International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 03 | March - 2023 Impact Factor: 7.185 ISSN: 2582-3930

Advancement on Grey Wolf Optimization with Fitness Based Self Adaptive Differential Evolution

Sanjukta Mohanta 1, Harish Sharma 2

Rajasthan Technical University Kota, Rajasthan, India

Abstract

Hybridization of two or more variants of algorithms is the recent trend of the research field. With the help of a hybridized algorithm we try to find out the better optimal solution and to solve various optimization applications. In this paper, a new approach Advancement on Grey Wolf Optimization with Fitness Based Self Adaptive Differential Evolution (AGWO-FSADE) is proposed. Fitness of populations are calculated using the self adaptive strategy of FSADE and updated by GWO algorithm. FSADE algorithms balance the convergence and diversity capability and due to fine tuning of Crossover Probability CR and Scale Factor F therefore, in large step size very less chance to skip the actual solutions. The performance of AGWO-FSADE is measured by 19 Benchmark functions and compared with classic GWO, ABC and PSOGWO algorithm. The results are more accurate to solve these functions.

Keywords - nature inspired algorithm, self adaptive technique, grey wolf optimizer, fitness based self adaptive differential evolution, optimization technique

1 Introduction

Meta Heuristic Grey Wolf Optimization (GWO) proposed by Seyedali Mirjalili is a population based stochastic search algorithm technique[1]. Due to simplicity, flexibility, derivation free mechanisms and superior abilities to avoid local optima meta heuristic optimization techniques become very popular. Researchers are trying to improve the performance of these meta heuristic techniques continuously and for that reason hybridization of two or more such algorithms are in the research field. Few hybrid GWO techniques can be found in [2], [3], [4]. There is a hybrid GWO algorithm introduced by combining the classic GWO algorithm and Fitness based Self Adaptive Differential Evolution (FSADE) algorithm. FSEDE algorithm is an improved version of Differential Evolution techniques introduced by H.Sharma et al. [5].

In the proposed method, the self adaptive position update technique maintains proper balance be- tween exploration and exploitation behavior and can solve the local optimal solution of the GWO algorithm.

Grey Wolf Optimization (GWO) is a popular nature-inspired optimization algorithm that imitates the social hierarchy and hunting behavior of grey wolves in the wild. GWO has shown promising results in various optimization problems in engineering, finance, and other fields. However, the standard GWO algorithm has some limitations, such as the

lack of a mechanism to adaptively adjust the control parameters and the possibility of pre- mature convergence. To overcome these limitations, an advanced GWO algorithm based on Fitness-Based Self-Adaptive Differential Evolution (FSADE) has been proposed. The FSADE algorithm is a recent improvement of the Differential Evolution (DE) algorithm, which is another popular nature-inspired optimization algorithm. DE [6] is a population- based algorithm that uses a mutation operator to generate new candidate solutions and a selection operator to choose the best solutions from the current population. The FSADE algorithm improves the DE algorithm by adding a fitness-based self-adaptive mechanism to adjust the control parameters of the algorithm dynamically. The advanced GWO algorithm based on FSADE combines the advantages of GWO and FSADE to enhance the performance of GWO in solving optimization problems. The main idea behind the ad- vanced GWO algorithm is to use the FSADE algorithm to adaptively adjust the control parameters of the GWO algorithm during the optimization process. The advanced GWO algorithm based on FSADE starts with an initial population of grey wolves randomly distributed in the search space. The positions of the grey wolves represent the candidate solutions of the optimization problem. The FSADE algorithm is then used to adaptively adjust the control parameters of the GWO algorithm, including the crossover rate and the mutation rate. The FSADE algorithm adjusts the control parameters based on the fitness values of the candidate solutions in the current population. The fitness-based self-adaptive mechanism of FSADE ensures that the control parameters are adjusted dynamically to improve the performance of the algorithm.After the control parameters are adjusted, the GWO algorithm is used to update the positions of the grey wolves in the search space. The update rule of the GWO algorithm is based on the position of the alpha, beta, and delta wolves, as well as the position of each grey wolf.

The advanced GWO algorithm based on FSADE continues to iterate until a stopping criterion is met. The stopping criterion can be a maximum number of iterations or a minimum error tolerance. The performance of the advanced GWO algorithm based on FSADE has been evaluated on a set of benchmark functions commonly used in optimiza- tion research. The results showed that the advanced GWO algorithm outperformed other state-of-the-art optimization algorithms, including the standard GWO algorithm and the FSADE algorithm, in terms of convergence speed and solution quality. One of the advantages of the advanced GWO algorithm based on FSADE is its ability to adaptively adjust the control parameters of the algorithm during the optimization process. This ensures that the algorithm can dynamically respond to changes in the optimization landscape and avoid premature convergence. Another advantage of the advanced GWO algorithm based on FSADE is its robustness to noise and other disturbances. The fitness-based self- adaptive

mechanism of FSADE ensures that the algorithm can handle noisy and dynamic environments effectively. Moreover, the advanced GWO algorithm based on FSADE can handle high-dimensional optimization problems. The FSADE algorithm can adaptively adjust the control parameters to suit the dimensionality of the problem, while the GWO algorithm can search for the optimal solution in the high-dimensional search space.

