Enhancing Mountain Gazelle Optimizer (MGO) with an Improved F-Parameter for Global Optimization

Toufic Seini[1], Abdul-Fatawu Seini Yussif[2], Iddrisu Mohammed Katali[3]

[1,3]Department of Physical Sciences, University for Development Studies, Ghana
[2]Department of Electrical and Electronic Engineering, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana

---***---

Abstract - *In this paper, a simplified alternative approach to determining the value of parameter F in Mountain Gazelle Optimizer (MGO) is proposed to enhance the algorithm's global performance on high-dimensional problems. The proposed improved MGO called Improved F-parameter Mountain Gazelle Optimizer (IFMGO), was tested on 13 standard high-dimensional benchmark functions while varying the problem dimensions. Further simulation test was run on other 10 fixed-dimensional benchmark functions. The MATLAB simulation results were compared to those of the original MGO and Particle Swarm Optimization (PSO) algorithms reported in the literature. The proposed IFMGO performed exceptionally better than the original MGO and PSO in solving high-dimensional optimization benchmark functions, as well as maintained excellent performance on fixed-dimensional optimization benchmark functions. The IFMGO also exhibited robustness, good convergence characteristics, and stability relative to the other algorithms.*

Key Words: optimization, mountain gazelle optimizer, benchmark functions, particle swarm optimization, improved F-parameter mountain gazelle optimizer.

1. INTRODUCTION

In recent years, nature-inspired optimization algorithms have gained significant attention for their ability to tackle complex optimization problems [1][2][3]. One such algorithm is the Mountain Gazelle Optimizer (MGO), inspired by the intelligence behind the wildlife of mountain gazelle species in their natural habitat [4]. While MGO has shown promising results in various optimization tasks, its performance in solving high-dimensional problems can be further enhanced to address the challenges presented by real-world problems characterized by a large number of variables [5].

In this paper, a new modification to the MGO is proposed by improving the F-parameter calculation to improve the algorithm's performance in handling high-dimensional optimization problems. The F-parameter plays a crucial role in controlling the exploration and exploitation trade-off during the search process, influencing the convergence speed and stability to escape local optima [5]. By modifying the F-parameter to the specific requirements of high-dimensional problems, the algorithm's ability to efficiently explore the search space and locate global optimal solutions is enhanced.

The primary objective of this research is to investigate the impact of the proposed F-parameter modification on the performance of the MGO algorithm when applied to a set of widely recognized high-dimensional benchmark functions. These benchmark functions have been extensively used in the literature to evaluate and compare the performance of various optimization algorithms [4][6][7]. By conducting a comprehensive experimental study, the aim is to assess the effectiveness of the proposed modification and provide empirical evidence of its benefits.

The remainder of this paper is organized as follows: Section 2 presents a detailed description of the MGO algorithm and its key components. Section 3 outlines the proposed modification to the F-parameter and explains its rationale. Section 4 describes the experimental setup and implementation to evaluate the performance. Section 5 presents the test results and discussions drawn from the test simulation. Finally, Section 6 concludes and outlines possible directions for future research.

2. ORIGINAL MGO ALGORITHM

2.1 Background

The mountain gazelle is a species of gazelle that naturally live in the Arabian Peninsula and the surrounding regions [4]. Despite having a wide distribution, the gazelle population density is relatively low. This species is closely linked to the habitat of the Robinia tree species. Mountain gazelles exhibit strong territorial behavior, establishing their territories at significant distances from each other. They form three distinct types of groups, which include herds consisting of mothers and offspring, herds of young males, and solitary males within their territories. Male gazelles engage in frequent battles, where the competition for resources is more intense than the competition for females. In these battles, immature males utilize their horns more frequently compared to adults or territorial males. Mountain gazelles undertake migrations of over 120km in search of food. They possess remarkable speed, being able to run 100 meters at a speed of 80km/h on average [4][5].

2.2 Mathematical Modelling

The MGO optimization algorithm is a mathematical model that draws inspiration from the social behavior and habitats of mountain gazelles. It incorporates essential elements of the gazelles' group dynamics, such as the behavior of bachelor male herds (BMH), maternity herds (MH), territorial and solitary males (TSM), and their migration pattern in search of food (MSF). These aspects are mathematically represented in the following manner.

Territorial Solitary Male (TSM):

The adult male gazelles' mechanism of protecting their territories against intruders is mathematically modeled in equation (1).

$$TSM = male_{gazelle} - |(ri_1 \times BH - ri_2 \times X(t)) \times F| \times Cof_r \quad (1)$$

Where;

ri_1 and *ri_2*: are random integers of either 1 or 2.

$male_{gazelle}$: is the position vector of the best male gazelle so far.

The values of *HB, F,* and *Cof_r* are determined using equations (2), (3), and (4).

$$BH = X_{ra} \times r_1 + M_{pr} \times r_2, \qquad ra = \left\{\tfrac{N}{3} .....N\right\} \quad (2)$$

The value of X_{ra} is a random solution (young male) in the range of *ra*, and that of M_{pr} is the average number of search agents. The value of *N* is the number of gazelles, and r_1 and r_2 are random values from a range of (0, 1).

$$F = N_1(D) \times \exp\left(2 - Iter \times \left(\tfrac{2}{MaxIter}\right)\right) \quad (3)$$

N_1 represents random values with the size of the problem dimension determined using a standard distribution. The *Iter* and *MaxIter* respectively represent the iteration count and the maximum iterations.

$$Cof_i = \begin{cases} (a+1)+r_3, \\ a \times N_2(D), \\ r_4(D), \\ N_3(D) \times N_4(D)^2 \times \cos((r_4 \times 2) \times N_3(D)), \end{cases} \quad (4)$$

Where;

r_3 and r_4: represent random values within the range (0, 1).

N_2, N_3, and N_4: are set of randomly generated values with the size of the problem function.

The value of *a* is determined using equation (5) below at every iteration.

$$a = -1 + Iter \times \left(\frac{-1}{MaxIter}\right) \quad (5)$$

Maternity Herd (MH):

The intelligence behind the mother gazelle's act of protecting its offspring is mathematically modeled in equation (6).

$$MH = (BH + Cof_{1,r}) + (ri_3 \times male_{gazelle} - ri_4 \times X_{rand}) \times Cof_{1,r} \quad (6)$$

Where;

X_{rand}: represents a vector position of a gazelle randomly selected from the population.

ri_3 and ri_4: are integers randomly chosen from (1, 2).

Bachelor Male Herds (BMH):

In part of the development process of the male gazelles, the young adult male ones create their territories and try winning female gazelles to join them. This behavior is modeled in equation (7).

$$BMH = (X(t) - D) + (ri_5 \times male_{gazelle} - ri_6 \times BH) \times Cof_r \quad (7)$$

Where;

X(t): is the position vector of the gazelle in the current iteration.

ri_5, and ri_6: are integers randomly from (1, 2).

r_6: is a randomly selected value from range (0 1).

