudo-code of the MOHHO algorithm

```
Inputs: Population size N, maximum iteration T and Archive size S
Initialize the population X_i(i = 1,2,...,N)

for each i:1≤T
    for each i:1≤N
    Calculate the related fitness values
    end for
/* If any search agents in out of the search space and then amend them*/
/* Calculate the related fitness values*/
/* Find out the non-dominated solutions. */
/* Update the archive based on the otained non-dominated solutions*/
            if the archive is full
                        /*Remove several solutions of the archive to update new
                        ones. */
                        /*Apply a roulette wheel and Eq.(18). */
            end if
            if the archive is full
                        /*Update the boundaries to cover the new solutions(S) */
            end if
for i:1≤N
            /*Update the energy Eq. (3). */
            /*Select a random hawk from the archive*/
            /*Select the elite using Roulette wheel from the archive */
            if |E|≥1
                    /*Update the location vector Eq. (1) */
            else if |E|<1
                    if r≥0.5 & |E|≥0.5
                        /*Update the location vector using Eq. (4) */
                    end if
                    if r≥0.5 & |E|<0.5
                        /*Update the location vector Eq. (5) */
                    end if
                    if r<0.5 & |E|≥0.5
                        /*Update the location  vector Eq.(10) */
                    end if
                    if r<0.5 & |E|<0.5
                        /*Update the location  vector Eq.(11) */
                    end if
            end if
end for
end for
Return archive
```

IV Experimental Results

The MHHO algorithm presented in this paper is tested with six unconstrained functions called ZDT taken from the literature. The benchmark functions are given in Table 1. These benchmarks have different Pareto optimal front. To show the performance of the MOHHO algorithm, we utilized MOALO (multi-objective ant lion optimizer) and MODA (multi-objective dragonfly optimizer) algorithms. In the benchmark experiments, we set the maximum iteration number to be 500, the number of search agents to be 100, and the maximum archive size to be 100. To obtain the statistical results, the MOHHO and others are run 10 times.

TABLE 1
Unconstrained multi-objective test problems

ZDT1	$Minimize \quad f_1(x) = x_1$ $Minimize \quad f_2(x) = g(x)\ x\ h(f_1(x), g(x))$ $where \quad g(x) = ((9/(N-1))\sum_{i=2}^{N} x_i$ $h(f_1(x), g(x)) = 1 - \sqrt{(f_1(x)/g(x))}$
ZDT2	$Minimize \quad f_1(x) = x_1$ $Minimize \quad f_2(x) = g(x)\ x\ h(f_1(x), g(x))$ $where \quad g(x) = ((9/(N-1))\sum_{i=2}^{N} x_i$ $h(f_1(x), g(x)) = 1 - \ (f_1(x)/g(x))^2$
ZDT3	$Minimize \quad f_1(x) = x_1$ $Minimize \quad f_2(x) = g(x)\ x\ h(f_1(x), g(x))$ $where \quad g(x) = ((9/(N-1))\sum_{i=2}^{N} x_i$ $h(f_1(x), g(x)) = 1 - \sqrt{(f_1(x)/g(x))}$ $- (f_1(x)/g(x))\ \sin(10\pi f_1(x))$
ZDT4	$Minimize \quad f_1(x) = x_1$ $Minimize \quad f_2(x) = g(x)\ x\ h(f_1(x), g(x))$ $where \quad g(x) = 1 + 10(N-1)$ $+ \sum_{i=2}^{N} (x_i^2 - 10\sin(4\pi x_i))$ $h(f_1(x), g(x)) = 1 - \sqrt{(f_1(x)/g(x))}$
ZDT5	$Minimize \quad f_1(x) = 1 + u_i$ $Minimize \quad f_2(x) = g(x)\ x\ h(x)$ $where \quad g(x) = \sum_{i=2}^{N} v_i$ $h(f_1(x), g(x)) = 1/\ (1 + \ u_i)$ $if \begin{cases} x_i = 0, u_i = 0 \\ \ not,\ u_i = 0 \end{cases}$ $if \begin{cases} u_i < 5, v_i = 2 + \ u_i \\ \ u_i = 5,\ v_i = 1 \end{cases}$
ZDT6	$Minimize \quad f_1(x) = 1$ $- \exp(-4\ x\ x_1)\ x\ \sin(6\pi x_1)^6$ $Minimize \quad f_2(x) = g(x)\ x\ h(f_1(x), g(x))$ $where \quad g(x) = 1$ $+ 9\left(\left(\sum_{i=2}^{N} x_i\right)/(N+1)^{0.25}\right)$ $h(f_1(x), g(x)) = 1 - ((f_1(x)/g(x)))^2$

