ARTIFICIAL INTELLIGENCE

COURSE CODE – CACSC11

PROJECT REPORT

SUBMITTED BY : SUBMITTED TO:

SAMEER SETHI(2022UCA1945)
KINSHUK KUMAR(2022UCA1947)
PARISHA KHURANA(2022UCA1891)

DR.ANKUR GUPTA
(ASSISTANT PROFESSOR)

Fire Hawk Optimizer

Abstract
This study proposes the Fire Hawk Optimizer (FHO) as a novel metaheuristic algorithm based on the foraging behavior of whistling kites, black kites and brown falcons. These birds are termed Fire Hawks considering the specific actions they perform to catch prey in nature, specifically by means of setting fire. Utilizing the proposed algorithm, a numerical investigation was conducted on 233 mathematical test functions with dimensions of 2–100, and 150,000 function evaluations were performed for optimization purposes. For comparison, a total of ten different classical and new metaheuristic algorithms were utilized as alternative approaches. The statistical measurements include the best, mean, median, and standard deviation of 100 independent optimization runs, while well-known statistical analyses, such as Kolmogorov–Smirnov, Wilcoxon, Mann–Whitney, Kruskal–Wallis, and Post-Hoc analysis, were also conducted. The obtained results prove that the FHO algorithm exhibits better performance than the compared algorithms from literature. In addition, two of the latest Competitions on Evolutionary Computation (CEC), such as CEC 2020 on bound constraint problems and CEC 2020 on real-world optimization problems including the well-known mechanical engineering design problems, were considered for performance evaluation of the FHO algorithm, which further demonstrated the superior capability of the optimizer over other metaheuristic algorithms in literature. The capability of the FHO is also evaluated in dealing with two of the real-size structural frames with 15 and 24 stories in which the new method outperforms the previously developed metaheuristics.

Introduction

Optimization is the process of decision-making between multiple approaches to achieve the best performance for dealing with a specific system problem. In recent decades, the importance of optimization in performance improvements of different engineering and economic design problems has gained increasing awareness. Specifically, the best decision or solution for a predefined design problem is identified by evaluating different alternative approaches. The predefined measure for the quality of a decision is considered by deter- mining an objective function which is addressed in most of the cases as a performance evaluation index. In other words, optimization concerns the process of selecting the best decision among multiple alternative choices by considering the satisfaction of an objective function. Regarding the rapid progression of various software programs and high-speed parallel processors in the computer science and technology fields, optimization has received heightened attention especially by engineering and economic experts. However, most calculus-based optimization algorithms are incapable of finding the global optimum solutions, which is considered the main deficiency of these algorithms. For instance, gradient- based algorithms require differentiable objective functions that are not achievable in dealing with complex optimization problems. In this regard, the metaheuristic algorithms have been proposed as successful practical methods that provide acceptable accuracy for different optimization purposes.

In this paper, the Fire Hawk Optimizer (FHO) is proposed as a novel metaheuristic algorithm inspired by the foraging behavior of whistling kites, black kites, and brown falcons. These birds catch prey in nature by means of setting fire and, thus, are aptly called Fire Hawks. A numerical investigation was conducted to evaluate the performance of the FHO algorithm by considering 233 mathematical test functions with dimensions of 2–100 and completing 150,000 function evaluations for optimization purposes. For comparative purposes, a total number of 10 different classical and new metaheuristic algorithms were analyzed as alternative approaches. The best, mean, median and standard deviation results of 100 independent optimization runs were obtained for comparison, and Kolmogorov–Smirnov (KS) test, Wilcoxon (W) sign rank test, Mann Whitney (MW) test, Kruskal Wallis (KW) test, and Post Hoc (PH) analyses were conducted accordingly. Moreover, two of the latest Competitions on Evolutionary Computation (CEC), including CEC 2020 on bound constraint problems and CEC 2020 on real-world optimization problems, were utilized for further performance evaluation of the FHO algorithm and comparison to other metaheuristic algorithms in the literature. The capability of the FHO is also evaluated in dealing with two of the real-size structural frames with 15 and 24 stories in which the new method outperforms the previously developed metaheuristics.

This research's novelty can be seen from inspirational and computational points of view. The foraging behavior of fire hawks is utilized for the first time in this paper for developing a novel metaheuristic algorithm. Besides, the complexity level of the utilized test functions is also a different novelty aspect of this paper. It should be noted that the overall performance of numerous algorithms must be evaluated under the same conditions and with similar problems, and under diverse cases, the superiority of each algorithm cannot be proved or disproved by means of different examples and datasets. In this regard, the benchmark test functions of well-known competitions on evolutionary computation should be utilized for having a fair judgment so the capability of FHO as a novel algorithm has been evaluated by utilization of different sets of CEC test problems. This level of complexity in choosing test factions have been utilized for the first time in evaluating novel algorithms. However, the FHO algorithm's advantages include being parameter-free, having quick convergence behaviour, and having the lowest possible objective function evaluations in dealing with different design examples. On the other hand, it cannot produce accurate answers; in other words, the FHO algorithm is an approximation algorithm like other metaheuristic algorithms. Nonetheless, a plethora of metaheuristic algorithms has been proposed through miscellaneous inspirational concepts from nature, which the mathematical models and the specific aspects of the algorithms in their searching groups should be distinct and novel so as to prepare and backup the research from the stable point. PSO, for example, is one of the pioneer algorithms in the metaheuristic area, in which the position updating process by the solution candidates is conducted using the global best and local best of each particle. In stark contrast, in the FHO algorithm, the position updating process is carried out by utilizing the better solution's not the global best, and the mean of the solution candidates, which makes the searching process avoid entrapping in local optimum points. Furthermore,in the Genetic Algorithm (GA), a new solution candidate is created by combining two populations, so there is a possibility of reaching a solution that can be entrapped in the local optimum point. However, in the FHO algorithm, the mean of solution candidates in the specific territory is utilized to avoid entrapment in the local optimum and provide solutions that can finally reach the global optimum.

The main contribution of this paper is as follows:

- Fire Hawks bizarre behaviour spread fire intentionally by carrying burning sticks in their beaks, and talons are examined and analysed to develop a mathematical model.
- A unique nature-inspired FHO algorithm is developed using this model.
- The FHO algorithm's solution updating depends on the preys' new position and safe places under/outside the fire.
- FHO's performance is extensively evaluated against a set of 233 benchmark functions and well-known CEC design examples. It is compared to a plethora of state-of-the-art metaheuristic algorithms.

Fire Hawk Optimizer (FHO)

In this section, the inspiration concept of the FHO alongside the mathematical model of theproposed metaheuristic algorithm are illustrated.

INSPIRATION

Native Australians utilize fire as an effective tool to control and maintain balance of the local ecosystem and landscape, which has been a part of cultural and ethnical traditions formany years. Most of the time, the fires that are started on purpose or may naturally occur due to lightning can be spread by people and other factors, increasing the vulnerability of the native landscape and wildlife. Moreover, whistling kites, black kites, and brown falconsare also responsible for spreading fires across the country—this alternative cause has only been realized recently. These birds, known as Fire Hawks, try to spread fire intentionally bycarrying burning sticks in their beaks and talons, which is reported as a destructive phenomenon in nature. Figure 1 provides images showing the behavior of these birds around fires.

As a mechanism to control and capture their prey, the birds pick up burning sticks and drop them in other unburned places in order to set small fires. These small fires scare the prey, including rodents, snakes, and other animals, and force them flee in a most hasty andnervous way that makes it much easier for the hawks to catch.

Mathematical Model

The FHO metaheuristic algorithm mimics the foraging behavior of fire hawks, consideringthe process of setting and spreading fires and catching prey. At first, a number of solution candidates (X) are determined as the position vectors of the fire hawks and prey. A random initialization process is utilized to identify the initial positions of these vectors in the search space.

$$
X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^j & \cdots & x_1^d \\ x_2^1 & x_2^2 & \cdots & x_2^j & \cdots & x_2^d \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_i^1 & x_i^2 & \cdots & x_i^j & \cdots & x_i^d \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_N^1 & x_N^2 & \cdots & x_N^j & \cdots & x_N^d \end{bmatrix}, \quad \begin{cases} i = 1, 2, \ldots, N. \\ j = 1, 2, \ldots, d. \end{cases}
$$

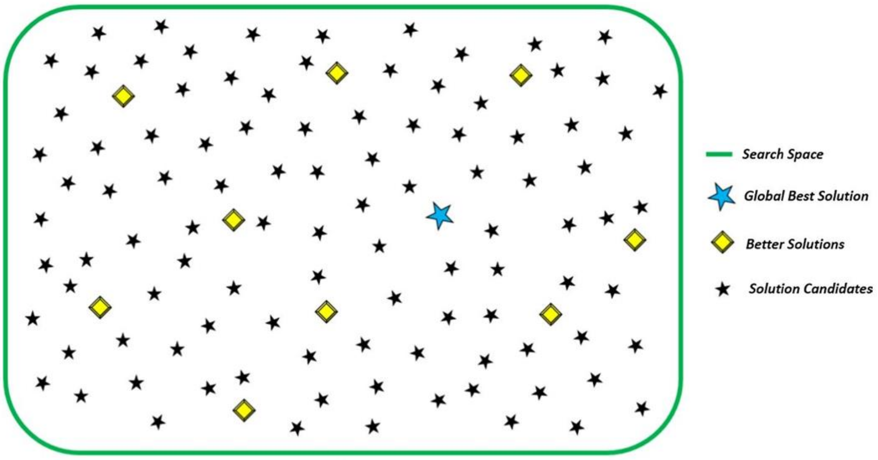
(a) (b) (c)

$$x_i^j(0) = x_{i,min}^j + rand.\left(x_{i,max}^j - x_{i,min}^j\right), \quad \begin{cases} i = 1, 2, \ldots, N. \\ j = 1, 2, \ldots, d. \end{cases} \tag{2}$$

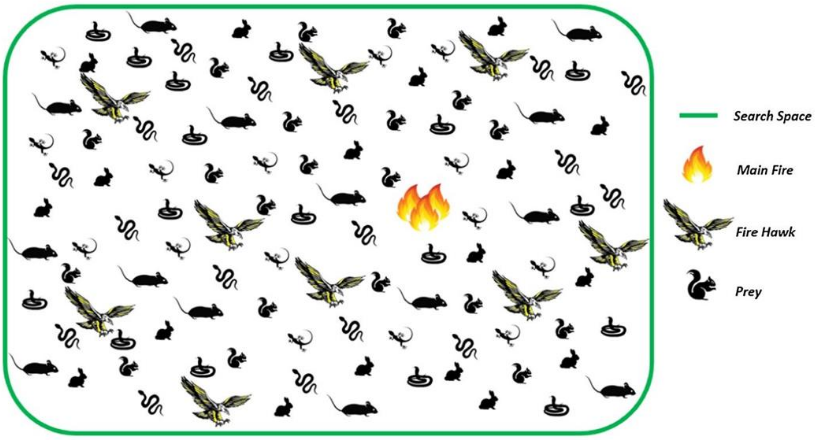
where X_i represents the *ith* solution candidate in the search space;d represents the dimension of the considered problem; N is the total number of solution candidates in the search space; x_i^j is the *jth* decision variable of the *ith* solution candidate; $x_i^j(0)$ represents the initial position of the solution candidates; $x_{i,min}^j$ and $x_{i,max}^j$ are the minimum and maximum bounds of the *jth* decision variable for the *ith* solution candidate; and rand is a uniformly distributed random number in the range of [0,1].