The value of *D* is determined using equation (8) below.

$$D = (|X(t)| + |male_{gazelle}|) \times (2 \times r_6 - 1) \quad (8)$$

Migration in Search of Food (MSF):

The foraging mechanism of mountain gazelles involves roaming to search the green pasture of their choice. This random movement is modeled in equation (9).

$$MSF = (ub - lb) \times r_7 + lb \quad (9)$$

lb and ub represent the lower search boundary and the upper search boundary respectively. The value of r_7 is randomly chosen from (0, 1).

Pseudocode of MGO Algorithm

Inputs: *iteration counter (Iter), maximum iteration (MaxIter), population size (N).*

Output: *gazelle's position, and its fitness value*

Initialize random gazelle populations, X_i(i=1, 2, ...N)

Evaluate the fitness values of the population.

While *(Iter < MaxIter),* ***do***

for *(every gazelle, X_i)* ***do***

Calculate TSM using equation (1)

Calculate MH using equation (6)

Calculate BMH using equation (7)

Calculate MSF using equation (9)

Evaluate the fitness values of TSM, MH, BMH, and MSF.

End for

Output best gazelle, X_{best}, and its fitness value.

End while

3. Proposed Modification in F-parameter

The territorial solitary males (TSM) phase is one of the major population update operators in the MGO algorithm, and it is mathematically presented in equation (1). The parameter F in the equation greatly influences the convergence behavior of the MGO algorithm. To enhance the algorithm's performance in producing better global solutions for high-dimensional problems, an alternative mathematical expression is proposed for the calculation of the parameter F in equation (1). The initial mathematical determination of the value parameter F, expressed in equation (3), has been replaced with a proposed expression presented in equation (10) below.

$$F = randn(1,d) \times \exp\left(-Iter\right) \quad (10)$$

Where; Iter is the iteration counter, d represents the problem dimension, and *randn(1,d)* generates normally distributed pseudorandom numbers.

The proposed mathematical expression for determining the value of parameter F, represented in equation (10), efficiently balanced the exploration and exploitation process of the MGO algorithm during the search process. The algorithm's ability to efficiently explore the search space and locate global optimal solutions is enhanced.

Flow Chart of IFMGO Algorithm

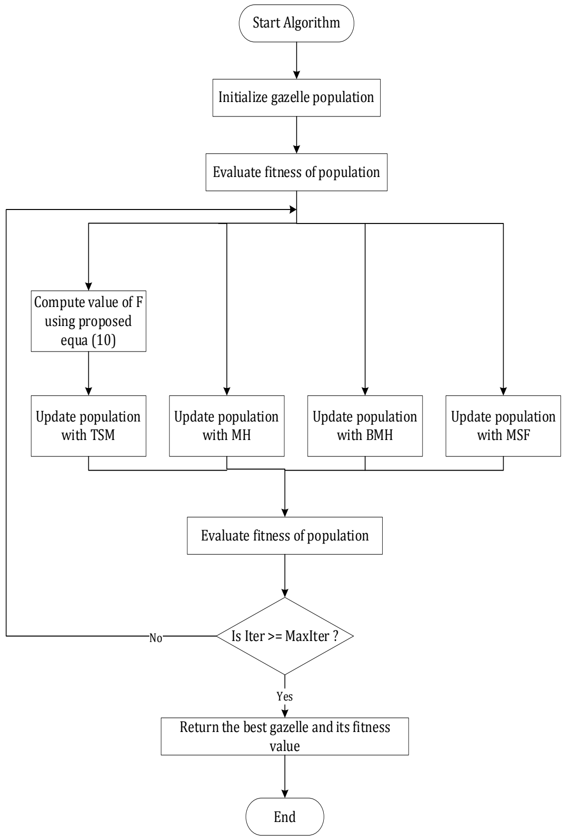


Figure 1: Flow Chart of IFMGO Algorithm

4. Testing Proposed IFMGO Algorithm

Twenty-three (23) standard benchmark test functions used in the original MGO algorithm are considered in this work to establish the effective performance of the proposed IFMGO algorithm [4][8]. The first thirteen (13) benchmark functions (F1-F13), representing high-dimensional optimization problems, are considered for various test cases based on varying problem dimensions. Case 1 considered the functions with the default problem dimension of 30, case 2 considered the same functions with an increased in dimension to 100, case 3 considered a 500-dimension size, and Case 4 considered the same problem functions with a dimension size of 1000. The remaining ten benchmark functions (F14-F23), representing standard fixed-dimensional optimization problems, are considered for case 5 [9]. The IFMGO algorithm is implemented in a MATLAB environment using a hp pavilion laptop computer. The simulation parameters settings are presented in Table 1.

Table - 1: Simulation Parameters

Item	Value
Population Size	30
Maximum Iteration	500
Number of Runs	30

The default information of the standard benchmark functions used to test the IFMGO algorithm is given in Table 2, while the function equations are presented in Table 3 [10].

Table - 2: Detail Information of Benchmark Functions

No	Function	Search Range	Global Optimum	Dim
1	F1	[-100, 100]	0	30
2	F2	[-10, 10]	0	30
3	F3	[-100, 100]	0	30
4	F4	[-100, 100]	0	30
5	F5	[-30, 30]	0	30
6	F6	[-100, 100]	0	30
7	F7	[-1.28, 1.28]	0	30
8	F8	[-500, 500]	-12,569	30
9	F9	[-5.12, 5.12]	0	30
10	F10	[-32, 32]	0	30
11	F11	[-600, 600]	0	30
12	F12	[-50, 50]	0	30
13	F13	[-50, 50]	0	30
14	F14	[-65.53, 65.53]	0.998	2
15	F15	[-5, 5]	0.00030	4
16	F16	[-5, 5]	-1.0316	2
17	F17	[-5, 0] [10, 15]	0.398	2
18	F18	[-5, 5]	3	2
19	F19	[0, 1]	-3.86	3
20	F20	[0, 1]	-3.32	6
21	F21	[0, 10]	-10.1532	4
22	F22	[0, 10]	-10.4028	4
23	F23	[0, 10]	-10.5363	4

5. Results and Discussion

The IFMGO algorithm was simulated on each of the test functions 30 times and the following statistical information; best, mean, worst, and standard deviation (Std) were extracted. The simulation test was repeated for all five (5) case scenarios. The results were compared to those reported in the base MGO algorithm under the same cases to illustrate the effectiveness of the performance of the proposed IFMGO algorithm in handling complex high-dimensional problems as well as fixed-dimensional problems [4]. The results comparison is presented under each case scenario.