Inverted Generational Distance (*IGD*) metric is used for ZDT benchmark comparison for all algorithms. The formulation of IGD is given below:

$$IGD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n} \tag{19}$$

where n is the number of Pareto optimal solutions, d_i stands for the Euclidean distance between the i^{th} real Pareto optimal solution and the Pareto optimal solutions obtained from the reference set. In IGD metric, Euclidean distance is calculated for each solution according to the closest Pareto optimal solutions obtained in objective space. The IGD metric provides a qualitative comparison between the proposed MOHHO algorithm and the MOALO and MODA algorithms. Fig. 5 shows the best Pareto optimal solutions obtained by MOHHO, MOALO, and MODA algorithms for ZDT benchmark functions. From these figures, the coverage of the MOHHO algorithm on ZDT benchmarks is better than MOALO, and MODA algorithms.

Table 2 summarizes the statistical results of the IGD metric obtained by MOHHO and other algorithms. The results in this table are ranked according to the mean IGD metric for the benchmarks. Also, the average and overall rankings of all algorithms are given in the last two rows of the table. As can be seen from these scores, the proposed MOHHO algorithm has outperformed MODA and MOALO algorithms. Pareto optimal solutions from Fig. 5 and IGD metric results from Table 2 show that the proposed MOHHO algorithm efficiently approaches the true front of the ZDT test functions.

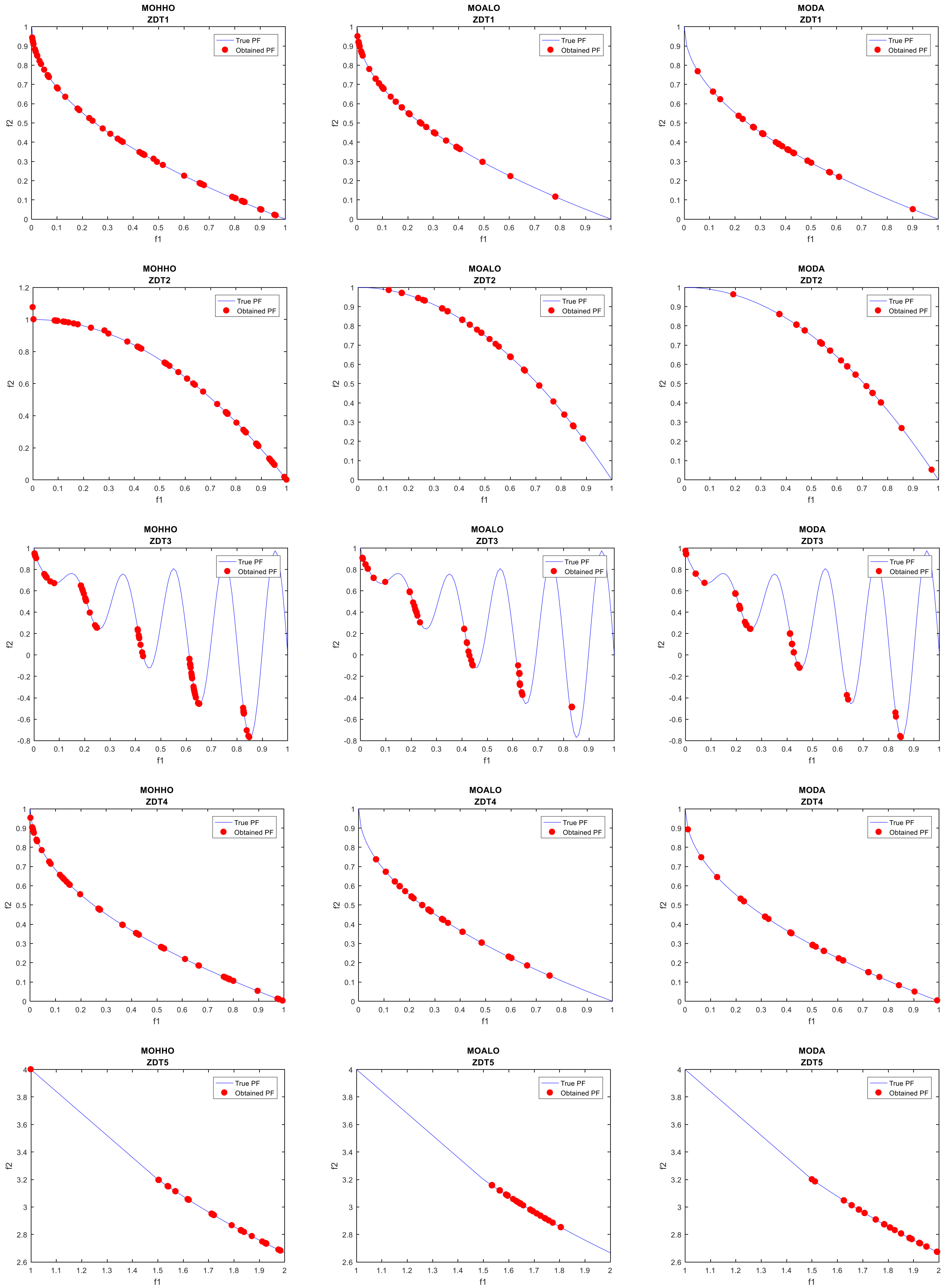
Fig. 6 presents the boxplots of the statistical results of the multi-objective algorithms used in this study. It is clear that the proposed MOHHO algorithm is superior to MALO and MODA algorithms in terms of median, best and worst statistical results. The reason for the successful results obtained here is the effective update mechanisms used in the exploration and exploitation phases of the MOHHO algorithm. In Table 3, computational time results of MOHHO and the other algorithms are summarized. Looking at the average computational time results, it is clear that the best result for ZDT functions belongs to the MOHHO algorithm.