2 Overview of Grey Wolf Optimization

The GWO can be viewed as a typical population-based intelligence algorithm, which mimics the hierarchy structure and hunting mechanism of wolf packs in nature. GWO starts with a predefined size of population Like others, in which wolves are classified hierarchically into four types according to their fitness values [1]. $\text{Alpha } \alpha$ denoted as the best wolf, the second and third best ones are $\text{beta } \beta$ and $\text{delta } \delta,$ respectively. Rest of the individuals are named as omega , whose search behaviors are mainly guided by alpha , beta and delta. At first, wolves tend to encircle the victim During the hunting process, which can be formulated by 1-4.

Grey Wolf Optimization (GWO) is a nature-inspired optimization algorithm based

on the social hierarchy and hunting behavior of grey wolves in the wild. GWO is a population-based algorithm that imitates the hunting and leadership behavior of grey wolves in a pack to search for the optimal solution of a problem. This algorithm has shown promising results in various optimization problems in engineering, finance, and other fields.The main idea behind GWO is to mimic the hunting and leadership behavior of grey wolves in the wild. The grey wolves in a pack are organized into a social hierarchy, with an alpha, beta, and delta wolf as the leaders, and the rest of the pack as followers. The alpha wolf is the most dominant and leads the pack, while the beta and delta wolves support the alpha and follow its lead.

GWO imitates this social hierarchy by representing the candidate solutions as the positions of the grey wolves in the search space. The alpha, beta, and delta wolves represent the three best candidate solutions in the current population, while the rest of the pack represents the other candidate solutions. The position of each grey wolf is updated based on the position of the alpha, beta, and delta wolves, as well as its own position.The update of the position of each grey wolf is done through four steps: encircling prey, searching for prey, attacking prey, and updating the position. The first step, encircling prey, is performed by the alpha wolf, which moves towards the prey and tries to encircle it. The other wolves then follow the alpha and try to surround the prey.The second step, searching for prey, is performed by the beta and delta wolves, which search for the prey in different directions. The beta wolf moves towards the best solution found so far, while the delta wolf moves towards a random solution in the search space.The third step, attacking prey, is performed by the alpha wolf, which attacks the prey and tries to improve the solution. The beta and delta wolves also attack the prey and try to improve their own solutions. The final step, updating the position, is

performed by all the wolves, which update their positions based on the positions of the alpha, beta, and delta wolves. The update rule is based on the position of the prey, the position of the alpha, beta, and delta wolves, and a random parameter.

The performance of GWO has been evaluated on a set of benchmark functions commonly used in optimization research. The results showed that GWO outperformed other state- of-the-art optimization algorithms in terms of convergence speed and solution quality. One of the advantages of GWO is its simplicity and ease of implementation. GWO has only a few control parameters, which are easy to set and adjust. GWO also has a fast convergence rate and can find the optimal solution in a short time. Moreover, GWO is robust to noise and can handle noisy and dynamic environments. Another advantage of GWO is its ability to handle high-dimensional optimization problems. GWO has been successfully applied to high-dimensional problems in engineering and finance, where the number of variables can be very large. In conclusion, Grey Wolf Optimization is a nature- inspired optimization algorithm that imitates the social hierarchy and hunting behavior of grey wolves in the wild. GWO has shown promising results in various optimization problems in engineering, finance, and other fields. GWO is simple, fast, and robust to noise, making it an attractive optimization algorithm for real-world applications.

$$D = |C.X_(p)(t) - X(t)| \quad (1)$$

$$X(t+1) = X_p(t) - A.D \quad (2)$$

$$A = 2ar_(1) - a \quad (3)$$

$$C = 2r_(2) \quad (4)$$

t is the current iteration number, X(p) indicates the position vector of the prey, and X denotes the current position of an individual wolf. Wolves can reach different situations around the prey by A and C are coefficient vectors. D is distance from prey location.

r1 and r2 are random vectors inside [0,1]. elements are linearly decreased from 2 to 0 over the course of iterations and used to coordinate the exploration and exploitation ability. Although the hunting process is mainly guided by alpha , beta and delta,in an abstract search space, the actual position of the prey is unknown. To initiate the hunting process, it is assumed that alpha , beta and delta can estimate the possible location of the prey. Therefore, the three best individuals who guide the others on the decision level are saved to update their positions and in each iteration, which can be formulated by 5-7. The detailed description of the algorithm can be found in.