Case 1: Results Comparison for F1-to-F13 (dimension = 30)

Table 4 contains the simulation results of the IFMGO algorithm on the high-dimensional benchmark functions with the default dimensions of 30, compared to those of the MGO algorithm and PSO algorithm reported in literature with the same dimension [5].

Table - 3: Results of Case 1 (d=30)

Func	Para	IFMGO	MGO	PSO
F1	Best	**5.5672E-273**	2.4152E-81	7.1071E-09
	Worst	**4.2677E-236**	4.9485E-71	7.3614E-05
	Mean	**1.4291E-237**	4.7455E-72	4.4590E-06
	STD	**0.0000E+00**	1.3401E-71	1.4568E-05
F2	Best	**2.3796E-154**	1.6760E-46	5.6941E-06
	Worst	**1.6278E-136**	6.0784E-41	2.0504E-02
	Mean	**5.4260E-138**	3.9067E-42	3.3272E-03
	STD	**2.9719E-137**	1.1893E-41	5.4084E-03
F3	Best	**3.9933E-67**	3.5309E-14	1.8171E+01
	Worst	**2.0790E-37**	1.6370E-07	3.4851E+03
	Mean	**6.9580E-39**	6.8224E-09	5.8934E+02
	STD	**3.7953E-38**	2.9791E-08	9.9009E+02
F4	Best	**2.2873E-117**	5.2537E-30	2.6406E-01
	Worst	**5.1373E-93**	4.1424E-22	2.1015E+00
	Mean	**1.7224E-94**	1.5909E-23	5.3311E-01
	STD	**9.3776E-94**	7.5435E-23	3.8014E-01
F5	Best	**0.0000E+00**	0.0000E+00	1.9874E+01
	Worst	**3.2121E-29**	2.5559E-22	1.0846E+02
	Mean	**2.7175E-30**	1.1953E-23	4.6616E+01
	STD	**7.5412E-30**	4.9592E-23	3.0547E+01
F6	Best	**1.2433e-13**	4.8095E-12	6.7447E-09
	Worst	**3.1703E-10**	3.5099E-08	4.1088E-05
	Mean	**2.1348E-11**	4.5398E-09	2.9114E-06
	STD	**6.9217E-09**	7.6544E-09	7.6033E-06
F7	Best	**3.2389E-05**	3.2450E-05	4.0726E-02
	Worst	**9.9062E-04**	1.5342E-03	1.5602E-01
	Mean	**2.3789E-04**	5.5958E-04	9.5056E-02
	STD	**2.1642E-04**	3.8895E-04	2.9977E-02
F8	Best	**-1.2569E+04**	**-1.2569E+04**	-3.315E+03
	Worst	**-1.2569E+04**	**-1.2569E+04**	-1.949E+03
	Mean	**-1.2569E+04**	**-1.2569E+04**	-2.590E+03
	STD	**1.7527E-08**	3.9992E-08	2.810E+02
F9	Best	**0.0000E+00**	**0.0000E+00**	1.9899E+01
	Worst	**0.0000E+00**	**0.0000E+00**	7.2632E+01
	Mean	**0.0000E+00**	**0.0000E+00**	3.8671E+01
	STD	**0.0000E+00**	**0.0000E+00**	1.3480E+01
F10	Best	**8.8818E-16**	**8.8818E-16**	3.7741E-06
	Worst	**4.4409E-15**	**4.4409E-15**	2.4083E+00
	Mean	**1.0066E-15**	1.7171E-15	3.1302E-01
	STD	**6.4863E-16**	1.5283E-15	7.3029E-01
F11	Best	**0.0000E+00**	**0.0000E+00**	6.3190E+01
	Worst	**0.0000E+00**	**0.0000E+00**	1.0377E+02
	Mean	**0.0000E+00**	**0.0000E+00**	8.3190E+01
	STD	**0.0000E+00**	**0.0000E+00**	1.0729E+01
F12	Best	**1.5705E-32**	**1.5705E-32**	9.3714E-11
	Worst	**1.6916E-32**	2.1956E-25	1.5674E+00
	Mean	**1.6313E-32**	1.6966E-26	2.4928E-01
	STD	**4.4173E-34**	4.5383E-26	3.7051E-01
F13	Best	**1.3498E-32**	**1.3498E-32**	2.9246E-11
	Worst	**3.5685E-32**	6.4034E-32	1.1006E-02
	Mean	**1.5618E-32**	1.8141E-32	2.5662E-03
	STD	**5.3197E-33**	9.9543E-33	4.7275E-03

The IFMGO algorithm performed exceptionally better than the base MGO algorithm and the PSO algorithm in all the benchmark functions, except in benchmark functions F9 and F11 where it produced the exact global optimal solution as the base MGO. With the default dimension for all 13 benchmark functions, the IFGMO algorithm is superior to the other two algorithms. In the remaining three test cases, the dimension is gradually increased to see the performance response to these changes in dimensions. The dimension is successively increased in the order of 100, 500, and 1000 respectively [4].

Case 2: Results Comparison for F1-F13 (dimension = 100)

Presented in Table 4 are the results of the proposed IFMGO algorithm, the base MGO algorithm, and the PSO algorithm with a problem dimension of hundred (100).

Table - 4: Results of Case 2 (d=100)

