

A Hybrid Algorithm Based on Firefly Algorithm and Differential Evolution for Global Optimization

S. Sarbazfard

Department of Mathematics
Urmia Branch, Islamic Azad University
Urmia, Iran

A. Jafarian

Department of Mathematics
Urmia Branch, Islamic Azad University
Urmia, Iran

Abstract—In this paper, a new and an effective combination of two metaheuristic algorithms, namely Firefly Algorithm and the Differential evolution, has been proposed. This hybridization called as HFADE, consists of two phases of Differential Evolution (DE) and Firefly Algorithm (FA). Firefly algorithm is the nature-inspired algorithm which has its roots in the light intensity attraction process of firefly in the nature. Differential evolution is an Evolutionary Algorithm that uses the evolutionary operators like selection, recombination and mutation. FA and DE together are effective and powerful algorithms but FA algorithm depends on random directions for search which led into retardation in finding the best solution and DE needs more iteration to find proper solution. As a result, this proposed method has been designed to cover each algorithm deficiencies so as to make them more suitable for optimization in real world domain. To obtain the required results, the experiment on a set of benchmark functions was performed and findings showed that HFADE is a more preferable and effective method in solving the high-dimensional functions.

Keywords—Differential Evolution; Firefly Algorithm; Global optimization; Hybrid algorithm

I. INTRODUCTION

One of the affective methods in finding the best solution in numerical problems is the Optimization technique. In optimization, only a few solutions are considered the best which are called as the goal. Classical optimization techniques have some deficiencies on solving the complex optimization problems. These deficiencies are primarily interdependent on their inherent search systems. These classical optimization methods are strongly under effects of choosing proper objectives, constraints functions and type of variables. They also do not grant a universal result approach that can be used to solve problems where various type of variables, objective and constraint functions, are used [1]. For covering these deficiencies, a new method with the name of metaheuristic was designed, which is mainly originated from artificial intelligence research that developed by researchers [2]. A metaheuristic is an algorithm designed for solving the various types of hard optimization problems without having to fully accommodate to each problem. The Greek word meta indicates that these methods are higher-level heuristics. The primary features of metaheuristic methods are as follows: they are nature-inspired (meaning that they have originated from nature physics, behavior and etc); stochastic components are one of the inseparable parts of these methods (involving random variables); they are not gradient base method and don't

use them; at the beginning of program, they have several parameters which needs to adjust properly. Metaheuristic algorithms combine various intelligent procedures and guide basic heuristic methods [3]. These algorithms are inspired from different things such as natural phenomena, natural selections and social behaviors and applied in solving the optimization problems. Examples of the recently metaheuristic algorithms are Vortex search [4], WOA (whale optimization algorithm) [5], MBA (mine blast algorithm) [6], WCA (water cycle algorithm) [10], and SFS (stochastic fractal search) [8].

The Firefly algorithm (FA) [9] is one of the nature-inspired algorithms presented to perform global optimization in complex search spaces. In fact, it uses the act of firefly in nature and simulates behavior of attraction to the flashing lights of fireflies.

One of the population based metaheuristic algorithms is the Differential evolution algorithm (DE) which is modeled on Darwin's evolutionary principle of Survival of the Fittest [10]. DE, like the Genetic algorithms, benefits from the Natural Selection Theory and uses its operators like the crossover, mutation and selection to create the new population for the next generation. Over the last decades, experiments on the DE algorithm have proven that it is the simplest algorithm which shows the best performance in metaheuristic algorithm for global optimization and in real parameter optimization. The most important difference between GA and DE is that DE uses distance and direction information from the current population to guide the search process.

Hybridization of DE with other algorithms has been investigated in many studies. DE-VNS [11] is a new type of hybrid method which combines two well-known metaheuristic approaches: Differential Evolution (DE) and Variable Neighborhood Search (VNS) [12], which has, in the last decade, attracted considerable attention in both academic circles and among practitioners. A promising new nature-inspired algorithm known as FA was recently proposed and has gained more attention in the research literature. The ACOFA [13] is the new hybridization for FA and ACO [14] algorithm. This hybrid algorithm has been designed to solve unconstrained optimization problems and FA works as a local search to refine the positions found by the ants. In this paper, we will combine DE and FA global optimization algorithms, and propose the novel hybrid algorithm based on these algorithms which are jointly called as HFADE. As DE has Operators like crossover and mutation, this could provide

more variant population for FA which could help in finding more lighter firefly algorithms.

In the real world, many problems have been proposed and optimization problems are one of them [15]. The optimization problems are single or multi-objective. The multi-objective is the problem with more than one objective function ($m > 1$) and single objective is a problem with one objective function ($m = 1$). The main goal in this procedure is to seek the global minimum or maximum. The function may have more than one minimum or maximum which is called as the local, but only one of them is the global maximum or minimum. The point x^* is the global minimum if $f(x^*) \leq f(x)$ for all the x in the searching space S . Optimization problem may consist of one or more mathematical functions which need to be optimized. The general form of the optimization problem is indicated in Eq. (1).

$$\text{minimize } F(f_1(x), \dots, f_m(x)), \quad x = (x_1, \dots, x_n) \in S \quad (1)$$

Where n is the decision variables, m is the number of objectives, x is decision vector and S is searching space. If the problem has one objective function ($m = 1$), then it should be indicated as Eq. (2).

$$\text{minimize } f(x), \quad x = (x_1, \dots, x_n) \in S \quad (2)$$

The rest of the paper is organized as follows: Section 2 illustrates the DE and FA algorithms, and section 3 discusses the HFADE algorithm, its parameters and boundary control. Section 4 presents 26 benchmark test functions applied for the experiments. Finally, the last section presents the concluding remarks.