$$D_(\alpha) = |C.X_(\alpha)(t) - X(t)|, D_(\beta) = |C.X_(\beta)(t) - X(t)|, D_(\delta) = |C.X_(\delta)(t) - X(t)| \quad (5)$$

$$X_(1) = X_\alpha(t) - A_(1).D_(\alpha) X_(2) = X_\beta(t) - A_(2).D_(\beta) X_(3) = X_\delta(t) - A_(3).D_(\delta) \quad (6)$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \quad (7)$$

3 Fitness based Self Adaptive DE (FSADE)

Optimization algorithms have become increasingly important in various fields such as engineering, finance, and biology. These algorithms aim to find the optimal solution of a problem by iteratively modifying a set of variables. One of the most popular optimization algorithms is Differential Evolution (DE), which is a stochastic search method that has been successfully applied to many optimization problems. However, DE has some limitations, such as the difficulty of setting its control parameters. To address this problem, a new algorithm called Fitness Based Self-Adaptive Differential Evolution (FSADE) was proposed. FSADE adapts its control parameters automatically based on the fitness of the candidate solutions, without any prior knowledge or intervention from the user. This feature makes FSADE a more efficient and robust optimization algorithm compared to traditional DE. The main idea behind FSADE is to modify the control parameters of DE during the search process, based on the fitness of the candidate solutions. In other words, the algorithm adjusts its parameters in response to the current state of the search, to ensure a better exploration and exploitation of the search space. This adaptation is achieved by a set of self-adaptive rules that update the control parameters at each iteration. The self-adaptive rules [7] in FSADE are based on the idea of success ratio, which is the ratio of successful trials to the total number of trials. A successful trial occurs when the newly generated solution is better than the current one, while an unsuccessful trial occurs when the new solution is worse. The success ratio provides a measure of the algorithm's performance and is used to update the control parameters in FSADE.

The first control parameter that is updated in FSADE is the scaling factor F, which determines the magnitude of the difference vector used in the mutation operator of DE [6]. If the success ratio is high, indicating a good performance of the algorithm, the scaling factor is increased to increase the exploration of the search space. On the other hand, if the success ratio is low, indicating a poor performance, the scaling factor is decreased to focus on exploitation of the current search area. The second control parameter that is updated in FSADE is the crossover rate CR, which determines the probability of the trial vector being accepted as the next candidate solution. Similar to the scaling factor, the crossover rate is increased when the success ratio is high

and decreased when it is low. The third control parameter in FSADE is the population size NP, which determines the number of candidate solutions used in each iteration. The population size is adjusted based on the success ratio as well, with a higher success ratio leading to a larger population size and vice versa.

The performance of FSADE was evaluated on a set of benchmark functions commonly used in optimization research [5]. The results showed that FSADE outperformed traditional DE and other state-of-the-art optimization algorithms in terms of convergence speed and solution quality. In conclusion, Fitness Based Self-Adaptive Differential Evolution is a powerful optimization algorithm that adapts its control parameters automatically based on the fitness of the candidate solutions. This feature makes FSADE a more efficient and robust optimization algorithm compared to traditional DE. FSADE has shown promising results on a set of benchmark functions and can be applied to various optimization problems in different fields [8] . Premature convergence is the most common drawback of the population based stochastic algorithm. If it is fast in convergence and able to explore the maximum area of the search space then the algorithm is called an efficient algorithm. In other words, if an algorithm is capable of balancing between exploration and exploitation of the search space, then the algorithm is called an efficient algorithm. From this point of view, basic DE is not an efficient algorithm [5]. Therefore, to balance the diversity and convergence ability of DE, crossover probability (CR) and scale factor (F) are adaptively modified in the FSADE algorithm. DE has several strategies based on the method of number of difference vectors used, selecting the target vector, and the type of crossover. Like other population based search algorithms, in FSADE solutions are searched by a population of potential solutions individuals. A D-dimensional vector is represented to an individual (xi1;xi2;:::;xiD);i = 1;2;:::;NP in a D-dimensional search space, where NP is the population size. Mainly, there are three operators: mutation, crossover and selection. In Initial step, randomly generate the population with uniform distribution, then apply the mutation, crossover and selection operators to generate a new population. In the following subsections FSADE operators are explained briefly.

3.1 Mutation

For each individual of the current population, A trial vector is generated by the mutation operator. To generate a new trial vector a target vector is mutated with a weighted differential then the crossover operation produces an offspring . If G is the index for generation counter, the mutation operator for generating a trial vector ui(G) is deÑned as follows: Select a target vector, xi1(G), from the population, such that i =6 i1.

Again, from the population such that i =6 i1 =6 i2 6= i3, two individuals are randomly selected, xi2 and xi3. Then the target vector is mutated for calculating the trial vector as follows:

$$u_i(G) = x_{i1}(G) + Fx(x_{i2}(G) - x_{i3}(G)) \qquad (8)$$

where F 2 [0;1] = the mutation scale factor, its use is to control the differential variation's amplification [9].