Func	Para	IFMGO	MGO	PSO
F1	Best	**5.8064E-262**	5.8529E-72	5.0972E-01
	Worst	**2.7701E-220**	5.0778E-59	4.6534E+00
	Mean	**9.2335E-222**	2.8822E-60	1.4251E+00
	STD	**0.0000E+00**	1.0124E-59	9.7808E-01
F2	Best	**9.3785E-152**	1.8226E-39	1.4002E+00
	Worst	**4.6887E-131**	1.1415E-34	6.9190E+00
	Mean	**2.8979E-132**	1.0590E-35	3.3344E+00
	STD	**9.7798E-132**	2.3627E-35	1.4074E+00
F3	Best	**9.5118E-58**	1.3729E-09	7.1514E+03
	Worst	**9.4411E-17**	1.6621E+00	5.9518E+04
	Mean	**3.3543E-18**	1.0307E-01	2.0883E+04
	STD	**1.7234E-17**	3.8330E-01	1.3515E+04
F4	Best	**1.7822E-113**	4.1801E-28	5.4860E+00
	Worst	**4.4171E-90**	5.5156E-20	8.1293E+00
	Mean	**1.6099E-91**	3.3318E-21	6.7348E+00
	STD	**8.0658E-91**	1.1114E-20	6.9549E-01
F5	Best	**0.0000E+00**	1.9281E-28	2.1878E+02
	Worst	**1.1618E-28**	2.4356E-24	8.4957E+02
	Mean	**2.1951E-29**	1.2127E-25	4.3733E+02
	STD	**3.9372E-29**	4.4925E-25	1.2317E+02
F6	Best	**8.3708E-10**	7.9256E-08	3.7238E-01
	Worst	**1.1378E-03**	1.3496E-03	3.1286E+00
	Mean	**1.4312E-04**	1.9521E-04	1.2662E+00
	STD	**1.0238E-04**	3.1368E-04	6.9627E-01
F7	Best	**9.2093E-06**	1.0821E-04	2.7694E+00
	Worst	**1.5500E-03**	1.7247E-03	1.8461E+01
	Mean	**2.5734E-04**	6.5441E-04	5.8645E+00
	STD	**3.3519E-04**	4.2507E-04	3.5868E+00
F8	Best	**-4.1898E+04**	-4.189E+04	-6.084E+03
	Worst	**-4.1898E+04**	-4.189E+04	-3.999E+03
	Mean	**-4.1898E+04**	-4.189E+04	-4.847E+03
	STD	**1.9854E-02**	5.4299E-04	5.5520E+02
F9	Best	**0.0000E+00**	0.0000E+00	1.1118E+02
	Worst	**0.0000E+00**	0.0000E+00	2.3001E+02
	Mean	**0.0000E+00**	0.0000E+00	1.5880E+02
	STD	**0.0000E+00**	0.0000E+00	2.4475E+01
F10	Best	**8.8818E-16**	8.8818E-16	2.2839E+00
	Worst	**8.8818E-16**	4.4409E-15	5.1474E+00
	Mean	**8.8818E-16**	1.9540E-15	3.4326E+00
	STD	**0.0000E+00**	1.6559E-15	7.7478E-01
F11	Best	**0.0000E+00**	0.0000E+00	2.9293E+02
	Worst	**0.0000E+00**	0.0000E+00	3.7786E+02
	Mean	**0.0000E+00**	0.0000E+00	3.3580E+02
	STD	**0.0000E+00**	0.0000E+00	2.2170E+01
F12	Best	**4.7116E-32**	4.5392E-28	7.2916E-02
	Worst	**4.8326E-32**	1.8790E-22	2.8015E+00
	Mean	**4.7552E-32**	1.4673E-23	1.0396E+00
	STD	**4.5567E-34**	3.7348E-23	6.8852E-01
F13	Best	**1.3498E-32**	7.1430E-32	1.0559E+01
	Worst	**1.7196E-32**	1.5033E-27	8.8300E+01
	Mean	**1.3621E-32**	9.3420E-29	4.4825E+01
	STD	**6.7512E-34**	2.8261E-28	2.0855E+01

Case 3: Results Comparison for F1-F13 (dimension = 500)

Table 5 contains simulation results of the IFMGO algorithm, MGO algorithm, and PSO algorithm with a problem dimension of 500.

Table - 5: Results of Case 3 (d=500)

Func	Para	IFMGO	MGO	PSO
F1	Best	**2.8209E-247**	6.3261E-65	2.2908E+02
	Worst	**3.9502E-218**	7.8225E-57	6.6223E+02
	Mean	**1.3172E-219**	8.3894E-58	4.0788E+02
	STD	**0.0000E+00**	1.9479E-57	9.3153E+01
F2	Best	**6.4132E-147**	4.0752E-37	1.3061E+02
	Worst	**4.3948E-126**	6.6739E-31	2.0196E+02
	Mean	**1.4727E-127**	3.7635E-32	1.6522E+02
	STD	**8.0223E-127**	1.3419E-31	1.7363E+01
F3	Best	**8.1929E-37**	1.0212E-03	2.2182E+05
	Worst	**1.4819E+00**	1.2988E+03	1.4161E+06
	Mean	**4.9624E-02**	9.5107E+01	5.6552E+05
	STD	**2.7051E-01**	3.0161E+02	2.7686E+05
F4	Best	**5.3363E-110**	2.5475E-24	1.0858E+01
	Worst	**1.0371E-90**	3.3913E-20	1.4880E+01
	Mean	**3.9329E-92**	4.6534E-21	1.3217E+01
	STD	**1.8911E-91**	8.4085E-21	9.9456E-01
F5	Best	**0.0000E+00**	1.1116E-26	2.1847E+04
	Worst	**4.8431E-28**	4.4281E-22	4.6978E+04
	Mean	**6.1804E-29**	3.2347E-23	3.4912E+04
	STD	**1.2812E-28**	8.4085E-23	6.9396E+03
F6	Best	**6.4686E-08**	2.2284E-05	2.6194E+02
	Worst	**9.5367E-02**	9.6604E-02	9.1820E+02
	Mean	**4.5669E-03**	1.3185E-02	4.1107E+02
	STD	**1.7451E-02**	2.5846E-02	1.5448E+02
F7	Best	**2.1416e-06**	7.5304E-05	1.9533E+03
	Worst	**2.0328E-03**	1.8466E-03	2.9580E+03
	Mean	**3.5801E-04**	6.9155E-04	2.3677E+03
	STD	**4.6525E-04**	4.7589E-04	2.8259E+02
F8	Best	**-2.0949E+05**	-2.0949E+05	-1.3732E+04
	Worst	**-2.0949E+05**	-2.0949E+05	-7.7850E+03
	Mean	**-2.0949E+05**	-2.0949E+05	-1.0804E+04
	STD	**1.3908E-02**	2.7521E-02	1.7772E+03
F9	Best	**0.0000E+00**	**0.0000E+00**	2.1294E+03
	Worst	**0.0000E+00**	**0.0000E+00**	2.8336E+03
	Mean	**0.0000E+00**	**0.0000E+00**	2.4039E+03
	STD	**0.0000E+00**	**0.0000E+00**	1.7236E+02
F10	Best	**8.8818E-16**	**8.8818E-16**	6.8260E+00
	Worst	**8.8818E-16**	4.4409E-15	8.4758E+00
	Mean	**8.8818E-16**	1.3619E-15	7.7063E+00

	STD	**0.0000E+00**	1.2283E-15	4.3292E-01
F11	Best	**0.0000E+00**	**0.0000E+00**	1.6763E+03
	Worst	**0.0000E+00**	**0.0000E+00**	1.8663E+03
	Mean	**0.0000E+00**	**0.0000E+00**	1.7885E+03
	STD	**0.0000E+00**	**0.0000E+00**	4.9130E+01
F12	Best	**9.4233E-34**	4.2975E-27	2.8703E+00
	Worst	**3.1399E-33**	2.6890E-21	4.6435E+00
	Mean	**1.4575E-33**	1.0060E-22	3.8666E+00
	STD	**5.5763E-34**	4.8930E-22	4.6638E-01
F13	Best	**1.3498E-32**	4.8797E-30	6.0221E+02
	Worst	**5.8419E-31**	4.3930E-25	9.0046E+02
	Mean	**2.1905E-31**	1.8353E-26	7.4543E+02
	STD	**1.2343E-31**	7.9783E-26	8.5221E+01

Case 4: Results Comparison for F1-F13 (dimension = 1000)

In Table 6 are the results of the IFMGO algorithm, MGO algorithm, and PSO algorithm with a problem dimension of 1000.