II. FIREFLY ALGORITHM AND DIFFERENTIAL EVOLUTION

A. Firefly Algorithm (FA)

Firefly algorithm was introduced by X.S Yang for the first time in 2007 [9]. This algorithm has been inspired from the firefly's behavior and its attraction to the light. It has more similarity to PSO [16] and therefore implementation for this algorithm is much easier. Flashing insects at nights are fireflies and they gather together when they have started the flashing. Each firefly releases small rhythmic light flashes which has a light intensity attraction I and this light will decrease by distance r increment. So, each firefly attracts to the other firefly which is lighter and nearer to them. Intensity

of light in the firefly depicts its fitness value, meaning the lighter the Firefly, the more fitness one has over the other. This light-intensity attractiveness of firefly is demonstrated by I in the firefly algorithm.

1) *Attractiveness and Light Intensity*: Each firefly has position in searching space and the light intensity for this firefly in distance r is $I(r)$ which is fitness value for distance r . The light intensity decrease base on the square of distance. So, the $I(r)$ varies according to the well-known inverse square law.

$$I(r) = \frac{I_s}{r^2} \quad (3)$$

Fireflies attractiveness is proportional to the $I(r)$ seen by surrounding fireflies can be defined as

$$\beta_0 e^{-\gamma r^2} \quad (4)$$

where γ is the light absorption coefficient.

2) *Distance*: The distance between fireflies i and j ($r_{i,j}$) is computed by euclidean distance formulation for d dimensional position x . This distance calculated as follow:

$$r_{i,j} = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \quad (5)$$

Movement: The new position for firefly calculated by Eq. (6). In this equation first term is previous position and second term is used for determining the attractiveness() of a firefly (attractive firefly), towards the attractive neighboring fireflies and third term causes movement in random direction toward the brighter firefly x_i .

$$x_i = x_i + \beta_0 e^{-\gamma r^2} (x_j - x_i) + \alpha \varepsilon_i \quad (6)$$

if the firefly i is not brighter than firefly 'j' then the algorithm will perform random walk with following formulation:

$$x_i = x_i + \alpha(\text{rand} - 1/2) \quad (7)$$

where the coefficient α is a randomization variable, and 'rand' is a random real number between interval [0 1].

Firefly algorithm is explained as follows:

Fireflyalgorithm

```
1: Generate the initial population randomly.
2: Calculate the fitness of initial population based on light intensity of fireflies.
3: While ( criterion )
4:   for i=1,2,...,n do (n firefly )
5:     for j=1,2,...,n do.
6:       Calculate  $\beta$  using Eq.(4).
7:       Distance between two fireflies is calculated using Eq.(5).
8:       If  $I(i) < I(j)$  then
9:         Firefly i is moved towards firefly j using Eq.(6).
10:        Determine new solutions..
11:       else
12:         Firefly i is moved randomly towards j using Eq.(7).
13:       endif
14:     endforj
15:   endfori
16: endwhile
17: Sort the fireflies according to light intensity values of the new solution.
```

B. Differential Evolution (DE)

Most of the metaheuristic algorithms start the searching space with initial population. These methods are population-based metaheuristic algorithms and DE is one of them [10]. The DE is simple to implement and requires a minimum parameter to be adjusted so as to make it the best for combinatorial optimization. It requires to be adjusted based on three parameters which are F, CR and N (population size). Parameter F is the weighting coefficient that is used to generate new trial solutions and CR is the crossover probability which is used to specify a rate of crossover. It has been argued that the DE algorithm is highly under the effects

of choosing proper values for the CR and F, and hence, changing the proper amounts of F and CR during algorithm execution can boost its efficiency. The mutation plays the most important role in the DE algorithm, and as a result, it is one of the main parts of the DE. The mutation equation is as follows:

$$X_{new_i} = a_i + F \times (b_i - c_i) \quad (8)$$

here a, b and c are random selected agents from population and i is the index from dimension d . The X_{new} is the new agent which is created from the mutation. The DE algorithm is described as follows:

DEalgorithm

```
1: Initialize each agent  $X_i$  in population with random position in searching space.
2: Calculate the fitness of each agent  $X_i$  in population.
3: While ( criterion)
4:   for each agent  $X_i$  in  $X_1 \dots X_n$  population do
5:     select three agents a, b and c randomly from population (  $a = b = c = X$  )
6:     select random index  $r_i \in \{1, \dots, d\}$  (disthe dimension of  $X$  )
7:     for  $i=1 \dots d$  do.
8:       pick a uniform distributed number from interval [0,1] as  $r_i$ 
9:       If  $r_i < CR$  or  $r_i = i$  then.
10:        update  $X_{new}$  with Eq.(8).
11:       else
12:         $X_{new} = X_i$ .
13:       endif
14:     endfori
15:   if  $X_{new}$  better than  $X$  then
16:      $X = X_{new}$ 
17:     if  $X_{new}$  better than  $X_{best}$  then
18:        $X_{best} = X_{new}$ 
19:     endif
20:   endif
21: endfor
22: endwhile
23: output the best solution
```