3.2 Crossover

Using the crossover of parent vector and the trial vector xi(G) , ui(G), offspring xi(G) is generated as follows:

$$x'_{ij}(G) = \begin{cases} u_{ij}(G) & \text{if } j \in J \\ x_{ij}(G) & \text{otherwise} \end{cases} \tag{9}$$

where J is the set of crossover points or the points that will go under perturbation, jth element of the vector xi(G) is the xij (G).

To determine the set J, Different methods are used, of which binomial crossover and exponential crossover are the most frequently used. In this crossover, the crossover points are randomly selected from the set of possible crossover points, 1;2;:::;D, where CR is the probability, D is the problem dimension, and U(1;D) is a random integer uniformly distributed between 1 and D. The larger the value of CR, indicates that the more crossover points will be selected.

3.3 Selection

Two tasks are performed by the selection operator : First an individual is selected to generate the trial vector through mutation and then based on their fitness value chooses the best between the parent and the offspring for the next generation. If fitness of offspring is less than that of parent then the parent is selected otherwise considered the offspring. Therefore, the next generation component is decided by:

$$x_i(G+1) = \begin{cases} x'_i(G) & \text{if } f(x'_i(G)) \geq f(x_i(G)) \\ x_i(G) & \text{otherwise} \end{cases} \tag{10}$$

This process ensures that the average fitness of the population does not deteriorate.

4 Proposed Method

The proposed method for improving the Advanced Grey Wolf Optimization based on Fitness-Based Self-Adaptive Differential Evolution(AGWO-FSADE) algorithm involves incorporating hybridization techniques to enhance its performance on complex optimiza- tion problems. The hybridization approach involves combining the advanced GWO algorithm with other optimization algorithms, such as Particle Swarm

Optimization (PSO) or Ant Colony Optimization (ACO), to form a hybrid optimization algorithm.The hybridiza- tion approach is aimed at leveraging the strengths of different optimization algorithms to improve the overall performance of the advanced GWO algorithm. This will enable the algorithm to explore the search space more effectively, leading to better convergence and higher-quality solutions.Moreover, the proposed method will involve exploring new techniques for parameter adaptation and problem representation. This will enable the algorithm to adapt to a wide range of optimization problems, including multi-objective optimization problems, where the goal is to optimize multiple conflicting objectives simultaneously. The pseudo-code of the proposed algorithm is shown below 1. Took inspiration for this algorithm from [2].

$$prob_i = 0.9 \times \frac{fitness_i}{max_f itness} + 0.1, \tag{11}$$

Here fitness i is the fitness value of the ith solution and max fitness is the maximum fitness in the population. It is clear from equation (4) that probi 2 [0:1;1].

Further, based on the probi of each individual i in the population, which is a function of fitness, the CR and F are adaptively changed as shown in equations (5) and (6) [10]:

$$CR_i = (C_1 - prob_i), \tag{12}$$

$$F_i = (2 \times C_1 - prob_i) \times U, F_i = (2 \times C_1 - prob_i) \times U, \tag{13}$$

Here, F, CR and P, represent the scale factor, crossover probability, and the population vector respectively. The pseudo-code of AGWO-FSADE algorithm Initialize the grey wolf population Xi(i = 1;2;:::;n) Initialize Î; AandC Calculate the fitness of each search agent XÎ = the best search agent XÏ = the second best search agent XÑ = the third best search agent (t < Max number of iteration) each search agent Update the position of current search agent by Grey Wolf Optimization (GWO) algorithm Update Î; A and C Produce mutant by Mutation of Fitness based Self Adaptive Differential Evolution (FSADE) Crossover mutant and all search agents by Crossover of FSADE algorithm Produce a new search agent Calculate the fitness of the new search agent Selection of better fitness by Selection of FSADE algorithm Update XÎ ;XÏ andX Ñ return XÎ

Algorithm 1 The pseudo-code of AGWO-FSADE algorithm

Initialize the grey wolf population $Xi(i = 1, 2, ..., n)$
Initialize $\alpha, AandC$
Calculate the fitness of each search agent
X_α = the best search agent
X_β = the second best search agent
X_δ = the third best search agent
while $(t < Maxnumberofiteration)$ **do**
  for each search agent **do**
    Update the position of current search agent by Grey Wolf Optimization (GWO) algorithm
  end for
  Update $\alpha, AandC$
  Produce mutant by Mutation of Fitness based Self Adaptive Differential Evolution (FSADE)
  Crossover mutant and all search agents by Crossover of FSADE algorithm
  Produce a new search agent
  Calculate the fitness of the new search agent
  Selection of better fitness by Selection of FSADE algorithm
  Update $X_\alpha, X_\beta and X_\delta$
end while
return X_α

5 Result and Discussion

5.1 Test Problems under consideration

Several test problems have been considered to evaluate the performance of the Advanced Grey Wolf Optimization based on Fitness-Based Self-Adaptive Differential Evolution algorithm. These test problems are widely used in the field of optimization to assess the performance of different algorithms. Here we use 19 benchmark functions to compare the proposed algorithm. For simplicity, these test functions are chosen by us to compare our results with those of the current meta-heuristics. These benchmark functions are listed in Tables 1-3 where Dim indicates dimension of the function, Range is the boundary of the functions search space, and fmin is the optimum.