TABLE 3
Computational time results of MOHHO, MOALO and MODA algorithms

	MOHHO	MOALO	MODA
ZDT1	28.46 sec	30.04 sec	31.97 sec
ZDT2	28.30 sec	28.01 sec	39.93 sec
ZDT3	21.59 sec	17.40 sec	32.88 sec
ZDT4	32.02 sec	37.15 sec	38.98 sec
ZDT5	27.80 sec	42.24 sec	37.75 sec
ZDT6	25.70 sec	21.27 sec	34.98 sec
MEAN	**27.31 sec**	**29.35 sec**	**36.08 sec**

V CONCLUSION

In this study, a multi-objective version of the HHO algorithm (MOHHO) proposed in 2019 has been presented. The MOHHO algorithm, which was developed by preserving the main structure of the HHO algorithm, was designed using an archive based on Pareto optimal dominance of HHO to store the best non-dominant Pareto optimal solution obtained during optimization. This algorithm was tested for six unconstrained benchmark functions. IGD performance metric quantitatively demonstrated that MOHHO has high convergence behavior by comparing MOHHO against MALO and MODA algorithms. As a result, it has been observed that the MOHHO algorithm provides much better competitive results in ZDT test functions compared to the compared MOALO and MODA algorithms. In future work, it is recommended to apply the MOHHO algorithm to other engineering design problems.


MOHHO ZDT1
MOALO ZDT1
MODA ZDT1
MOHHO ZDT2
MOALO ZDT2
MODA ZDT2
MOHHO ZDT3
MOALO ZDT3
MODA ZDT3
MOHHO ZDT4
MOALO ZDT4
MODA ZDT4
MOHHO ZDT5
MOALO ZDT5
MODA ZDT5
True PF
Obtained PF
f1
f2