Table - 6: Results of Case 4 (d=1000)

Func	Para	IFMGO	MGO	PSO
F1	Best	**4.6815E-246**	2.0680E-65	2.4923E+03
	Worst	**2.6767E-215**	3.3469E-54	5.9108E+03
	Mean	**9.5208E-217**	2.0920E-55	3.5190E+03
	STD	**0.0000E+00**	7.1240E-55	6.6606E+02
F2	Best	**2.4812E-148**	4.3762E-36	8.0454E+02
	Worst	**5.5287E-124**	7.0749E-31	Inf
	Mean	**1.8919E-125**	3.1598E-32	Inf
	STD	**1.0088E-124**	1.2894E-31	NaN
F3	Best	**7.0647E-17**	8.2558E-03	7.4284E+05
	Worst	**1.7013E+01**	1.8423E+04	8.7022E+06
	Mean	**5.6711E+00**	2.0856E+03	2.6108E+06
	STD	**9.8227E+00**	4.8253E+03	1.7905E+06
F4	Best	**1.3899E-114**	7.6613E-26	1.4682E+01
	Worst	**6.8124E-83**	1.1359E-18	1.7249E+01
	Mean	**2.2708E-84**	5.2719E-20	1.5704E+01
	STD	**1.2438E-83**	2.0792E-19	6.2236E-01
F5	Best	**0.0000E+00**	6.9113E-26	2.4047E+05
	Worst	**2.2480E-28**	6.6826E-22	4.3051E+05
	Mean	**2.1532E-29**	7.6341E-23	3.3618E+05
	STD	**5.2756E-29**	1.5242E-22	4.4091E+04
F6	Best	**9.5648E-08**	1.2772E-06	2.4440E+03
	Worst	**1.6199E-01**	1.7625E-01	4.7158E+03
	Mean	**7.5226E-03**	2.1746E-02	3.4883E+03
	STD	**3.5105E-02**	4.4438E-02	5.0329E+02
F7	Best	**3.3092E-05**	1.8704E-04	1.5771E+04
	Worst	**1.5667E-03**	2.8759E-03	2.8577E+04
	Mean	**2.9676E-04**	1.0528E-03	2.1410E+04
	STD	**3.7261E-04**	6.8893E-04	3.0007E+03
F8	Best	**-4.1898E+05**	-4.1898E+05	-1.9784E+04
	Worst	**-4.1898E+05**	-4.1898E+05	-1.1276E+04
	Mean	**-4.1898E+05**	-4.1898E+05	-1.4824E+04
	STD	**5.7911E-02**	4.0104E-01	2.0745E+03
F9	Best	**0.0000E+00**	**0.0000E+00**	5.8035E+03
	Worst	**0.0000E+00**	**0.0000E+00**	7.0748E+03
	Mean	**0.0000E+00**	**0.0000E+00**	6.4039E+03
	STD	**0.0000E+00**	**0.0000E+00**	2.5631E+02
F10	Best	**8.8818E-16**	**8.8818E-16**	7.7746E+00
	Worst	**8.8818E-16**	4.4409E-15	9.3308E+00
	Mean	**8.8818E-16**	1.3619E-15	8.5188E+00
	STD	**0.0000E+00**	1.2283E-15	3.2110E-01
F11	Best	**0.0000E+00**	**0.0000E+00**	3.4845E+03
	Worst	**0.0000E+00**	**0.0000E+00**	3.7033E+03
	Mean	**0.0000E+00**	**0.0000E+00**	3.5917E+03
	STD	**0.0000E+00**	**0.0000E+00**	6.0923E+01
F12	Best	**4.7116E-34**	7.0959E-28	4.2959E+00
	Worst	**1.6716E-33**	3.5238E-22	9.0145E+00
	Mean	**7.9361E-34**	2.2783E-23	6.3190E+00
	STD	**4.0264E-34**	6.6733E-23	1.0388E+00
F13	Best	**1.3498E-32**	1.5909E-29	1.6508E+03
	Worst	**7.0868E-31**	5.0494E-25	3.1058E+03
	Mean	**4.1762E-31**	5.3591E-26	2.2124E+03
	STD	**2.0839E-31**	1.1590E-25	3.8699E+02

From Table 3 to Table 6, the proposed IFMGO algorithm showed consistently exceptional performance on the high-dimensional benchmark functions compared to the MGO algorithm and PSO algorithm. Comparing the performance of the IFMGO algorithm in all four case scenarios, it maintained very good performance on almost all the benchmark functions, given an indication that the changes in problem dimension do not affect the performance of the proposed modified version of the MGO algorithm. For the base MGO algorithm and the PSO algorithm, the performances declined as the problem dimension increased from 30 gradually up to 1000.

Case 5: Results Comparison for F14-F23 (fixed-dimensions)

The performance of the four algorithms on fixed-dimensional benchmark functions is presented in Table 7. This illustrates the influence of the proposed modification on the performance of the algorithm on fixed-dimensional problems.

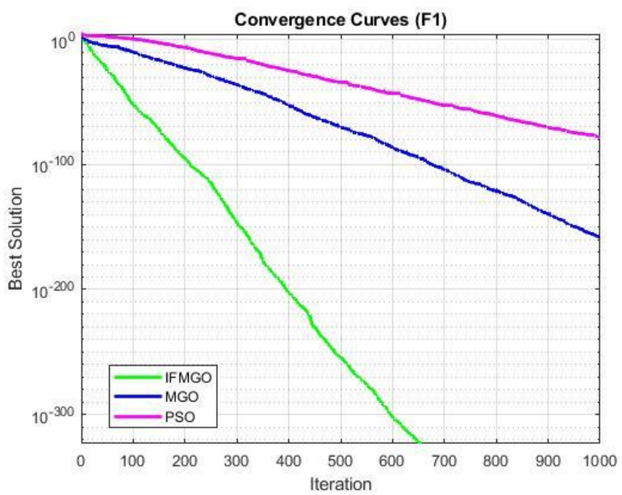
Table - 7: Results of Case 5 (Fixed-dimension)