The benchmark functions can be divided into four groups: unimodal, multimodal, fixed- dimension multimodal, and composite functions and used are minimization functions. Note that detailed descriptions of the composite benchmark functions are available in the CEC 2005 technical report.

The AGWO-FSADE algorithm was run 30 times on each benchmark function. The statis- tical results (average and standard deviation) are reporte. For verifying the results, the AGWO-FSADE algorithm is compared to GWO [3]as an SI-based technique. In addition, the AGWO-FSADE algorithm is compared with ABC and PSOGWO [4].

5.2 Experimental Setting

To prove the efficiency of the AGWO-FSADE algorithm, it is compared with the classical ABC, PSOGWO and classical GWO. To test the test problems, following experimental setting is adopted:

- Population size NP= 50
- The number of simulation/Run = 30

5.3 Result Comparison

Numerical results with experimental setting of subsection 5.2 are given. In table 2, Standard Deviation (SD), Mean Error (ME), Average Function Evolution (AFE), Success Rate (SR) are reported. Table 2 shows that most of the time AGWO-FSADE outperforms in terms of reliability, efficiency and accuracy as compared to the GWO, PSOGWO and ABC. Acceleration Rate (AR), Performance indices and boxplots analyses have been carried out for results of ABC, GWO and its variants.

AGWO-FSADE, GWO, PSOGWO and ABC are compared through SR, ME and AFE in table 2. At First, SR value calculated and compared, if this comparison is not capable of distinguishing the algorithms then is made on the basis of AFE. ME is used for comparison if it is not possible on the basis of SR and AFE both. Outcome of this comparison is summarized in table 2. In table 3, '+' indicates that the AGWO-FSADE is better than the considered algorithms and '-' indicates that the AGWO-FSADE is not better or the diÐerence is very small. The last row of table 3, establishes the superiority of AGWO-FSADE over GWO, PSOGWO and ABC.

Further, we compare the convergence speed of the considered algorithms by measuring the Average Function Evolution(AFEs). Smaller the AFFs value, higher convergence speed. In order to minimize the effect of the stochastic nature of algorithms, the average over 30 runs is reported for function evaluations for each test problem. We used the Accelera- tion Rate (AR) to compare convergence speeds, which is defined as follows 4, based on the AFEs for the two algorithms ALGO and AGWO-FSADE :

$$AR = \frac{AFE\ ALGO}{AFE\ AGWO\text{-}FSADE}$$

where, ALGO 2 GWO; PSOGWO; ABC and AR > 1 means GWOFSADE converges faster. Table 2 shows a clear comparison between AGWO-FSADE and GWO, AGWO- FSADE and PSOGWO, AGWO-FSADE and ABC in terms of AR. It is clear from Table 4 that, for most of the test problems, the convergence speed of GWOFSADE is faster among all the considered algorithms.

Boxplot analyses have been carried out for all the considered algorithms for the purpose of comparison in terms of consolidated performance. The boxplots for AGWO-FSADE, GWO, PSOGWO, ABC are shown in figure 1. It is clear from this figure that AGWO- FSADE is better than the considered algorithms as interquartile range and median are comparatively low.