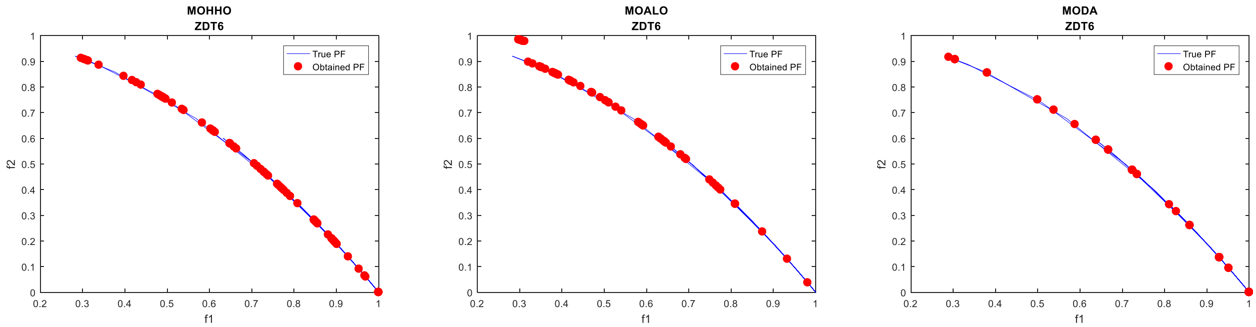


Figure 5. Best Pareto optimal solutions of ZDT1-6 functions of MOHHO, MOALO, and MODA algorithms.

TABLE 2

IGD metric results of MOHHO, MOALO and MODA algorithms

Function Name		MOHHO	MOALO	MODA
ZDT1	Min	2.1558e-03	7.5053e-03	2.6238e-03
	Max	3.1261e-03	2.8622e-02	3.5280e-02
	Mean	2.6590e-03	1.6085e-02	1.0384e-02
	Std	3.1904e-04	6.5678e-03	9.2860e-03
	Rank (Mean)	1	3	2
ZDT2	Min	2.0533e-03	5.8187e-03	3.8275e-03
	Max	4.8615e-03	3.7475e-02	1.7363e-02
	Mean	2.7954e-03	2.0062e-02	8.1583e-03
	Std	7.8483e-04	1.0799e-02	3.9199e-03
	Rank (Mean)	1	3	2
ZDT3	Min	2.3504e-02	2.4158e-02	2.5003e-02
	Max	2.5167e-02	3.2551e-02	3.2223e-02
	Mean	2.4399e-02	2.7230e-02	2.6289e-02
	Std	4.2077e-04	2.8361e-03	2.1390e-03
	Rank (Mean)	1	3	2
ZDT4	Min	2.5782e-03	9.2187e-03	2.8312e-03
	Max	8.8654e-02	1.0964e-01	1.4854e-02
	Mean	2.0324e-02	3.2029e-02	7.0687e-03
	Std	3.5679e-02	2.9537e-02	4.2805e-03
	Rank (Mean)	2	3	1
ZDT5	Min	1.4355e-03	5.8342e-02	3.8521e-02
	Max	5.5223e-02	6.6722e-02	5.9623e-02
	Mean	3.9140e-02	6.1894e-02	5.4596e-02
	Std	2.2496e-02	2.7238e-03	5.8588e-03
	Rank (Mean)	1	3	2
ZDT6	Min	1.6015e-03	3.9523e-03	1.5319e-03
	Max	3.6562e-03	3.8021e-02	3.3530e-03
	Mean	2.3745e-03	1.2621e-02	2.4403e-03
	Std	6.5828e-04	1.0242e-02	6.4006e-04
	Rank (Mean)	1	3	2
Average Rank		**1.167**	**3.000**	**1.833**
Overall Rank		**1**	**3**	**2**