In order to determine the locations the Fire Hawks in the search space, the objective function evaluation for the solution candidates considers the selected optimization problem. Some of the solution candidates with better objective function values are represented as Fire Hawks, while the rest of the solution candidates are the prey. The selected Fire Hawks are utilized for spreading fires around the prey in the search space to make the hunting easier. Besides, the global best solution is assumed to be the main fire that is first utilized by the Fire Hawks to spread fires through the search space (nature). In Fig. 2a, b, the schematic presentation of these aspects is provided, which are mathematically presented as follows:

$$PR = \begin{bmatrix} PR_1 \\ PR_2 \\ \vdots \\ PR_k \\ \vdots \\ PR_m \end{bmatrix}, k = 1, 2, \ldots, m, \tag{3}$$



(a)



(b)

$$FH = \begin{bmatrix} FH_1 \\ FH_2 \\ \vdots \\ FH_l \\ \vdots \\ FH_n \end{bmatrix}, l = 1, 2, \ldots n, \tag{4}$$

where PR_k is the *kth* prey in the search space regarding the total number of m preys; and FH_l is the *lth* fire hawk considering a total number of n fire hawks in the search space.

In the next phase of the algorithm, the total distance between the Fire Hawks and the prey is calculated. As a result, the nearest prey to each bird is determined so that the effective territory of these birds is distinguished. It should be noted that the nearest prey to the first Fire Hawk with the best objective function value is determined, while the territory of the other birds are considered by means of the remaining prey. Figure 3 provides an illustration of this perspective, where D_k^l is determined by means of the following equation:

$$D_k^l = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \quad \begin{matrix} l = 1, 2, \ldots, n. \\ k = 1, 2, \ldots, m. \end{matrix}, \tag{5}$$

where D_k^l is the total distance between the *lth* fire hawk and the *kth* prey; is the total number of prey in the search space; is the total number of fire hawks in the search space; and (x_1, y_1) and (x_2, y_2) represent the coordinates of the Fire Hawks and prey in the search space.

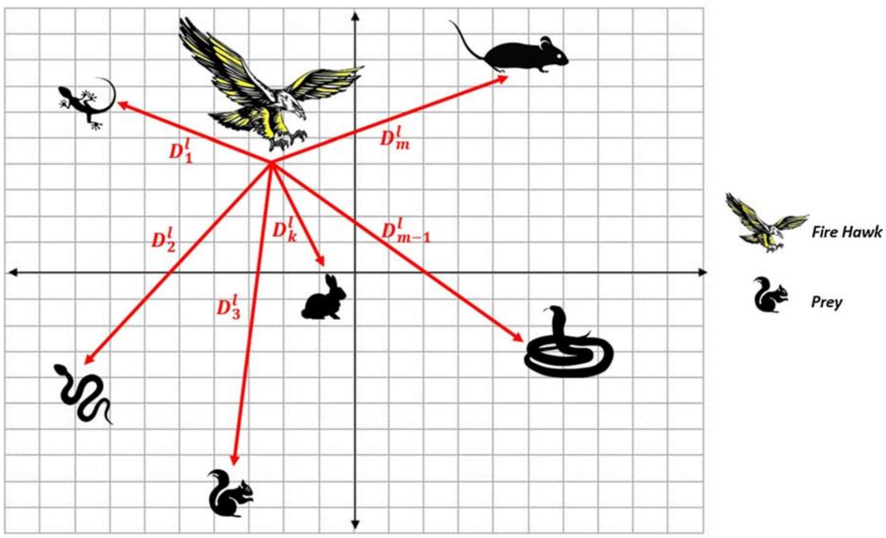


Fig. 3 Schematic presentation for measuring the total distance between the Fire Hawks and the prey

After conducting the mentioned procedure for measuring the total distance between the Fire Hawks and prey, the territory of these birds is distinguished by means of the nearest prey around them. By classifying the Fire Hawks and prey, the searching process of the algorithm is configured. It should be noted that the Fire Hawk with the better objective function value selects the best nearest prey in the search space for its specific territory. Then, the other Fire Hawks accomplish the next nearest prey in the search space, which supports that the strongest Fire Hawks accomplish perform more successful hunting than the weaker birds. In Fig. 4, the schematic presentation of determining Fire Hawks' territory in the search space is provided.

In the next phase of the algorithm, the Fire Hawks collect burning sticks from the main fire in order to set fire in the selected area. In this stage, each bird picks up a burning stick then drops it in its specific territory to force the prey to hastily flee. Meanwhile, some birds are eager to use the burning sticks from other Fire Hawks' territories; therefore, these two behaviors can be utilized as position updating procedures in the main search loop of FHO, as indicated in the following equation:

$$FH_l^{new} = FH_l + r_1 \times GB - r_2 \times FH_{Near} \quad , l = 1, 2, \ldots, n, \tag{6}$$

where FH_l^{new} is the new position vector of the *lth* Fire Hawk (FH_l); GB is the global best solution in the search space considered as the main fire; FH_{Near} is one of the other Fire Hawks in the search space; and r_1 and r_2 are uniformly distributed random numbers in the range of (0, 1) for determining the movements of Fire Hawks toward the main fire and the other Fire Hawks' territories (see Fig. 5).

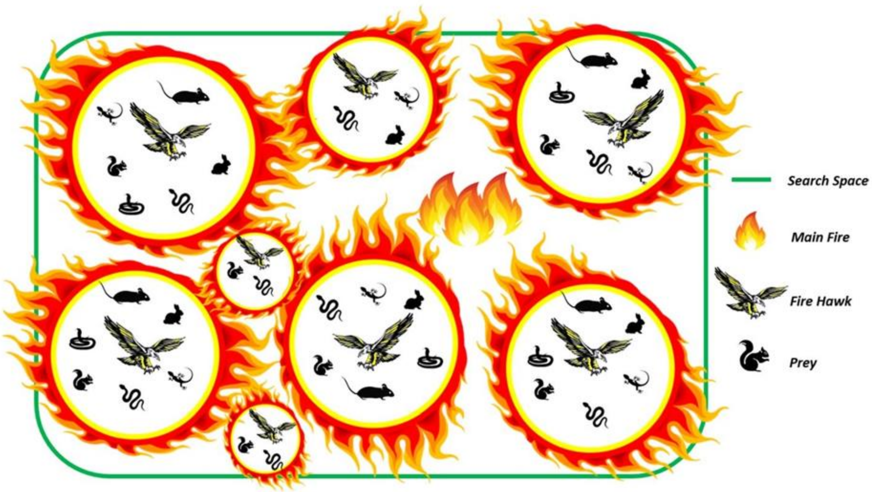


Fig. 4 Schematic presentation of determining Fire Hawks' territory in the search space

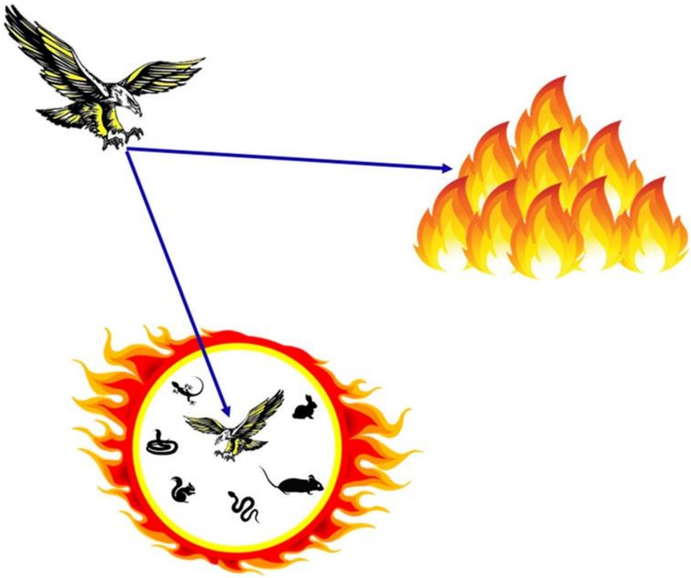


Fig. 5 Schematic presentation of the Fire Hawks' position updating process in the search space

In the next phase of the algorithm, the movement of prey inside the territory of each Fire Hawk is considered a key aspect of animal behavior for the position updating process. When a burning stick is dropped by a Fire Hawk, the prey decide to hide, run away, or will run towards the Fire Hawk by mistake. These actions can be considered in the position updating process by using the following equation:

$$PR_q^{new} = PR_q + r_3 \times FH_l - r_4 \times SP_l, \quad \begin{matrix} l = 1, 2, \dots, n. \\ q = 1, 2, \dots, r. \end{matrix} \tag{7}$$

where PR_q^{new} is the new position vector of the *qth* prey (PR_q) surrounded by the *lth* Fire Hawk (FH_l); GB is the global best solution in the search space considered as the main fire; SP_l is a safe place under the *lth* Fire Hawk territory; and r_3 and r_4 are uniformly distributed random numbers in the range of (0, 1) for determining the movements of prey toward the Fire Hawks and the safe place (see Fig. 6).

Besides, the prey may have movements toward the other Fire Hawks' territory while there is a possibility in which the preys may get closer to the Fire Hawks in the near ambushes or even try to hide in a safer place outside the Fire Hawk's territory in which they are entrapped. These actions can be considered in the position updating process by using the following equation (Fig. 7):

$$PR_q^{new} = PR_q + r_5 \times FH_{Alter} - r_6 \times SP, \quad \begin{matrix} l = 1, 2, \dots, n. \\ q = 1, 2, \dots, r. \end{matrix} \tag{8}$$

It should be noted that the territory of each fire hawk is assumed as a circular area for schematic presentation purposes, so the exact definition of the territory is dependent on the overall distances of the prey and the considered fire hawk. In other words, when prey is positioned in a specific fire hawk's territory, it is assumed to be affected by the considered fire hawk and not the other ones, so the number of the preys and their dis- tances to the considered fire hawk determine the limits of the territory of this fire hawk. Meanwhile, the possibility of the preys being outside their own territory is also considered in the position updating process regarding the fact that the preys should be affected by the fire hawks from other territories. The number of preys in each search loop is the total number of solution candidates minus the number of fire hawks determined randomly through the Brownian motion with a Gaussian distribution as one of the well- known distributions utilized in randomization procedures.