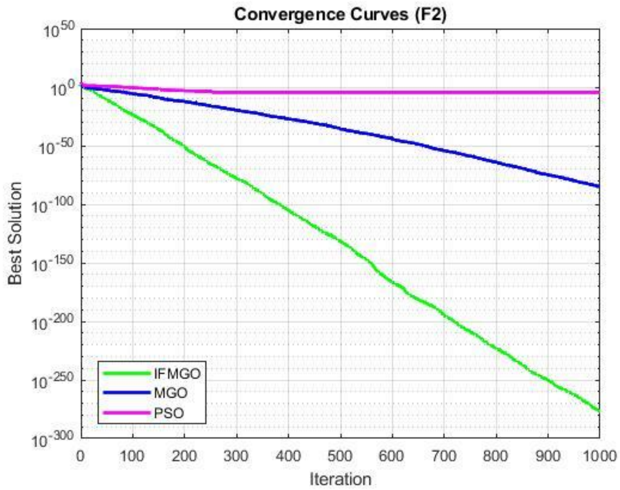
Func	Para	IFMGO	MGO	PSO
F14	Best	**9.9800E-01**	**9.9800E-01**	**9.9800E-01**
	Worst	**9.9800E-01**	**9.9800E-01**	1.9926E+00
	Mean	**9.9800E-01**	**9.9800E-01**	1.3294E+00
	STD	**5.9168E-17**	1.8440E-16	4.7662E-01
F15	Best	**3.0605E-04**	**3.0749E-04**	**3.0749E-04**
	Worst	**1.2343E-03**	**1.2232E-03**	2.0363E-02
	Mean	**3.0779E-04**	3.7059E-04	1.2877E-03
	STD	**1.9014E-04**	2.3182E-04	3.6337E-03
F16	Best	**-1.0316E+00**	**-1.0316E+00**	**-1.0316E+00**
	Worst	**-1.0316E+00**	**-1.0316E+00**	**-1.0316E+00**
	Mean	**-1.0316E+00**	**-1.0316E+00**	**-1.0316E+00**
	STD	**6.9914E-17**	4.7908E-16	6.3877E-16
F17	Best	**3.9789E-01**	**3.9789E-01**	**3.9789E-01**
	Worst	**3.9789E-01**	**3.9789E-01**	**3.9789E-01**
	Mean	**3.9789E-01**	**3.9789E-01**	**3.9789E-01**
	STD	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
F18	Best	**3.0000E+00**	**3.0000E+00**	**3.0000E+00**
	Worst	**3.0000E+00**	**3.0000E+00**	**3.0000E+00**
	Mean	**3.0000E+00**	**3.0000E+00**	**3.0000E+00**
	STD	**1.1019E-15**	1.4092E-15	2.0550E-15
	Best	**-3.8628E+00**	**-3.8628E+00**	**-3.8628E+00**

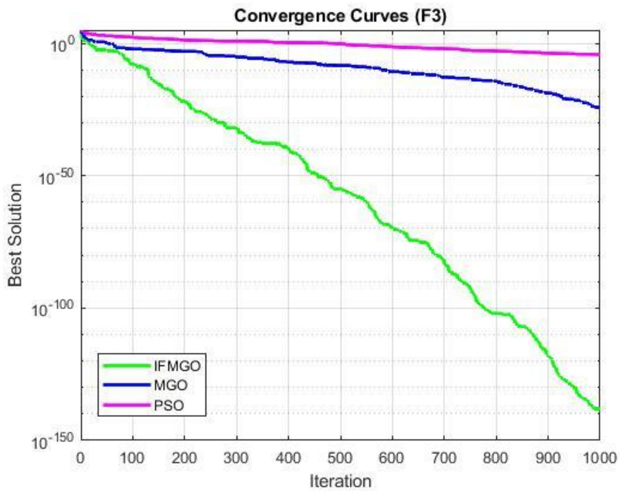
F19	Worst	**-3.8628E+00**	**-3.8628E+00**	**-3.8549E+00**
	Mean	**-3.8628E+00**	**-3.8628E+00**	**-3.8620E+00**
	STD	**1.8436E-15**	2.2584E-15	2.4049E-03
F20	Best	**-3.3220E+00**	**-3.3220E+00**	**-3.3220E+00**
	Worst	**-3.3220E+00**	**-3.3220E+00**	**-2.9564E+00**
	Mean	**-3.3220E+00**	**-3.3220E+00**	**-3.2389E+00**
	STD	**0.0000E+00**	6.0328E-02	1.0146E-01
F21	Best	**-1.01532E+1**	**-1.01532E+1**	**-1.01532E+1**
	Worst	**-1.01532E+1**	**-1.01532E+1**	-2.6305E+00
	Mean	**-1.01532E+1**	**-1.01532E+1**	-6.7321E+00
	STD	**0.0000E+00**	**0.0000E+00**	3.5630E+00
F22	Best	**-1.04029E+1**	**-1.04029E+1**	**-1.04029E+1**
	Worst	**-1.04029E+1**	**-1.04029E+1**	-2.7519E+00
	Mean	**-1.04029E+1**	**-1.04029E+1**	-6.7370E+00
	STD	**0.0000E+00**	**0.0000E+00**	3.5639 E+00
F23	Best	**-1.05364E+1**	**-1.05364E+1**	**-1.05364E+1**
	Worst	**-1.05364E+1**	**-1.05364E+1**	-2.4217E+00
	Mean	**-1.05364E+1**	**-1.05364E+1**	-7.2984E+00
	STD	**0.0000E+00**	**0.0000E+00**	3.7994E+00

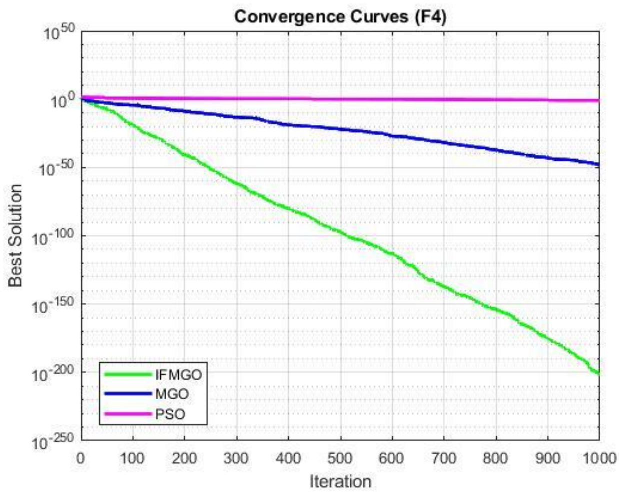
The IFMGO algorithm showed excellent performance by producing global optimum solutions for functions F14 to F23. Though the MGO algorithm produced very competitive results, the IFMGO algorithm produced its results on all the functions with a high level of robustness and stability through better values in mean, worst, and standard deviation. Hence, the proposed IFMGO algorithm outperformed the MGO algorithm and PSO algorithm.