III. A HYBRID ALGORITHM BASED ON FIREFLY ALGORITHM AND DIFFERENTIAL EVOLUTION

The main bulk of research on the FA algorithm has been demonstrated that, in FA algorithm, each firefly moves to the brighter one and when it does not find any brighter firefly, it performs random walk. The random walk is a simple and weak operator that has been performed in FA. Then, this can be changed, and instead, one can use another strong and better operator so as to make the variation in the desired firefly. To circumvent this deficiency, the hybrid algorithm of FA and DE, hence forth, is called (HFADE) has been proposed. The DE algorithm performs the mutation and crossover on the one firefly which couldnt find a brighter firefly. When the supposed firefly couldnt find a brighter one around itself, then, it might be assumed that the supposed firefly is the local best. So, the DE Crossover and mutation operators apply variation to that local best which could help in find another better place around the firefly under question, and thus, avoiding the trap in local solutions. The general steps of (HFADE) algorithm are as follows: First, The initial population ought to be created, and then, each firefly should be allocated the random position and the cost (light intensity) for that position should also be computed. The next step is the process of firefly algorithm. In this step, based on the value of cost (light intensity) in each firefly, if the preconditions of moving to the another firefly is not satisfied, then, it will be used to perform the process of updating in firefly algorithm by Eq. (6). Otherwise, we go to the next step which performs the process of the DE algorithm. In DE algorithm phase, the mutation and crossover operators are applied on those fireflies which couldnt find their brighter counterparts and the conditions didnt apply to them. The new firefly which has been produced by the DE is compared with the previous one. If the cost (light-intensity) for the newly-produced firefly is better than

the previous one, then, it will be swapped with the previous one, and therefore, the new firefly can occupy a position and if the new firefly is better than the global best, then, the global best will also be swapped with the new firefly. Finally, in the last step, if the termination criterion is satisfied, then, the global best firefly should be considered the output. Otherwise, the next iteration will be started from firefly algorithm again. Figure 1 shows the simple flowchart for the proposed algorithm (HFADE). As can be seen in this figure, the (HFADE) is consisted of two main phases, and in each phase, it runs in parallel forms. The proposed hybrid (HFADE) algorithm is as follows:

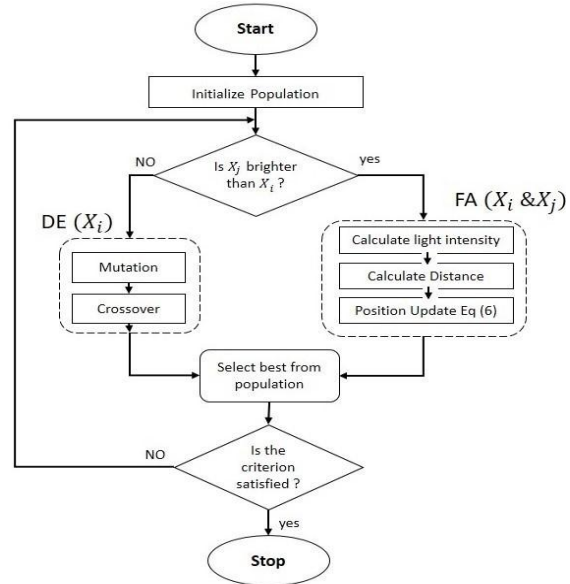


Fig. 1. Flowchart of HFADE algorithm

A. HFADE algorithm

HFADE algorithm:

```
1: Input: objective function  $f$ , constraints and problem dimension  $d$ 
2: Initialization
3: Initial parameters  $n$  (population size),  $\gamma, \beta_0, \alpha, CR$  and  $F$ 
4: Let population be the set of firefly  $1, 2, \dots, n$ 
5: for each firefly do
6: Assign random real number between  $[L, U]$  to the firefly position
7: Calculate fitness (cost) for assigned position
8: endfor
9: Main iterations
10: while (the stopping criterion is not met) do
11: for each firefly  $i$  do
12: for each firefly  $j$  do
13: if  $I(X_i) < I(X_j)$  then
14: Calculate distance  $X_i$  and  $X_j$  by Eq.(5)
15: Calculate  $\beta$  by Eq.(4)
16: Create  $X_{new}$  by Eq.(6)
17: if  $X_{new}$  better than  $X_i$  then
18:  $X_i = X_{new}$ 
19: endif
20: else
21: select three firefly  $a, b$  and  $c$  from population randomly
22: select random index  $q$  from  $q \in \{1, \dots, d\}$  (the dimension of  $X$ )
23: for  $k=1 \dots d$  do,
24: pick a uniformly distributed number from interval  $[0, 1]$  as  $r_k$ 
25: if  $r_k < CR$  or  $k=q$  then
26: update  $X_{new_k}$  with Eq.(8).
27: else
28:  $X_{new_k} = X_i^k$ .
29: endif
30: endfor
31: if  $X_{new}$  better than  $X_i$  then
32:  $X_i = X_{new}$ 
33: if  $X_{new}$  better than  $G_{best}$  then
34:  $G_{best} = X_{new}$ 
35: endif
36: endif
37: endif
38: endfor
39: endfor
40: decrease the mutation coefficient ( $\alpha * adamp$   $\alpha$  is a real value between  $[0, 1]$ )
41: end while
42: the final stage
43: output the minimum value found  $G_{best}$ 
```

B. Complexity and Convergence analysis of HFADE

1) Complexity analysis

HFADE algorithm consists of 2 main parts and these parts are executed in parallel form in each cycle which are shown at the flow chart Fig. 1. In each part process is performed on agent of population with size of n and agent position is d dimensional vectors. The main operator for each part effects on time complexity. The sections that alter time complexity are: DE Crossover and Mutation, FA position update equation. We can explain complexity analysis of HFADE in worst case and prove the fastest execution of these two algorithms combination like this: As the HFADE

algorithms are executed in parallel form, then in each cycle only one of the algorithms is executed. The time complexity for FA is $O(nd)$ and for DE is also $O(nd)$, therefore the HFADE algorithm is run in $O(nd)$ complexity because of the parallel form of execution in FA and DE algorithms.

2) Convergence analysis

For analyzing convergence of the metaheuristic algorithm, Markov chain Monte Carlo method is one of the preferable methods for this task [17]. Most metaheuristic algorithms can be sighted in the framework of Markov chain from a static viewpoint. Now if we look at the proposed hybrid method closely using the framework of Markov chain Monte

Carlo, each firefly in HFADE essentially forms a Markov chain and the appropriate better solutions which created in each phase, replace with previous one. Convergence analyzing based on the Markov chain for algorithm HFADE is performed as follows:

Definition 1. Assume that the best firefly is shown by $X^* := \{x^* \in X : f(x^*) = \min(f(x) \mid x \in X)\}$ where X is probable solution and f is fitness function. The number of best fireflies in firefly population is shown by $\lambda(N) : |N \cap O^*|$.