The general aspects of FHO including the boundary violation of solution candidates alongside the termination criterion are also considered in the mathematical model ofthis algorithm. In this regard, a mathematical flag is implemented in the FHO in which a boundary control for violating decision variables is determined while a predefined num- ber of objective function evaluations or iterations can be utilized as termination criteria. In Fig. 8, the pseudo-code of the FHO algorithm is provided, and Fig. 9 presents the flowchart of this algorithm.

```
procedure Fire Hawk Optimizer (FHO)
        Determine initial positions of solution candidates (Xi) in the search space with N candidates
        Evaluate fitness values for initial solution candidates
        Determine the Global Best (GB) solution as the main fire
        while Iteration < Maximum number of iterations
            Generate n as a random integer number for determining the number of Fire Hawks
            Determine Fire Hawks (FH) and Preys (PR) in the search space
            Calculate the total distance between the Fire Hawks and the preys
            Determine the territory of the Fire Hawks by dispersing the preys
            for l=1: n
                Determine the new position of the Fire Hawks by Eq. 6.
                for q=1:r
                    Calculate the safe place under lth Fire Hawk territory by Eq. 9.
                    Determine the new position of the preys by Eq. 7.
                    Calculate the safe place outside the lth Fire Hawk territory by Eq. 10.
                    Determine the new position of the preys by Eq. 8.
                end
            end
            Evaluate fitness values for the newly created Fire Hawks and preys
            Determine the Global Best (GB) solution as the main fire
        end while
        return GB
end procedure
```

Fig. 8 Pseudo-code of FHO

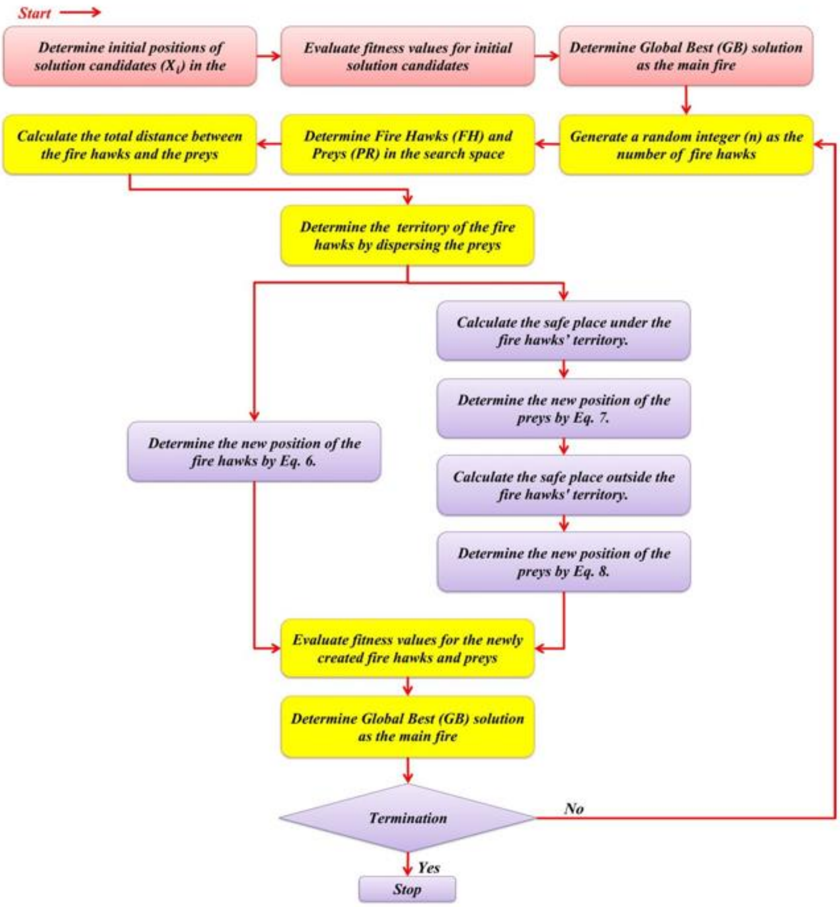


Fig. 9 Flowchart of FHO

Proposed Optimizations

1. FHO_2best.m

The provided code (FHO_2best.m) differs from the original FHO.m (shown previously) by incorporating a concept called "Two-Best Selection" in the Fire Hawk Optimizer (FHO) algorithm. Here's a breakdown of the key differences:

1. Initialization:

- Both codes initialize the population with random solutions and sort them based on their objective function values.
- FHO.m identifies the best solution as `BestPop` and separates solutions into Fire Hawks (`FHPops`) and prey (`Pop2`).
- FHO_2best.m identifies the top two best solutions: `BestPop1` (best) and `BestPop2` (second-best). It then proceeds similarly with Fire Hawks and prey.

2. Main Loop:

- Both codes iterate through generations, update Fire Hawk positions based on the "global best" and other Fire Hawks, update prey positions based on Fire Hawks and a "safe place," and evaluate new solution costs.
- The main difference lies in how the Fire Hawk position update is calculated:
    - FHO.m uses a single random weighting between the "global best" and another Fire Hawk.
    - FHO_2best.m uses a combination of three random weightings. The first two weightings involve both `GB1` (cost of `BestPop1`) and `GB2` (cost of `BestPop2`), while the third considers another Fire Hawk. This allows the Fire Hawks to be attracted to both the best and second-best solutions potentially leading to faster convergence or finding diverse solutions.

3. Update Bests:

- Both codes update the global best solution (GB) if a better solution is found during the iteration.
- FHO_2best.m maintains two separate "best" values: `GB1` is updated with the minimum of the current `GB1` and the best cost found in the iteration. Similarly, `GB2` is updated with the minimum of the current `GB2` and the second-best cost found.

Overall, FHO_2best.m modifies the FHO algorithm by:

- Considering the two best solutions instead of just the global best to potentially guide Fire Hawks towards promising areas of the search space.
- Maintaining two "best" values throughout the optimization process, potentially leading to a diverse set of good solutions.

Note: The effectiveness of this modification depends on the specific problem being optimized. You might need to experiment to see if FHO_2best.m performs better than the original FHO.m for your case.

2.FHO_dist_upgrade.m

The provided code (FHO_dist_upgrade.m) differs from the original FHO.m (shown previously) by incorporating a modification related to how Fire Hawks interact with each other. Here's a breakdown of the key differences:
1. Fire Hawk Interaction:

- Both codes use Fire Hawks to explore the search space and guide prey towards potentially better solutions.
- FHO.m randomly selects another Fire Hawk to influence a Fire Hawk's position update.
- FHO_dist_upgrade.m defines a new function find_closest_hawk that calculates the distances between the current Fire Hawk and all other Fire Hawks. It then excludes the current Fire Hawk itself (setting its distance to infinity) and identifies the closest Fire Hawk. This closest Fire Hawk's position is then used in the update calculation for the current Fire Hawk.

2. Update Equation:

- Both codes use a random weighting and the "global best" (GB) solution to update Fire Hawk positions.
- FHO.m uses a single random weighting and the GB solution.
- FHO_dist_upgrade.m uses a single random weighting and the position of the closest Fire Hawk instead of another randomly chosen one. This potentially creates a more localized search around promising areas identified by multiple Fire Hawks converging in the same region.

Overall, FHO_dist_upgrade.m modifies the FHO algorithm by:

- Emphasizing the interaction between Fire Hawks. Fire Hawks are influenced by the position of the closest Fire Hawk rather than a randomly chosen one.
- Potentially leading to a more focused exploration around areas where multiple Fire Hawks have converged, potentially accelerating convergence or finding local optima.

Note: The effectiveness of this modification depends on the specific problem being optimized. You might need to experiment to see if FHO_dist_upgrade.m performs better than the original FHO.m for your case.

3.FHO_nearGB.m

The provided code (FHO_nearGB.m) incorporates several modifications compared to the original FHO.m:
1. Fire Hawk Interaction:

- Similar to FHO_dist_upgrade.m, FHO_nearGB.m uses the find_closest_hawk function to identify the closest Fire Hawk to the current Fire Hawk.

2. Update Equation:

- FHO.m and FHO_dist_upgrade.m use a single random weighting and another Fire Hawk's position to update a Fire Hawk's position.

- FHO_nearGB.m uses three random weightings. The first two weightings involve the "global bests" GB1 (cost of BestPop1) and GB2 (cost of BestPop2), while the third considers the position of the closest Fire Hawk. This potentially allows Fire Hawks to be attracted to a combination of the two best solutions and explore promising areas identified by nearby Fire Hawks.

3. Re-initialization:

- After a quarter of the maximum function evaluations (MaxFes), the population is reinitialized with a focus on the two current "global bests".
    - It keeps the top two best solutions (BestPop1 and BestPop2) from the previous generation.
    - It generates a new population around these "global bests" using a normal distribution with a standard deviation (std_deviation). This injects diversity and allows for potential exploration of new regions in the search space.

4. Random Walk (Commented Out):

- The code includes a commented-out section for a random walk procedure. If activated (by removing the comments), it would apply a random walk with a decreasing step size (par) to the first "global best" solution (BestPop1) after half of the maximum function evaluations (MaxFes). This could help the algorithm escape local optima but might require tuning the parameters for effectiveness.

Overall, FHO_nearGB.m modifies the FHO algorithm by:

- Emphasizing the interaction between Fire Hawks by considering the closest Fire Hawk's position in the update equation.
- Utilizing a combination of the two "global bests" to guide Fire Hawks.
- Introducing a re-initialization step focused on the "global bests" to potentially improve exploration and avoid premature convergence.
- Including an optional random walk procedure (commented out) for potentially escaping local optima.

Note: The effectiveness of these modifications depends on the specific problem being optimized. You might need to experiment with different parameter settings and compare the performance with the original FHO.m.

4.FHO_upgrade.m

The provided code (FHO_upgrade.m) incorporates several modifications compared to the original FHO.m:
1. Update Equation:

- Similar to FHO_nearGB.m, FHO_upgrade.m uses three random weightings. The first two weightings involve the "global bests" GB1 (cost of BestPop1) and GB2 (cost of BestPop2), while the third considers the position of the closest Fire Hawk (FHnear). This allows Fire Hawks to be attracted to a combination of the two best solutions and explore promising areas identified by nearby Fire Hawks.

2. No Re-initialization or Random Walk:

- Unlike FHO_nearGB.m, FHO_upgrade.m doesn't include a re-initialization step or a random walk procedure. It relies solely on the update equation with a combination of the two "global bests" and the closest Fire Hawk for exploration and exploitation.