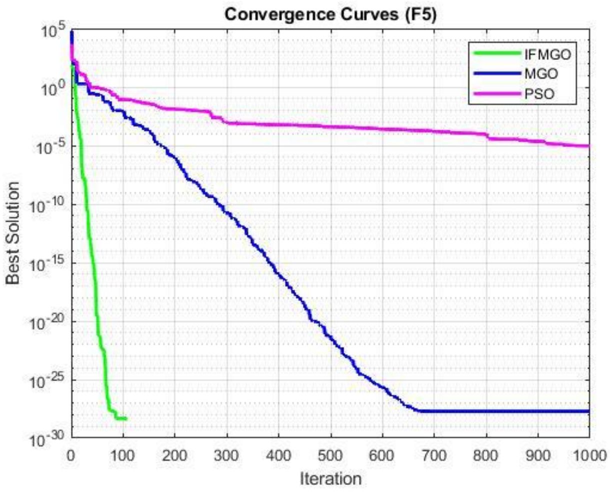
Additionally, a comparison of the convergence curves for the three (3) algorithms, IFMGO, MGO, and PSO, on the high-dimensional benchmark functions (F1 to F13) are presented below as figures F1 to F13. From the curves, it is obvious that the IFMGO algorithm has better convergence characteristics on all the test functions than the other algorithms. It exhibited fast convergence to the global solutions without getting trapped in local optimal solutions. This confirms the superiority of the proposed IFMGO algorithm in terms of convergence behavior relative to the other algorithms.