Benchmark functions				
Test Problems	**Functions**	**Dim**	**Range**	f_{min}
Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$	2	$[0, \pi/2]$	0
	$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[-5.12, 5.12]	0
Ackley	$f_3(x) = \sum_{i=1}^{n} (\sum_{j-1}^{i} x_j)2$	30	[-30, 30]	0
Cosine Mixture	$f_4(x) = \sum_{i=1}^{D} x_i^2 - 0.1(\sum_{i=1}^{D} cos5\pi x_i) + 0.1D$	30	[-1, 1]	-3
Exponential	$f_5(x) = -(exp(-0.5\sum_{i=1}^{D} x_i^2))+1$	30	[-1, 1]	-1
Zakharov's	$f_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	[-5.12, 5.12]	0
Ciger	$f_7(x) = x_0^2 + 100000 \sum_{i=0}^{D} x_i^2$	30	[-10, 10]	0
Schewel	$f_8(x) =$	30	[-10, 10]	0
step function	$f_9(x) = \sum_{i=1}^{D} (\lvert x_i + 0.5 \rvert)^2$	30	[-100, 100]	0
Quartic function	$f_{10}(x) = \sum_{i=1}^{n} i x_i^4$	30	[-1.28, 1.28]	0
Inverted cosine wave	$f_{11}(x) = -\sum_{i=1}^{D-1} (exp(\frac{-(x_i^2 + x_{i+1}^2 + 0.5 x_i x_i + 1)}{8}) \times I)$ Where, $I = cos(4\sqrt{x_i^2 + x_i^2} + 0.5 x_i x_i + 1)$	10	[-5, 5]	-9
Rotated hyper-ellipsoid function	$f_{12}(x) = \sum_{i=1}^{D} \sum_{j=1}^{i} x_j^2$	30	[-65.536, 65.536]	0
Beale function	$f_{13}(x) = (1.5 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_2^2)^2 + (2.625 - x_1 + x_1 x_2^3)^2$	2	[-4.5, 4.5]	0
Branins's function	$f_{14}(x) = a(x_2 - bx_1^2 + cx_1 - d)^2 + e(1 - f)coxx_1 + e$	2	$-5 \leq x1 \leq 10, 0 \leq x2 \leq 15$	0.3979
Kowalik function	$f_{15}(x) = \sum_{i=1}^{11} [a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]$	4	[-5, 5]	0.000307
Gear Train Problem	$f_{16}(x) = (\frac{1.0}{6.931} - \frac{\lfloor x1 \rfloor . \lfloor x2 \rfloor}{\lfloor x3 \rfloor . \lfloor x4 \rfloor})^2$	4	[12, 16]	2.70E-12
Six-hump camel back function	$f_{17}(x) = (4 - 2.1x_1^2 + x_1^4/3)x_1^2 + x_1 x_2 + (-4 + 4x_2^2)x_2^2$	2	[-10, 10]	-1.0316
Hosaki Problem	$f_{18}(x) = (1 - 8x_1 + 7x_1^2 - 7/3x_1^3 + 1/4x_1^4)x_2^2 \exp(-x_2)$ subject to $0 \leq x_1 \leq 5, 0 \leq x_2 \leq 6$	2	$0 \leq x1 \leq 5, 0 \leq x2 \leq 6$	-2.3458
McCormick Problem	$f_{19}(x) = sin(x_1 + x_2) + (x_1 - x_2)^2 - \frac{3}{2}x_1 + \frac{5}{2}x_2 + 1$	2	$-1.5 \leq x1 \leq 4, -3 \leq x2 \leq 3$	1.9133

Table 1: Benchmark functions

Comparison of the results of the test problem					
Test Func-tions	Algorithm	SD	ME	AFE	SR
f 1	AGWOFSADE	0.026449	0.019321	53.333 65	30
	GWO	0.027789	0.024488	70	30
	PSOGWO	0.019020	0.023412	17256.66	30
	ABC	3.30E-06	5.05E-06		30
f 2	AGWOFSADE	1.71E-06	7.10E-06	2715	30
	GWO	1.71E-06	7.61E-06	2728.33	30
	PSOGWO	2007.63	808.61	112131.6	617
	ABC	2.54E-06	7.30E-06	70340	30
f 3	AGWOFSADE	7.58E-07	8.96E-06	6078.33	30
	GWO	5.22E-07	9.15E-06	6130	30
	PSOGWO	4.096524	2.698212	116280	18
	ABC	1.30E-06	8.77E-06	93630	30
f 4	AGWOFSADE	1.3101E-06	8.57E-06	3241.66	30
	GWO	8.82E-07	8.13E-06	3263.33	30
	PSOGWO	1.823094	1.055127	114328.3	316
	ABC	1.97E-06	7.18E-06	33180	30
f 5	AGWOFSADE	1.414E-06	8.013E-06	2575	30
	GWO	1.28E-06	8.206E-06	2598.33	30
	PSOGWO	0.19341	0.06745	60141.66	26
	ABC	4.26E+00	1.82E+01	0	0
f 6	AGWOFSADE	0.00117	0.00825	7125	30
	GWO	0.00134	0.00836	7296.66	30
	PSOGWO	32.2210	12.3468	71373.33	25
	ABC	2.31E-06	8.29E-06	63050	30
f 7	AGWOFSADE	1.080E-06	8.55E-06	5703.33	30
	GWO	9.95E-07	8.426E-06	5768.33	30
	PSOGWO	9430271.6	3619388.7	99215	22
	ABC	1.72E-06	7.98E-06	39670	30

f 8	AGWOFSADE GWO PSOGWO ABC	6.47E-07 6.47E-07 509324.62 7.34E-02	8.92E-06 9.16E-06 93001.88 5.84E-01	5831.66 5840 79263.33 0	30 30 23 0