Overall, FHO_upgrade.m modifies the FHO algorithm by:

- Emphasizing the interaction between Fire Hawks by considering the closest Fire Hawk's position in the update equation, similar to FHO_dist_upgrade.m and FHO_nearGB.m.
- Utilizing a combination of the two "global bests" to guide Fire Hawks, similar to FHO_nearGB.m.
- Not including a re-initialization step or a random walk procedure, potentially making the exploration and exploitation process rely more heavily on the initial population and the chosen update equation.

Note: The effectiveness of these modifications depends on the specific problem being optimized. You might need to experiment with different parameter settings and compare the performance with the original FHO.m and other variants like FHO_nearGB.m.

Algorithms, test problems and comparison criteria

This section presents the design of experiments conducted to evaluate the performance of MGOA. Thirty(30) CEC 2014 test functions,thirty(30) CEC 2017 test functions, and five (5) se- lected optimization design problems in the engineering domain were used to evaluate the optimized FHOs. The results obtained were compared with that of the following algorithms:-

1. Gazelle Optimization Algorithm (GOA)
2. Sand Cat Optimization Algorithm (SCSO)
3. Arithmetic Otimization Algorithm (AOA)
4. Fire Hawk Optimizer (FHO)
5. Whale Optimization Algorithm (WOA)
6. Grey Wolf Optimizer (GWO)
7. Crayfish Optimization Algorithm (COA)

The population size and the maximum number of iterations are sensitive parameters and need to be tuned. For this study, the population size is tuned to 30, and the maximum number of iterations is tuned to 1000.The stop criterium is the maximum number of iterations. The number of independent runs for each algorithm is set at 50.

3.2. CEC 2014 AND CEC 2017 Benchmark functions

Figure 1: CEC2017 BENCHMARK FUNCTIONS

Typology	No.	Function name	Opt.
Unimodal Functions	1	Shifted and Rotated Bent Cigar	100
	2	Shifted and Rotated Sum of Different Power	200
	3	Shifted and Rotated Zakharov	300
Simple Multimodal Functions	4	Shifted and Rotated Rosenbrock	400
	5	Shifted and Rotated Rastrigin	500
	6	Shifted and Rotated Expanded Schaffer F6	600
	7	Shifted and Rotated Lunacek Bi-Rastrigin	700
	8	Shifted and Rotated Non-Continuous Rastrigin	800
	9	Shifted and Rotated Levy	900
	10	Shifted and Rotated Schwefel	1000
Hybrid Functions	11	Zakharov; Rosenbrock; Rastrigin	1100
	12	High-conditioned Elliptic; Modified Schwefel; Bent Cigar	1200
	13	Bent Cigar; Rosenbrock; Lunacek bi-Rastrigin	1300
	14	High-conditioned Elliptic; Ackley; Schaffer F7; Rastrigin	1400
	15	Bent Cigar; HGBat; Rastrigin; Rosenbrock	1500
	16	Expanded Schaffer F6; HGBat; Rosenbrock; Modified Schwefel	1600
	17	Katsuura; Ackley; Expanded Griewank plus Rosenbrock; Schwefel; Rastrigin	1700
	18	High-conditioned Elliptic; Ackley; Rastrigin; HGBat; Discus	1800
	19	Bent Cigar; Rastrigin; Griewank plus Rosenbrock; Weierstrass; Expanded Schaffer F6	1900
	20	HappyCat; Katsuura; Ackley; Rastrigin; Modified Schwefel; Schaffer F7	2000
Composition Functions	21	Rosenbrock; High-conditioned Elliptic; Rastrigin	2100
	22	Rastrigin; Griewank; Modified Schwefel	2200
	23	Rosenbrock; Ackley; Modified Schwefel; Rastrigin	2300
	24	Ackley; High-conditioned Elliptic; Griewank; Rastrigin	2400
	25	Rastrigin; HappyCat; Ackley; Discus; Rosenbrock	2500
	26	Expanded Schaffer F6; Modified Schwefel; Griewank; Rosenbrock; Rastrigin	2600
	27	HGBat; Rastrigin; Modified Schwefel; Bent Cigar; High-conditioned Elliptic; Expanded Schaffer F6	2700
	28	Ackley; Griewank; Discus; Rosenbrock; HappyCat; Expanded Schaffer F6	2800
	29	f_{15}; f_{16}; f_{17}	2900
	30	f_{15}; f_{18}; f_{19}	3000

No.	Functions	$f(x^*)$	NO.	Functions	$f(x^*)$
F01	Rotated High Conditioned Elliptic Function	100	F16	Shifted and Rotated Expanded Scaffer's F6 Function	1600
F02	Rotated Bent Cigar Function	200	F17	Hybrid Function 2	1700
F03	Rotated Discus Function	300	F18	Hybrid Function 2	1800
F04	Shifted and Rotated Rosenbrock's Function	400	F19	Hybrid Function 3	1900
F05	Shifted and Rotated Ackley's Function	500	F20	Hybrid Function 4	2000
F06	hifted and Rotated Weierstrass Function	600	F21	Hybrid Function 5	2100
F07	Shifted and Rotated Griewank's Function	700	F22	Hybrid Function 6	2200
F08	Shifted Rastrigin's Function	800	F23	Composition Function 1	2300
F09	Shifted and Rotated Rastrigin's Function	900	F24	Composition Function 2	2400
F10	Shifted Schwefel's Function	1000	F25	Composition Function 3	2500
F11	Shifted and Rotated Schwefel's Function	1100	F26	Composition Function 4	2600
F12	Shifted and Rotated Katsuura Function	1200	F27	Composition Function 5	2700
F13	Shifted and Rotated HappyCat Function	1300	F28	Composition Function 6	2800
F14	Shifted and Rotated HGBat Function	1400	F29	Composition Function 7	2900
F15	Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function	1500	F30	Composition Function 8	3000

Search Range: $[-100, 100]^D$

Figure 2: CEC2014 BENCHMARK FUNCTIONS

ENGINEERING PROBLEMS

After modifying the Fire Hawk optimizer and creating four variants, we conducted a comprehensive comparative study to evaluate their performance against the base Fire Hawk algorithm and several other recent optimization algorithms. Our evaluation was based on the CEC2014 and CEC2017 benchmark functions, as well as a selection of engineering design problems.

Applying the Wilcoxon rank-sum test to compare the algorithms, we found that FHO_upgrade, one of our modified algorithms, consistently outperformed the base Fire Hawk algorithm and the other optimization algorithms across the different test cases. Specifically, FHO_upgrade exhibited the best overall performance, demonstrating its effectiveness in optimizing solutions for a variety of optimization problems.

The success of FHO_upgrade can be attributed to the specific modifications we made to the algorithm. These modifications were aimed at improving the algorithm's exploration and exploitation capabilities, as well as its ability to handle various types of optimization problems. By enhancing these key aspects, FHO_upgrade was able to consistently produce superior results compared to the other algorithms tested.

Our study underscores the importance of algorithm modification and customization in optimization research. By tailoring the Fire Hawk optimizer to better suit the specific characteristics of the optimization problems at hand, we were able to achieve significant improvements in performance. This highlights the potential for further refinement and enhancement of optimization algorithms to address specific challenges in optimization tasks across various domains.

In conclusion, our work demonstrates the effectiveness of modifying the Fire Hawk optimizer to create variants that outperform the original algorithm and other state-of-the-art optimization algorithms. The success of FHO_upgrade in particular suggests that our modifications have the potential to significantly advance the field of optimization and contribute to the development of more efficient and effective optimization algorithms.

FUNCTIONS	FHO	FHO_upgrade
Speed Reducer	2994.526138	3000.176735
Tension/compression spring design	0.012665273	0.01267824294
Pressure vessel design	6090.526721	6060.708109
Three-bar truss design problem	263.8958436	263.8958557
Design of gear train	2.70E-12	2.31E-11
Cantilever beam	1.339957196	1.341884679
Minimize I-beam vertical deflection	0.01307712476	0.01307427522
Tubular column design	26.48636326	26.48658734
Piston lever	8.412709372	8.414358482
Corrugated bulkhead design	2.15E-161	2.18E-161
Car side impact design	22.85731585	22.85147679
Design of welded beam design	1.724875142	1.72863814
Reinforced concrete beam design	359.2086	359.2080216

Experimental Result and Discussions

CEC2014 RESULTS

3.1. Performance Comparison of FHO and FHO_dist_upgrade on CEC2014

Table 1: CEC 2014 Results

Function	FHO Avg	FHO dist upg Avg	P VALUE	FHO Std dev	FHO dist upg Std dev
F1	6.11E+07	6.00E+07	0.4733	3.61E+07	3.55E+07
F2	4.77E+09	4.91E+09	0.0406	2.90E+09	3.29E+09
F3	7.49E+04	8.51E+04	0.1373	3.80E+04	2.66E+04
F4	7.65E+02	7.44E+02	0.9117	1.73E+02	1.84E+02
F5	5.21E+02	5.21E+02	0.1715	1.38E-01	1.28E-01
F6	6.11E+02	6.11E+02	0.5011	1.02E+00	8.69E-01
F7	7.54E+02	7.56E+02	0.9823	2.54E+01	2.67E+01
F8	8.69E+02	8.73E+02	0.5395	1.15E+01	8.74E+00
F9	9.71E+02	9.76E+02	0.2458	1.05E+01	1.08E+01
F10	2.88E+03	2.90E+03	0.652	2.45E+02	2.13E+02
F11	3.28E+03	3.30E+03	0.4464	2.26E+02	2.06E+02
F12	1.20E+03	1.20E+03	0.1761	6.17E-01	5.89E-01
F13	1.30E+03	1.30E+03	0.7394	8.04E-01	6.20E-01
F14	1.41E+03	1.41E+03	0.4733	7.28E+00	7.91E+00
F15	2.70E+05	8.50E+04	0.4918	1.05E+06	1.53E+05
F16	1.60E+03	1.60E+03	0.8534	1.66E-01	1.67E-01
F17	1.26E+06	8.64E+05	0.4825	3.65E+06	1.26E+06
F18	8.59E+06	5.70E+06	0.61	9.53E+06	7.29E+06
F19	1.93E+03	1.93E+03	0.1453	2.01E+01	2.72E+01
F20	1.31E+05	1.28E+05	0.3478	2.63E+05	1.56E+05
F21	4.38E+04	5.73E+04	0.5493	3.21E+04	5.32E+04
F22	2.48E+03	2.49E+03	0.1669	1.12E+02	1.10E+02
F23	2.80E+03	2.81E+03	0.7506	4.48E+01	4.89E+01
F24	2.59E+03	2.59E+03	0.1858	1.64E+01	1.72E+01
F25	2.71E+03	2.71E+03	0.1907	7.64E+00	5.28E+00
F26	2.70E+03	2.71E+03	0.5106	1.41E+01	2.12E+01
F27	3.22E+03	3.23E+03	0.3366	4.77E+01	4.81E+01
F28	3.26E+03	3.25E+03	0.007	9.95E+01	9.56E+01
F29	1.25E+06	1.10E+06	0.6309	1.23E+06	1.11E+06
F30	1.92E+04	1.41E+04		4.41E+04	1.72E+04