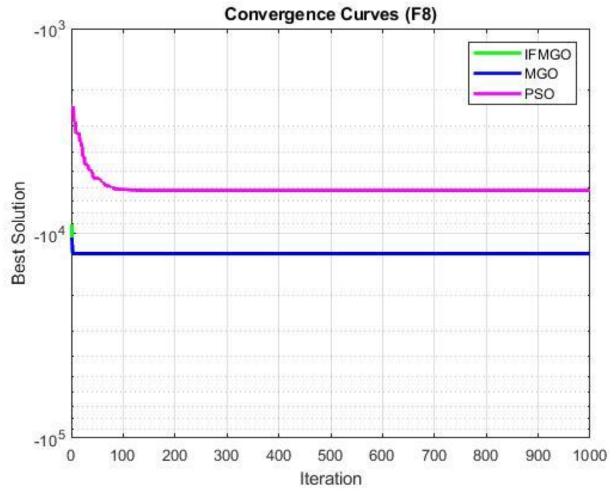


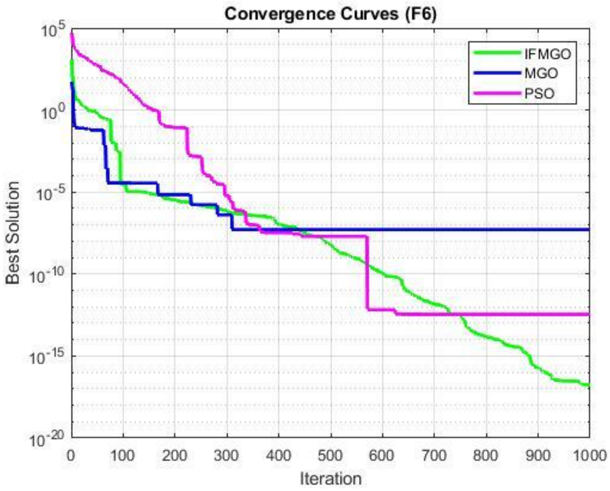


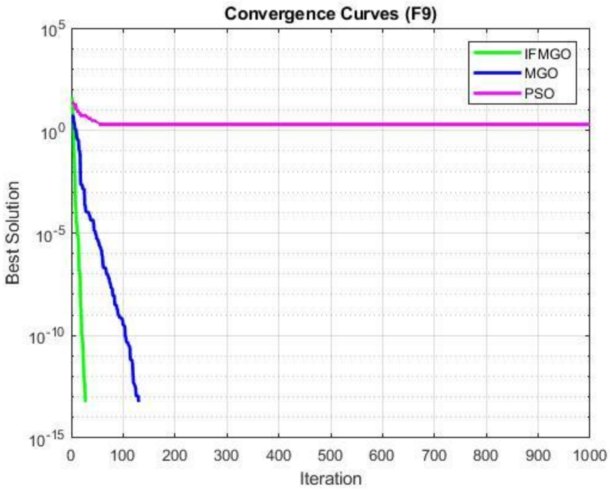


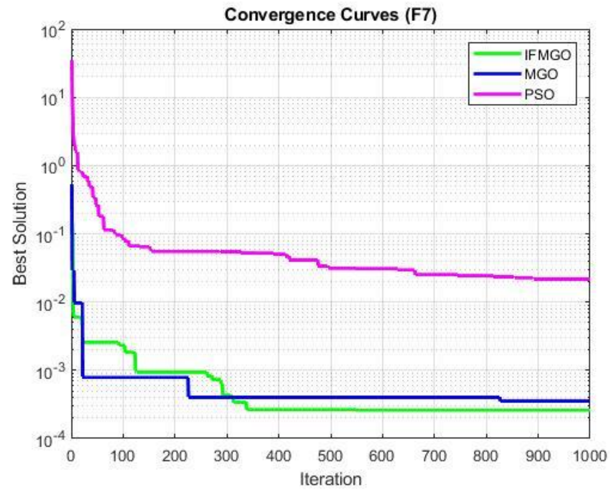


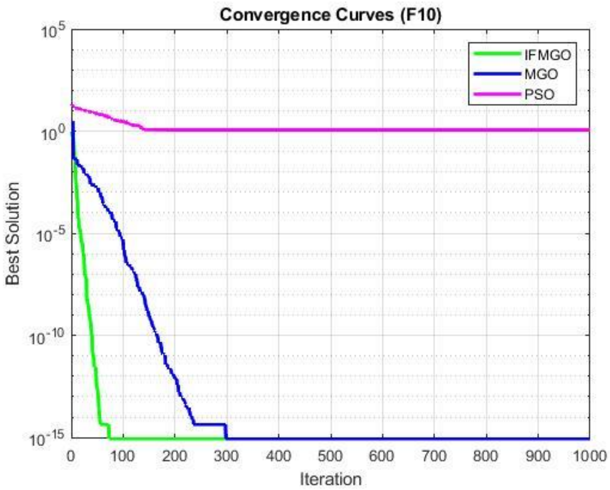

Convergence Curves (F5)
IFMGO
MGO
PSO
Best Solution
Iteration

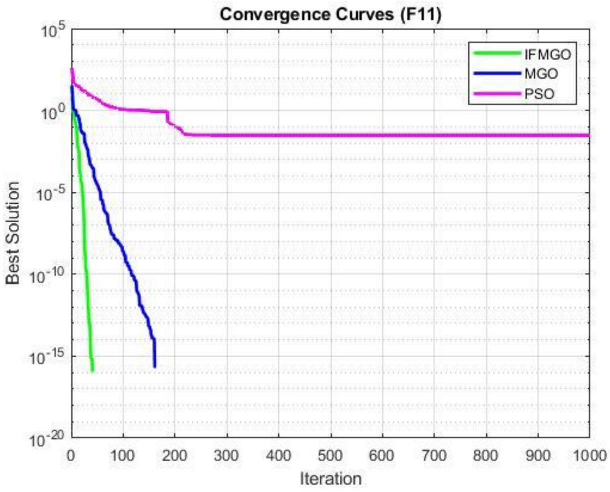

Convergence Curves (F8)
IFMGO
MGO
PSO
Best Solution
Iteration

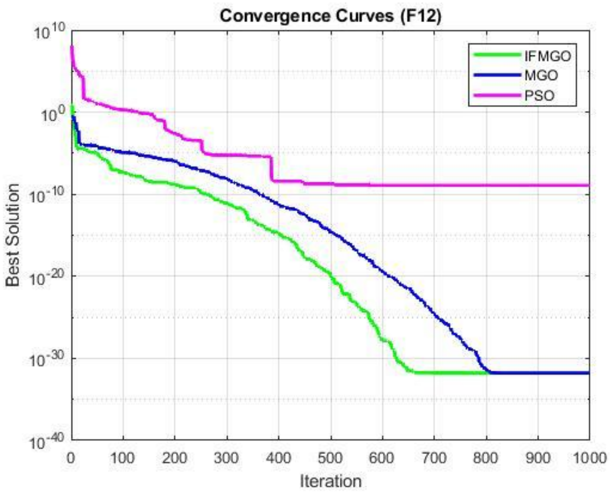

Convergence Curves (F6)
IFMGO
MGO
PSO
Best Solution
Iteration

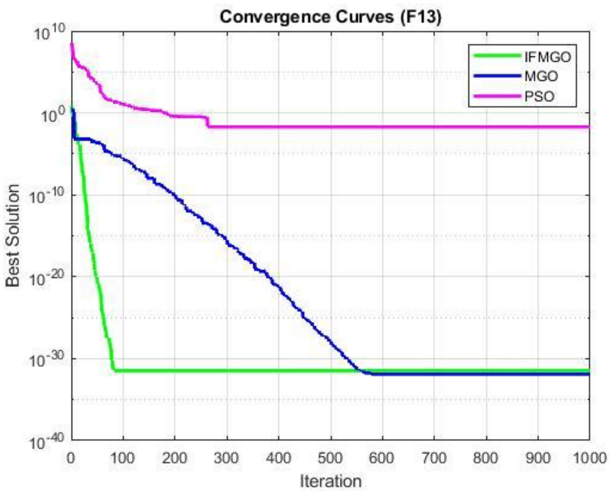

Convergence Curves (F9)
IFMGO
MGO
PSO
Best Solution
Iteration


Convergence Curves (F7)
IFMGO
MGO
PSO
Best Solution
Iteration


Convergence Curves (F10)
IFMGO
MGO
PSO
Best Solution
Iteration







6. CONCLUSION AND RECOMMENDATION

An improved version of the MGO algorithm called the Improved F-factor Mountain Gazelle Optimizer (IFMGO algorithm) is presented. A new calculation of the F-parameter in the MGO is proposed to enhance the algorithm's performance on high-dimensional problems without jeopardizing the performance on fixed-dimensional problems as well. The proposed IFMGO algorithm is tested on 13 standard high-dimensional benchmark functions and 10 standard fixed-dimensional benchmark functions. Performance comparison of the proposed IFMGO algorithm, MGO algorithm, and PSO algorithm is carried out and the IFMGO algorithm outperformed the other two algorithms exceptionally across all the test functions considered.

IFMGO is an algorithm with good qualities and is therefore recommended for application in real-life optimization problems in fields such as engineering, mathematics, health, and so on.

REFERENCES

[1] Y. Xiao, X. Sun, Y. Guo, S. Li, Y. Zhang, and Y. Wang, "An Improved Gorilla Troops Optimizer Based on Lens Opposition-Based Learning and Adaptive β-Hill Climbing for Global Optimization," *C. - Comput. Model. Eng. Sci.*, vol. 130, no. 3, 2022, doi: 10.32604/cmes.2022.019198.

[2] O. Findik, "Bull optimization algorithm based on genetic operators for continuous optimization problems," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 23, pp. 2225–2239, 2015, doi: 10.3906/elk-1307-123.

[3] C. Sakthi Gokul Rajan and K. Ravi, "Optimal placement and sizing of DSTATCOM using Ant lion optimization algorithm," *8th Int. Conf. Comput. Power, Energy, Inf. Commun. ICCPEIC 2019*, no. 7, pp. 31–34, 2019, doi: 10.1109/ICCPEIC45300.2019.9082382.

[4] B. Abdollahzadeh, F. S. Gharehchopogh, N. Khodadadi, and S. Mirjalili, "Mountain Gazelle Optimizer: A new Nature-inspired Metaheuristic Algorithm for Global Optimization Problems," *Adv. Eng. Softw.*, vol. 174, no. May, p. 103282, 2022, doi: 10.1016/j.advengsoft.2022.103282.

[5] A. S. Yussif, E. Twumasi, and E. A. Frimpong, "Modified Mountain Gazelle Optimizer Based on Logistic Chaotic Mapping and Truncation Selection," *IRJET*, vol. 10, no. May, pp. 1769–1776, 2023.

[6] P. Trojovský and M. Dehghani, "A new optimization algorithm based on mimicking the voting process for leader selection," *PeerJ Comput. Sci.*, vol. 8, pp. 1–40, 2022, doi: 10.7717/peerj-cs.976.

[7] C. Ouyang, D. Zhu, and Y. Qiu, "Lens Learning Sparrow Search Algorithm," *Math. Probl. Eng.*, vol. 2021, 2021, doi: 10.1155/2021/9935090.

[8] P. Trojovský and M. Dehghani, "Pelican Optimization Algorithm: A Novel Nature-Inspired Algorithm for Engineering Applications," *Sensors*, vol. 22, no. 3, 2022, doi: 10.3390/s22030855.

[9] M. F. F. Mohd Fadzil, "Tiki-taka algorithm: a novel metaheuristic inspired by football playing style," *Eng. Comput. (Swansea, Wales)*, vol. 38, no. 1, pp. 313–343, 2021, doi: 10.1108/EC-03-2020-0137.

[10] Y. Yuan, J. Ren, S. Wang, Z. Wang, X. Mu, and W. Zhao, "Alpine skiing optimization: A new bio-inspired optimization algorithm," *Advances in Engineering Software*, vol. 170. 2022, doi: 10.1016/j.advengsoft.2022.103158.