Definition 2. Algorithm convergence with probability 1 to the best if this condition is true :

$\lim_{g \rightarrow \infty} P\{\omega(N(g)) \geq 1 \mid N(0) = N_0\} = 1$, where g indicate generation number and N_0 is random initial population.

Theorem 1. HFADE algorithm converges to its globally best solution with probability 1. **Proof:** Let $P_0(g) = P\{\omega(N(g)) = 0\}$ then the probability due to the Bayesian conditional probability of $P_0(g + 1)$ is

$$P_0(g+1) = P\{\omega(N(g+1)) = 0\} \Rightarrow P\{\omega(N(g+1)) = 0 \mid \omega(N(g)) = 0\} + P\{\omega(N(g+1)) = 0 \mid \omega(N(g)) \neq 0\}$$

Since the best solution replace with previous one in memory, this expression $P\{\omega(N(g + 1)) = 0 \mid \omega(N(g)) \neq 0\}$ is true.

Hence, $P_0(g + 1) = P\{\omega(N(g + 1)) = 0 \mid \omega(N(g)) = 0\} \times P\{\omega(N(g)) = 0\}$. $P\{\omega(N(g + 1)) = 1 \mid \omega(N(g)) = 0\} > 0$ is true because of the HFADE algorithm by two main phases FA and DE store the best solution.

Make $\Gamma = \min P\{\omega(N(g + 1)) = 1 \mid \omega(N(g)) = 0\}_{\min, g} = 0, 1, 2, \dots$

Then

$$\begin{aligned} &P\{\omega(N(g + 1)) = 0 \mid \omega(N(g)) = 0\} \\ &= 1 - P\{\omega(N(g + 1)) \neq 0 \mid \omega(N(g)) = 0\} \\ &= 1 - P\{\omega(N(g + 1)) \geq 1 \mid \omega(N(g)) = 0\} \\ &\leq 1 - P\{\omega(N(g + 1)) = 1 \mid \omega(N(g)) = 0\} \leq 1 - \Gamma < 1 \end{aligned}$$

Therefore,

$$0 \leq P_0(g + 1) = P\{\omega(N(g + 1)) = 0\} \leq (1 - \Gamma) \times P\{\omega(N(g)) = 0\} = (1 - \Gamma) \times P_0(g).$$

Hence,

$$0 \leq P_0(g + 1) \leq (1 - \Gamma) \times (1 - \Gamma) \times P_0(g - 1) \leq \dots \leq (1 - \Gamma)^{g+1} \times P_0(0).$$

Given that $\lim_{g \rightarrow \infty} (1 - \Gamma)^{g+1} = 0$ and $0 \leq P_0(0) \leq 1$.

Hence $0 \leq \lim_{g \rightarrow \infty} P_0(g) \leq \lim_{g \rightarrow \infty} (1 - \Gamma)^g \times P_0(0) = \lim_{g \rightarrow \infty} P_0(g) = 0$

Then $\lim_{g \rightarrow \infty} P\{\omega(N(g)) \geq 1 \mid \omega(N(0)) = N_0\} = 1$

$1 - \lim_{g \rightarrow \infty} P\{\omega(N(g)) = 0 \mid \omega(N(0)) = N_0\} = 1 - \lim_{g \rightarrow \infty} P_0(g) = 1$.

Therefore, when $g \rightarrow \infty, P\{\omega(N(g)) \geq 1\} \rightarrow 1$. HFADE algorithm could reach to best solution and assurance convergence with probability 1. □

C. Parameter adjustments and boundary control

Parameter adjustment is a non-negligible task which is required to be performed properly in order to get a better result in solving various problems. Besides, the parameter adjustment also is necessary for controlling the boundary whenever the algorithm finds a new solution [18]. The (HFADE) needs boundary control for a Firefly Xi, because its position is required to be in the searching space, which is a boundary between [L U], (L is the lower bound and U is the upper bound of the searching space). The method that controls the boundary is as follows:

$$p = \text{Max}(X, L), q = \text{Min}(p, U) \quad (9)$$

Where Min and Max are the functions that select the minimum and maximum among the input pairs, X is the input firefly position and q is the output which have been controlled in the boundary range [L U]. As it was alluded to previously, The DE algorithm is a simple algorithm and enjoys two primary parameters, F and Crossover rate (CR), which is required to be adjusted properly and FA has three main parameters, Light Absorption Coefficient (gamma), Attraction Coefficient Base Value (beta0), and Mutation Coefficient (alpha). So, the combination of two algorithms of FA and DE also enjoys these parameters.