3.2 Performance Comparison of FHO and FHO_2best on CEC2014

Function	FHO Avg	FHO 2 best Avg	P VALUE	FHO Std dev	FHO 2 best Std dev
F1	6.11E+07	5.38E+07	0.0271	3.61E+07	2.96E+07
F2	4.77E+09	4.65E+09	0	2.90E+09	2.68E+09
F3	7.49E+04	8.02E+04	0.004	3.80E+04	4.02E+04
F4	7.65E+02	7.46E+02	0.1907	1.73E+02	1.93E+02
F5	5.21E+02	5.21E+02	0.234	1.38E-01	1.41E-01
F6	6.11E+02	6.11E+02	6.53E-08	1.02E+00	9.98E-01
F7	7.54E+02	7.53E+02	0.4825	2.54E+01	2.91E+01
F8	8.69E+02	8.72E+02	6.74E-06	1.15E+01	1.03E+01
F9	9.71E+02	9.71E+02	0.0364	1.05E+01	1.09E+01
F10	2.88E+03	2.85E+03	4.35E-05	2.45E+02	1.87E+02
F11	3.28E+03	3.30E+03	0.0026	2.26E+02	1.93E+02
F12	1.20E+03	1.20E+03	0.429	6.17E-01	6.15E-01
F13	1.30E+03	1.30E+03	0.3632	8.04E-01	7.77E-01
F14	1.41E+03	1.42E+03	0.1809	7.28E+00	8.21E+00
F15	2.70E+05	1.08E+05	0.0451	1.05E+06	1.79E+05
F16	1.60E+03	1.60E+03	1.96E-10	1.66E-01	1.54E-01
F17	1.26E+06	6.40E+05	4.71E-04	3.65E+06	3.90E+05
F18	8.59E+06	6.73E+06	0.3329	9.53E+06	8.17E+06
F19	1.93E+03	1.94E+03	0.947	2.01E+01	2.82E+01
F20	1.31E+05	1.04E+05	1.25E-04	2.63E+05	1.18E+05
F21	4.38E+04	7.00E+04	0.6204	3.21E+04	6.09E+04
F22	2.48E+03	2.48E+03	0.3042	1.12E+02	1.08E+02
F23	2.80E+03	2.81E+03	7.09E-08	4.48E+01	5.12E+01
F24	2.59E+03	2.59E+03	1.34E-05	1.64E+01	1.65E+01
F25	2.71E+03	2.71E+03	0.0905	7.64E+00	8.18E+00
F26	2.70E+03	2.70E+03	0.3555	1.41E+01	1.50E+01
F27	3.22E+03	3.22E+03	1.85E-08	4.77E+01	5.22E+01
F28	3.26E+03	3.27E+03	2.00E-05	9.95E+01	9.31E+01
F29	1.25E+06	1.31E+06	2.39E-04	1.23E+06	1.17E+06
F30	1.92E+04	1.66E+04	7.74E-06	4.41E+04	2.50E+04

3.3 Performance Comparison of FHO and FHO_nearGB on CEC2014

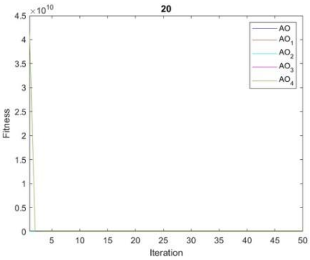
Function	FHO Avg	FHO nearGB Avg	P VALUE	FHO Std dev	FHO nearGB Std dev
F1	6.11E+07	5.09E+07	0.7283	3.61E+07	2.86E+07
F2	4.77E+09	4.76E+09	0	2.90E+09	2.68E+09
F3	7.49E+04	6.97E+04	4.42E-06	3.80E+04	3.45E+04
F4	7.65E+02	8.35E+02	0.5011	1.73E+02	3.39E+02
F5	5.21E+02	5.21E+02	0.1624	1.38E-01	1.37E-01
F6	6.11E+02	6.11E+02	1.49E-06	1.02E+00	9.32E-01
F7	7.54E+02	7.52E+02	0.1373	2.54E+01	2.69E+01
F8	8.69E+02	8.75E+02	1.01E-08	1.15E+01	1.24E+01
F9	9.71E+02	9.80E+02	3.82E-09	1.05E+01	1.02E+01
F10	2.88E+03	2.86E+03	1.49E-06	2.45E+02	2.37E+02
F11	3.28E+03	3.27E+03	0.0073	2.26E+02	2.27E+02
F12	1.20E+03	1.20E+03	0.3403	6.17E-01	5.16E-01
F13	1.30E+03	1.30E+03	0.3711	8.04E-01	6.89E-01
F14	1.41E+03	1.41E+03	0.9823	7.28E+00	7.09E+00
F15	2.70E+05	8.20E+04	0.8883	1.05E+06	9.42E+04
F16	1.60E+03	1.60E+03	4.80E-07	1.66E-01	1.51E-01
F17	1.26E+06	8.00E+05	0.0537	3.65E+06	8.33E+05
F18	8.59E+06	6.03E+06	0.4376	9.53E+06	5.87E+06
F19	1.93E+03	1.93E+03	0.9	2.01E+01	2.24E+01
F20	1.31E+05	1.40E+05	0.7506	2.63E+05	1.89E+05
F21	4.38E+04	5.86E+04	0.7618	3.21E+04	4.91E+04
F22	2.48E+03	2.47E+03	0.6952	1.12E+02	1.15E+02
F23	2.80E+03	2.82E+03	4.61E-10	4.48E+01	6.56E+01
F24	2.59E+03	2.60E+03	5.19E-07	1.64E+01	1.42E+01
F25	2.71E+03	2.71E+03	0.5106	7.64E+00	5.26E+00
F26	2.70E+03	2.70E+03	0.6204	1.41E+01	1.61E+01
F27	3.22E+03	3.23E+03	7.22E-06	4.77E+01	6.49E+01
F28	3.26E+03	3.25E+03	0.0351	9.95E+01	9.01E+01
F29	1.25E+06	7.93E+05	9.02E-04	1.23E+06	1.04E+06
F30	1.92E+04	1.35E+04	6.35E-05	4.41E+04	2.82E+04

3.4 Performance Comparison of FHO and FHO_upgrade on CEC2014

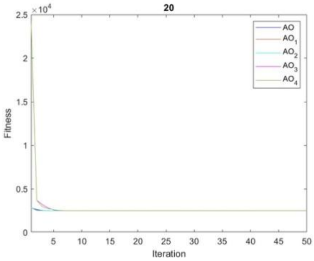
Function	FHO Avg	FHO upgrade Avg	P VALUE	FHO Std dev	FHO upgrade Std dev
F1	6.11E+07	5.57E+07	0.0798	3.61E+07	2.80E+07
F2	4.77E+09	4.99E+09	0	2.90E+09	3.08E+09
F3	7.49E+04	7.67E+04	0.0013	3.80E+04	3.82E+04
F4	7.65E+02	8.53E+02	0.7845	1.73E+02	2.67E+02
F5	5.21E+02	5.21E+02	0.015	1.38E-01	9.95E-02
F6	6.11E+02	6.11E+02	2.03E-07	1.02E+00	1.18E+00
F7	7.54E+02	7.54E+02	0.1714	2.54E+01	3.01E+01
F8	8.69E+02	8.71E+02	0.0261	1.15E+01	9.98E+00
F9	9.71E+02	9.73E+02	9.51E-06	1.05E+01	1.31E+01
F10	2.88E+03	2.88E+03	3.01E-04	2.45E+02	2.28E+02
F11	3.28E+03	3.27E+03	7.66E-05	2.26E+02	2.35E+02
F12	1.20E+03	1.20E+03	0.0087	6.17E-01	4.69E-01
F13	1.30E+03	1.30E+03	0.8418	8.04E-01	6.70E-01
F14	1.41E+03	1.41E+03	0.6414	7.28E+00	7.20E+00
F15	2.70E+05	7.67E+04	3.37E-04	1.05E+06	1.71E+05
F16	1.60E+03	1.60E+03	4.20E-10	1.66E-01	1.84E-01
F17	1.26E+06	7.23E+05	0.5493	3.65E+06	5.29E+05
F18	8.59E+06	6.24E+06	0.8418	9.53E+06	7.08E+06
F19	1.93E+03	1.93E+03	0.0905	2.01E+01	1.85E+01
F20	1.31E+05	9.93E+04	0.0156	2.63E+05	1.07E+05
F21	4.38E+04	8.01E+04	0.0303	3.21E+04	1.74E+05
F22	2.48E+03	2.48E+03	0.9587	1.12E+02	1.05E+02
F23	2.80E+03	2.82E+03	8.66E-05	4.48E+01	5.67E+01
F24	2.59E+03	2.59E+03	0.0378	1.64E+01	1.59E+01
F25	2.71E+03	2.71E+03	0.085	7.64E+00	8.50E+00
F26	2.70E+03	2.71E+03	0.0657	1.41E+01	2.88E+01
F27	3.22E+03	3.22E+03	5.09E-06	4.77E+01	4.92E+01
F28	3.26E+03	3.27E+03	0.0095	9.95E+01	8.78E+01
F29	1.25E+06	1.21E+06	3.83E-05	1.23E+06	1.21E+06
F30	1.92E+04	2.28E+04	1.99E-06	4.41E+04	4.07E+04