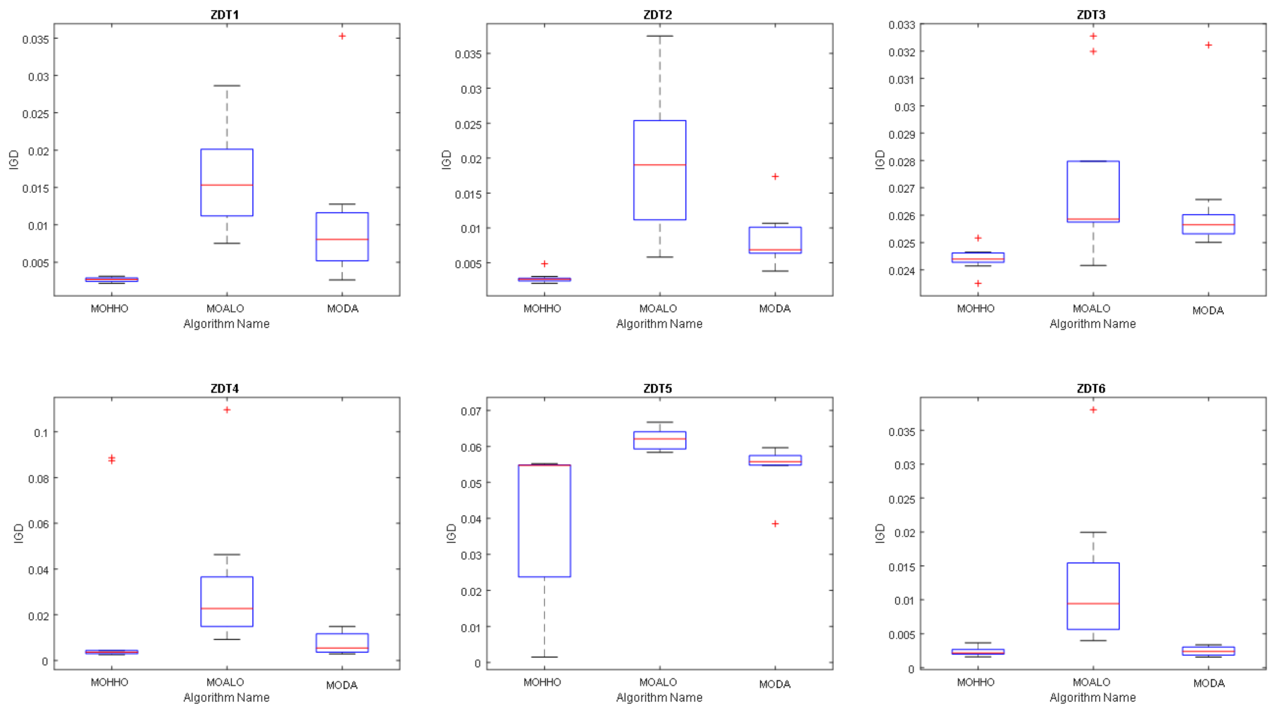


Figure 6. Boxplot of the statistical results for IGD metric from ZDT1 to ZDT6.

Acknowledgment

This paper was conducted within the scope of a postgraduate study conducted with Computer Engineering Department at Institute of Graduate, Bilecik Seyh Edebali University.

References

[1] Y. Çelik, İ. Yıldız, A.T., Karadeniz, "A Brief Review of Metaheuristic Algorithms Improved in the Last Three Years", *European Journal of Science and Technology* Special Issue, pp. 463-477, October 2019.

[2] Kaya S., Fığlalı N., " ÇAO Problemlerinde Pareto Optimal Kullanımı", *Social Sciences Research Journal*, vol. 5, no. 2, pp. 9-18, 2016.

[3] K. Deb., *Multi-objective optimization using evolutionary algorithms*. vol. 16. John Wiley & Sons, 2001.

[4] Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, "Harris hawks optimization: Algorithm and applications", *Future Generation Computer Systems*, vol. 97, pp. 849-872, 2019.

[5] G. Wu, W. Pedrycz, P.N. Suganthan, R. Mallipeddi, "A variable reduction strategy for evolutionary algorithms handling equality constraints", *Applied Soft Computing*, vol. 37, pp. 774-786, 2015.

[6] J Dréo, A Pétrowski, P Siarry, E Taillard *Metaheuristics for Hard Optimization: Methods and Case Studies*, Springer Science & Business Media, 2006.

[7] E.G. Talbi, *Metaheuristics: from Design to Implementation*, vol. 74 John Wiley & Sons, 2009.

[8] H. Kılıç and U. Yüzgeç, "Improved antlion optimization algorithm," 2017 International Conference on Computer Science and Engineering (UBMK), Antalya, 2017.

[9] H. Kılıç, U. Yüzgeç, "Tournament selection based antlion optimization algorithm for solving quadratic assignment problem", *Engineering Science and Technology, an International Journal*, vol. 22, no. 2, pp. 673-691, 2019.