TABLE I. UNIMODAL TEST FUNCTIONS (D: DIMENSIONS)

Function	D	Range	Min	Formulation
F1(Beale)	2	[-4.5,4.5]	0	$-x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + x_1x_2^3)^2$ $(x_1) \cos(x_2) \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$
F2(Easom)	2	[-100,100]	-1	
F3(Matyas)	2	[-10,10]	0	$f(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$
F4(Colville)	4	[-10,10]	0	
F5(Zakharov)	10	[-5,10]	0	
F6(Schwefel 2.22)	30	[-10,10]	0	

				$f(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2$ $+ 90(x_3^2 - x_4)^2 + 10.1(x_2 - 1)^2 + (x_4 - 1)^2$ $+ 19.8(x_2 - 1)(x_4 - 1)$
				$f(x) = \sum_{i=1}^D x_i^2 + \left(\sum_{i=1}^D 0.5ix_i\right)^2 + \left(\sum_{i=1}^D 0.5ix_i\right)^4$
				$f(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $
F7(Schewefel 1.2)	30	[-100,100]	0	$f(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j\right)^2$
				$f(x) = (x_1 - 1)^2 + \sum_{i=2}^D i(2x_i^2 - x_{i-1})^2$
				$f(x) = \sum_{i=1}^D (x_i + 0.5)^2$
				$f(x) = \sum_{i=1}^D x_i^2$
				$f(x) = \sum_{i=1}^D ix_i^2$
				$f(x) = \sum_{i=1}^D ix_i^4 + Rand$
F8(Dixon-price)	30	[-10,10]	0	
F9(Step)	30	[-5.12,5.12]	0	
F10(Sphere)	30	[-100,100]	0	
F11(SumSquares)	30	[-10,10]	0	
F12(Quartic)	30	[-1.28,1.28]	0	

IV. BENCHMARK TEST FUNCTIONS

In this paper, the proposed algorithm has been evaluated with a set of benchmark functions which are a subgroup of unimodal and multimodal functions. These functions have various dimensions such as 2, 4, 10 and 30. Tables 2 highlights a multimodal test functions and Table 1 shows unimodal test functions. Multimodal test functions have many local minimums, and therefore, they are hard to be solved simply because of the trapping in local solutions. The (HFADE) has been evaluated by these functions and search the global minimum for them. In Table 3, the results for algorithms GA [19], DE, PSO, BA [20], PBA [21] and FA that compared with HFADE. Conditions for experiment and parameter settings for these algorithms is explained in reference [21]. The parameter settings for HFADE are as follows: n=20 (populationsize), $\gamma=2$, $\beta_0=2$, $\alpha=0.2$, CR=0.2 and $F \in [0.2, 0.8]$. The experiment has been performed on the computer with following features: CPU 2.1 GHZ, Ram 8 GB and Matlab 2013 running on computer with windows 10. The NFE=

500,000 (number of function evaluation) was set as Stopping criteria and the values minimum than 1E-12, presented as 0 same as other methods. The mean value and Std Dev (standard deviation) have been calculated from 30 independent runs. In Table 3, the HFADE found the minimum results for most of the functions with best standard deviations. It only did not reach the minimum average value in functions Quartic (F12), Dixon-price (F8), Michalewicz 10 (F25) and schewefel 1.2 (F7). The results of Friedman non-parametric test [22] are also presented in this table and HFADE could rank the best with value 3.15. The low p-value indicates that the results are remarkably different with each other and figure 3 depicts the results for this test with a bar diagram. For analyzing the procedure of convergence in (HFADE), figure 2 has been presented. This figure shows the Convergence diagram for functions F19, F25, F18 and F20 in algorithms FA, DE and (HFADE). As it stands, HFADE has reached the desired minimum with minimum number of iteration and faster against the other algorithms FA.

TABLE II. MULTIMODAL TEST FUNCTIONS (D: DIMENSIONS)

Function	D	Range	Min	Formulation
F13(Schaffer)	2	[-100,100]	0	$f(x) = 0.5 + \frac{\sin^2(\sqrt{x_1^2 + x_2^2}) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$
F14(6 H Camel)	2	[-5,5]	-1.03163	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$
F15(Bohachevsky2)	2	[-100,100]	0	$f(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1)(4\pi x_2) + 0.3$
F16(Bohachevsky3)	2	[-100,100]	0	$f(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1 + 4\pi x_2) + 0.3$

F17(Shubert)	2	[-10,10]	-186.73	$f(x) = \left(\prod_{i=1}^5 \cos((i+1)x_1 + i) \right) \left(\prod_{i=1}^5 \cos((i+1)x_2 + i) \right)$
F18(Rosenbrock)	30	[-30,30]	0	
F19(Griewank)	30	[-600,600]	0	
				$f(x) = \sum_{i=1}^{D-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$
				$f(x) = \frac{1}{4000} \left(\sum_{i=1}^D (x_i - 100)^2 \right) - \left(\prod_{i=1}^D \cos \left(\frac{x_i - 100}{\sqrt{i}} \right) \right) + 1$
F20(Ackley)	30	[-32,32]	0	$f(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^D x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 + e$
				$f(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) + 0.7$
F21(Bohachevsky1)	2	[-100,100]	0	
F22(Booth)	2	[-10,10]	0	$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$
F23(Michalewicz2)	2	[0,π]	-1.8013	$f(x) = - \sum_{i=1}^D \sin(x_i) (\sin(ix_i^2/\pi))^{20}$
F24(Michalewicz5)	5	[0,π]	-4.6877	$f(x) = - \sum_{i=1}^D \sin(x_i) (\sin(ix_i^2/\pi))^{20}$
				$f(x) = - \sum_{i=1}^D \sin(x_i) (\sin(ix_i^2/\pi))^{20}$
F25(Michalewicz10)	10	[0,π]	-9.6602	
F26(Rastrigin)	30	[-5.12,5.12]	0	$f(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$

DE. Also, the test of normality of Kolmogorov-Smirnova and Shapiro-Wilk [22] has been performed for the three functions of F7, F12 and F25 which are hard to be solved, and here, standard deviation is not zero for them. Table 4 presents the results for this test and the p-value, df (degree of freedom), and the statistics for this test are also presented. Accordingly, the p-value which is higher than the significant =0.05 is considered a normal distribution and the lower than that value

is supposed to be an abnormal distribution. Based on the p-value and the test of normality of Kolmogorov-Smirnova, the function of F7 result is normal and other functions have the abnormal distribution. Fig.4 shows the normal and abnormal distributions, the histogram, QQ-plot and Box-plot for the two functions, F7 and F12. F7 is a normal distribution and F12 is considered abnormal. As can be inferred, in normal distributions, the results are in one diagonal line for QQ-plot while in abnormal distributions, this fact does not hold true