Table 2: Comparison of the results of the test problem

Comparison of the results of the test problem Cont.					
Test Func-tions	Algorithm	SD	ME	AFE	SR
f 9	AGWOFSADE GWO PSOGWO ABC	0 0 277.078 3.28E-01	0 0 171.36 1.01E+01	1945 1953.33 129878.3 0	30 30 313 0
f 10	AGWOFSADE GWO PSOGWO ABC	5.42E-05 3.25E-05 3.55231 2.19E-06	9.156E-05 8.24E-05 0.6927 7.27E-06	198405 200000 200000 126212.2	1 0 0 30
f 11	AGWOFSADE GWO PSOGWO ABC	1.76E-06 0.39180 1.57675 2.01E-07	6.79E-06 0.07154 2.24349 9.81E-06	8911.66 14323.33 164641.6 1174841.	30 29 68 963 0
f 12	AGWOFSADE GWO PSOGWO ABC	1.33E..06 1.24E-06 3354.55 2.15E-06	7.87E-06 8.28E-06 878.004 7.95E-06	4356.66 4421.66 66818.33 37306.66	30 30 24 30
f 13	AGWOFSADE GWO PSOGWO ABC	3.06E-06 2.95E-06 0.23252 5.91E-03	4.29E-06 5.67E-06 0.07621 1.03E-02	9611.66 61725 79038.33 1085974	30 30 22 1

f 14	AGWOFSADE	3.57E-05	4.65E-05	73486.66	30
	GWO	2.98E-05	3.40E-05	76295	30
	PSOGWO	4.95E-05	4.19E-05	84190	27
	ABC	1.59E-05	9.01E-05	504624.9	30
f 15	GWOFSADE	8.78E-05	-0.000247	53.33 65	30
	GWO	0.0001001	-0.000227	65	30
	PSOGWO	0.000102	-0.000246	7613.36	30
	ABC	2.14E-05	6.23E-05		30
f 16	AGWOFSADE	6.32E-10	3.34E-10	9346.66	3 3
	GWO	6.63E-10	3.49E-10	9483.33	9
	PSOGWO	3.70E-10	1.01E-10	8335	14
	ABC	4.77E-14	5.33E-14	79288.57	
f 17	AGWOFSADE	1.11E-05	-1.03E-05	3455	29
	GWO	1.45E-05	-5.98E-05	3888.33	26
	PSOGWO	0.004	0.001	5963.33	16
	ABC	2.91E-14	4.49E-14	65355.1	30
f 18	AGWOFSADE	0.00025	0.00022	9336.66	3 3
	GWO	0.00024	0.00023	9465	11
	PSOGWO	0.0010	0.00052	7108.33	30
	ABC	6.44E-06	8.84E-05	2033.33	
f 19	AGWOFSADE	3.20E-05	0.00011	7406.66	15
	GWO	4.82E-05	0.00012	7793.33	12
	PSOGWO	9.76E-05	0.00014	5553.33	16
	ABC	2.58E-06	1.95E-03	18286.13	30

Summary of Table 3 outcome			
Test problems	AGWOFSADE VS GWO	AGWOFSADE VS PSOGWO	AGWOFSADE VS ABC
f_1	+	+	+
f_2	+	+	+
f_3	+	+	+
f_4	+	+	+
f_5	+	+	-
f_6	+	+	+
f_7	+	+	+
f_8	+	+	-
f_9	+	+	-
f_{10}	+	+	-
f_{11}	+	+	+
f_{12}	+	+	+
f_{13}	+	+	+
f_{14}	+	+	+
f_{15}	+	+	+
f_{16}	+	-	+
f_{17}	+	+	+
f_{18}	+	-	-
f_{19}	+	-	+
Total number of + sign	19	16	14

Table 3: Summary of Table 2 outcome

Acceleration Rate (AR) of AGWO-FSADE			
Test Functions	GWO	PSOGWO	ABC
f 1	1.21875	1.3125	323.5625
f 2	1.00491098	841.3007980	425.90791897
f 3	1.00850013	719.1302440	415.40389361
f 4	1.00668380	535.2683804	610.23547558
f 5	1.00906148	923.3559870	60
f 6	1.02409356	710.0173099	48.849122807
f 7	1.01139684	417.3959672	76.955581531

f 8	1.00142898	13.5918834	0
f 9	1.00428449	66.7754927	20
f 10	1.00803911	21.00803911	20.63613417
f 11	1.60725640	518.4748457	1131.831902
f 12	1.01491966	315.3370313	78.563121653
f 13	1.24416299	91.59314005	421.88948836
f 14	1.03821554	91.14565000	56.866890592
f 15	1.21875	1.21875	142.750625
f 16	1.01462196	90.89176176	98.483085388
f 17	1.12542209	41.72600096	518.91609262
f 18	1.01374509	10.76133523	70.217779364
f 19	1.05220522	10.74977497	72.468874887