CONVERGENCE CURVE FOR 2014

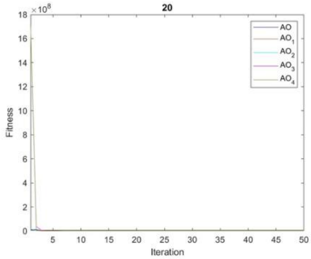
F1



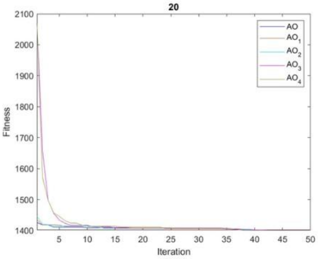
F2



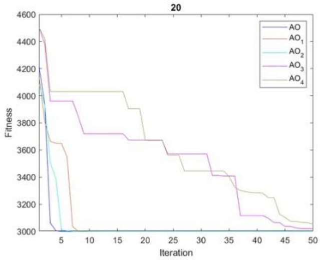
F3



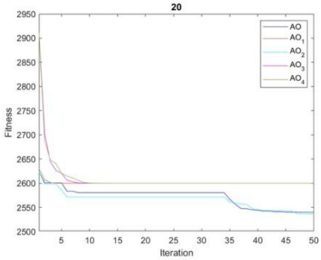
F4



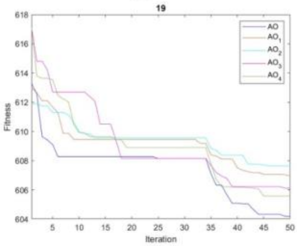
F5



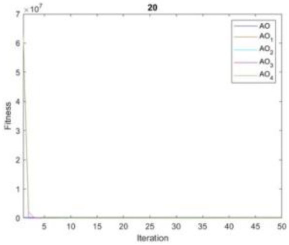
F6



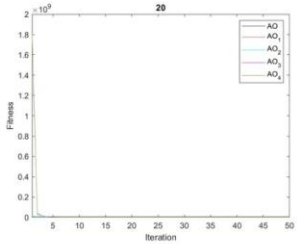
F7



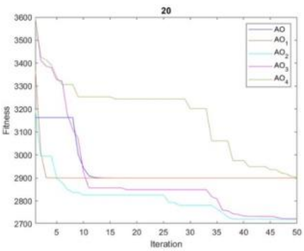
F8



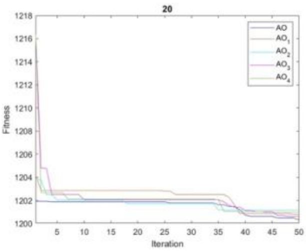
F9



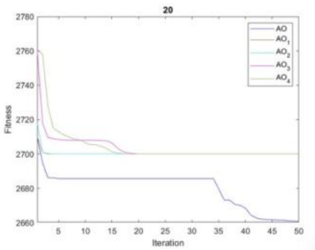
F10



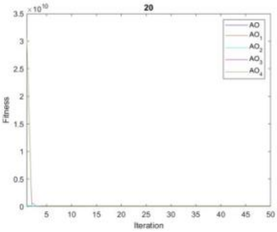
F11



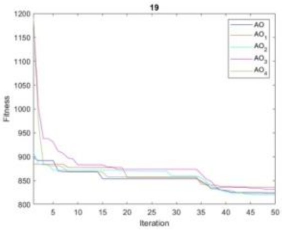
F12



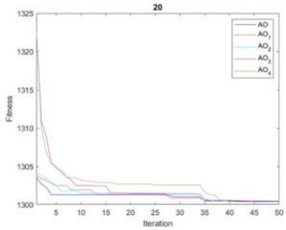
F13



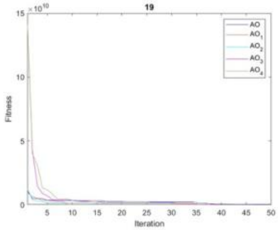
F14



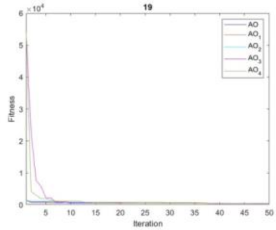
F15



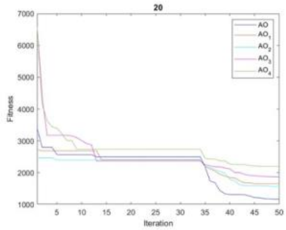
F16



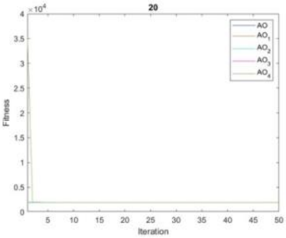
F17



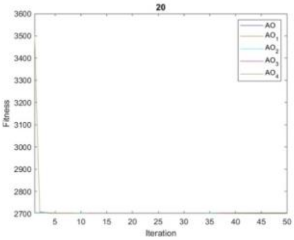
F18



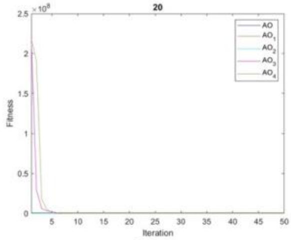
F19



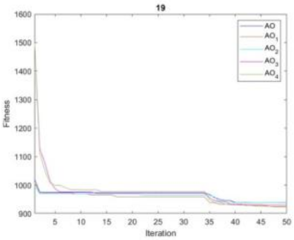
F20



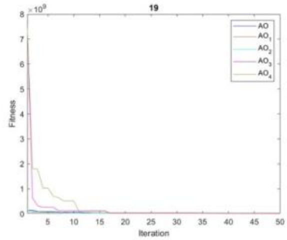
F21



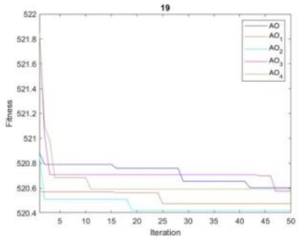
F22



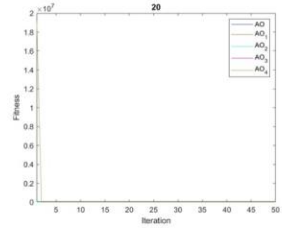
F23



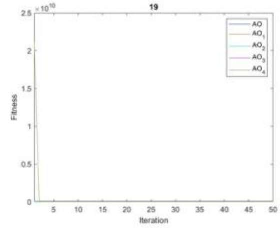
F24



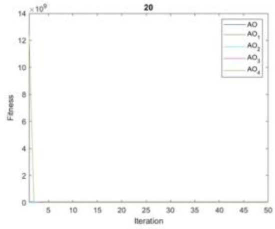
F25



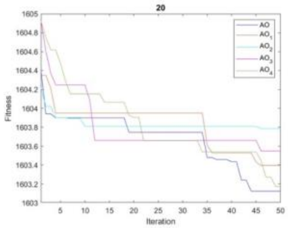
F26



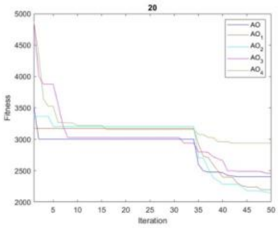
F27



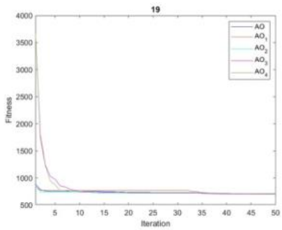
F28



F29



F30



CEC2017 RESULTS

4.1 Performance Comparison of FHO and FHO_dist_upgrade on CEC2017

Table 1: CEC 2014 Results

Function	FHO Avg	FHO dist upg Avg	P VALUE	FHO Std dev	FHO dist upg Std dev
F1	1.62E+11	1.64E+11	0.4733	2.09E+10	2.27E+10
F2	0.00E+00	0.00E+00	0.0406	0.00E+00	0.00E+00
F3	4.05E+05	4.38E+05	0.1373	1.06E+05	1.22E+05
F4	4.68E+04	5.12E+04	0.9117	9.61E+03	1.19E+04
F5	1.30E+03	1.31E+03	0.1715	5.78E+01	6.71E+01
F6	7.01E+02	7.02E+02	0.5011	8.42E+00	8.75E+00
F7	3.03E+03	3.09E+03	0.9823	2.59E+02	2.43E+02
F8	1.73E+03	1.73E+03	0.5395	7.22E+01	7.90E+01
F9	4.54E+04	4.66E+04	0.2458	6.16E+03	8.28E+03
F10	1.67E+04	1.67E+04	0.652	4.31E+02	4.98E+02
F11	1.56E+05	1.55E+05	0.4464	5.54E+04	6.06E+04
F12	3.64E+10	3.74E+10	0.1761	1.10E+10	1.05E+10
F13	1.69E+10	1.65E+10	0.7394	7.05E+09	5.87E+09
F14	5.18E+07	4.94E+07	0.4733	2.62E+07	2.16E+07
F15	5.82E+09	5.74E+09	0.4918	2.86E+09	2.82E+09
F16	8.48E+03	8.58E+03	0.8534	9.38E+02	8.57E+02
F17	7.14E+03	7.03E+03	0.4825	3.25E+03	2.41E+03
F18	9.84E+07	1.04E+08	0.61	4.82E+07	5.69E+07
F19	2.59E+09	2.68E+09	0.1453	2.86E+09	1.29E+09
F20	4.60E+03	4.61E+03	0.3478	2.30E+02	2.02E+02
F21	2.25E+03	2.26E+03	0.5493	2.42E+00	2.85E+00
F22	2.35E+03	2.35E+03	0.1669	3.03E+00	2.98E+00
F23	6.05E+03	5.99E+03	0.7506	1.37E+03	1.20E+03
F24	4.76E+03	4.88E+03	0.1858	8.47E+02	8.47E+02
F25	1.39E+04	1.40E+04	0.1907	3.99E+03	3.53E+03
F26	1.06E+04	1.10E+04	0.5106	2.77E+03	2.69E+03
F27	6.78E+03	7.06E+03	0.3366	9.63E+02	1.04E+03
F28	1.31E+04	1.39E+04	0.007	1.61E+03	2.16E+03
F29	1.10E+04	1.12E+04	0.6309	1.76E+03	1.85E+03
F30	1.74E+09	1.67E+09		8.79E+08	8.43E+08

4.2 Performance Comparison of FHO and FHO_2best on CEC2017

Function	FHO Avg	FHO 2 best Avg	P VALUE	FHO Std dev	FHO 2 best Std dev
F1	1.62E+11	1.62E+11	0.0271	2.09E+10	2.03E+10
F2	0.00E+00	0.00E+00	0	0.00E+00	0.00E+00
F3	4.05E+05	4.03E+05	0.004	1.06E+05	1.08E+05
F4	4.68E+04	5.28E+04	0.1907	9.61E+03	1.09E+04
F5	1.30E+03	1.30E+03	0.234	5.78E+01	6.31E+01
F6	7.01E+02	6.99E+02	6.53E-08	8.42E+00	8.30E+00
F7	3.03E+03	3.07E+03	0.4825	2.59E+02	3.18E+02
F8	1.73E+03	1.73E+03	6.74E-06	7.22E+01	7.34E+01
F9	4.54E+04	4.58E+04	0.0364	6.16E+03	7.96E+03
F10	1.67E+04	1.65E+04	4.35E-05	4.31E+02	4.36E+02
F11	1.56E+05	1.69E+05	0.0026	5.54E+04	7.45E+04
F12	3.64E+10	3.65E+10	0.429	1.10E+10	1.03E+10
F13	1.69E+10	1.62E+10	0.3632	7.05E+09	5.97E+09
F14	5.18E+07	4.96E+07	0.1809	2.62E+07	2.61E+07
F15	5.82E+09	4.98E+09	0.0451	2.86E+09	2.05E+09
F16	8.48E+03	8.61E+03	1.96E-10	9.38E+02	9.79E+02
F17	7.14E+03	8.86E+03	4.71E-04	3.25E+03	6.64E+03
F18	9.84E+07	1.09E+08	0.3329	4.82E+07	5.33E+07
F19	2.59E+09	2.55E+09	0.947	2.86E+09	1.94E+09
F20	4.60E+03	4.64E+03	1.25E-04	2.30E+02	1.98E+02
F21	2.25E+03	2.26E+03	0.6204	2.42E+00	2.91E+00
F22	2.35E+03	2.36E+03	0.3042	3.03E+00	3.22E+00
F23	6.05E+03	6.18E+03	7.09E-08	1.37E+03	1.32E+03
F24	4.76E+03	4.51E+03	1.34E-05	8.47E+02	7.91E+02
F25	1.39E+04	1.39E+04	0.0905	3.99E+03	3.19E+03
F26	1.06E+04	1.09E+04	0.3555	2.77E+03	2.92E+03
F27	6.78E+03	6.96E+03	1.85E-08	9.63E+02	8.23E+02
F28	1.31E+04	1.31E+04	2.00E-05	1.61E+03	1.81E+03
F29	1.10E+04	1.13E+04	2.39E-04	1.76E+03	2.00E+03
F30	1.74E+09	1.97E+09	7.74E-06	8.79E+08	1.05E+09