TABLE III. HFADE COMPARISON WITH GA, DE, PSO, BA, PBA AND FA (UNIMODAL FUNCTION SET), BOLD VALUES REPRESENT THE BEST

Function	Criteria	GA	DE	PSO	BA	PBA	FA
(F1)	Mean	0	0	0	1.88E-05	0	0
	StdDev	0	0	0	1.94E-05	0	0
(F2)	Mean	-1	-1	-1	-0.99994	-1	-1
	StdDev	0	0	0	4.50E-05	0	0
(F3)	Mean	0	0	0	0	0	0
	StdDev	0	0	0	0	0	0
(F4)	Mean	0.01494	0.04091	0	1.11760	0	0

	StdDev	0.00736	0.08198	0	0.46623	0	0	0
(F5)	Mean	0.01336	0	0	0	0	0	0
	StdDev	0.00453	0	0	0	0	0	0
(F6)	Mean	11.0214	0	0	0	7.59E-10	2.73028E-10	0
	StdDev	1.38686	0	0	0	7.10E-10	1.1535E-11	0
(F7)	Mean	7.40E+03	0	0	0	0	147.401395	8.514535
	StdDev	1.14E+03	0	0	0	0	448.571186	8.768288147
(F8)	Mean	1.22E+03	0.66667	0.66667	0.66667	0.66667	0.66667	0.66667
	StdDev	2.66E+02	E-9	E-8	1.16E-09	5.65E-10	0	0
(F9)	Mean	1.17E+03	0	0	5.12370	0	0	0
	StdDev	76.56145	0	0	0.39209	0	0	0
(F10)	Mean	1.11E+03	0	0	0	0	0	0
	StdDev	74.21447	0	0	0	0	0	0
(F11)	Mean	1.48E+02	0	0	0	0	0	0
	StdDev	12.40929	0	0	0	0	0	0
(F12)	Mean	0.18070	0.00136	0.00116	1.72E-06	0.00678	3.66E-03	9.70E-04
	StdDev	0.02712	0.00042	0.00028	1.85E-06	0.00133	0.001401347	0.00125
(F13)	Mean	0.00424	0	0	0	0	0	0
	StdDev	0.00476	0	0	0	0	0	0
(F14)	Mean	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163
	StdDev	0	0	0	0	0	0	0
(F15)	Mean	0.06829	0	0	0	0	0	0
	StdDev	0.07822	0	0	0	0	0	0
(F16)	Mean	0	0	0	0	0	0	0
	StdDev	0	0	0	0	0	0	0
(F17)	Mean	-186.73	-186.73	-186.73	-186.73	-186.73	-186.73	-186.73
	StdDev	0	0	0	0	0	0	0
(F18)	Mean	1.96E+05	18.20394	15.088617	28.834	4.2831	2.02E+01	1.04E-07
	StdDev	3.85E+04	5.03619	24.170196	0.10597	5.7877	1.147947126	2.95E-07
(F19)	Mean	10.63346	0.00148	0.01739	0	0.00468	0	0
	StdDev	1.16146	0.00296	0.02081	0	0.00672	0	0
(F20)	Mean	14.67178	0	0.16462	0	3.12E-08	6.56E-10	0
	StdDev	0.17814	0	0.49387	0	3.98E-08	1.24159E-09	0
(F21)	Mean	0	0	0	0	0	0	0
	StdDev	0	0	0	0	0	0	0
(F22)	Mean	0	0	0	0.00053	0	0	0
	StdDev	0	0	0	0.00074	0	0	0
(F23)	Mean	-1.8013	-1.8013	-1.57287	-1.8013	-1.8013	-1.8013	-1.8013
	StdDev	0	0	0.11986	0	0	0	0
(F24)	Mean	-4.64483	-4.68348	-2.4908	-4.6877	-4.6877	-4.60E+00	-4.6877

	StdDev	0.09785	0.01253	0.25695	0	0	0.092696359	0	
(F25)	Mean	-9.49683	-9.59115	-4.0071	-9.6602	-9.6602	-9.29521729	-9.653525	
	StdDev	0.14112	0.06421	0.50263	0	0	0.282019302	0.014947	
(F26)	Mean	52.92259	11.71673	43.97714	0	0	47.88406904	0	
	StdDev	4.56486	2.53817	11.72868	0	0	16.13200446	0	
<hr/>									
Friedman	Test	Rank	5.65	3.65	4.08	3.85	3.56	4.06	3.15
		p-value	3.45E-07						
		Statistic	40.609						

TABLE IV. TEST OF NORMALITY KOLMOGOROV-SMIRNOVA AND SHAPIRO-WILK FOR FUNCTIONS F7,F25 AND F12

Kolmogorov-Smirnova			Shapiro-Wilk				
	Statistic		df	p-value	Statistic	df	p-value
F7	0.1719999		20	0.125	0.858999	20	8.0002E-3
F25	0.4219999		20	0.000	0.4839999	20	0.000
F12	0.5270000		20	0.000	0.3579999	20	0.000