[10] C. Karakuzu, F. Karakaya, M. A. Çavuşlu, "FPGA implementation of neuro-fuzzy system with improved PSO learning", *Neural Networks*, vol. 79, pp. 128-140, 2016.

[11] M. Mafarja, I. Aljarah, A.A. Heidari, H. Faris, P. Fournier-Viger, X. Li, S. Mirjalili, "Binary dragonfly optimization for feature selection using time-varying transfer functions", *Knowledge Based System*, vol. 161, pp. 185-204, 2018.

[12] I. Aljarah, M. Mafarja, A.A. Heidari, H. Faris, Y. Zhang, S. Mirjalili, "Asynchronous accelerating multi-leader salp chains for feature selection", *Applied Soft Computing*, vol. 71, pp. 964-979, 2018.

[13] L. Lefebvre, P. Whittle, E. Lascaris, A. Finkelstein "Feeding innovations and forebrain size in birds", Animal Behaviour, vol.53, pp. 549-560, 1997.

[14] D. Sol, R.P. Duncan, T.M. Blackburn, P. Cassey, L. Lefebvre, "Big brains, enhanced cognition, and response of birds to novel environments", Proceedings

of the National Academy of Sciences, vol. 102, no. 15, pp. 5460-5465, 2005.

[15] F. Dubois, L.A. Giraldeau, I.M. Hamilton, J.W. Grant , L. Lefebvre, "Distraction sneakers decrease the expected level of aggression within groups: a game-theoretic model", The American Naturalist, vol. 164, no.2, pp.32-45, 2004.

[16] D. Sol, T. Székely, A. Liker, L. Lefebvre, " Big-brained birds survive better in nature ", Proceedings of the Royal Society B: Biological Sciences, vol. 274, no. 1611, pp. 763-769, 2007.

[17] E. H. Houssein, M. E. Hosney, D. Oliva, W. M. Mohamed, M. Hassaballah, "A novel hybrid Harris hawks optimization and support vector machines for drug design and discovery", *Computers & Chemical Engineering*, vol. 133, 2020.

[18] H. Chen, S. Jiao, M. Wang, A. A. Heidari, X. Zhao, "Parameters identification of photovoltaic cells and modules using diversification-enriched Harris hawks optimization with chaotic drifts", *Journal of Cleaner Production*", vol. 244, 2020.

[19] D. T. Bui, H. Moayedi, B. Kalantar, A. Osouli, M. Gör, B. Pradhan, et. al. "Harris hawks optimization: A novel swarm intelligence technique for spatial assessment of landslide susceptibility", *Sensors*, vol. 19, no. 16, 2019.

[20] H. Chen, A. A. Heidari, H. Chen, M. Wang, Z. Pan, A. H. Gandomi, "Multi-population differential evolution-assisted Harris hawks optimization: Framework and case studies", *Future Generation Computer Systems*, 2020.

[21] N. A. Golilarz, A. Addeh, H. Gao, L. Ali, A. M. Roshandeh, H. M. Munir, R. U. Khan, "A New Automatic Method for Control Chart Patterns Recognition Based on ConvNet and Harris Hawks Meta Heuristic Optimization Algorithm", *IEEE Access*, vol. 7, pp. 149398-149405, 2019.

[22] N. A. Golilarz, H. Gao and H. Demirel, "Satellite Image De-Noising With Harris Hawks Meta Heuristic Optimization Algorithm and Improved Adaptive Generalized Gaussian Distribution Threshold Function", *IEEE Access*, vol. 7, pp. 57459-57468, 2019.

[23] V. K. Kamboj, A. Nandi, A. Bhadoria, S. Sehgal, "An intensify Harris Hawks optimizer for numerical and engineering optimization problems", *Applied Soft Computing*, vol. 89, 2020.

[24] A. A. Ewees, M. A. Elaziz, "Performance analysis of Chaotic Multi-Verse Harris Hawks Optimization: A case study on solving engineering problems", *Engineering Applications of Artificial Intelligence*, vol. 88, 2020.