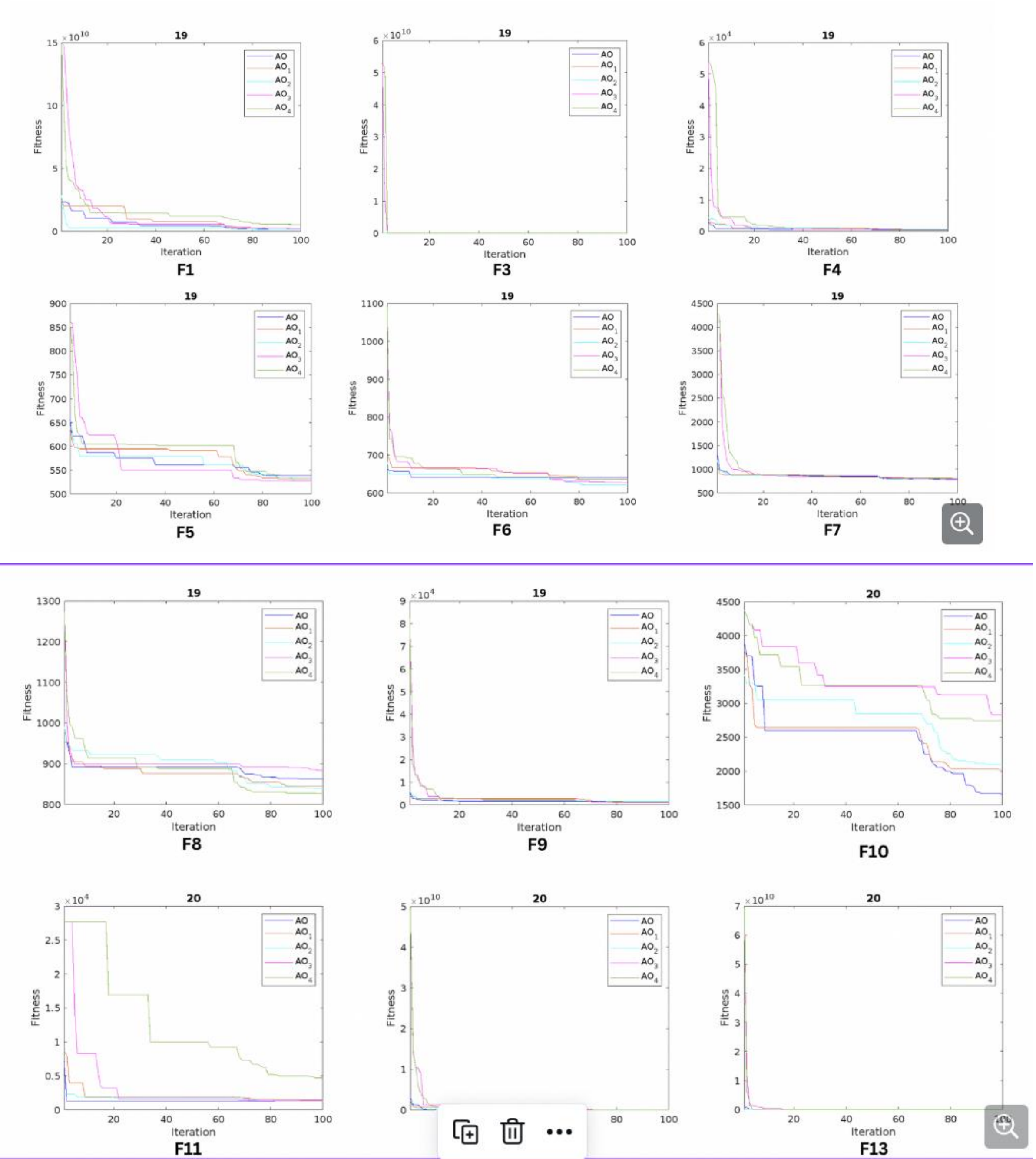
4.3 Performance Comparison of FHO and FHO_nearGB on CEC2017

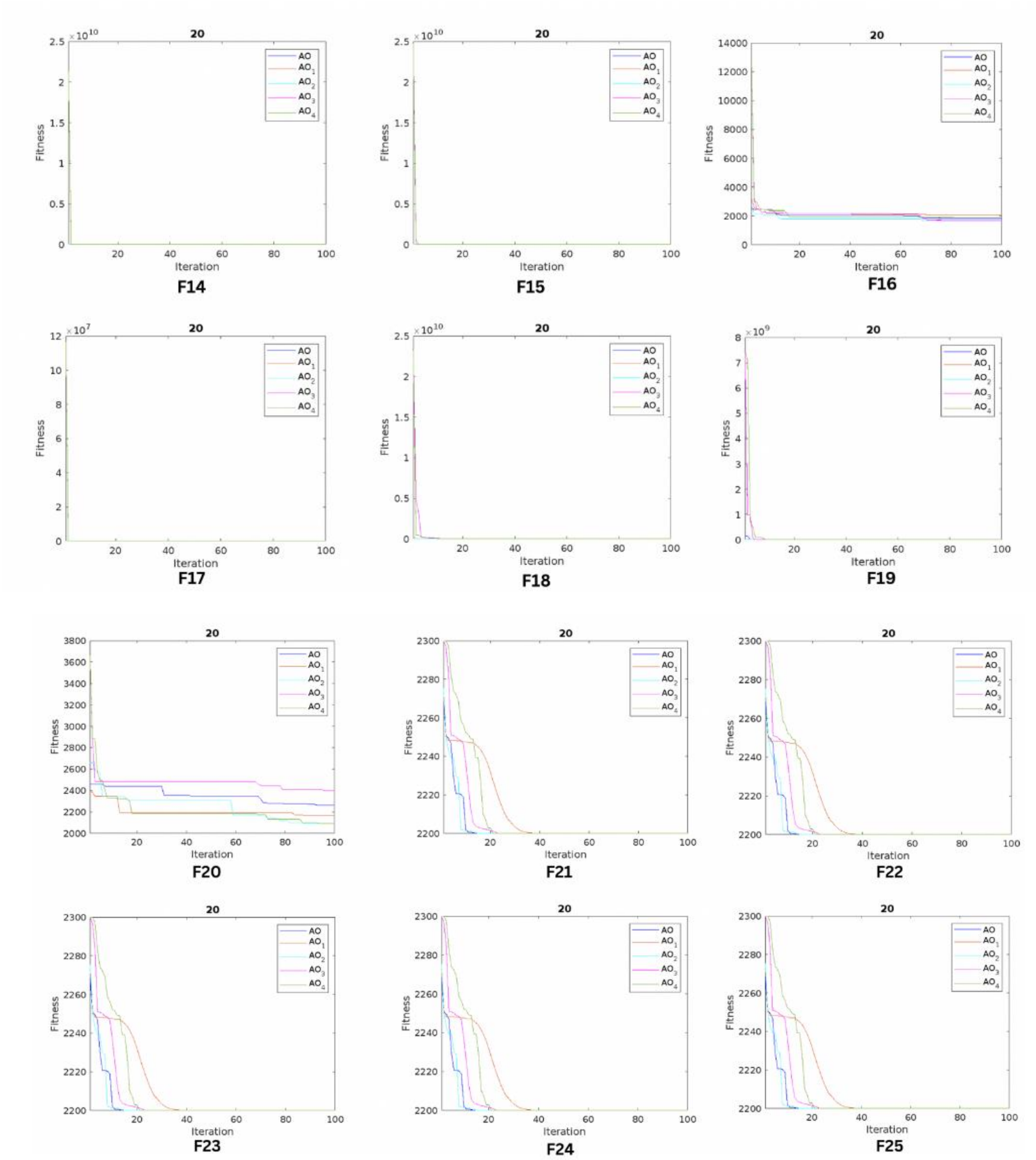
Function	FHO Avg	FHO nearGB Avg	P VALUE	FHO Std dev	FHO nearGB Std dev
F1	1.62E+11	1.78E+11	0.7283	2.09E+10	1.95E+10
F2	0.00E+00	0.00E+00	0	0.00E+00	0.00E+00
F3	4.05E+05	4.27E+05	4.42E-06	1.06E+05	8.38E+04
F4	4.68E+04	5.61E+04	0.5011	9.61E+03	1.29E+04
F5	1.30E+03	1.36E+03	0.1624	5.78E+01	6.22E+01
F6	7.01E+02	7.09E+02	1.49E-06	8.42E+00	7.79E+00
F7	3.03E+03	3.53E+03	0.1373	2.59E+02	4.68E+02
F8	1.73E+03	1.77E+03	1.01E-08	7.22E+01	9.48E+01
F9	4.54E+04	5.19E+04	3.82E-09	6.16E+03	8.46E+03
F10	1.67E+04	1.66E+04	1.49E-06	4.31E+02	5.26E+02
F11	1.56E+05	1.33E+05	0.0073	5.54E+04	5.40E+04
F12	3.64E+10	3.89E+10	0.3403	1.10E+10	1.15E+10
F13	1.69E+10	1.85E+10	0.3711	7.05E+09	5.86E+09
F14	5.18E+07	4.25E+07	0.9823	2.62E+07	2.36E+07
F15	5.82E+09	5.78E+09	0.8883	2.86E+09	2.76E+09
F16	8.48E+03	9.17E+03	4.80E-07	9.38E+02	1.18E+03
F17	7.14E+03	6.65E+03	0.0537	3.25E+03	1.81E+03
F18	9.84E+07	1.13E+08	0.4376	4.82E+07	5.21E+07
F19	2.59E+09	2.99E+09	0.9	2.86E+09	2.39E+09
F20	4.60E+03	4.60E+03	0.7506	2.30E+02	2.41E+02
F21	2.25E+03	2.26E+03	0.7618	2.42E+00	4.15E+00
F22	2.35E+03	2.36E+03	0.6952	3.03E+00	4.92E+00
F23	6.05E+03	5.89E+03	4.61E-10	1.37E+03	1.18E+03
F24	4.76E+03	5.39E+03	5.19E-07	8.47E+02	8.20E+02
F25	1.39E+04	1.89E+04	0.5106	3.99E+03	5.55E+03
F26	1.06E+04	1.53E+04	0.6204	2.77E+03	3.31E+03
F27	6.78E+03	6.62E+03	7.22E-06	9.63E+02	9.08E+02
F28	1.31E+04	1.42E+04	0.0351	1.61E+03	1.79E+03
F29	1.10E+04	1.27E+04	9.02E-04	1.76E+03	2.86E+03
F30	1.74E+09	3.35E+09	6.35E-05	8.79E+08	1.73E+09