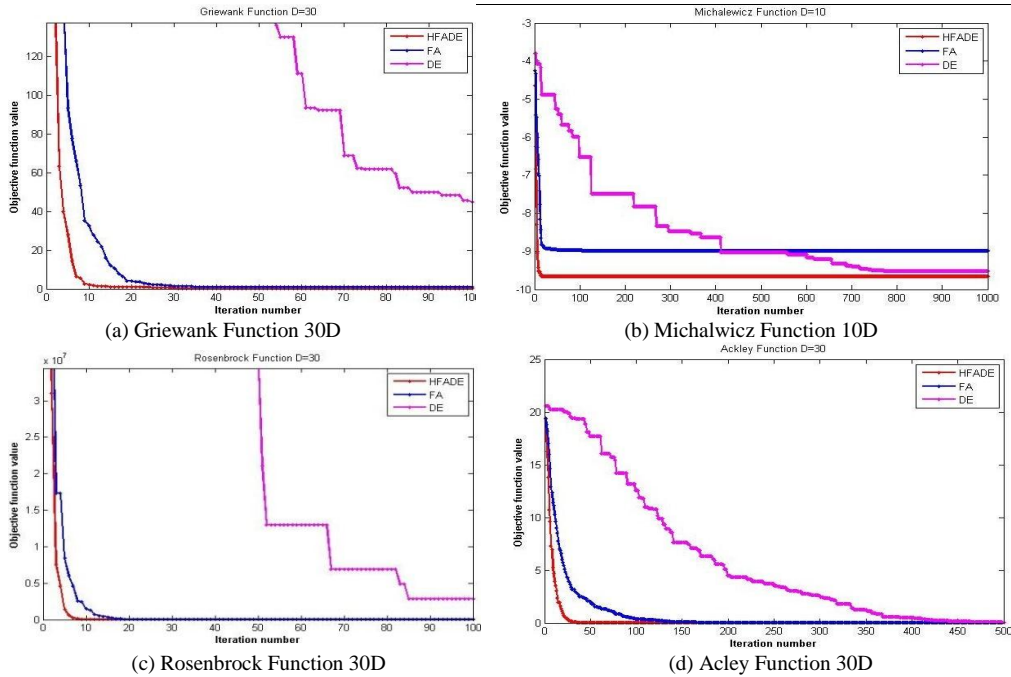


Fig. 2. Convergence diagram for functions F19, F25, F18 and F20 in algorithms FA, DE and HFADE

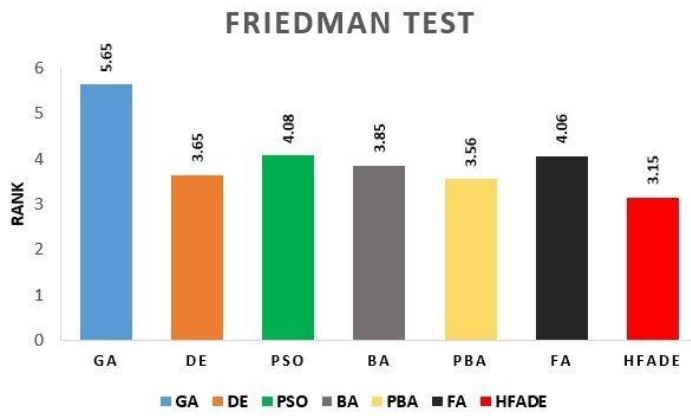


Fig. 3. Bar diagram for non-parametric Friedman test results for Functions F1-F26