[25] X. Bao, H. Jia and C. Lang, "A Novel Hybrid Harris Hawks Optimization for Color Image Multilevel Thresholding Segmentation", *IEEE Access*, vol. 7, pp. 76529-76546, 2019.

[26] P. Du, J. Wang, Y. Hao, T. Niu, W. Yang, "A novel hybrid model based on multi-objective Harris hawks optimization algorithm for daily PM2.5 and PM10 forecasting", *Applied Soft Computing*, vol. 96, 2020.

[27] Z. M. Elgamal, N. B. M. Yasin, M. Tubishat, M. Alswaitti and S. Mirjalili, "An Improved Harris Hawks Optimization Algorithm With Simulated Annealing for Feature Selection in the Medical Field", *IEEE Access*, vol. 8, pp. 186638-186652, 2020.

[28] O. Akdağ, A. Ates, C. Yeroglu, "Harris Şahini Optimizasyon Algoritması ile Aktif Güç Kayıplarının Minimizasyonu", *Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi*, vol. 22, no. 65, pp. 481-490, 2020.

[29] M.Z. Islam, N.I.A. Wahab, V. Veerasamy, H. Hizam, N.F. Mailah, J.M. Guerrero and M.N. Nasir, "A Harris Hawks Optimization Based Single-and Multi-Objective Optimal Power Flow Considering Environmental Emission", *Sustainability*, vol. 12, no. 13, 2020.

[30] E.H. Houssein, M. E. Hosney, M. Elhoseny, D. Oliva, W.M. Mohamed and M. Hassaballah, "Hybrid Harris hawks optimization with cuckoo search for drug design and discovery in chemoinformatics", *Scientific Reports*, vol. 10, no. 1, pp. 1-22, 2020.

[31] Harris Hawks Optimization (HHO), Evoml research group, http://evo-ml.com/2019/03/02/hho, (2020).

[32] Harris Hawks Optimizer, http://mdm.wzu.edu.cn/HHO.html, 2020.

[33] O.M. Ismael, O.S. Qasim and Z.Y. Algamal, "Improving Harris hawks optimization algorithm for hyperparameters estimation and feature selection in v-support vector regression based on opposition-based learning", *Journal of Chemometrics*, 2020.

[34] H. Jia, C. Lang, D. Oliva, W. Song and X. Peng, "Dynamic harris hawks optimization with mutation mechanism for satellite image segmentation", *Remote sensing*, vol. 11, no. 12, 2019.

[35] Deb, K. *Multi-objective optimization using evolutionary algorithms*, vol. 16, John Wiley & Sons, 2001.

[36] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002.

[37] C.A.C. Coello, G.B. Lamont and D.A.van Veldhuizen, *Evolutionary algorithms for solving multi-objective problems*, vol. 5, pp. 79-104, Springer, 2007.

[38] S. Mirjalili, S. Saremi, S.M. Mirjalili and L.D.S. Coelho, "Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization", *Expert Systems with Applications*, vol. 47, pp. 106-119, 2016.

[39] M. Abdel-Basset, R. Mohamed and S. Mirjalili, "A novel Whale Optimization Algorithm integrated with Nelder–Mead simplex for multi-objective optimization problems", *Knowledge-Based Systems* 2020.

Meryem Kusoğlu was born in Karatay, Konya, Turkey, in January 1994. She received the B.S. degree from the Computer Engineering Department, Faculty of Engineering, Bilecik Seyh Edebali University, Bilecik, Turkey, in 2018.

Uğur Yüzgeç was born in Adilcevaz, Bitlis, Turkey, in May 1974. He received the B.S. degree from the Electronics and Communication Engineering Department, Yıldız Technical University, Istanbul, Turkey, in 1995, and the M.S. and Ph.D. degrees from the Electronics and Communication Engineering Department, Kocaeli University, Kocaeli, Turkey, in 1999 and 2005, respectively. From 1998 to 2010, he was a Research Assistant with Electronics and Communication Engineering Department, Kocaeli University. Since August 2020, He has been a Professor with the Computer Engineering Department, Faculty of Engineering, Bilecik Seyh Edebali University, Turkey. His research interest includes intelligent systems and control, fuzzy, neuro-fuzzy systems, meta-heuristic algorithms, unmanned aerial vehicle and numeric techniques in optimization problem.