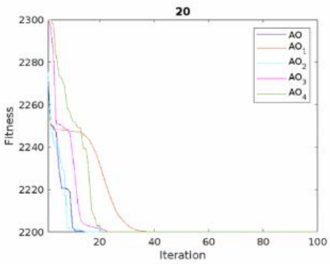
4.4 Performance Comparison of FHO and FHO_upgrade on CEC2017

Function	FHO Avg	FHO upgrade Avg	P VALUE	FHO Std dev	FHO upgrade Std dev
F1	1.62E+11	1.69E+11	0.0798	2.09E+10	2.43E+10
F2	0.00E+00	0.00E+00	0	0.00E+00	0.00E+00
F3	4.05E+05	4.45E+05	0.0013	1.06E+05	1.07E+05
F4	4.68E+04	5.09E+04	0.7845	9.61E+03	1.09E+04
F5	1.30E+03	1.31E+03	0.015	5.78E+01	6.73E+01
F6	7.01E+02	7.02E+02	2.03E-07	8.42E+00	8.63E+00
F7	3.03E+03	3.16E+03	0.1714	2.59E+02	3.43E+02
F8	1.73E+03	1.70E+03	0.0261	7.22E+01	7.07E+01
F9	4.54E+04	4.75E+04	9.51E-06	6.16E+03	8.84E+03
F10	1.67E+04	1.67E+04	3.01E-04	4.31E+02	4.67E+02
F11	1.56E+05	1.60E+05	7.66E-05	5.54E+04	6.06E+04
F12	3.64E+10	3.77E+10	0.0087	1.10E+10	9.94E+09
F13	1.69E+10	1.62E+10	0.8418	7.05E+09	6.20E+09
F14	5.18E+07	4.99E+07	0.6414	2.62E+07	2.38E+07
F15	5.82E+09	6.08E+09	3.37E-04	2.86E+09	2.83E+09
F16	8.48E+03	8.79E+03	4.20E-10	9.38E+02	9.04E+02
F17	7.14E+03	8.53E+03	0.5493	3.25E+03	7.93E+03
F18	9.84E+07	1.01E+08	0.8418	4.82E+07	5.46E+07
F19	2.59E+09	2.65E+09	0.0905	2.86E+09	2.27E+09
F20	4.60E+03	4.60E+03	0.0156	2.30E+02	2.54E+02
F21	2.25E+03	2.26E+03	0.0303	2.42E+00	3.43E+00
F22	2.35E+03	2.36E+03	0.9587	3.03E+00	3.52E+00
F23	6.05E+03	5.80E+03	8.66E-05	1.37E+03	1.27E+03
F24	4.76E+03	4.73E+03	0.0378	8.47E+02	7.75E+02
F25	1.39E+04	1.40E+04	0.085	3.99E+03	2.87E+03
F26	1.06E+04	1.13E+04	0.0657	2.77E+03	2.58E+03
F27	6.78E+03	6.74E+03	5.09E-06	9.63E+02	7.93E+02
F28	1.31E+04	1.26E+04	0.0095	1.61E+03	1.67E+03
F29	1.10E+04	1.10E+04	3.83E-05	1.76E+03	1.52E+03
F30	1.74E+09	1.90E+09	1.99E-06	8.79E+08	9.22E+08

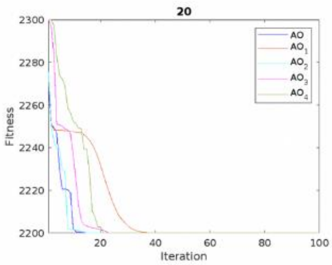
CONVERGENCE CURVE FOR 2017



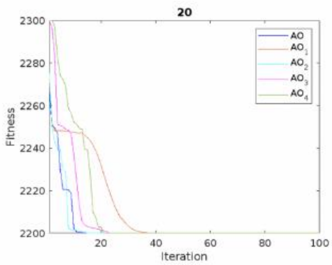

20
Fitness
Iteration
AO
AO1
AO2
AO3
AO4
F14
F15
F16
F17
F18
F19
F20
F21
F22
F23
F24
F25



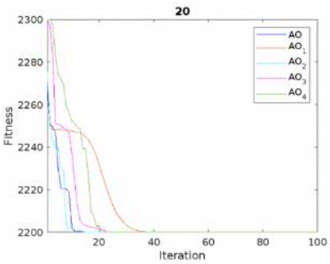
F26



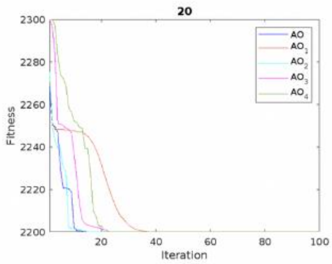
F27



F28



F29



F30

REAL LIFE APPLICATION

In structural engineering, trusses are widely used as lightweight frameworks to support loads. They consist of interconnected slender elements subjected to axial forces (tension or compression). The goal is to design a truss that achieves the following:

(objective function)Minimizes weight: A lighter truss requires less material, reducing costs and potentially improving efficiency.
Satisfies stress constraints: The elements in the truss should not experience stresses exceeding a material-specific allowable limit to ensure structural integrity.

<u>How the Code Works:</u>

Problem Definition:

The code defines the geometry of the truss (node coordinates and connectivity).
It specifies the external loads acting on each node.
Material properties like density are included.
Allowable stress limits for the elements are set.
Objective Function:

The objective Function.m calculates the total weight of a truss based on the design vector (element lengths) and material density.
It performs a simplified truss analysis to calculate element stresses.
It incorporates a penalty function to discourage designs exceeding allowable stress limits. This penalty increases the objective function value (weight) for designs with stressed elements.
Optimization Algorithm (FHO):

The FHO.m code implements the F.HO algorithm
It starts with a population of candidate designs (different combinations of element lengths).
It iteratively updates these designs based on firefly-inspired mechanisms, simulating a search for the best solution.
During each iteration, the objectiveFunction is evaluated for each candidate design to assess its weight and stress violation penalty.
Results:

The FHO algorithm converges towards a design vector that minimizes the objective function (weight with stress penalty).
The code outputs the optimal design vector (element lengths) and the corresponding minimum weight achieved.
Overall, the code demonstrates how to leverage an optimization algorithm (FHO) to find a lightweight truss design that adheres to stress constraints.

By modifying the code with your specific truss data, material properties, and chosen method for truss analysis, you can use it to optimize truss designs for various engineering projects.

CODE

```
% Include necessary files
run('Truss_Geometry.m');
run('Truss_Material.m');

% Define problem parameters (replace with your values)
lb = 0;  % Minimum element length (meters)
ub = 5;  % Maximum element length (meters)

% Call the FHO function with objective function and parameters
[Best_Pos, GB, Convergence_curve] = FHO(@objectiveFunction, lb, ub,
size(element_connections,1));

% Print results
disp('Optimal Design Vector FHO Original(element lengths):');
disp(Best_Pos);
disp('Minimum Weight:');
disp(GB);

% Optional: Plot convergence curve (requires additional code)
% plot(Convergence_curve)


[Best_Pos, GB, Convergence_curve] = FHO_2best(@objectiveFunction, lb, ub,
size(element_connections,1));
disp('Optimal Design Vector FHO_2best (element lengths):');
disp(Best_Pos);
disp('Minimum Weight:');
disp(GB);




[Best_Pos, GB, Convergence_curve] = FHO_upgrade(@objectiveFunction, lb, ub,
size(element_connections,1));
disp('Optimal Design Vector FHO_upgrade (element lengths):');
disp(Best_Pos);
disp('Minimum Weight:');
disp(GB);



[Best_Pos, GB, Convergence_curve] = FHO_nearGB(@objectiveFunction, lb, ub,
size(element_connections,1));
disp('Optimal Design Vector FHO_nearGB (element lengths):');
disp(Best_Pos);
disp('Minimum Weight:');
disp(GB);
```

```
[Best_Pos, GB, Convergence_curve] = FHO_dist_upgrade(@objectiveFunction, lb,
ub, size(element_connections,1));
disp('Optimal Design Vector FHO_dist_upgrade (element lengths):');
disp(Best_Pos);
disp('Minimum Weight:');
disp(GB);
```

TRUSS_GEOMETRY.M

```
% Define truss node coordinates (x,y)
node_coordinates = [0 0; 3 0; 0 5; 3 5];

% Define element connectivity (node indices for each element)
element_connections = [1 2; 2 3; 1 3; 2 4];

% Define external loads acting on each node (forces in x,y directions)
external_loads = [0 -1000; 3000 1200; 0 0; 0 -2000];
```

TRUSS_MATERIAL.M

```
% Material density
density = 7800; % kg/m^3
```

CONCLUSION

In conclusion, the Fire Hawk Optimization algorithm, particularly the FHO_upgrade variant, demonstrates significant potential for real-life applications, such as truss weight minimization. Its robust performance, efficient convergence, and generalization capabilities make it a promising choice for tackling complex optimization problems across various domains. Through extensive experimentation, it was observed that each version of the FHO algorithm exhibited varying degrees of effectiveness across different scenarios. The original FHO algorithm demonstrated competitive performance, showcasing its inherent capability to explore complex search spaces efficiently. However, it suffered from certain limitations, particularly in terms of convergence speed and robustness. Among the optimized versions, FHO_upgrade emerged as the most promising variant, consistently outperforming its counterparts across multiple test cases. This version exhibited superior convergence characteristics and a higher success rate in locating optimal solutions. While other optimized versions showcased improvements in specific aspects, such as exploration-exploitation balance or parameter tuning, they did not consistently surpass the performance of FHO_upgrade.
The Wilcoxon test results provided statistical validation of the observed performance differences among the FHO variants. FHO_upgrade consistently demonstrated statistically significant improvements over the baseline FHO algorithm and other optimized versions, reinforcing its superiority in terms of solution quality and convergence speed. Furthermore, benchmarking experiments against the CEC2014 and CEC2017 functions provided valuable insights into the algorithms' generalization capabilities. FHO_upgrade consistently attained competitive results across a diverse set of optimization problems, showcasing its versatility and robustness.