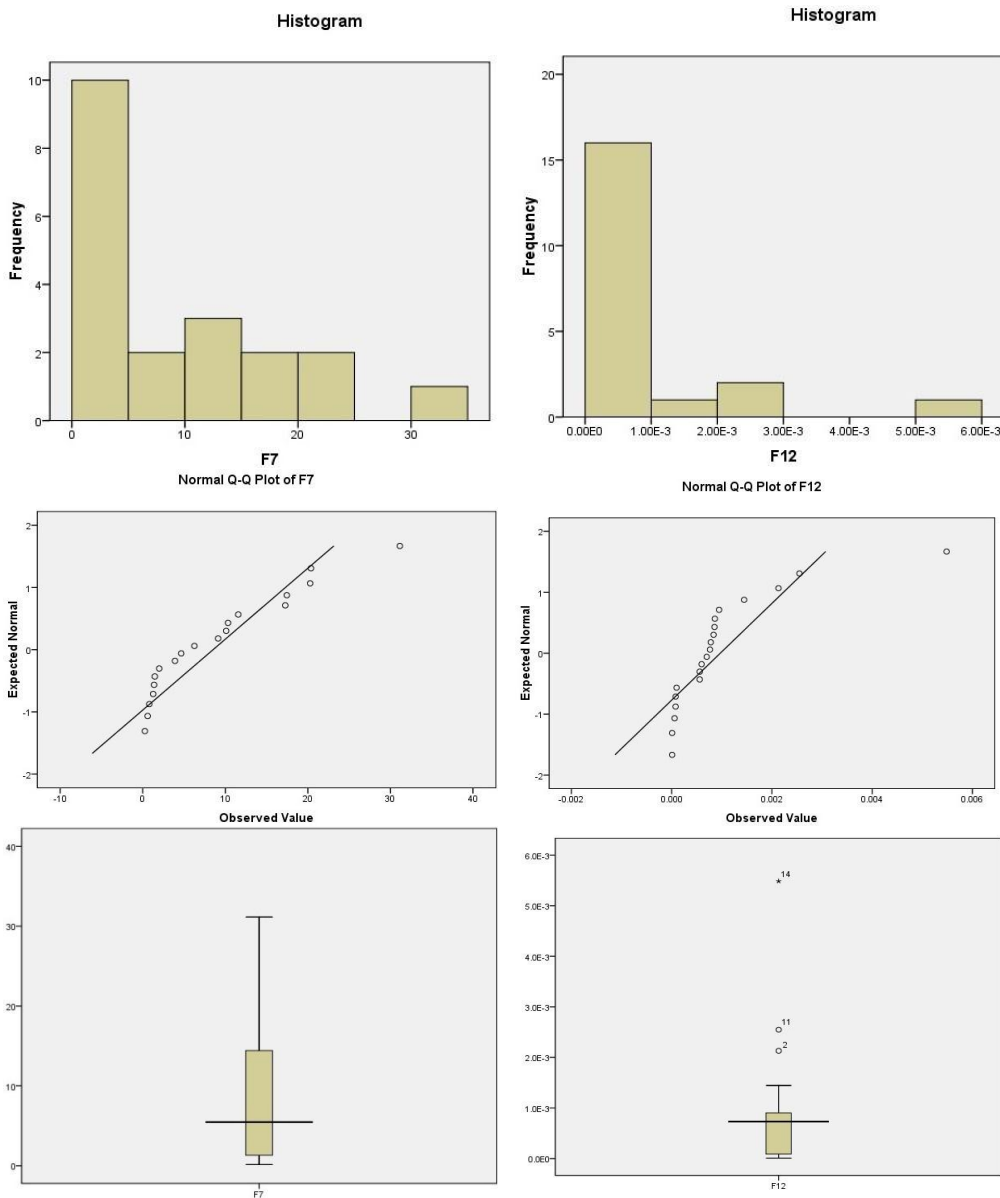


Fig. 4. From top to down respectively: Histogram, Q-Q Graphic and Box plot for functions F7 and F12 results on HFADE algorithm. Left plots are normal for F7 and the Right are abnormal for F12

V. CONCLUSION

The FA is, arguably, one of the most efficient nature-inspired metaheuristic algorithms, which has outperformed most of the algorithms in solving the various optimizing numerical problems. Furthermore, one of the practical metaheuristic algorithms which have been used most widely in the optimization is the DE. The DE algorithm is a simple and practical algorithm which is more amenable to combination with others. In the current study, the FA and DE algorithms are combined in order to design a new hybrid method for seeking the global solution. The proposed hybrid algorithm begins to search from the FA algorithm if the precondition for entering the FA algorithm is not satisfied the process is passed to the DE which makes the alteration with crossover and mutation operators on the agent of population. The experiment based on benchmark functions and non-parametric ranking shows that the proposed hybrid is more dominant and competent than other famous algorithms. Moreover, based on the results of the test of normality and convergence, this proposed hybrid algorithm mostly has abnormal distributions for results and converges to the optimum solutions in minimum number of iterations.

REFERENCES

- [1] C. Grosan, A. Abraham, Intelligent systems a modern approach, Intelligent Systems Reference Library, Springer Berlin Heidelberg, 2011.
- [2] X-S Yang, Introduction to Mathematical Optimization: From Linear Programming to Metaheuristics. Cambridge International Science Publishing, 2008.
- [3] P. Hansen, N. Mladenovic, J. Brimberg, J. Moreno-Prez, Variableneighbor- hood search. In: Handbook of Metaheuristics. Springer US, pp. 61-86, 2010.
- [4] B. Doğan, T. Olmez, A new metaheuristic for numerical function optimization: Vortex search algorithm, Information Sciences, pp. 125-145, 2015.
- [5] S. Mirjalili, A. Lewis, The Whale Optimization Algorithm, Advances in Engineering Software, pp. 51-67, 2016.
- [6] A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, Mine blast algorithm for optimization of truss structures with discrete variables, Computers & Structures, pp. 49-63, 2012.
- [7] H. Eskandar, A. Sadollah, A. Bahreininejad, M. Hamdi, Water cycle algorithm a novel metaheuristic optimization method for solving constrained engineering optimization problems, Computers & Structures, pp. 151-166, 2012.
- [8] H. Salimi, Stochastic fractal search: A powerful metaheuristic algorithm, Knowledge-Based Systems, pp. 1-18, 2015.
- [9] X.-S. Yang, Nature-Inspired Metaheuristic Algorithms: Second Edition, Luniver Press, 2010.
- [10] R. Storn, K. Price, Differential evolution a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization, pp. 341-359, 1997.
- [11] D. Kovaevi, N. Mladenovi, B. Petrovi, P. Miloevi, DE-VNS: Self-adaptive Differential Evolution with crossover neighborhood search for continuous global optimization, Computers & Operations Research, pp. 157-169, 2014.
- [12] Mladenovic N, Hansen E. Variableneighbor- hood search. Comput Oper Res, pp. 1097-1100, 1997.
- [13] R.M. Rizk-Allah, Elsayed M. Zaki, Ahmed Ahmed El-Sawy, Hybridizing ant colony optimization with firefly algorithm for unconstrained optimization problems, Applied Mathematics and Computation, pp. 473-483, 2013.
- [14] M. Dorigo, T. Sttzle, Ant Colony Optimization, MIT Press, London, 2004.
- [15] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
- [16] J. Kennedy, R. Eberhart, Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks. Perth, Australia, pp. 1942-1948, 1999.
- [17] H. Duan, D. Wang, X. Yu, Markov Chains and Martingale Theory Based Convergence Proof of Ant Colony Algorithm and Its Simulation Platform. in Intelligent Control and Automation, pp. 3057-3061, 2006.
- [18] W. Chu, X. Gao, S. Sorooshian, Handling boundary constraints for particle swarm optimization in high-dimensional search space. Information Sciences, pp. 4569-4581, 2011.
- [19] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, 1st Edition, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, pp. 1989.
- [20] D. Pham, A. Ghanbarzadeh, E. Ko, S. Otri, S. Rahim, M. Zaidi, The bees algorithm a novel tool for complex optimisation problems, in: D. Pham, E. Eldukhri, A. Soroka (Eds.), Intelligent Production Machines and Systems, Elsevier Science Ltd, Oxford, pp. 454-459, 2006.
- [21] M. Cheng, L. Lien, Hybrid artificial intelligence based pba for benchmark functions and facility layout design optimization, Journal of Computing in Civil Engineering, pp. 612-624, 2002.
- [22] S. Garcia, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization. Journal of Heuristics, pp. 617-644, 2009.