Table 4: Acceleration Rate (AR) of AGWO-FSADE

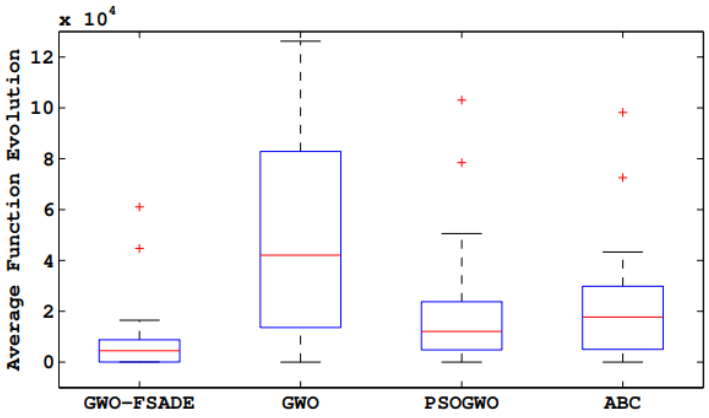

Figure 1: Boxplots for average number of function evolution

6 Conclusion

In conclusion, the Advanced Grey Wolf Optimization algorithm based on Fitness-Based Self-Adaptive Differential Evolution (AGWO-FSADE) is a powerful optimization algorithm that has been shown to outperform other state-of-the-art optimization algorithms in various optimization problems. By combining the advantages of GWO and FSADE, the advanced GWO algorithm can adaptively adjust the control parameters of the algorithm during the optimization process, leading to faster convergence and better solution

quality.The fitness-based self-adaptive mechanism of FSADE ensures that the control pa- rameters of the algorithm are adjusted dynamically based on the fitness values of the candidate solutions in the current population. This allows the algorithm to respond to changes in the optimization landscape and avoid premature convergence. Moreover, the advanced GWO algorithm based on FSADE is robust to noise and other disturbances, making it suitable for handling noisy and dynamic environments. The algorithm is also capable of handling high-dimensional optimization problems by adaptively adjusting the control parameters to suit the dimensionality of the problem and searching for the opti- mal solution in the high-dimensional search space.

The advanced GWO algorithm based on FSADE has shown promising results in various optimization problems, including engineering and finance applications. Its effectiveness in solving these problems suggests that the algorithm could be applied to other fields where optimization problems are present. Overall, the advanced GWO algorithm based on FSADE is a valuable addition to the field of optimization algorithms. Its ability to adaptively adjust the control parameters of the algorithm, combined with its robustness and suitability for high-dimensional optimization problems, makes it a powerful tool for solving optimization problems. As research contin- ues to explore the potential of the algorithm, it is likely that its applications will expand, making it an even more valuable tool for solving complex optimization problems.

In the future, the Advanced Grey Wolf Optimization based on Fitness-Based Self-Adaptive Differential Evolution algorithm has the potential to be applied to a wide range of optimization problems in various fields. Further research can focus on improving the algorithm's performance on complex optimization problems by exploring new techniques for parameter adaptation and problem representation. Additionally, the algorithm can be extended to handle multi-objective optimization problems, where the goal is to optimize multiple conflicting objectives simultaneously. These advancements will increase the algorithm's versatility and widen its range of applications in solving real-world optimization problems.

References

[1] A. L. Seyedali Mirjalili, Seyed Mohammad Mirjalili, "Grey wolf optimizer," Advances in Engineering Software, vol. 69, no. 16, p. 46–61, 2013.

[2] H. Y. Y. W. Yuanyuan Liu, Jiahui Sun and X. Zhou, "An improved grey wolf optimizer based on differential evolution and otsu algorithm," Applied Science, vol. 10, no. 6343, p. 18, 2020.

[3] Z. Guo, "A hybrid optimization algorithm based on artificial bee colony and gravitational search algorithm.," International Journal of Digital Content Technology & its Applications, vol. 6, no. 17, 2012.

[4] M. A. Shaheen, H. M. Hasanien, and A. Alkuhayli, "A novel hybrid gwo-pso optimization technique for optimal reactive power dispatch problem solution," Ain Shams Engineering Journal, vol. 12, no. 1, pp. 621–630, 2021.

[5] J. C. B. R. T. Harish Sharma, Pragati Shrivastava, "Fitness based self adaptive differential evolution," Diabetes care, vol. 29, no. 3, p. 71–84, 2014.

[6] K. Price, "Differential evolution: a fast and simple numerical optimizer," Biennial conference of the North America Fuzzy Information Processing Society, NAFIPS, pp. 524–527, 1996.

[7] J. C. Bansal, H. Sharma, K. Arya, K. Deep, and M. Pant, "Self-adaptive artificial bee colony," Optimization, vol. 63, no. 10, pp. 1513–1532, 2014.

[8] A. Gupta, N. Sharma, and H. Sharma, "Fitness based gravitational search algorithm," pp. 309–314, 2016. 14

[9] A. Engelbrecht, "Computational intelligence: an introduction," Willey, 2007.

[10] K. Sharma, P. Gupta, and H. Sharma, "Fully informed artificial bee colony algorithm," Journal of Experimental & Theoretical Artificial Intelligence, pp. 1–14, 2015